

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ODSTRANĚNÍ HLUKU MAGNETICKÉ REZONANCE V NAHRÁVKÁCH ŘEČI

CANCELLING NOISE OF MAGNETIC RESONANCE IN RECORDINGS OF SPEECH

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Vrba

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vojtěch Zvončák

BRNO 2021

Diplomová práce

magisterský navazující studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Bc. Filip Vrba

ID: 186641

Ročník: 2

Akademický rok: 2020/21

NÁZEV TÉMATU:

Odstranění hluku magnetické rezonance v nahrávkách řeči

POKYNY PRO VYPRACOVÁNÍ:

Pomocí různých přístupů provedte odstranění šumu v nahrávkách řeči, které byly zaznamenány v prostředí magnetické rezonance. K tomuto účelu využijte technologii Nvidia RTX Voice, VST plug-in modul a veřejně dostupnou knihovnu pro odstranění šumu dle vlastního uvážení. Součástí práce bude také krátké měření v prostředí magnetické rezonance dvěma mikrofony. S využitím tohoto měření vyzkoušejte metodu odstranění šumu odečtením nahrávky přímo z magnetické rezonance a nahrávky mimo magnetickou rezonanci. Kvalitu upravené nahrávky otestujte pomocí metody STOI (Short-Time Objective Intelligibility Measure). V rámci práce bude implementován program v Pythonu pro načtení nahrávek, následné odšumění a otestování zmíněnými algoritmy.

DOPORUČENÁ LITERATURA:

[1] LYONS, Richard G. Understanding digital signal processing. Reading: Prentice Hall, 2011. ISBN 01-370-27-1-9.

[2] How To Build a Deep Audio De-Noiser Using TensorFlow 2.0 [online]. [cit. 2020-09-10]. Dostupné z: <https://medium.com/better-programming/how-to-build-a-deep-audio-de-noiser-using-tensorflow-2-0-79c1c1aea299>

Termín zadání: 1.2.2021

Termín odevzdání: 24.5.2021

Vedoucí práce: Ing. Vojtěch Zvončák

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

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.

ABSTRAKT

Tato práce se zabývá odstraněním šumu v nahrávkách řeči, které byly zaznamenány v prostředí magnetické rezonance. K tomuto účelu je využita technologie Nvidia RTX Voice, VST plug-in modul Noisereduce a vlastní navržená metoda subtraktivního odhlučování nahrávek. V rámci práce je implementován program s jednoduchým grafickým rozhraním v jazyce Python pro načtení nahrávek a následné odhlučnění pomocí navržených metod. Součástí práce je měření v prostředí magnetické rezonance dvěma mikrofony. Kvalita zpracovaných nahrávek je v rámci programu otestována pomocí metody STOI (Short-Time Objective Intelligibility Measure) a také metody subjektivní analýzy v rámci poslechových testů.

KLÍČOVÁ SLOVA

MR, Python, RTX Voice, Noisereduce, odstranění hluku

ABSTRACT

This thesis deals with the removal of noise in speech recordings that have been recorded in an MRI environment. For this purpose, the Nvidia RTX Voice technology, the VST plug-in module Noisereduce and a self-designed method of subtractive de-noising of recordings are used. A program with a simple graphical interface in Python is implemented within the work to retrieve the recordings and then de-noise them using the proposed methods. The work includes measurements in a magnetic resonance environment with two microphones. The quality of the processed recordings is tested within the program using the STOI (Short-Time Objective Intelligibility Measure) method as well as the subjective analysis method within listening tests.

KEYWORDS

MRI, Python, RTX Voice, Noisereduce, de-noising

VRBA, Filip. *Odstranění hluku magnetické rezonance v nahrávkách řeči*. Brno, 2021, 67 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Vojtěch Zvončák

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Odstranění hluku magnetické rezonance v nahrávkách řeči“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené 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 autorského 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.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Vojtěchu Zvončákovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	10
1 Výzkum řeči v magnetické rezonanci	12
1.1 Parkinsonova nemoc	12
1.2 Magnetická rezonance	13
1.3 Studie řečové odezvy	15
2 Návrh systému	16
2.1 Funkce navrhovaného programu	16
2.2 Blokové schéma systému	17
3 Využití nástroje	19
3.1 Nvidia RTX Voice	19
3.2 Virtuální kabel	20
3.2.1 VB-Cable	20
3.3 Knihovna Noisereduce	22
3.4 Subtraktivní metoda	23
3.5 VST	25
3.5.1 ReaFIR	26
3.6 DAW	27
3.6.1 Software Reaper	27
3.7 Metrika STOI	28
3.8 Subjektivní analýza nahrávek	29
4 Nahrávání v přístroji MR	30
5 Programové řešení práce	31
5.1 Struktura navrženého programu	31
5.1.1 Funkce main	32
5.1.2 Funkce load_denoise	32
5.1.3 Funkce rtx_denoise	32
5.1.4 Funkce rtx_input	33
5.1.5 Funkce rtx_record	35
5.1.6 Funkce noisereduce_denoise	35
5.1.7 Funkce subtraction_denoise	36
5.1.8 Funkce stoi_module	36
5.1.9 Soubor config.cfg	38
5.1.10 Soubor requirements.txt	38

6 Průvodce navrženým programem	39
6.1 Menu základních funkcí	39
6.2 Funkce DENOISE	39
6.2.1 Metoda RTX Voice	41
6.2.2 Metoda NOISEREDUCE	41
6.2.3 Metoda SUBTRACTION	42
6.3 Funkce SCORE	42
7 Odšumování pomocí DAW	44
7.1 Nastavení DAW a využitého VST	44
7.2 Hromadné odšumění a export nahrávek	46
8 Výsledky práce	48
8.1 Výsledky metriky STOI	48
8.2 Výsledky subjektivní analýzy	49
9 Diskuze	52
9.1 Objektivní analýza	52
9.1.1 Výsledky STOI - DB	52
9.1.2 Výsledky STOI - CEITEC	52
9.2 Subjektivní analýza	53
9.2.1 Výsledky subjektivní analýzy - DB	53
9.2.2 Výsledky subjektivní analýzy - CEITEC	53
Závěr	55
Literatura	57
Seznam symbolů, veličin a zkratk	59
Seznam příloh	60
A Testování metriky STOI	61
B Příklady zdrojových kódů	64
B.1 Zdrojový kód metody speaker_output funkce rtx_input	64
B.2 Zdrojový kód metody choose_rtx_output funkce rtx_record	65
B.3 Zdrojový kód funkce funkce rtx_denoise	66
C Obsah přílohy	67

Seznam obrázků

1.1	Ukázka přístroje MR	13
1.2	Ukázka snímku z přístroje MR	14
2.1	Blokový diagram navrhovaného programu	17
3.1	Uživatelské rozhraní aplikace RTX Voice	19
3.2	Kontrolní panel aplikace VB-Cable	21
3.3	Pravidlo pro nalezení vhodné maximální latence	21
3.4	Blokové schéma subtraktivní metody	24
3.5	Grafické rozhraní VST ReaFIR	26
3.6	Ukázka grafického rozhraní programu Reaper	27
4.1	Využitý přístroj MR ve výzkumném centru CEITEC	30
5.1	Volba vstupního kanálu v rámci souboru <code>config</code>	34
6.1	Menu pro volbu funkcí	39
6.2	Menu pro načtení nahrávek k dalšímu zpracování	40
6.3	Menu pro volbu metody odstranění hluku	40
6.4	Menu pro výběr úseku šumu	41
6.5	Menu pro výběr šumové nahrávky	42
6.6	Okno pro načtení originálních nahrávek	43
6.7	Okno pro načtení zpracovaných nahrávek	43
7.1	Okno pro načtení referenční nahrávky	44
7.2	Volba FX na panelu kanálu nahrávky	45
7.3	Správně nastavený plugin ReaFIR	45
7.4	Okno pro hromadné zpracování nahrávek	46
8.1	Výsledky subjektivní analýzy – DB	49
8.2	Výsledky subjektivní analýzy – CEITEC	50
8.3	Výsledky subjektivní analýzy v procentech – DB	50
8.4	Výsledky subjektivní analýzy v procentech – CEITEC	51
A.1	Signál zatížený šumem o úrovni –70dbW	61
A.2	Signál zatížený šumem o úrovni –30dbW	62
A.3	Signál zatížený šumem o úrovni –15dbW	63

Seznam výpisů

5.1	Zdrojový kód metody SUBTRACTION	36
5.2	Srovnání nahrávek pomocí metody STOI	37
B.1	Zdrojový kód metody <code>speaker_output</code> funkce <code>rtx_input</code>	64
B.2	Zdrojový kód metody <code>choose_rtx_output</code> funkce <code>rtx_record</code>	65
B.3	Zdrojový kód funkce <code>rtx_denoise</code>	66

Úvod

Zkoumání řeči a principů jejího vzniku je důležitou součástí nejen lékařského odvětví. Pro její studium existuje řada podnětů a aplikace takto získaných informací se rozprostírá napříč mnohými vědními obory. Jedním z významných zaměření tohoto zkoumání je v současné době výzkum řečových poruch či dysfunkcí projevujících se jako symptom řady onemocnění. Nemocí pro které jsou patologické projevy onemocnění s vlivem na řečovou tvorbu typické je Parkinsonova nemoc. Typické symptomy ovlivňující řeč a doprovázející toto onemocnění jsou snížená hybnost mluvidel, zhoršená schopnost kontroly svalů či artikulace a také snížení plynulosti řeči. Pro potřebu zkoumání tvorby řeči a řečového ústrojí (nejen) pacientů s Parkinsonovou nemocí se využívá řada lékařských zobrazovacích metod. Významnou metodou s vysokou rozlišovací schopností je magnetická rezonance. Pomocí tohoto přístroje je možné neinvazivně a detailně zkoumat strukturu lidského těla a jeho orgánů i na trojrozměrné úrovni.

Současný výzkum ukazuje, že další zásadní oblastí pro zkoumání Parkinsonovy choroby je kromě vlivu nemoci na řečové ústrojí také sledování neurologických (především mozkových) změn či abnormalit v souvislosti s reakční dobou řečníků postižených touto nemocí. V rámci přístrojů magnetické rezonance je možné zkoumat řeč pacientů předcítajících stanovená (existující i smyšlená) slova či fráze a na základě jejich reakční doby a změn v různých oblastech mozku zkoumat vliv onemocnění na řeč i samotnou mozkovou aktivitu.

Zásadní nevýhodou zobrazovací metody magnetické rezonance je její vysoká hluchost, která negativně ovlivňuje následné zpracování získaných řečových nahrávek. V rámci této práce je proto cílem navrhnout program s jednoduchým grafickým rozhraním pro odstranění hluku z nahrávek řeči pořízených v tomto přístroji.

Cílem práce je v rámci vytvořeného programu využít několika různých metod pro odstranění hluku a to: aplikaci RTX Voice společnosti Nvidia, která pomocí grafických čipů karet Nvidia a jejich využití umělé inteligence dokáže v reálném čase odstraňovat hluk či šum ze vstupního zvukového signálu, knihovnu Noisereduce, zaměřenou na odšumování nahrávek ve frekvenční oblasti, subtraktivní metodu (metodu vzájemného odečítání nahrávek) a také metodu využívající VST plugin pomocí vybraného DAW software.

Takto navržený program je dále testován na databázi 24 řečových nahrávek z přístroje MR silně zatížených nežádoucím hlukem a také na nahrávkách pořízených při vlastním nahrávání v přístroji MR ve výzkumném ústavu CEITEC. Pro hodnocení takto vytvořených nových a odhlučněných nahrávek je využito objektivní metriky hodnocení srozumitelnosti řeči STOI (*Short-Time Objective Intelligibility measure*). Ze získaných hodnot je vytvořena statistická analýza hodnotící výsledky

této metriky. Dále je provedena subjektivní analýza kvality odhlučnění založená na poslechových testech.

V první kapitole této práce je věnována pozornost výzkumu řeči v magnetické rezonanci. Je zde popsána problematika Parkinsonovy nemoci a její vliv na řeč, dále magnetická rezonance jakožto zobrazovací metoda a také výzkum řeči v MR zaměřující se na časovou odezvu řečníků.

Druhá kapitola se věnuje návrhu budoucího systému, jeho blokovému návrhu a definici požadovaných funkcí.

V další kapitole jsou popsány využití nástroje jako jsou Nvidia RTX Voice, virtuální kabel, využitá knihovna Noisereduce, navržená subtraktivní metoda, problematika DAW a VST, metrika objektivní analýzy srozumitelnosti řeči STOI a také popis využití formy subjektivní analýzy.

Čtvrtá kapitola se v krátkosti věnuje proběhlému nahrávání v přístroji MR v rámci výzkumného centra CEITEC.

V páté kapitole je shrnuta praktická tohoto programu v jazyce Python a popsána struktura programu, tedy jeho jednotlivé části a dílčí funkce.

Šestá kapitola slouží jako průvodce navrženým programem a dává do souvislosti jednotlivé části uživatelského rozhraní s vnitřním programovým řešením.

Sedmá kapitola se věnuje dodatečné metodě odstranění šumu za využití VST v DAW softwaru Reaper a zároveň slouží jako návod a průvodce tímto programem a metodou.

V osmé kapitole je věnována pozornost výsledkům získaným pomocí metriky STOI a jejich jednoduché statistické analýze. Jsou zde také prezentovány výsledky provedené subjektivní analýzy formou poslechových testů.

Poslední kapitola se věnuje rekapitulaci a zhodnocení získaných výsledků, jejich vysvětlení a zasazení těchto výsledků do kontextu dle očekávaných vlastností navržených metod.

1 Výzkum řeči v magnetické rezonanci

Tato kapitola se věnuje problematice výzkumu řeči u pacientů s Parkinsonovou nemocí v přístrojích magnetické rezonance (MR). V rámci následujících kapitol je v krátkosti popsáno toto onemocnění a jeho symptomy ovlivňující tvorbu řeči. Dále je věnována pozornost samotné zobrazovací metodě MR a v poslední sekci je nastíněna metoda zkoumání řeči s ohledem na časovou odezvu řečníka.

1.1 Parkinsonova nemoc

Parkinsonova nemoc (PN) je neurodegenerativní onemocnění při kterém dochází k progresivnímu odumírání dopaminergních neuronů v mozku, především v oblasti bazálních ganglií *substantia nigra pars compacta*. Poprvé byla popsána Jamesem Parkinsonem v roce 1817 a po něm nese tato své pojmenování. Toto onemocnění je spojeno s řadou motorických i nemotorických fyziologických poruch spojených s postupnou progresí této choroby. [4].

Prevalence této nemoci se odhaduje na 160 pacientů na 100 000 obyvatel s největším podílem postižených pacientů mezi 40. až 70. rokem života. Patologické změny při PN se mohou objevit až několik desítek let před manifestací prvních klinických symptomů a jelikož jsou v tomto projevy PN velmi nevýrazné, je téměř nemožné nemoc v tomto období diagnostikovat [1].

Příznaky tohoto onemocnění se u jednotlivých postižených projevují různě a progresse nemoci je obvykle nelineární [6]. Typickými projevy této choroby jsou pocit únavy, ztuhlost, křeče, spánkové poruchy, poruchy rovnováhy, inkontinence, třes a další. Tyto symptomy se obvykle projevují nárazově a dochází tak ke střídání období se silnějšími a slabšími projevy příznaků.

Hypokinetická dysartrie je porucha motorických aspektů řeči, která se vyskytuje u přibližně 70% pacientů s PN. Je to souhrnné označení pro neurogenní poškození nervového systému způsobující neschopnost kontroly řečových mechanismů [5]. Její vliv se vyskytuje v oblastech řečové fonace (vytváření zvuku ve hlasivkách a hrtanu), artikulace či prozodie (přízvuk, melodie, intonace a plynulost) řeči.

Etiologie (příčina) tohoto onemocnění není doposud dostatečně vědecky objasněna. Žádná z mnoha dosavadních vědeckých teorií dostatečně neobjasňuje veškeré patologické změny a klinické příznaky této choroby. Významný vliv na nemoc má dle dosavadního zkoumání dědičná genetická dispozice, ale také životní prostředí či samotný proces stárnutí [1].

1.2 Magnetická rezonance

Magnetická rezonance (MR či anglicky MRI) je zobrazovací technika využívající se ve zdravotnictví jako neinvazivní a bezbolestná metoda zobrazení vnitřních orgánů či jiných částí lidského těla. K zobrazování v rámci MR je využito kombinace velmi silného statického magnetického pole (v několika jednotkách T) a elektromagnetické vlnění s frekvencemi až stovek MHz. K zobrazení tkání lidského těla je využito faktu, že se jejich přístrojem pozorovatelné vlastnosti mění vlivem vnějšího působení silného magnetického pole rozdílně.



Obr. 1.1: Ukázka přístroje MR

Princip přístroje magnetické rezonance využívá vlastnosti atomů, které díky faktu, že mají protony a neutrony odlišný moment hybnosti (spin), dosahují nenulového magnetického momentu [17]. Vlivem kolmo působícího pole B_t se se jejich spiny natáčí kolmo ke směru působení původní magnetické indukce B a dochází k vychýlení z rovnovážné polohy. Tímto způsobem vzniká proměnné magnetické pole, které následně vytváří indukované napětí ve sledované cívce. Standardní velikosti magnetických polí využívaných v přístrojích MR ve zdravotnictví se pohybují v rozmezí 1 T až 7 T [8].



Obr. 1.2: Ukázka snímku z přístroje MR

Ukázkový snímek z přístroje MR se nachází na obr.1.2. Samotný obraz poté vzniká za využití gradientních polí v závislosti na sledované oblasti lidského těla. Změnou velikosti magnetického pole B se mění rezonanční frekvence jader a pomocí měření frekvencí indukovaných napětí na sledované cívice je možné získat informaci o amplitudě v daném bodě. Následně se tak na základě jejich poloh v prostoru dá vypočítat a složit obraz s využitím vyobrazení pomocí šedé škály (*grayscale*). Odstíny v obraze poté indikují různé vlastnosti sledovaných tkání [8].

Výhodami magnetické rezonance jsou: vysoká přesnost, absence negativního ionizujícího záření a také možnost tzv. MRI-fluoroskopie, díky které je možné zobrazovat pohybující se orgány v reálném čase [17].

Nevýhodami této metody jsou vysoké pořizovací náklady přístroje a také vysoké časové nároky, které se však s příchodem modernějších metod zkracují. Nebezpečím metody je náchylnost na přítomnost kovových předmětů v jeho blízkosti, které se vlivem silného magnetického pole zahřívají a mohou tak způsobit poranění. Nebezpečí přístroj činí také pro pacienty s kardiostimulátory, kovovými implantáty či např. tetováním, u kterých obvykle není možné vyšetření provádět. Oproti jiným metodám s sebou MR nese také častější výskyt obrazových artefaktů negativně ovlivňujících výsledné snímky [8].

Zásadní nevýhodou MR, které věnuje právě tato práce je její velká hlučnost. Ta negativně ovlivňuje možnost zkoumání tvorby řeči pomocí řečových nahrávek, které jsou tak silně znehodnoceny nežádoucím hlukem či šumem.

1.3 Studie řečové odezvy

Klíčovou experimentální metodou v kognitivní psychologii i neurovědě je experiment, při kterém jsou účastníci klinických testů vyzváni ke specifickým úkonům testujícím přesnost dané činnosti a jejich odezvu v reálném čase. Statistickým porovnáním výsledků úspěšnosti v těchto zadáních je možné získávat čím dál specifičtější informace o kognitivních a neurologických procesech doprovázejících tyto úkony [16].

V řadě dosavadních neurovědeckých studiích zkoumajících (nejen) řeč, byl naproti studiím psychologickým častokrát záměrně vyloučen faktor časové odezvy (RT – *response time*), pro údajně matoucí vliv při srovnání různých zkoumání mozkové aktivity [16].

V rámci neurovědecké studie řeči a s ní spojené mozkové aktivity byla ve spolupráci britské univerzity *Royal Hollowy University of London* a *Medical Research Council Cognition and Brain Sciences Unit* v Cambridge zkoumána změna této aktivity a také zmíněné časové odezvy v závislosti na daném úkonu. Při hlasitém předčítání bylo použito sady existujících obyčejných i složitějších slov a frází, stejně tak jako tzv. pseudoslov, tedy slov bez jakéhokoli významu, která více či méně připomínají slova existující. Studií byla prokázána změna mozkové aktivity na základě povahy daných úkonů a s ní úzce spojený vliv časové odezvy vzniklé při těchto úkonech [16].

Program navržený v rámci této práci by měl zlepšit podmínky pro zjištění počátku promluv v řečových nahrávkách silně zatížených šumem či negativním hlukem a napomout tak ke snazšímu zkoumání řeči v přístrojích MR s ohledem na časovou odezvu řečníka.

2 Návrh systému

2.1 Funkce navrhovaného programu

V této diplomové práci bylo úkolem navrhnout systém pro hromadné odstranění nežádoucího šumu či hluku pozadí ze dvojice databází řečových nahrávek silně zatížených tímto nežádoucím jevem. V prvním případě šlo o existující databázi řečníků nahranou v přístroji MR, ve druhém případě sadu nahrávek z vlastního nahrávání v tomto přístroji. Pro odstranění negativního hluku mělo být využito kombinace několika různých nezávislých metod a to:

- Programu Nvidia RTX Voice
- Open source knihovny
- Vlastní subtraktivní metody
- Využití VST pomocí DAW

První z metod by měla využívat software RTX Voice fungující za využití grafických karet společnosti Nvidia. Ten je určený k odstraňování negativního šumu či hluku ze signálu daného zvukového vstupu (obvykle mikrofonu) v reálném čase.

V rámci druhé metody by mělo být využito zvolené a volně dostupné knihovny pro odstranění šumu. V případě této práce se jedná o využití knihovny Noisereduce, navržené v jazyce Python. Noisereduce je volně dostupná a otevřená (open-source) knihovna pro odstraňování hluku pozadí za pomoci FFT (rychlé Fourierovy transformace) s využitím vzorku nežádoucího šumu v rámci jedné nahrávky.

V rámci třetí metody bylo úkolem odstranit šum z nahrávek za pomoci subtrakce (vzájemného odečtení) signálů. Odstranění šumu pomocí vlastní subtraktivní metody využívá dodatečné referenční nahrávky šumového pozadí k nahrávce hlavní. Odstranění šumu funguje na principu převodu do frekvenční oblasti (SFFT – krátkodobá Fourierova transformace), odečtení referenčního šumového pozadí od hlavní nahrávky a následném převodu zpět do oblasti časové.

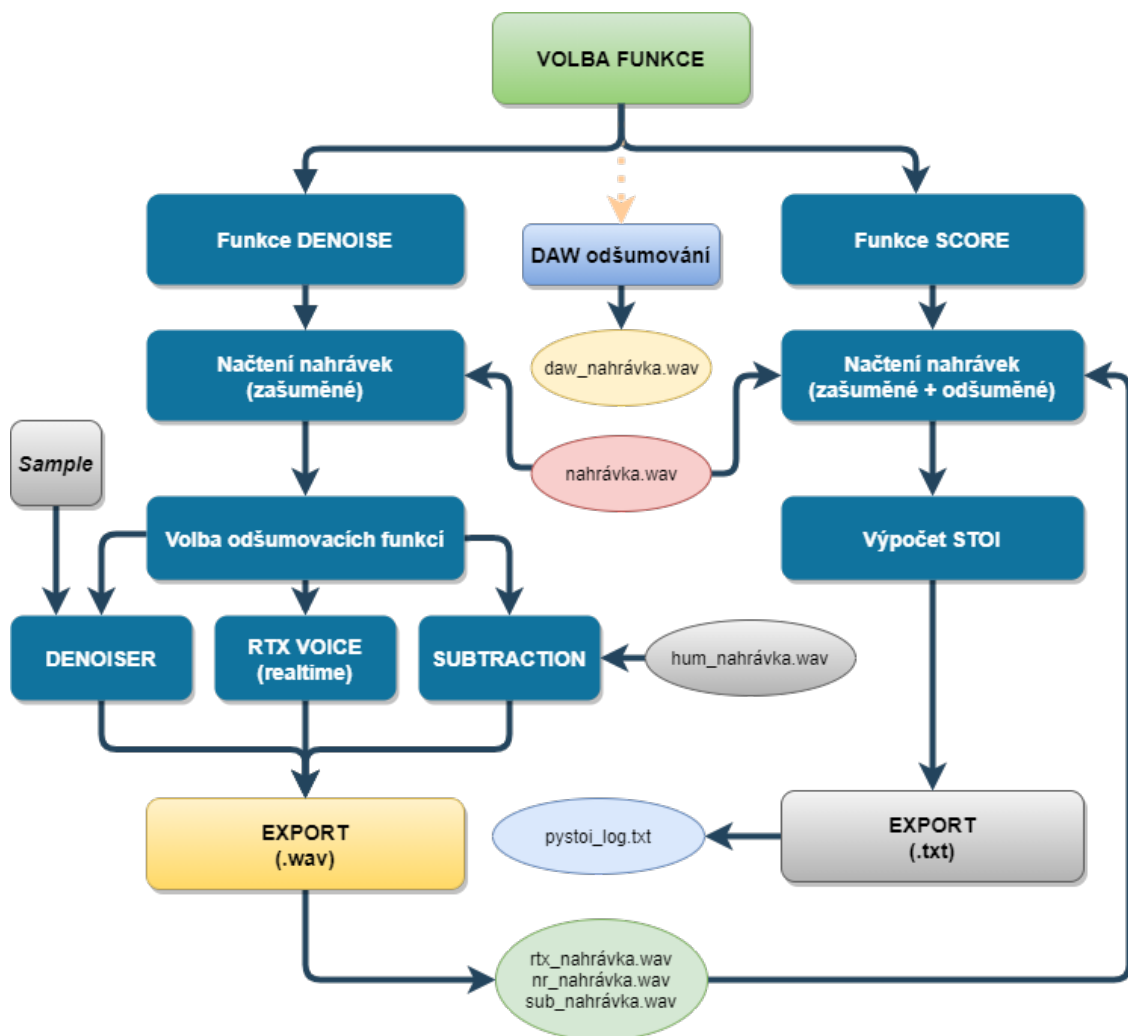
V rámci poslední metody bylo úkolem pro odšumování využít vybranou VST knihovnu (Virtual Studio Technology). Tato metoda není z důvodu vysoké komplikovanosti a částečné zbytečnosti implementace realizována v rámci navrhovaného programu. Využití VST pluginů je realizováno pomocí tzv. DAW softwaru (digital audio workstation) Reaper, tedy mimo samotný navrhovaný program.

Vytvořený program by měl obsahovat jednoduché grafické rozhraní, které bude nabízet možnost hromadného načtení libovolného množství zvukových nahrávek a jejich následné zpracování pomocí jednoho až tří algoritmů pro jejich odšumění. Volba požadovaných funkcí by měla být libovolná, včetně možnosti využití více možností

najednou. Zároveň by měl program obsahovat funkci pro hodnocení kvality odšumění již hotových nahrávek, tedy v případě této práce metriku STOI. V následující sekci je v blokovém diagramu nastíněna základní struktura navrhovaného programu.

2.2 Blokové schéma systému

Na blokovém diagramu 2.1 lze vidět, že po spuštění navrhovaného programu by měl být uživatel vyzván k volbě funkce. Na výběr je v rámci návrhu tohoto programu



Obr. 2.1: Blokový diagram navrhovaného programu

funkce DENOISE a funkce SCORE. Výběrem funkce DENOISE program přistoupí k vyzvání uživatele k výběru nahrávek určených ke zpracování pomocí vybrané funkce pro odstranění negativního hluku. V rámci programu je možné vybrat mezi třemi

volbami a to metodami RTX VOICE, SUBTRACTION a NOISEREDUCE s možností výběru všech těchto metod zároveň. Po načtení složky pro výběr nahrávek program přejde k jejich odšumování, v případě metody RTX VOICE jde o zpracování v reálném čase. Zpracované nahrávky jsou poté postupně exportovány do určené složky programu jako soubory ve formátu *.wav*.

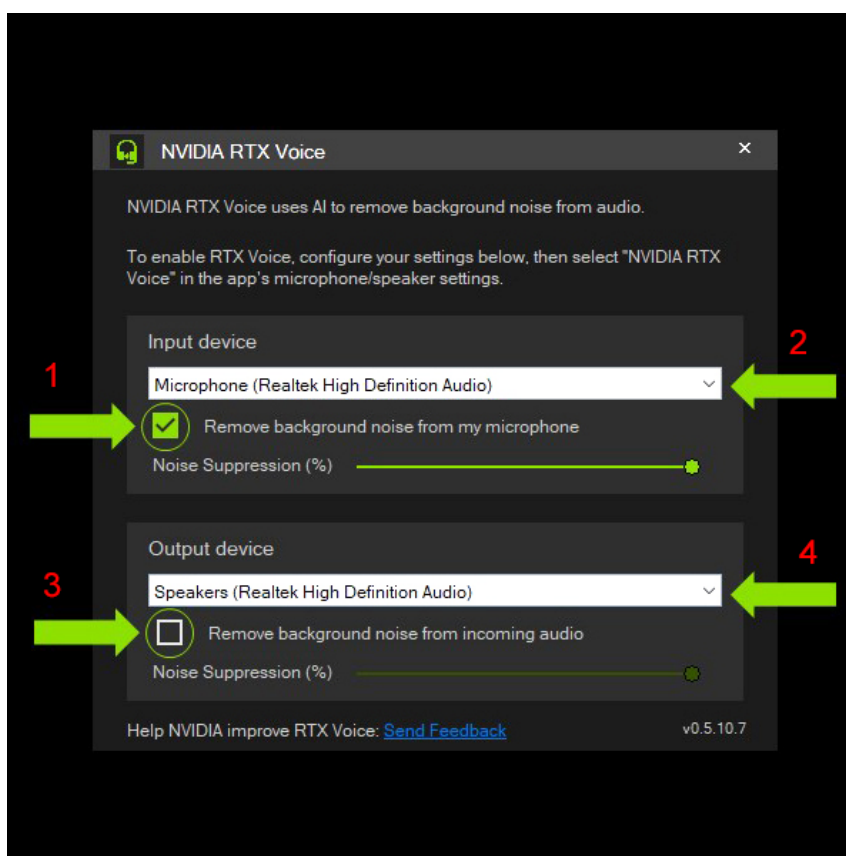
Funkce SCORE je určena k hodnocení již zpracovaných nahrávek a v rámci této volby dojde ke srovnání odhlučněných nahrávek s nahrávkami originálními. K tomuto hodnocení je využito metriky objektivní řečové analýzy STOI a její výsledky jsou programem automaticky zapsány do textového či tabulkového souboru.

3 Využité nástroje

V rámci této kapitoly jsou popsány nástroje a funkce vybrané pro realizaci navrhovaného systému. Pozornost je zde věnována aplikacím jako *VB-Cable* (virtuální kabel), programu Nvidia RTX Voice, knihovně Noisereduce a navržené subtraktivní metodě odšumování. Dále se kapitola zabývá problematikou tzv. VST pluginů a v rámci práce využitého DAW software Reaper. V poslední sekci je popsána využitá metrika hodnocení srozumitelnosti řeči STOI.

3.1 Nvidia RTX Voice

RTX Voice je desktopová programová aplikace (či plugin v rámci některých softwarových aplikací) využívající čipy (GPU) grafických karet Nvidia. Za využití umělé inteligence (AI) v rámci grafických karet umožňuje odstranění nežádoucího hluku či šumu z vysílání, hlasových rozhovorů či vzdálených videokonferenčních schůzek. Ukázka uživatelského rozhraní aplikace se nachází na obr. 3.1



Obr. 3.1: Uživatelské rozhraní aplikace RTX Voice

Jednotlivé funkce popsané na obrázku: 1) Volba odhlučnění vstupního kanálu 2) Výběr vstupu 3) Volba odhlučnění výstupního kanálu 4) Výběr výstupu.

Pomocí interních algoritmů tento program odstraňuje hluk či šum pozadí především za účelem zvýraznění a zvýšení srozumitelnosti řeči v reálném čase. K využití aplikace RTX Voice je zapotřebí počítač s operačním systémem Windows 10, grafická karta Nvidia s čipy řady GTX či RTX a odpovídající grafické ovladače minimální verze 410.18 a vyšší [10].

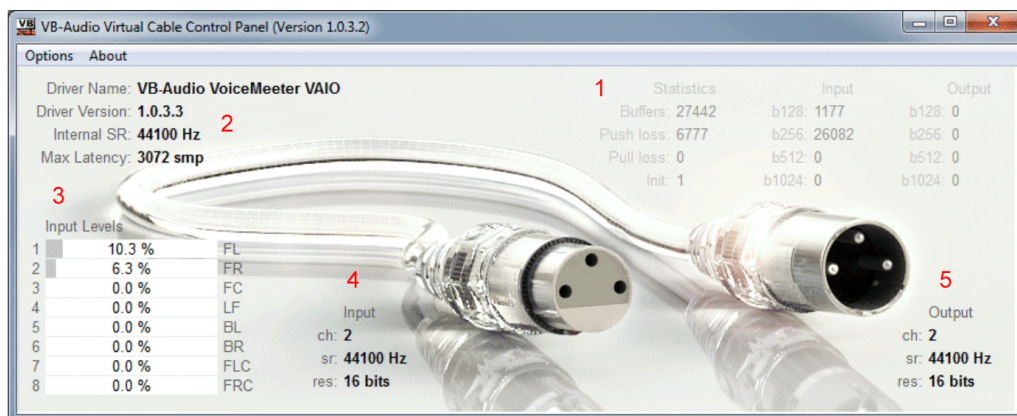
RTX Voice funguje na principu vytvoření virtuálního vstupního (mikrofonního) a výstupního (reproduktorového) kanálu. V rámci virtuálního řetězce program provádí zpracování a odhlučnění procházejícího signálu v reálném čase. Tohoto programu je tak pro komerční účely obvykle využíváno k potlačení hluku pozadí v komunikačních či streamovacích platformách jako jsou např. Twitch, Discord, Skype, Microsoft Teams apod.. Instalace aplikace je pro majitele karet Nvidia RTX/GTX bezplatná a veškeré informace ke stažení a návod k instalaci lze nalézt na webové stránce [10].

3.2 Virtuální kabel

V rámci programu navrhovaného v této práci je aplikace *RTX Voice* využito jakožto prostředku k odstranění negativního šumu a hluku z hotových nahrávek řeči pořízených v přístrojích magnetické rezonance (MR). Z povahy této funkce není možné proces provádět na již hotových nahrávkách v rámci aplikace samotné, ale je pro tento účel zapotřebí využít funkce virtuálního kabelu. Ten umožňuje vytvořit virtuální vstup a výstup potřebný pro implementaci pomocí (pro tento účel vytvořeného) skriptu v jazyce Python. Díky virtuálnímu kabelu je s pomocí programu RTX Voice dále možné libovolně a opakovaně zpracovávat již existující zvukové záznamy. Touto metodou se tak není třeba omezovat pouze na základní funkci odhlučňování jen z mikrofonního a reproduktorového výstupu a či systému.

3.2.1 VB-Cable

V rámci programového řešení této práce byl vybrán program *VB-Cable* od společnosti VB-Audio Software. Jedná se o virtuální zvukové zařízení pracující na principu zvukového ovladače. Veškerý signál přiváděný na vstup virtuálního kabelu je přenesen na jeho výstup. Na obrázku 3.2 je možné vidět kontrolní panel aplikace v systému Windows.



Obr. 3.2: Kontrolní panel aplikace VB-Cable

Na kontrolním panelu je možné v bodě **1** spravovat dvojici důležitých nastavení a to maximální latenci (*max latency*) a vnitřní vzorkovací kmitočet kabelu (*internal sample rate*). Dále je také možné zobrazit název používaného ovladače a jeho verzi. Virtuální kabel podporuje veškeré I/O vzorkovací kmitočty od hodnot 8kHz do 96kHz, avšak pracuje pouze s interními vzorkovacími kmitočty o hodnotách 44.1kHz, 48kHz, 88.2kHz, and 96kHz [18].

V bodě **2** se nachází statistické informace o počtu využitých vyrovnávacích pamětí (bufferů), a jejich velikostí. Hodnoty b128, b256 atd. značí počet přenesených bufferů o velikostí vyšší nebo rovnou tomuto číslu.

V bodě **3** se nachází informace o využití vstupních kanálů 1–8.

V bodech **4** a **5** nalezneme informace o počtu zvukových kanálů, vzorkovacím kmitočtu a bitovém rozlišení vstupu (input) a výstupu (output) virtuálního kabelu.

Virtuální kabel dokáže nezávisle zpracovávat různé nastavení hodnot vstupních a výstupních vzorkovacích kmitočtů a interně je konvertovat na zvolené hodnoty. Pro dosažení nejlepších výsledků (nejlepší kvality přenosu) a snížení výpočetních nároků je vhodné nastavit interní vzorkovací kmitočet na hodnotu shodnou se vzorkovacím kmitočtem I/O.

Pro nastavení nejvhodnější maximální latence a velikosti vyrovnávací paměti (bufferu) je dobré se řídit pravidlem uváděným výrobcem [18] viz obr. 3.6.

$$MaxLatency = 3 \times (MaxBufferSize) \times \frac{InternalSR}{OutputSR}$$

Obr. 3.3: Pravidlo pro nalezení vhodné maximální latence

Pro zjištění vhodné velikosti bufferu se stačí řídit údaji o jejich přeneseném počtu uvedené na kontrolním panelu v bodě **2**. Nejvhodnější je vybrat největší používaný z nich (nejvyšší s nenulovým počtem přenosů).

Návod na instalaci programu je možné nalézt na oficiálních webových stránkách výrobce [18].

3.3 Knihovna Noisereduce

Noisereduce je volně dostupná knihovna v jazyce Python využívající metody spektrálního odšumování, při kterém je ze zašuměné nahrávky získán vzorek nežádoucího šumu a na jeho základě je šum v rámci celé nahrávky potlačen. Skript návrhové vychází z interního VST efektu DAW softwaru Audacity v jazyce C++ s názvem NoiseReduction. V rámci skriptu je nutné předem definovat referenční časovou oblast nahrávky obsahující šum. Na jejím základě je poté vypočítán šumový profil využitý k následnému odečtení hlukového pozadí od originální nahrávky.

Průběh algoritmu se dá rozdělit do následujících kroků:

1. Výpočet FFT z vybraného vzorku šumu
2. Výpočet statistických hodnot z FFT šumu (frekvenční oblast)
3. Výpočet prahu ze statistických hodnot šumu
4. Výpočet FFT z celé nahrávky
5. Stanovení masky srovnáním FFT nahrávky s vytvořeným prahem
6. Vyhlazení masky filtrem (časová i frekvenční oblast)
7. Maska je invertovaně aplikována na FFT nahrávky

Algoritmus nejprve provádí FFT analýzu pouze šumového signálu za využití Hannova okna o velikosti 2048 vzorků. Z těchto dat jsou následně vypočítány statistické hodnoty jako průměrný výkon pro každé z 1025 takto získaných frekvenčních pásem. V následující fázi je z na základě těchto dat a také nastavení citlivosti algoritmu vypočítán práh pro každé frekvenční pásmo.

Během fáze odšumování je signál sledované nahrávky srovnán s takto vytvořeným prahem, jestliže signál nahrávky překročí stanovený práh v daném pásmu, je zisk pro dané pásmo nastaven na 0dB. V opačném případě je zisk v tomto pásmu odpovídajícím rozdílem vůči šumovému prahu snížen za účelem potlačení šumu v tomto pásmu.

Následně je provedeno časové vyhlazování, které zaručuje, že se změna potlačení na různých frekvenčních pásmech nemění příliš skokově. Dále je provedeno také frekvenční vyhlazování, které zajistí, že se jednotlivá frekvenční pásma nikdy nepotlačují či nezesilují izolovaně.

Data o vypočítaných potlačeních na jednotlivých pásmech jsou následně aplikovány na FFT data odšumovaného signálu a poté je za pomoci IFFT (inverzní Fourierovy transformace) celý signál opětovně složen a převeden zpět do časové oblasti. Výsledný signál je v rámci funkce doplněn nulovými vzorky pro zajištění shodné délky s originální nahrávkou. Tato funkce je důležitá pro pozdější analýzu nahrávek pomocí metriky STOI. V rámci skriptu je nutné definovat časovou oblast nahrávky pro vyhodnocení šumové masky. Toho je v navrhovaném systému docíleno pomocí jednoduchého GUI okna s možností zadání časového rozsahu v sekundách.

Veškeré informace o knihovně a její zdrojový kód je možné nalézt na oficiálním webu [9].

3.4 Subtraktivní metoda

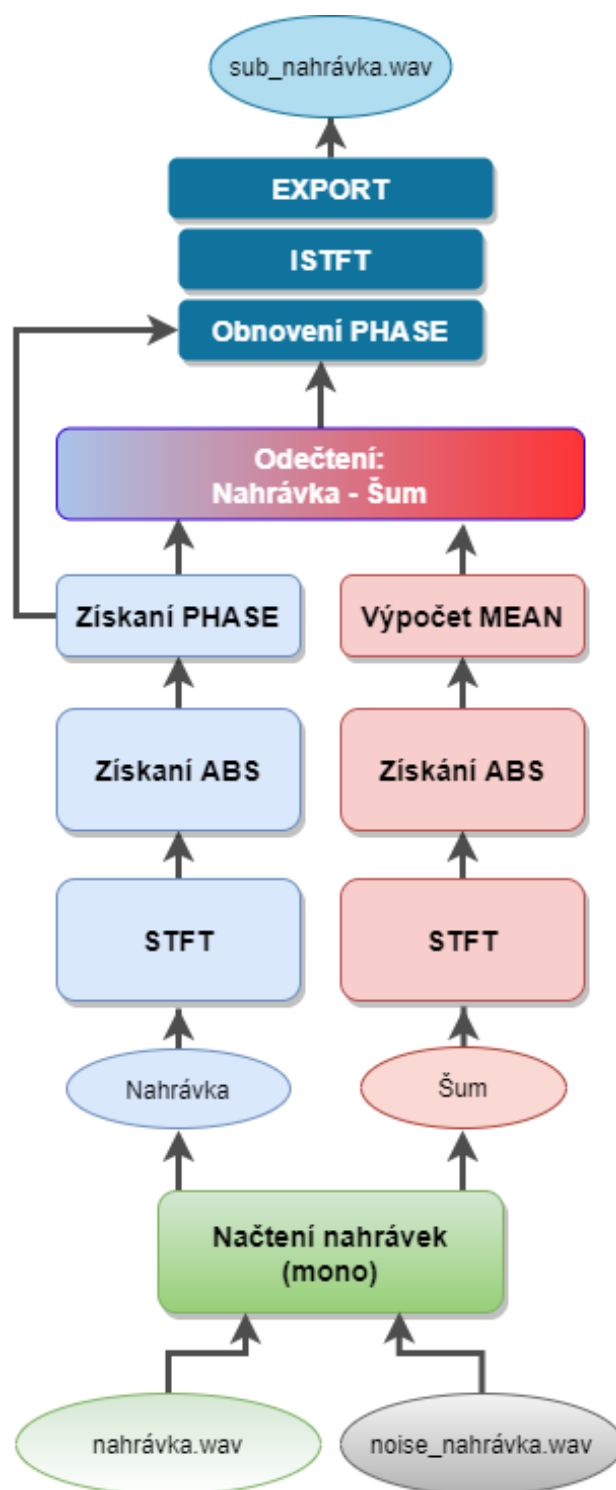
Subtraktivní metoda je v rámci práce navržený způsob odšumění nahrávek za použití vzájemně časově korelované dvojice signálů. Tato metoda byla navržena v rámci této práce a to specificky pro potřebu odstraňování šumu v přístrojích MR.

Při nahrávání řečníků v tomto přístroji je využito dvojice mikrofónů: jeden s co nejbližším kontaktem zaznamenávající řečníka, druhý zaznamenávající šumové pozadí. Algoritmus probíhá jako současné zpracování zvlášť hlavní nahrávky a zvlášť nahrávky šumového pozadí. U těchto nahrávek je předpokladem, že jsou časově korelované, respektive byly pořízeny současně. Pro ideální funkci metody je tak potřeba dodat hlavní nahrávku, tedy nahrávku obsahující především užitečný signál se šumovým pozadím a poté nahrávku se zaměřením právě na toto šumové pozadí.

Metoda funguje na principu převodu signálu do spektrální oblasti pomocí FFT (rychlé Fourierovy transformace), následně jejich vzájemného odečtení v této oblasti a poté finálního složení zpět do oblasti časové. Takto vytvořená nahrávka je tak ochuzena o šum pozadí získaný na základě referenční šumové nahrávky.

Pro co nejspokojivější výsledky je nutné, aby mikrofón zaznamenávající šumové pozadí nebyl příliš vzdálený mikrofónu hlavnímu, respektive příliš vzdálený zdroji nežádoucího hluku. Tímto způsobem je zajištěna co největší shoda mezi frekvenčními spektry obou signálů pro co nejkvalitnější odečtení. Zároveň však nesmí šumový mikrofón obsahovat příliš užitečného signálu, tedy promluví řečníka v přístroji MR. Při odečtení by tak došlo k nechtěnému potlačení signálu řeči.

Blokové schéma navržené metody se nachází na následující stránce na obrázku 3.4.



Obr. 3.4: Blokové schéma subtraktivní metody

3.5 VST

VST, neboli *Virtual Studio Technology* je audio softwarové plug-in (zásuvný modul) rozhraní spojující simulace reálných efektových jednotek, digitálních jednotek či virtuálních nástrojů a syntezátorů. Samotné rozhraní a jeho SDK (*Software Development Kit*) byly vyvinuty a poprvé představeny firmou Steinberg v roce 1996 jako součást firmou vyvíjeného DAW Steinberg Cubase 3.02 [13].

VST se dají rozdělit do tří základních kategorií:

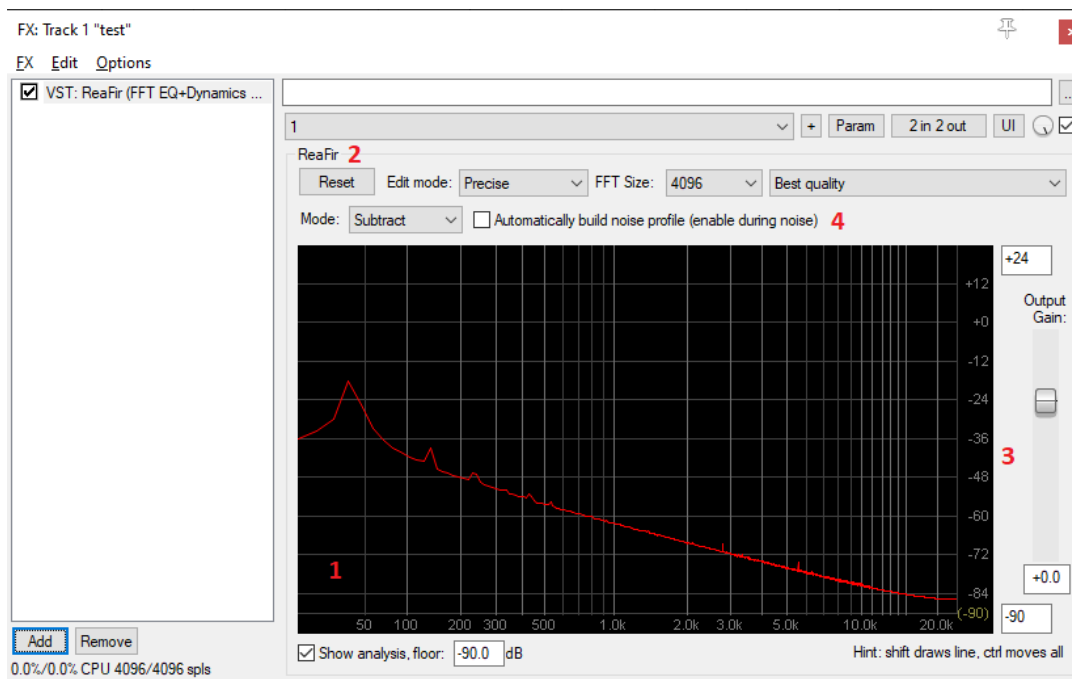
- **VST instruments** - neboli VST nástroje, tedy efekty generující zvuk. Obvykle virtuální syntezátory či samplery simulující existující nástroje, či tvořící úplně nové digitální nástroje.
- **VST effects** - neboli efekty zpracovávající či modifikující audio. Často simulující reálné analogové či digitální efekty a efektové jednotky (reverb, delay, phaser, chorus, kompresor a mnoho dalších) či tvořící úplně nové. VST efekty je obvykle možné libovolně řetězit a vytvářet tak celé kaskády digitálního zpracování.
- **VST MIDI effects** - neboli efekty využívající technologie MIDI. Obvykle přijímají zprávy MIDI hardwarových nástrojů či kontrolerů a ty dále zpracovávají či za využití dalších VST fungují jako komplexní digitální hudební nástroje.

VST obvykle fungují jako součást a rozšíření v rámci DAW (viz následující kapitola 3.6) přidávající širokou paletu funkcí. Software v rámci kterého program VST pracuje se nazývá VST host. Obvykle se jedná o různé druhy již zmíněných DAW jako jsou Audacity, Cubase, Sonar, Pro Tools a mnohé další. VST obsahují více či méně komplexní grafické rozhraní často odpovídající vzhledu reálných simulovaných nástrojů a efektů, nebo představující vlastní a nová rozhraní. V rámci svého těchto rozhraní téměř standardně nabízí možnost uložení zvoleného nastavení v rámci tzv. presetů. Ty umožňují uložení a následné vyvolání těchto nastavení pro potřebu zrychlení a zjednodušení práce s pluginem.

VST představují způsob jak získat řadu funkcí v rámci audio zpracování bez nutnosti investice do jinak mnohdy velmi nákladných nástrojů či hardwarové techniky. Existují tisíce VST pluginů v různých cenových relacích, od bezplatných (freeware) po placený komerční software. Za dobu své existence prošlo VST řadou aktualizací, rozšíření a vylepšení. Současná platforma VST3 nabízí oproti předchůdcům například vyšší stabilitu a výkon, podporu 64-bitového zpracování, nelimitované I/O (vstupy a výstupy) pro efekty a vícenásobné I/O pro MIDI, automatizaci zpracování audia až na úroveň samotných vzorků a řadu dalších [14].

3.5.1 ReaFIR

Za účelem odstranění šumu z nahrávek v rámci této práce bylo využito VST pluginu *ReaFIR*, který je bezplatnou součástí využitého DAW Reaper. Jedná se o FFT ekvalizér a dynamický procesor s několika různými funkcemi jako *EQ*, *Gate*, *Compressor*, *L/R convolute*, či mód *Subtract* využitý právě v rámci této práce.



Obr. 3.5: Grafické rozhraní VST ReaFIR

Na obrázku 3.5 je možné vidět grafické rozhraní pluginu *ReaFIR*. Hlavní část rozhraní tvoří FFT spektrální analyzátor v bodě **1**, panel nastavení v bodě **2** a jezdec pro nastavení výstupního zesílení v bodě **3**.

V rámci zpracování signálu v této práci je využit mód *Subtract*. V okně analyzátoru je možné vidět masku vstupního signálu znázorněnou červenou křivkou. Samotný vstupní signál je ve chvíli spuštění přehrávání s aplikovaným efektem znázorněn křivkou žlutou. Panel funkcí dále nabízí možnost nastavení velikosti FFT okna, výběr *Edit mode* pro různé způsoby zakreslení bodů pro ekvalizér a další. S využitím módu *Subtract* a následným zaškrtnutím funkce *Automatically build noise profile* v bodě **4** program vytváří masku ve frekvenční oblasti na základě spektrální obálky přijatého signálu. Dále se zde nachází tlačítko *Reset* pro restartování navolených křivek či vytvořené masky vstupního signálu.

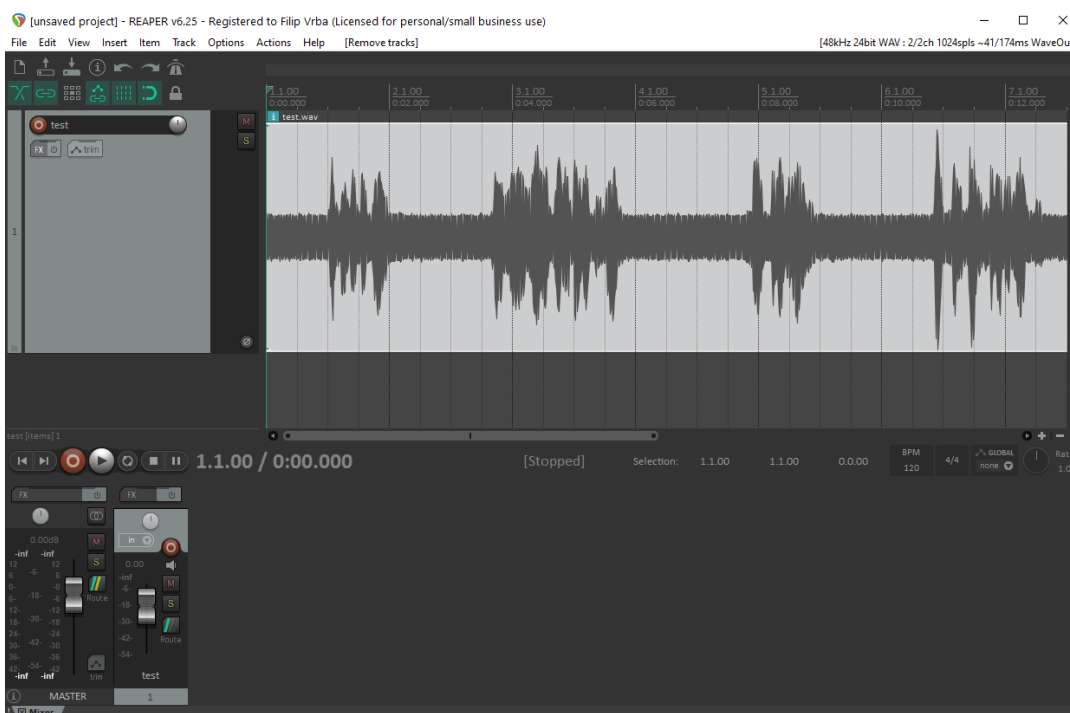
Popisu konkrétního využití programu pro odšumění nahrávek se věnuje sekce v rámci kapitoly 5.

3.6 DAW

Z důvodu velké komplikovanosti programové implementace a potenciálního načítání VST pluginů v rámci navrhovaného programu v jazyce Python je pro odšumování nahrávek pomocí VST v této práci využito tzv. DAW softwaru. DAW (*Digital Audio Workstation*) je elektronické zařízení nebo aplikační software sloužící k nahrávání, editaci či produkci zvukových souborů. DAW existují v řadě konfigurací, od samostatných aplikačních programů (software) či integrovaných jednotek (hardware), po komplexní spojení řady komponentů řízených samostatným počítačem [7].

3.6.1 Software Reaper

Pro účely odstranění šumu za pomoci VST pluginů v této práci je využito softwaru Reaper od společnosti Cockos, který pro svoji uživatelskou přívětivost, spolehlivost, široké možnosti modifikací a nízkou cenu oproti konkurenci patří mezi nejoblíbenější DAW současnosti.



Obr. 3.6: Ukázka grafického rozhraní programu Reaper

Software zdarma nabízí řadu kvalitních interních rozšíření (pluginů) a také 30denní zkušební verzi bez omezení funkcí. Tento program kromě základních funkcí společných pro většinu DAW (přehrávání, editace zvukových souborů, míchání, aplikace efektů, export atd.) nabízí také možnost hromadného zpracování (batch editing)

nahrávek s možností aplikace vybraného FX-chain (tzv. efektového řetězce) a nastavitelného formátování výstupních souborů. Pro (nejen) tuto užitečnou funkci byl tento software zvolen jako nejvhodnější software pro účel hromadného zpracování nahrávek v této práci.

Oficiální stránka programu Reaper a jeho technická dokumentace je dostupná na oficiální webové adrese vývojářů Cockos [12]. Praktická realizace odšumování za využití softwaru Reaper je popsána v kapitole 5.

3.7 Metrika STOI

STOI, neboli *Short-Time Objective Intelligibility measure* je metodou objektivní analýzy řečové srozumitelnosti (OIM) se zaměřením na hodnocení degradovaného řečového signálu. Je vhodná zejména pro hodnocení řečových signálů zpracovaných časově-frekvenčním (TF) váhováním, tedy vhodná pro hodnocení nahrávek u kterých došlo např. k odstraňování šumu či řečové separaci. Je založena na pokročilé metodě měření srozumitelnosti pro krátké (přibližně 400ms) časové segmenty a využívá jednoduchou metodu TF dekompozice za využití diskrétní Fourierovy transformace (DFT) [15].

Tato metoda využívá porovnání originálního a zpracovaného zvukového souboru (řečové nahrávky) a je navržena s interním vzorkovacím kmitočtem 10kHz pokrývajícím relevantní kmitočtovou oblast pro zkoumání řeči. Předpokladem pro správné zpracování a přesné vyhodnocení je vzájemné časové srovnání (korelace) obou nahrávek [15].

Po segmentaci obou signálů za využití Hanningova okna jsou signály rozděleny do sady třetinooktákových pásem. d_j , neboli *intermediate intelligibility measure* (tedy pokročilá míra srozumitelnosti) je parametr definovaný jakožto odhad lineárních korelačních koeficientů mezi originální a zpracovanou nahrávkou. Výsledná hodnota metriky d je poté získána jakožto:

$$d = \frac{1}{JM} \sum_{j,m} d_j(m), \quad (3.1)$$

kde M značí počet segmentů a J celkový počet třetinooktákových pásem. Takto získaná normalizovaná bezrozměrná hodnota d se na základě kvality srozumitelnosti řeči ve zpracované nahrávce pohybuje v rozsahu mezi 0 a 1. Čím vyšší je toto číslo (čím více se blíží 1), tím vyšší je vyhodnocená srozumitelnost řeči v zpracované nahrávce.

Metrika STOI byla v rámci své publikace srovnána s dalšími třemi sofistikovanějšími metodami objektivní analýzy řečové srozumitelnosti (OIM) a to metodou

CSTI, DAU a NSEC. V rámci vzájemného srovnání těchto metod dosáhla metrika STOI výrazně přesvědčivějších výsledků [15]. I z tohoto důvodu byla vybrána jako vhodná metoda pro hodnocení srozumitelnosti odhlučněných nahrávek řeči v rámci této práce.

V příloze A je metrika STOI otestována na zvukové nahrávce, do které byl ve třech příkladech uměle přidán bílý šum (white noise) o výkonech -70dbW , -30dbW a -15dbW . V každém z uvedených grafů je vidět časový průběh čisté nahrávky, přidaného bílého šumu a následně nahrávky získané jejich sečtením. Se zvyšujícím se zatížením šumem můžeme pozorovat snižující se hodnotu d .

Na základě dostupného skriptu pro programové prostředí Matlab je odvozena knihovna *PyStoi* určená pro jazyk Python. Právě této knihovny je využito pro hodnocení řečových nahrávek v programovém řešení vytvořeném v rámci této práce.

3.8 Subjektivní analýza nahrávek

V rámci práce je kromě metody objektivní analýzy pomocí metriky STOI zpracována také analýza subjektivní. Ta je realizována pomocí dvojice poslechových testů realizovaných v rámci online dotazníků.

Soubor respondentů je náhodně vybrán mezi zkušenými i nezkušenými posluchači a navržený test je realizován metodou párového srovnávání. Posluchač v testu srovnává vždy mezi dvojicí nahrávek, přičemž se vždy jedná o srovnání stejné referenční nahrávky zpracované každou s navržených metod. Mezi tyto nahrávky je také zařazena nahrávka originální (nezpracovaná). Délka hodnocených nahrávek je v obou případech v rámci několika vteřin. Počet nahrávek v každém testu odpovídá metodě „každý s každým“, tedy počtu dle vzorce:

$$N_c = \frac{N * (N - 1)}{2}, \quad (3.2)$$

kde N_c je celkový počet kombinací a N je počet srovnávaných metod včetně originální nahrávky.

V rámci hodnotící fáze jsou tak respondenti postaveni před sadu 6 a 10 otázek v každém z testů. V prvním se jedná o srovnání v rámci zpracování nahrávek z poskytnuté databáze a ve druhém případě k hodnocení nahrávek pořizovaných ve výzkumném centru CEITEC. Pořadí otázek je voleno vždy náhodně a test není nijak časově omezen. Respondent je požádán o volbu nahrávky, která je dle jeho subjektivního cítění více srozumitelná a zároveň méně zatížená hlukem či šumem pozadí. Vytvořená poslechová analýza je provedena nad rámec zadání této práce a je vypracována především pro získání ucelenějšího pohledu na chování jednotlivých navržených metod a způsobů odstranění šumu v prostředí MR.

4 Nahrávání v přístroji MR

V rámci této diplomové práce bylo za účelem testování vytvořených metod provedeno samostatné nahrávání řeči v přístroji MR. Nahrávání proběhlo v brněnském výzkumném centru CEITEC. Jednou z hlavních motivací pro provedení tohoto nahrávání byla možnost testování všech metod, především však subtraktivní metody odšumování za využití nahrávek ze dvojice mikrofونů.



Obr. 4.1: Využitý přístroj MR ve výzkumném centru CEITEC

Zkoumaný řečník byl prvním z mikrofونů snímán kontaktně v oblasti úst pro maximalizaci užitečného řečového signálu v nahrávce. Druhý z mikrofونů byl umístěn vedle přístroje MR a nasměrován ke zdroji hluku. Z důvodu nemožnosti využití konvenčně využívaných dynamických či kondenzátorových mikrofонů obsahujících magnetické součástky v prostředí silného magnetického pole přístroje MR nám bylo poskytnuto a následně využito tzv. optických vláknových mikrofонů s optickým vedením signálu bez obsahu nežádoucích magnetických materiálů. Vlivem pandemické situace v době tvorby této práce bylo možné uskutečnit tato nahrávání až po dostatečném rozvolnění vládních opatření. Nebylo tak možné provést množství nahrávání potřebné pro optimalizaci tohoto postupu. Využití takto získaných nahrávek v rámci testovaných metod je proto spíše experimentální.

5 Programové řešení práce

V této kapitole jsou popsány jednotlivé funkce a části výsledného programového řešení práce. Vytvořený program je realizován v jazyce Python, konkrétně v překladači verze 3.7 a byl navržen a testován ve vývojovém prostředí PyCharm 64bit (2019). Tento programovací jazyk byl vybrán především pro jeho uživatelskou přívětivost a také díky velké dostupnosti vhodných rozšíření pro realizaci navrhovaného systému. Externím dodatkem k navrženému systému je využití DAW softwaru Reaper a jeho funkce umožňující hromadné zpracování nahrávek za využitím rozšíření (pluginu) VST.

V kapitole přílohy C se nachází výčet přiložených souborů nutných pro úspěšné spuštění a správnou funkci programu. Pro tvorbu jednoduchého grafické rozhraní (GUI) v rámci tohoto programu bylo využito Python knihovny EeasyGUI [3]. Program je kromě samotného skriptu realizován také jako samostatně spustitelný *.exe* soubor pro co nejsnazší využití vytvořený za pomoci knihovny *pyinstaller* dostupné z oficiální webové stránky vývojářů [11].

5.1 Struktura navrženého programu

Následující kapitola se věnuje popisu jednotlivých programových funkcí v rámci jazyka Python. Popsány jsou zde využití knihovny a balíčky programovacího jazyka Python a také funkce jednotlivých metod v rámci navrženého programu. Tento program byl vytvořen ve vývojovém prostředí (IDE) PyCharm a je rozdělen do následujících funkcí, jejich metod a dodatečných souborů:

- `main`
- `load_denoise`
- `rtx_denoise`
- `rtx_input`
- `rtx_record`
- `noisereduce_denoise`
- `subtraction_denoise`
- `stoi_module`
- `config`
- `requirements`

5.1.1 Funkce `main`

Hlavní funkcí celého programu je funkce `main`. V rámci této funkce je využita knihovna *EasyGui* pro vytvoření jednoduchého dialogového okna s výzvou pro výběr mezi funkcí odšumování pomocí vytvořených metod či objektivní analýzy zpracovaných signálů pomocí metriky STOI. V rámci tzv. *choiceboxu* program získává informaci o vybrané metodě a dále se odkazuje na funkce `load_denoise` v případě odšumování, či funkci `stoi_module` v případě objektivní analýzy.

5.1.2 Funkce `load_denoise`

Funkce `load_denoise` je nejkompexnější součástí programu a v jejím rámci probíhá volání a využití významné části celého navrženého systému. V rámci funkce je pevně stanovena hodnota `output_directory`, tedy výchozí výstupní cesta potřebná k uložení exportovaných souborů v rámci všech odšumovacích metod. Tato cesta je modifikovatelná pouze v rámci této funkce. Ta v počátku zavolání vytváří dialogové okno žádající uživatele o načtení složky se soubory ke zpracování. Informace o vybrané složce je uložena a program dále přistupuje k výzvě o volbu odšumovací metody. Na výběr je pak z již zmíněných metod RTX Voice, SUBTRACTION a NOISEREDUCE.

Program dále na základě výběru metod přistupuje k volání odpovídajících funkcí, tedy funkcí `rtx_denoise`, `subtraction_denoise` a `noisereduce_denoise`. Pomocí cyklů `for` je obsah vybrané složky prohledán za účelem nalezení všech souborů s koncovkou *.wav*, které jsou následně postupně zpracovány zmíněnými funkcemi. Funkcím je v argumentech předána informace o samotném zpracovávaném souboru a také požadované výstupní složce.

V případě volání funkce `noisereduce_denoise` je potřeba v argumentu funkce předat časový rámeček pro získání vzorku šumu, tedy informaci `tmin` a `tmax`. Toho je docíleno vytvořením dialogového okna, které uživatele požádá o zadání minimální a maximální hodnoty v sekundách. Ze signálu obsaženém v tomto časovém rozsahu nahrávky je poté metodou zjištěna povaha šumového pozadí k odstranění, viz 3.3. V rámci této funkce je také vytvořena ochrana proti zadání neplatných hodnot. Ošetřeny jsou zde případy kdy uživatel nezadal číslo, zadal minimum větší než maximum, nebo úplně vynechal zadání. O úspěšném dokončení odšumování v rámci jednotlivých metod je uživatel informován odpovídající konzolovou zprávou.

5.1.3 Funkce `rtx_denoise`

Funkce `rtx_denoise` je hlavní funkcí v rámci které je realizována metoda RTX Voice. Doplňujícími funkcemi pro její správný chod jsou pak soubory `rtx_input`

a `rtx_record`. Vstupními argumenty této funkce jsou `input_song_path` a také `output_directory`, tedy informace o cestě ke zpracovávanému souboru a informace o požadované výstupní složce.

Funkce automaticky načítá veškeré nahrávky s příponou `.wav`. Program dále kontroluje, zda v kořenovém adresáři programu existuje soubor `config.cfg`, který obsahuje informace o `mic_input` a `speaker_output` tedy informace vstupním a výstupním zařízením počítače. Za předpokladu že soubor neexistuje, program přistoupí k jeho vytvoření pomocí metody `choose_rtx_output` v rámci funkce `rtx_record`, viz 5.1.5. Jestliže soubor existuje a obsahuje požadované informace, funkce přistupuje k načtení vstupní nahrávky pomocí knihoven `wave` a `PyGame`.

Z načteného zvukového souboru je poté zjištěn vzorkovací kmitočet souboru a také samotný počet vzorků. Časová délka nahrávky je dále vypočítána jako jejich podíl. Tato informace je poté konvertována pro zobrazení ve formátu hodiny, minuty, sekundy pomocí metody `convert` viz 5.1.4. Do konzole jsou v této fázi vypsány informace o názvu právě zpracovávané nahrávky, jejím vzorkovacím kmitočtu a také její délce. Tyto informace jsou mimo jiné předány jako argument metodě `record`, která spolu s metodou `play` knihovny `PyGame` spouští nahrávání s pomocí nastaveného vstupního a výstupního zařízení dle souboru `config.cfg`.

Odšumování v rámci této metody jako jediné v navrženém programu probíhá v reálném čase. Pro správnou funkci metody je zapotřebí spuštěná a správně nastavená aplikace *RTX Voice* a *VB-Cable* viz 3.1 a 3.2. Ukázka zdrojového kódu této funkce se nachází v příloze B.3.

5.1.4 Funkce `rtx_input`

Funkce `rtx_input` slouží jako jedna ze dvou doplňujících funkcí pro funkci `rtx_denoise`. Na rozdíl od zbytku funkcí v programu obsahují tyto soubory řadu dalších metod nutných pro nahrávání, zpracování a export v reálném čase pomocí metody *RTX Voice*.

Tato funkce využívá zejména knihoven *PyGyme* či *configparser* a obsahuje metody:

- `convert`
- `check_cfg_exists`
- `cfg_create`
- `cfg_write`
- `cfg_read`
- `select_speaker_output`

Metoda `convert` slouží k převodu celkového času nahrávky do formátu hodiny, minuty, sekundy a navrácí ho pak hlavní funkci `rtx_denoise`.

Metoda `check_cfg_exists` slouží ke kontrole existence konfiguračního souboru `config.cfg` a navrácí boolean hodnotu, tedy `True` či `False`.

Metoda `cfg_create` slouží k vytvoření zmíněného konfiguračního souboru a založení sledovaného nastavení výstupního zvukového zařízení `speaker_output`. Ke čtení v takto vytvořeném souboru poté slouží metoda `cfg_read`, která je v případě neúspěšného načtení informace z konfiguračního souboru ošetřena chybovou zprávou uživateli.

Metoda `select_speaker_output` slouží k načtení výstupního zvukového zařízení programu *VB-Cable* potřebného pro nahrávání zpracovaného signálu metodou *RTX Voice*. Metoda v počátku kontroluje přítomnost konfiguračního souboru a v případě jeho absence přistupuje k jeho vytvoření pomocí zmíněné metody `cfg_create`. Program vytváří výčet dostupných audio zařízení se zaměřením na přítomnost výrazu „AUX“ popřípadě „Stereo Mix“ v názvu zařízení. Uživatel je poté v konzole vyzván k potvrzení správnosti výběru v případě úspěšného nalezení odpovídajícího kanálu pomocí volby „y/n“ (yes/no), tedy ano či ne. Za předpokladu, že nedojde k nalezení zařízení pomocí zmíněných hesel, je uživatel o této skutečnosti informován a vyzván k manuálnímu výběru zařízení pomocí zadání odpovídajícího čísla zařízení do konzole.

```
Config file not found... creating
--- SPEAKER OUTPUT ---

Could not find VB Audio AUX

Audio Options:
0. Speakers (NVIDIA RTX Voice)
1. CABLE Input (VB-Audio Virtual Cable)
2. Speakers/Headphones (Realtek(R) Audio)

If VB Audio is not found here, please check it is not disabled.

Please select the VB Audio speaker input: |
```

Obr. 5.1: Volba vstupního kanálu v rámci souboru `config`

Na obrázku 5.1 je vidět ukázka konzolové zprávy pro výběr požadovaného kanálu. Vybrané zařízení je poté předáno k uložení do konfiguračního souboru `config.cfg` pomocí metody `cfg_write`. Ukázka zdrojového kódu této metody se nachází v příloze B.1.

5.1.5 Funkce `rtx_record`

Funkce `rtx_record` je druhou z dvojice dodatečných funkcí pro funkci `rtx_denoise`. Obsahuje metody `choose_rtx_output` a `record`.

První ze zmíněných metod slouží ke zjištění výstupního kanálu z programu *RTX Voice*. Za využití knihovny *PyAudio* je načten výčet dostupných výstupních zvukových zařízení a v jejich rámci je hledáno zařízení obsahující název „NVIDIA RTX Voice“. Pokud je programem toto výstupní zařízení nalezeno, je uživatel informován o jeho využití a následně je pomocí metody `cfg_write` uloženo do souboru `config.cfg`. Za předpokladu, že k úspěšnému nalezení dle zmíněného hesla nedojde, je uživatel vyzván k manuálnímu výběru obdobně jako v případě metody `select_speaker_output`.

Druhou z metod v rámci funkce je metoda `record`. Ta slouží k nahrávání výstupu z programu *RTX Voice*, tedy již k nahrávání zpracovaného signálu v reálném čase. Argumenty funkce jsou:

- časová délka (implicitně 10s)
- výstupní název nahrávky (předán dle metody)
- výstupní zařízení k nahrávání (dle konfiguračního souboru)
- vzorkovací kmitočet (implicitně 48kHz)

Metoda dále implicitně nastavuje hodnoty jako jsou datový formát (UInt16), počet kanálů (1) či velikost okna pro nahrávání (1024). Metoda po získání hodnot nahrává výstup programu *RTX Voice* a pomocí knihoven *PyAudio* a *Wave* vytváří odpovídající soubory `.wav` s požadovanými parametry.

O úspěšném zpracování každé z nahrávek je uživatel následně informován pomocí konzolové zprávy. Ukázka navrženého zdrojového kódu této metody se nachází v příloze B.2

5.1.6 Funkce `noisereduce_denoise`

V rámci této funkce dochází k odstranění šumu pomocí knihovny *Noisereduce*. Obsahuje pouze jednu metodu `noisereduce_denoise` s argumenty `t_min`, `t_max`, `input_song_path` a `output_directory`.

Za využití knihovny *Soundfile* je načtena nahrávka z dodané souborové cesty, která je poté načtena spolu se zjištěním jejího vzorkovacího kmitočtu. V rámci metody je dále zjištěna maximální a minimální časová hodnota pro stanovení rozsahu šumové složky v nahrávce. Na jejím základě je poté v dalším kroku zavolána funkce samotné knihovny, která na základě zjištěného šumového úseku odstraňuje odpovídající šumové pozadí ze zpracovávané nahrávky. Tento proces na rozdíl od metody *RTX Voice* neprobíhá v reálném čase a k odšumění každé nahrávky tak dochází v rámci několika vteřin.

5.1.7 Funkce `subtraction_denoise`

Funkce `subtraction_denoise` implementuje třetí z navržených metod a to metodu SUBTRACTION. Na rozdíl od ostatních funkcí je v případě využití metody subtraktivního odšumování kromě samotné nahrávky zatížené šumem zapotřebí dodat také nahrávku obsahující převážně nežádoucí šum. Pro správnou funkci programu je zapotřebí nahrávek, které jsou souhlasné délky a časové korelace.

Argumenty jediné stejnojmenné funkční metody `subtraction_denoise` jsou souborová cesta k odšumované nahrávce a její odpovídající nahrávce šumového pozadí. Stejně jako u ostatních metod je zapotřebí také informace o požadované složce pro uložení zpracované nahrávky.

Principu samotné funkce se blíže věnuje kapitola 3.4. V následující ukázce zdrojového kódu 5.1 se nachází její programová realizace.

Výpis 5.1: Zdrojový kód metody SUBTRACTION

```
def subtraction_denoise(input_song_path, input_noise_path,
                        output_directory):

    input, sr = sf.read(input_song_path)
    input_stft = librosa.stft(input)
    mag = np.abs(input_stft)
    angle = np.angle(input_stft)
    b = np.exp(1.0j * angle)

    noise, nsr = sf.read(input_noise_path)
    noise_stft = librosa.stft(noise)
    noise_stft_abs = np.abs(noise_stft)
    mean_ns = np.mean(noise_stft_abs, axis=1)

    reshape = mag - mean_ns.reshape((mean_ns.shape[0], 1))
    phased = reshape * b
    y = librosa.istft(phased)

    write(output_directory + "sub_" + os.path.basename(input_song_path)
          , sr, y.astype(np.float32))
```

5.1.8 Funkce `stoi_module`

Funkce `stoi_module` je realizací programové funkce SCORE, tedy realizace objektivní metody hodnocení nahrávek STOI. Obsahuje metodu `pystoi` a v ní vnořenou metodu `pystoi_write`.

Po zavolání funkce je uživatel vyzván k výběru složky s originálními a poté odšuměnými nahrávkami. Algoritmus poté sám zjišťuje přítomnost zpracovaných na-

hrávek na základě odpovídajících předpon v jejich souborových názvech. Ke každé této nahrávce je v rámci cyklu nalezena odpovídající nezpracovaná (zašuměná) nahrávka. Nahrávky jsou načteny pomocí knihovny *librosa* a je tak zároveň zjištěn jejich vzorkovací kmitočet. Následně je zavolána funkce knihovny *PyStoi* jejímž vstupem je dvojice takto spárovaných nahrávek. Pro každou nahrávku zpracovanou některou z metod v dané složce je poté v rámci funkce vypočítána hodnota d pomocí srovnání s nahrávkami originálními.

Podmínkou pro úspěšné srovnání nahrávek touto metodou je jejich správná časová korelace a totožná délka, tedy stejný vzorkovací kmitočet a počet vzorků. Tato podmínka je v rámci navržené funkce ošetřena pomocí načítání shodné délky nahrávek. Toho je docíleno načtením takové délky obou nahrávek, která odpovídá délce kratší z nich. Tohoto je využito zejména u funkce RTX Voice, kde se výsledná nahrávka (vzhledem ke způsobu zpracování) v počtu vzorků ne vždy shoduje s nahrávkou originální. Tento rozdíl je v řádech maximálně desítek vzorků, avšak pomocí zmíněné metody je dosaženo vyloučení vlivu této potenciální neshody na hodnocení nahrávek. V následujícím příkladu zdrojového kódu se nachází ukázka srovnání délky nahrávek pro hodnocení metody RTX Voice pomocí STOI.

Výpis 5.2: Srovnání nahrávek pomocí metody STOI

```

if glob.glob(get_directory_denoised + "\\rtx_*"):
    for clean_file in glob.glob(get_directory_clean + "/*.wav"):
        denoised_file = get_directory_denoised + "\\rtx_" + os.path.
            basename(clean_file)
        denoised_file_load, fs = sf.read(denoised_file)
        s_duration = librosa.get_duration(denoised_file_load, sr=fs)
        clean_file_load_reduced, fs = librosa.load(clean_file,
            duration=s_duration, sr=fs)
        d = stoi(clean_file_load_reduced, denoised_file_load, fs,
            extended=False)
        pystoi_write(d, denoised_file)

```

V rámci metody `pystoi_write` je pro každou nahrávku zpracovanou touto funkcí zapsána hodnota d do textového souboru `pystoi_log.txt`. Za předpokladu, že tento soubor neexistuje, je funkcí vytvořen. Pokud je pak funkce opětovně volána, je obsah tohoto souboru přemazán a nahrazen novými výsledky.

5.1.9 Soubor `config.cfg`

Tento jednoduchý konfigurační soubor je potřebný pro správnou funkci metody RTX Voice. Je využit v rámci funkcí `rtx_record` a `rtx_input` a obsahuje informaci o přednastavených vstupních a výstupních zařízeních programu *VB-Cable* a *RTX Voice*. V případě jeho absence v kořenovém adresáři programu je pomocí zmíněných funkcí vytvořen a uživatelem nastaven viz sekce 5.1.4 a 5.1.5.

5.1.10 Soubor `requirements.txt`

Soubor `requirements.txt` slouží jako dodatečný soubor ke snazšímu spuštění a nastavení celého programu. V případě spouštění programu v rámci IDE PyCharm slouží jako seznam požadovaných knihoven a rozšíření potřebných k jeho spuštění. PyCharm uživateli na základě tohoto souboru v kořenovém adresáři programu nabídne automatickou aktualizaci či stažení požadovaných rozšíření a uživatel tak nemusí tento proces provádět manuálně.

Seznam požadovaných rozšíření a jejich verzí obsahuje tyto knihovny jazyka Python:

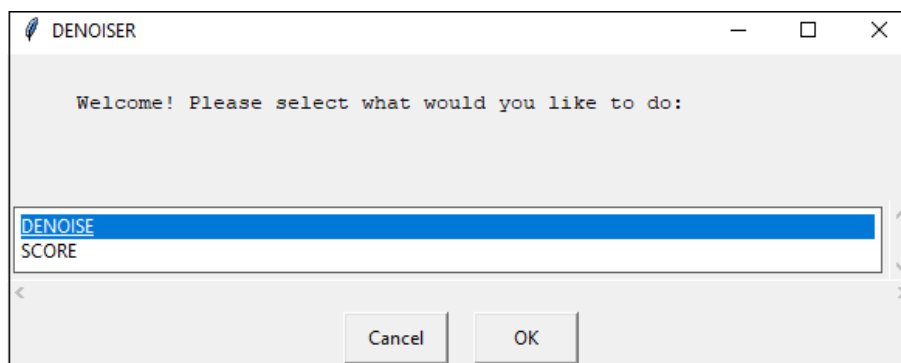
- EasyGUI, v. 0.98.1
- Librosa, v. 0.8.0
- Noisereduce v. 1.1.0
- Numpy 1.20.1
- PyAudio 0.2.11
- PySTOI 0.3.3
- Pygame 2.0.0.dev8
- Scipy 1.6.1
- Soundfile 0.10.3.post1
- Wheel 0.35.1

6 Průvodce navrženým programem

Tato kapitola slouží jako průvodce vytvořeným programem po jeho spuštění uživatelem, zejména s ohledem na spuštění v rámci vývojového prostředí (IDE). Pozornost je zde věnována popisu jednotlivých funkcí a možností v rámci navrženého grafického rozhraní se zaměřením na souvislosti s navrženými funkčními součástmi programu již popsány v kapitole 5.1.

6.1 Menu základních funkcí

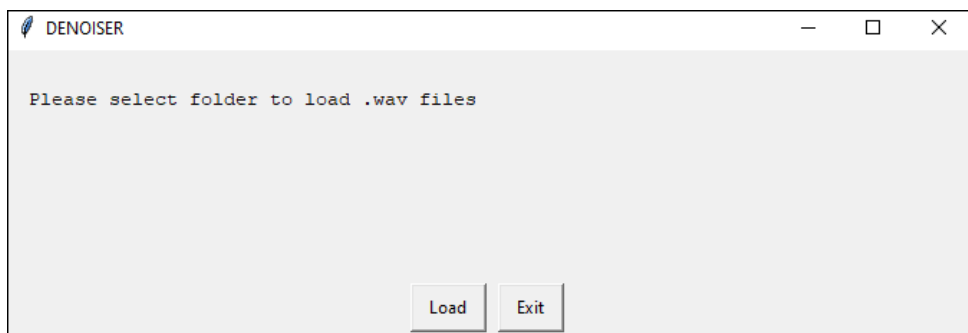
Hlavním souborem programu je soubor `main`, který po spuštění otevře jednoduché grafické menu nabízející dvojici možností `DENOISE` a `SCORE`. Funkce `main` dále využívá dalších navržených funkcí `load_denoise` a `stoi_module` na základě volby funkce. Na základě volby uživatele program přechází k odšumování nahrávek pomocí jednoho až tří vybraných algoritmů (volba `DENOISE`), nebo k hodnocení kvality odšuměných nahrávek pomocí metriky `STOI` (volba `SCORE`). Na obrázku 6.1 se nachází ukázka úvodního menu programu s těmito volbami.



Obr. 6.1: Menu pro volbu funkcí

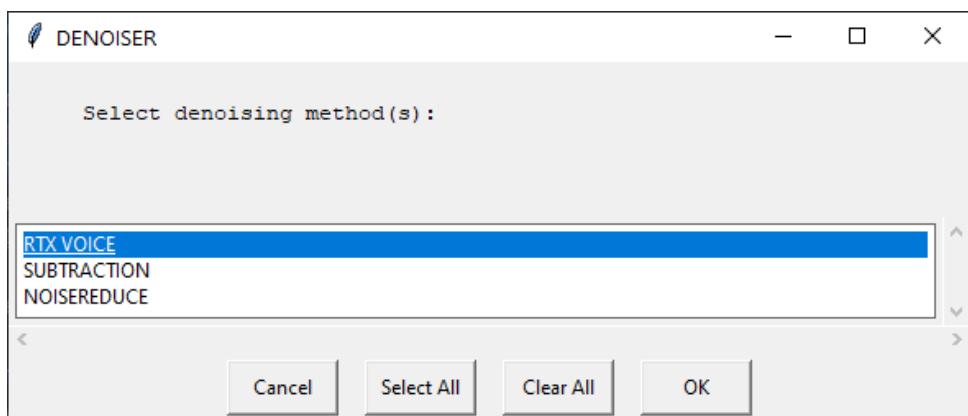
6.2 Funkce DENOISE

Pokud se uživatel rozhodne pro funkci `DENOISE`, je požádán o výběr adresáře z něhož budou postupně načítány a odšumovány veškeré obsažené nahrávky ve formátu `.wav`. Ukázka menu pro načtení nahrávek se nachází na obrázku 6.2.



Obr. 6.2: Menu pro načtení nahrávek k dalšímu zpracování

Po vybrání nahrávek ke zpracování je uživatel vyzván k výběru metody odstranění šumu. Vytvořený software nabízí trojici metod odšumování a to: RTX Voice, SUBTRACTION a NOISEREDUCE. Po vybrání libovolného množství metod program přechází ke zpracování všech nahrávek obsažených ve zvolené složce pomocí vybraných algoritmů. V případě volby funkce RTX Voice toto zpracování probíhá jako první a to (vzhledem k povaze metody) v reálném čase. Nutné je využití programů *VB-Cable* a *RTX Voice*, které musí být korektně spuštěny a nastaveny pro správný průběh této metody. Doporučené nastavení využitě pro testování na vývojovém zařízení se nachází v příloženém textovém souboru *README.txt*. Ukázka uživatelského menu pro výběr metody se nachází na obr. 6.3.



Obr. 6.3: Menu pro volbu metody odstranění hluku

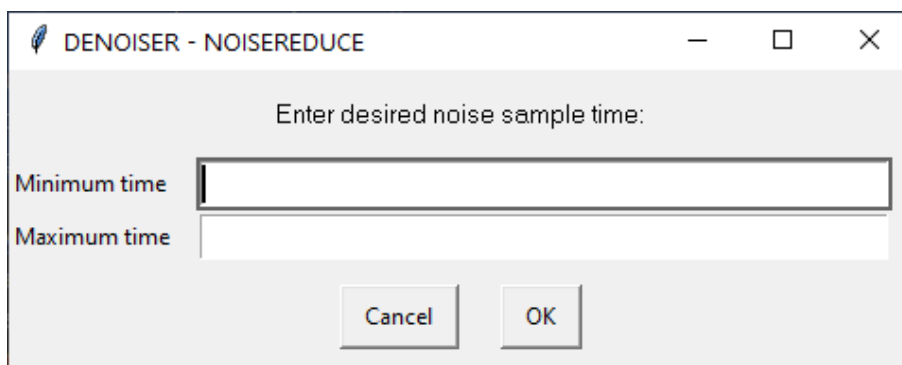
6.2.1 Metoda RTX Voice

V případě volby RTX Voice hlavní soubor `main` využívá funkce `rtx_denoise`, pomocí které dochází k načtení, postupnému odhlučnění a následnému uložení nových nahrávek. Funkce `rtx_input` nejprve načítá data o vstupních a výstupních (I/O) zařízeních počítače uložených v souboru `config.cfg`. Pokud tento soubor neexistuje, či doposud nebyl vytvořen, je uživatel informován o jeho neúspěšném načtení a následně vyzván k výběru výstupního zvukového zařízení virtuálního kabelu odpovídajícího nastavení RTX Voice. Pokud je vybrán výstup odpovídající výstupu virtuálního kabelu, program následně plynule pokračuje k odšumování a vytvoření nových nahrávek pomocí funkce `rtx_record`. Pro budoucí využití navrženého programu na stejném zařízení je toto I/O nastavení uchováno v souboru `config.cfg` a tím pádem ho již není třeba provádět znovu.

V rámci funkce `rtx_record` dochází k postupnému nahrávání zvukového signálu z výstupu RTX Voice v reálném čase a dále k tvorbě a uložení nově odšuměné nahrávky s předponou `rtx` do kmenového adresáře programu. Pomocí konzolových zpráv je uživatel během zpracování nahrávek průběžně informován o názvu právě zpracovávaného souboru, jeho vzorkovacím kmitočtu a také o jeho délce. Po úspěšném průběhu celého procesu je uživatel informován zprávou o jeho dokončení.

6.2.2 Metoda NOISEREDUCE

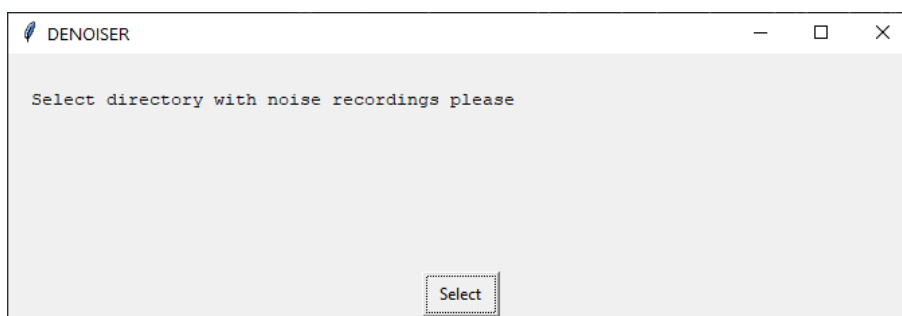
V případě volby funkce NOISEREDUCE je pomocí menu s dvojicí vstupních parametrů uživatel vyzván k zadání požadovaného časového rozsahu z nahrávky v sekundách, na jehož základě bude stanoven vzorek šumu k následnému odstranění ze zpracovávaných nahrávky. Ukázka tohoto menu se nachází na obr. 6.4.



Obr. 6.4: Menu pro výběr úseku šumu

6.2.3 Metoda SUBTRACTION

V případě volby funkce SUBTRACTION je zapotřebí využít párované dvojice nahrávek. Uživatel je nejprve vyzván k vybrání složky obsahující nahrávky šumu a až poté k výběru složky s nahrávkami k odšumění. Bez načtení takto spárovaných nahrávek není možné tuto metodu využít. Ukázka menu pro volbu šumové nahrávky se nachází na obr. 6.5.



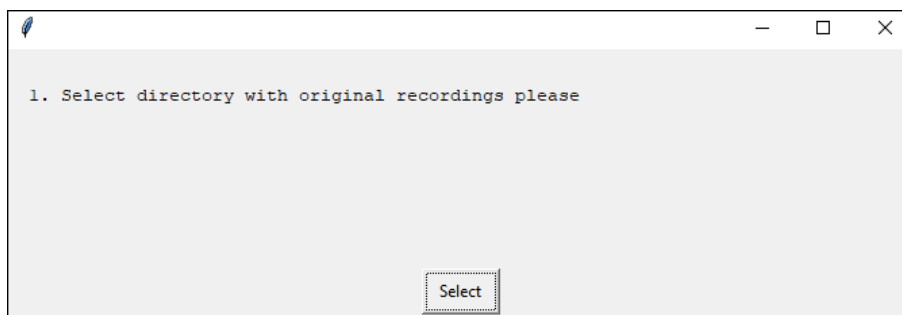
Obr. 6.5: Menu pro výběr šumové nahrávky

Dokončení odšumování pomocí metod NOISEREDUCE a SUBTRACTION probíhá zpravidla výrazně rychleji než pomocí metody RTX Voice, jelikož na rozdíl od ní není nutné nahrávky zpracovávat v reálném čase. O dokončení obou metod je uživatel stejně jako v případě metody první informován konzolovou zprávou.

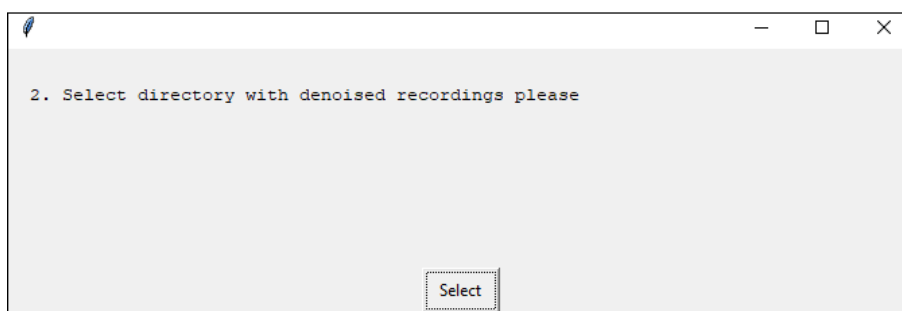
6.3 Funkce SCORE

Pomocí funkce SCORE je možné provést hodnocení srozumitelnosti odšuměné nahrávky za využití metriky STOI. K ní je kromě zpracované nahrávky zapotřebí také nahrávka originální. Pro tuto funkci je využito upravené externí knihovny jazyka Python *PyStoi*. Po zvolení funkce SCORE v menu je uživatel vyzván k výběru adresáře obsahujícího původní nahrávky viz obr. 6.6. Následně je vyzván k výběru složky s nahrávkami již zpracovanými viz obr. 6.7.

Pokud se v adresáři s již zpracovanými nahrávkami nachází odpovídající soubory vygenerované některým z implementovaných algoritmů, tedy soubory s předponou *rtx*, *sub*, *nr* či externě generované soubory z programu Reaper, tedy s předponou *daw* ve formátu .wav, jsou načteny a srovnány s odpovídajícími originálními nahrávkami. V případě, že se v adresáři nachází soubory pouze s některou z těchto předpon, metoda STOI je vypočítána pouze pro tyto funkce a úspěšné dokončení jednotlivých metod ohlásí pomocí konzolové zprávy.



Obr. 6.6: Okno pro načtení originálních nahrávek



Obr. 6.7: Okno pro načtení zpracovaných nahrávek

Pro správnou funkci programu je nutné, aby se délky originálních nahrávek přesně shodovaly s délkami již zpracovaných. V případě nedodržení tohoto požadavku je nedostatek automaticky ošetřen tak, že je z obou nahrávek načteno jen tolik vzorků, kolik odpovídá délce kratší nahrávky. To u většiny nahrávek znamená buď žádný, nebo rozdíl maximálně několika desítek vzorků. Tímto způsobem je zajištěna maximální korelace obou nahrávek a nejpřesnější vyhodnocení v rámci metriky.

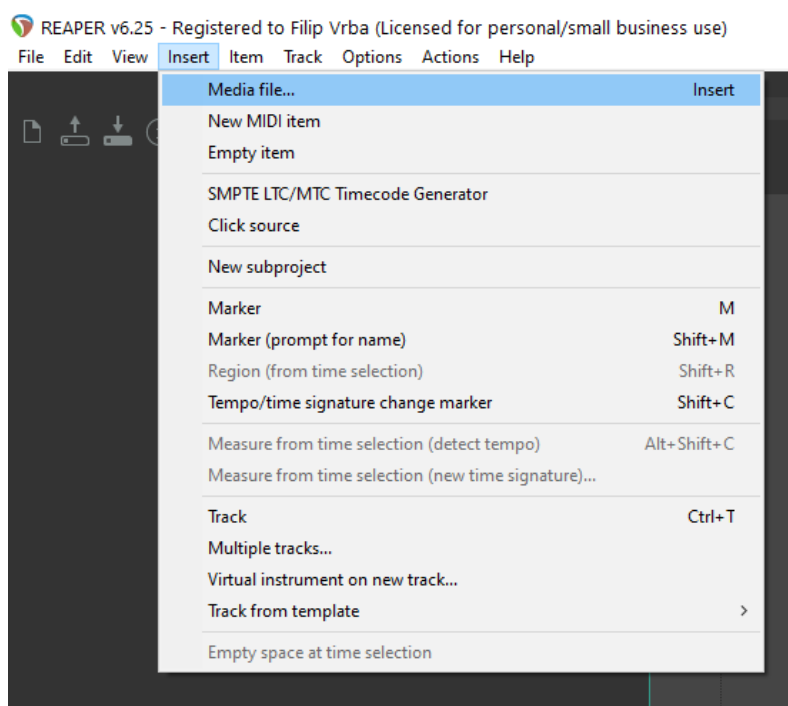
Následně je programem vytvořen soubor `pystoi.log`, který obsahuje název zpracované nahrávky a také odpovídající výslednou hodnotu d metriky STOI (viz kap. 3.7) pro každou z nich. Při opětovném spuštění funkce SCORE je tento soubor a informace v něm pro přehlednost přepsán.

7 Odšumování pomocí DAW

Doplňující metodou je v této práci odstranění šumu pomocí DAW, konkrétně za pomoci softwaru Reaper. Tento program zdarma nabízí 30denní zkušební verzi a tuto metodu tak lze využít bez nutnosti finanční investice, využití externích VST či dalších komerčních programů. Problematice DAW a konkrétně programu Reaper se blíže věnuje kapitola 3.6. Následující sekce se věnují postupu pro odšumění nahrávek metodou DAW využitím v rámci této práce.

7.1 Nastavení DAW a využitého VST

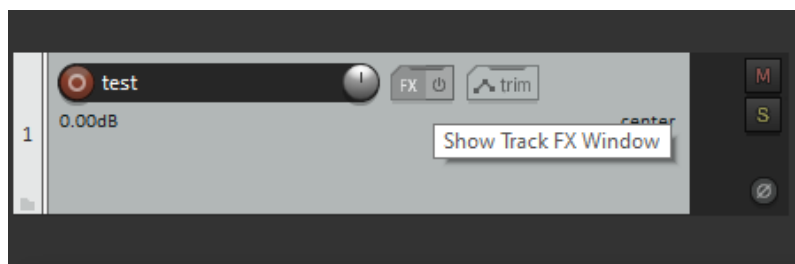
Prvním krokem po spuštění programu je načtení referenční nahrávky zatížené šumem. Toho lze docílit jednoduchým přetažením souboru do hlavního okna programu, či pomocí klávesové zkratky *Insert*, či volbě *Insert* v okně programu s následným výběrem *Media file* viz obr. 7.1.



Obr. 7.1: Okno pro načtení referenční nahrávky

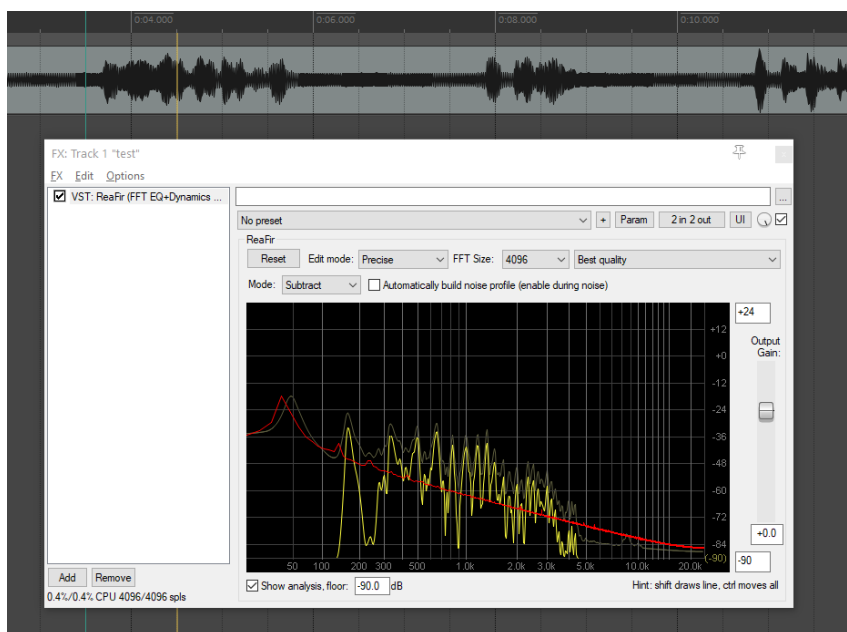
Dalším krokem po načtení nahrávky do programu je přiřazení a nastavení VST pluginu *ReaFIR*. Toho docílíme zakliknutím tlačítka *FX* na panelu kanálu vytvořeného spolu s načtením nahrávky.

V nově zobrazeném okně se seznamem instalovaných VST vybereme pomocí volby *VST:ReaFir*. Nyní je potřeba v novém okně pluginu vybrat *Mode: Subtract* a



Obr. 7.2: Volba FX na panelu kanálu nahrávky

zaškrtnout volbu *Automatically build noise profile*. Od tohoto momentu bude program pomocí pluginu schopný vytvářet šumový profil (obálku) na základě frekvenčního spektra signálu přehrávaného ve sledovaném kanálu. Následně je potřeba v hlavním okně vybrat úsek nahrávky obsahující typický šum pro daný typ nahrávek. Z tohoto místa je zapotřebí krátkodobě spustit přehrávání. V okně pluginu se tímto způsobem vytvoří práh šumové obálky. Pro dosažení nejlepších výsledků je zapotřebí zachovat krátké přehrávání pouze v oblasti nežádoucího šumu. Po odkliknutí volby *Automatically build noise profile* je možné testovací nahrávku kontrolně přehrát a ověřit si tak správnost provedení metody. Pokud nebyl v rámci tvorby šumové obálky přehrán užitečný signál, měl by být šum pozadí v testované nahrávce výrazně potlačen. Modelová ukázka správně nastaveného pluginu *ReaFIR* se nachází na obr. 7.3.

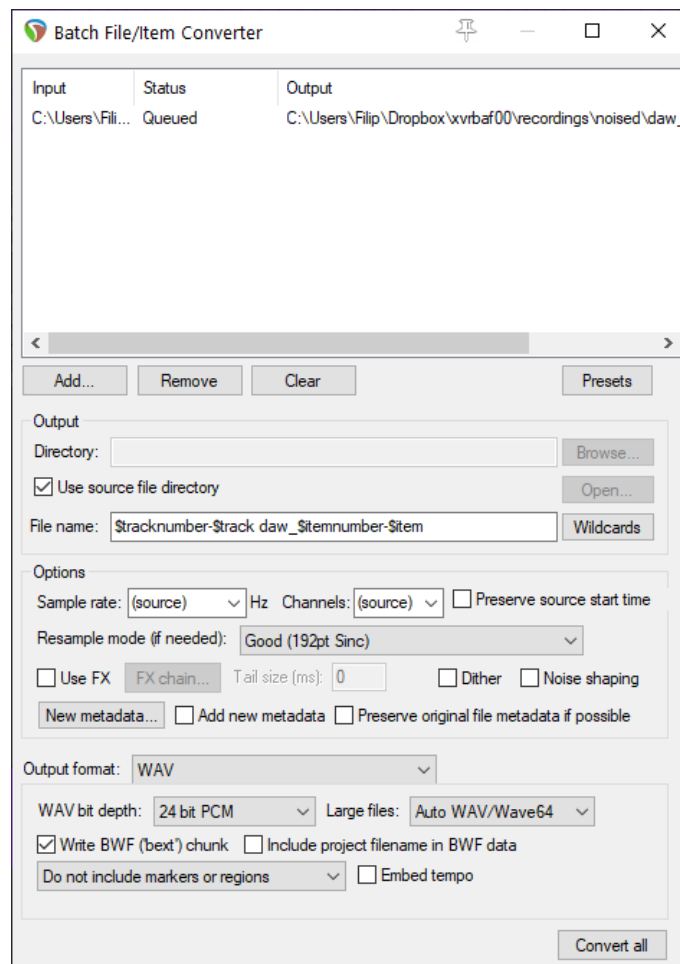


Obr. 7.3: Správně nastavený plugin ReaFIR

Opětovným spuštěním nahrávky je možné míru a kvalitu odšumění subjektivně posoudit. V této fázi je zapotřebí využít možnosti uložení takto vytvořeného přednastavení pluginu VST ReaFIR. Toho docílíme stisknutím tlačítka „+“ na panelu pluginu a následnou volbou *Save preset*. Takto vytvořené přednastavení bude dále využito k hromadnému odšumění libovolného množství nahrávek. Vzhledem ke způsobu tvorby šumové obálky na základě typického šumového pozadí se tato metoda hodí především pro nahrávky obsahující stejný, nebo velmi podobný zdroj nežádoucího hluku pozadí.

7.2 Hromadné odšumění a export nahrávek

V tomto kroku je hromadně načteno libovolné množství nahrávek pro odšumění. Zde je využito funkce *Batch File/Item Converter* dostupné v záložce *File* či pod klávesovou zkratkou *Ctrl+Shift+F*.



Obr. 7.4: Okno pro hromadné zpracování nahrávek

Pomocí tlačítka *Add* se otevře okno pro výběr nahrávek k odšumění. V rámci tohoto okna je dále možné pomocí volby *Directory* vybrat cílovou složku pro uložení zpracovaných nahrávek. V řádku *File name* je možné specifikovat formát pojmenování a číslování nově vzniklých nahrávek. V rámci nastavení v ukázce je předem připraven export s předponou *daw*, který tak nahrávky formátuje na snadné budoucí vyhodnocení metrikou STOI. Na ukončení hromadného odšumování je uživatel upozorněn novým oknem s programovou hláškou.

8 Výsledky práce

V následující kapitole se nachází výsledky vyhodnocení kvality nově získaných nahrávek pomocí zpracování v rámci navržených metod RTX Voice, NOISEREDUCE, SUBTRACTION a DAW. Jedná se o výsledky testování těchto metod na databázi 24 řečových nahrávek z přístroje MR, které byly zpracovány metodami RTX Voice, NOISEREDUCE a DAW a také výsledky testování všech navržených metod na nahrávkách ze samostatného nahrávání, včetně využití metody SUBTRACTION.

V první tabulce se nachází výsledky pro poskytnutou databázi, v druhé tabulce se nachází hodnocení nahrávek pořízených ve výzkumném centru CEITEC. Všechny tyto nahrávky jsou hodnoceny pomocí metriky STOI v rámci navrženého programu. Na základě jejích výsledků je provedena jednoduchá statistická pomocí vybraných statistických metod jako: aritmetický průměr, směrodatná odchylka a další.

V rámci této práce byla dále provedena subjektivní poslechová analýza na základě poslechových testů. Té se věnuje druhá sekce této kapitoly.

8.1 Výsledky metriky STOI

V rámci práce byla vytvořena tabulka výsledků metriky STOI všech zpracovaných nahrávek a jejich odpovídajících hodnot d . Soubor *stoi_results.xlsx* se nachází na příloženém CD viz příloha C. Dle předpon *rtx*, *nr*, *sub* a *daw* lze rozlišit o kterou z metod se v případě daného výsledku se jedná.

Tab. 8.1: Statistické výsledky metriky STOI nahrávek z databáze

	$\bar{x}(d)$	$\tilde{x}(d)$	$\sigma(d)$	$\min(d)$	$\max(d)$	$D(d)$
RTX Voice	0,206	0,201	0,089	0,067	0,416	0,008
Noisereduce	0,736	0,743	0,092	0,612	0,947	0,008
Subtraction	-	-	-	-	-	-
DAW	0,881	0,877	0,025	0,837	0,949	0,001

V rámci tabulky 8.1 jsou shrnuty výsledky statistické analýzy výsledných hodnot d metriky STOI pro všech 24 řečových nahrávek z přístroje MR. V rámci využitých metod zde chybí metoda SUBTRACTION z důvodu absence potřebné odpovídající šumové nahrávky k nahrávkám z využití databáze. Výsledky využití této metody je možné vidět v následující tabulce v rámci zpracování nahrávek z výzkumného centra CEITEC.

V rámci tabulky 8.2 jsou shrnuty výsledky statistické analýzy výsledných hodnot d metriky STOI pro samostatné nahrávky pořízené ve výzkumném centru CEITEC.

Tab. 8.2: Statistické výsledky metriky STOI nahrávek z centra CEITEC

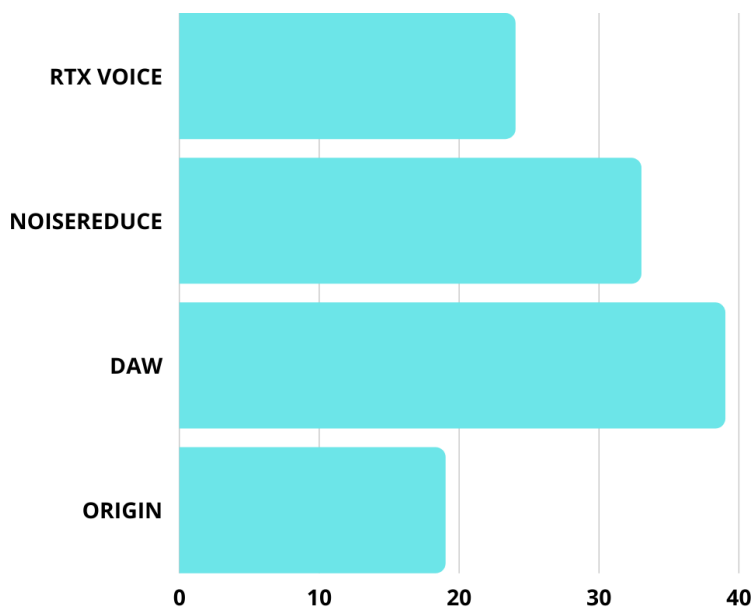
	$\bar{x}(d)$	$\tilde{x}(d)$	$\sigma(d)$	$\min(d)$	$\max(d)$	$D(d)$
RTX Voice	0,221	0,168	0,100	0,142	0,396	0,010
Noisereduce	0,629	0,644	0,060	0,544	0,698	0,004
Subtraction	0,758	0,760	0,014	0,732	0,772	0,000
DAW	0,219	0,236	0,050	0,169	0,285	0,003

Použité metody v obou případech jsou: $\bar{x}(d)$ (aritmetický průměr), $\tilde{x}(d)$ (medián), $\sigma(d)$ (směrodatná odchylka), minimum, maximum a $D(d)$ (rozptyl). Význam čísla d a metrika STOI jako taková je blíže popsána v kapitole 3.7.

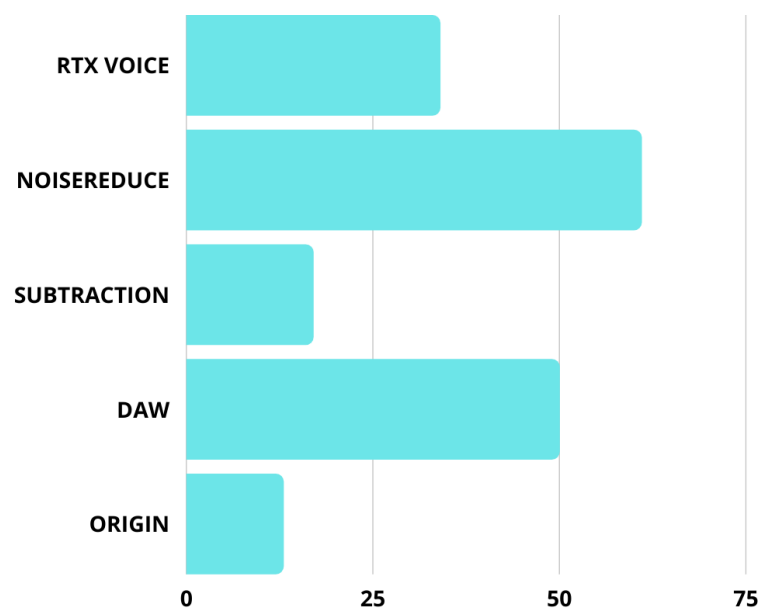
8.2 Výsledky subjektivní analýzy

V rámci následující sekce jsou prezentovány výsledky subjektivní analýzy založené na poslechových testech. Metodika a realizace této analýzy je blíže popsána v kapitole 3.8.

V rámci grafů 8.1 a 8.2 jsou vyneseny výsledky poslechových testů pro jednotlivé metody. Dané hodnoty v grafech reprezentují celkový počet odpovědí v testu ve prospěch dané metody. Každá z testovaných metod je v celkovém počtu hodnocených nahrávek zastoupena stejným počtem zpracovaných nahrávek.

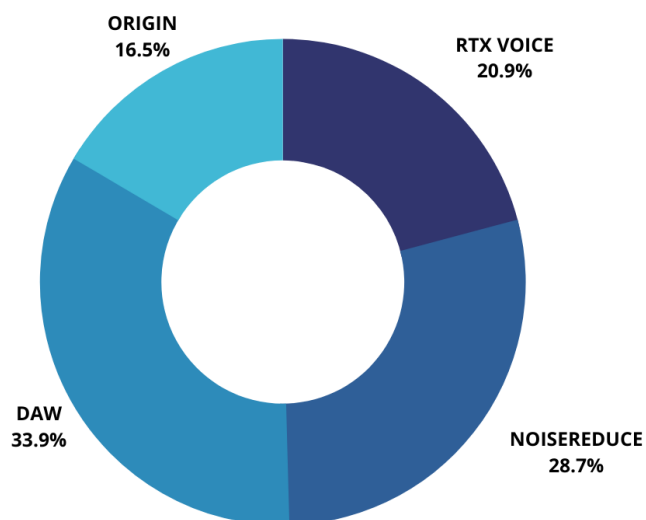


Obr. 8.1: Výsledky subjektivní analýzy – DB

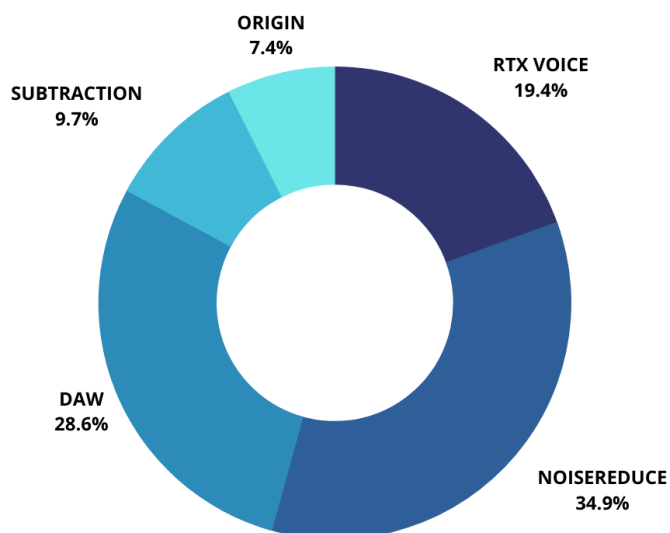


Obr. 8.2: Výsledky subjektivní analýzy – CEITEC

V grafech 8.3 a jsou získané výsledky prezentovány jakožto jejich procentuální podíl z celkového počtu získaných odpovědí.



Obr. 8.3: Výsledky subjektivní analýzy v procentech – DB



Obr. 8.4: Výsledky subjektivní analýzy v procentech – CEITEC

V rámci poslechových testů bylo celkově získáno hodnocení od více než 20 nezávislých respondentů. Tato analýza slouží jako dodatková metoda pro objektivní hodnocení navržených metod. Poslechové testy bylo možné realizovat až po provedení nahrávání v přístroji MR ve výzkumném centru CEITEC. To nebylo vzhledem k pandemické situaci a pozdní době rozvolnění podmínek v období psaní této práce provedeno v dostatečném počtu a výsledky této metody jsou tak spíše orientační.

9 Diskuze

Tato kapitola se věnuje sumarizaci a zhodnocení výsledků získaných metodami objektivní a subjektivní analýzy v rámci testování zpracovaných nahrávek navrženým programem. Pozornost je zde věnována vysvětlení výsledků a poukázání na zjištěné silné a slabé stránky navržených metod a zvolených způsobů analýzy výsledků.

9.1 Objektivní analýza

V rámci této analýzy bylo využito metriky STOI pro objektivní zhodnocení srozumitelnosti nahrávek po odstranění šumu pozadí. Tato sekce se zabývá zhodnocením statistických výsledků získaných touto metrikou.

9.1.1 Výsledky STOI - DB

V případě hodnocení nahrávek z databáze (DB) dosáhla nejvyššího hodnocení v rámci této metriky metoda DAW, tedy metoda odstranění šumu za využití VST pluginu s průměrnou hodnotou $d = 0,881$. Velmi dobrých výsledků dosáhla také metoda NOISEREDUCE, dosahující průměrné hodnoty $d = 0,736$. Naproti tomu metoda RTX Voice zde dosahuje výrazně menších hodnot. To je pravděpodobně způsobeno faktem, že tato metoda dosahuje subjektivně téměř dokonalého potlačení šumu pozadí za cenu poměrně výrazného subjektivního snížení srozumitelnosti řeči oproti ostatním metodám. Dalším faktorem může být menší míra časového zpoždění nahrávek zpracovaných touto metou oproti nahrávkám originálním. Tento občasný jev, který se projevuje především u zpracování delších nahrávek nemá vliv na subjektivní vnímání kvality odšumění či nahrávky jako takové, může mít ovšem vliv na výsledek metriky STOI. V případě potřeby správné časové korelace nahrávek je proto vhodnější využít některou z ostatních navržených metod.

9.1.2 Výsledky STOI - CEITEC

V rámci hodnocení nahrávek získaných při nahrávání ve výzkumném centru CEITEC dosáhla nejlepšího výsledku metoda SUBTRACTION s průměrnou hodnotou $d = 0,758$. Druhou metodou s příznivými výsledky je metoda NOISEREDUCE, dosahující průměrné hodnoty $d = 0,629$. Dvojice zbývajících metod získala výrazně nižší hodnocení s hodnotami d pod 0,250. Tento výsledek je pravděpodobně způsoben několika faktory. V případě metody DAW dochází k získávání odečítané šumové složky na základě sledování stacionárního hluku v pozadí nahrávky. Při nahrávání řeči v rámci této sady nahrávek byly získány promluvy s výrazně nižším odstupem

užitečného signálu od hluku pozadí naproti databázi DB. Ve zpracované nahrávce tak dochází k větší míře filtrace užitečného signálu spolu s nežádoucím šumem a dochází tak k následnému zhoršení srozumitelnosti výsledné nahrávky. Vzhledem k tomuto způsobu odstranění hluku jsou pravděpodobně nahrávky zpracované metodou DAW hodnoceny metrikou STOI hůře než nahrávky z databáze (DB). V případě metody RTX Voice způsobují horší výsledek STOI pravděpodobně stejné faktory jako v případě nahrávek databáze (DB).

9.2 Subjektivní analýza

V případě subjektivní analýzy bylo využito dvojice poslechových testů pro každou z testovaných sad nahrávek. V rámci této části jsou zhodnoceny výsledky získané na základě odpovědí oslovených respondentů získaných pomocí vytvořených online dotazníků.

9.2.1 Výsledky subjektivní analýzy - DB

Dle poslechových testů zpracovaných nahrávek z databáze byla nejlépe hodnocena metoda DAW, s celkovým zastoupením 33,9%. Se zastoupením 28,7% následuje metoda NOISEREDUCE a 20,9% hlasů získala metoda RTX Voice. S 16,5% hlasů skončil vzorek originální nahrávky. Tyto výsledky tak po odstupňování úspěšnosti odpovídají hodnocení v rámci objektivní analýzy metrikou STOI. V případě metody RTX je výsledek s nejnižším hodnocením pravděpodobně způsoben stejnou příčinou jako u metody objektivní, tedy celkovou nižší srozumitelností nahrávky způsobené silným potlačením hluku zasahujícím do oblasti řeči.

Poměrně velké zastoupení má ve výsledcích i samotná nezpracovaná nahrávka (16,5%). To je pravděpodobně způsobeno faktem, že i přes silné hlukové pozadí nahrávky se může nezávislým respondentům originální záznam jevit „srozumitelnější“ než nahrávky zpracované. U těch se totiž při odstranění šumu téměř vždy v menší či větší míře zasáhne do frekvenční oblasti užitečného signálu, tedy řeči a její srozumitelnosti. Takto způsobené snížení srozumitelnosti se může některým respondentům mnohdy zdát závadnější než samotný hluk pozadí.

9.2.2 Výsledky subjektivní analýzy - CEITEC

V případě poslechových testů nahrávek pořizovaných ve výzkumném centru CEITEC dosáhla nejlepších výsledků metoda NOISEREDUCE s celkově 34,9% hlasů. Podobně jako v případě nahrávek z databáze následuje metoda DAW (28,6%) a metoda RTX (19,4%). Navíc se tu objevuje metoda SUBTRACTION, která ovšem

získala pouze 9,7% všech hlasů, tedy jen o dvě procenta více než originální, tedy nezpracovaná nahrávka (7,4%). Tento výsledek je tak částečně v rozporu s výsledkem objektivní analýzy metrikou STOI, kde si tato metoda naopak vysloužila nejvyšší hodnocení v rámci daného srovnání. Tato skutečnost je pravděpodobně způsobena tím, že v rámci subjektivního dojmu není v případě metody SUBTRACTION natolik dobře odstraněn hluk pozadí jako u ostatních navržených metod. Tento fakt ovšem nemá žádný zásadní vliv na srozumitelnost hodnocenou danou metrikou.

Lepších subjektivních i objektivních výsledků navržené metody SUBTRACTION by bylo pravděpodobně dosaženo vhodnějším nastavením a umístěním využitých mikrofonů pro získání shodnějšího šumového pozadí, tedy shodnějšího signálu hluku pozadí ve frekvenční oblasti. Následným odečtením by tak došlo k přesnější kompenzaci tohoto šumu za možného zachování užitečného signálu nahrávky. Vzhledem k omezené možnosti nahrávání pro testování metody SUBTRACTION z důvodu vládních omezení v době tvorby práce je tak provedené testování této metody pouze omezené a spíše experimentální.

Závěr

Tato diplomová práce se zabývá odstraněním nežádoucího hluku z nahrávek řeči pořízených v přístroji magnetické rezonance. K tomuto účelu jsou zde navrženy a následně využity metody RTX Voice, NOISEREDUCE, SUBTRACTION a DAW, které slouží k odstraňování šumu či hluku pozadí ze zpracovávaného daného signálu.

V rámci práce je navržen a vytvořen program v programovacím jazyce Python, který disponuje jednoduchým grafickým rozhraním pro načtení a následné odhlučnění vybraných nahrávek. Tato funkce je v programu reprezentována názvem DENOISE.

První z navržených metod RTX VOICE využívá program *RTX Voice* od společnosti Nvidia pro odstranění hluku pozadí v reálném čase za využití grafické karty. Metoda NOISEREDUCE je implementací existující open-source knihovny NoiseReduce v jazyce Python pro odstraňování šumu z nahrávek. Tato knihovna funguje na principu odstraňování hluku na základě referenčního vzorku šumu z dané nahrávky. Metoda SUBTRACTION je vlastní navržená metoda využívající odečítání hluku ve frekvenční oblasti za využití dvojice spárovaných nahrávek. Metoda DAW je jako jediná realizována mimo navržený program za využití DAW softwaru Reaper s pomocí tzv. VST pluginu. Využitý VST plugin *ReaFir* v rámci této práce slouží k odstraňování hluku pozadí pomocí vytvoření šumové spektrální masky na základě poskytnuté sekvence hluku pozadí.

Program dále umožňuje hodnocení zpracovaných nahrávek objektivní metodou analýzy srozumitelnosti STOI, jejíž implementace je v rámci této práce reprezentována názvem SCORE. Metrika STOI (*Short-Time Objective Intelligibility measure*) využitá v této práci slouží k objektivnímu hodnocení srozumitelnosti řeči v nově získaných nahrávkách pomocí jejich srovnání s původními záznamy silně zatíženými negativním hlukem.

V rámci práce bylo dále provedeno nahrávání v přístroji MR realizované ve výzkumném centru CEITEC v Brně. Toto nahrávání bylo s ohledem na vládní opatření v době tvorby této práce provedeno spíše experimentálně. Pro plnohodnotné otestování metody SUBTRACTION využívající těchto nahrávek je zapotřebí větší množství nahrávání a testování v přístroji MR pro plnohodnotnou optimalizaci metody.

Pomocí vytvořeného programu byl dále odstraněn nežádoucí hluk z databáze 24 řečových nahrávek pořízených v přístroji MR (databáze DB) a také ze sady nahrávek pořízených ve výzkumném centru (sada CEITEC). Tyto nahrávky byly následně zpracovány všemi vytvořenými algoritmy v rámci navrženého programu a také zhodnoceny zmíněnou metrikou STOI. Výsledky objektivní analýzy pomocí metriky STOI jsou v rámci programu shrnuty a exportovány do jednoduchého textového

souboru. V rámci této práce byly tyto výsledky pro všechny hodnocené nahrávky zpracovány několika základními statistickými metodami a shrnuty do jednoduchých tabulek.

V této práci bylo dále využito metody subjektivní analýzy formou poslechových testů pro dodatečné a robustnější zhodnocení kvality odhlučnění danými metodami v rámci vytvořených nahrávek. Tyto testy byly provedeny formou online poslechových testů s více jak 20 nezávislými respondenty. Výsledky subjektivní analýzy jsou prezentovány pomocí grafů reprezentujících celkové zastoupení metod na základě jejich preference respondenty v poslechových testech.

Na základě objektivní a subjektivní analýzy byla následně provedena diskuze sumarizující a zhodnocující získané výsledky a také zjištěné klady a zápory jednotlivých metod. Celkově nejlepších výsledků dle objektivní analýzy pomocí metriky STOI i subjektivní analýzy v rámci poslechových testů dosáhly navržené metody NOISEREDUCE a DAW, které dosahovaly podobných výsledků v rámci obou analýz. Metoda RTX VOICE dosáhla v rámci obou analýz o poznání horších výsledků než obě výše zmíněné metody. Metoda SUBTRACTION byla provedena a hodnocena pouze na jedné ze zpracovaných sad nahrávek. Zde dosáhla nejvyššího výsledku v rámci objektivní analýzy, ale zároveň nejhoršího výsledku v rámci analýzy subjektivní. Za účelem optimalizace této metody by bylo zapotřebí provést větší počet měření a celkových testování vhodného způsobu její realizace.

V rámci této práce byla vytvořena programová aplikace pro odstranění hluku z libovolných zvukových nahrávek. Program nabízí několik metod s různou povahou a kvalitou výsledného zpracování. Tento program byl navržen především pro účely odstranění hluku přístrojů MR, avšak teoretické uplatnění může nalézt v řadě dalších odvětví.

Literatura

- [1] BAREŠ, M. *Diagnostika a klinické příznaky Parkinsonovy nemoci* Centrum pro abnormální pohyby a parkinsonismus, I. neurologická klinika LF MU, FN u sv. Anny, Brno Neurologie pro praxi 2001 / 1 Dostupné z URL: <<https://www.neurologiepropraxi.cz/pdfs/neu/2001/01/08.pdf>>
- [2] DEJONCKERE, P. H. *Assessment of Voice and Respiratory Function* [online]. Surgery of Larynx and Trachea, editace M. Remacle; H. E. Eckel, Springer Berlin Heidelberg, 2010, ISBN 978-3-540-79135-5, s. 11–26. [cit. 17. 11. 2014]. Dostupné z URL: <http://dx.doi.org/10.1007/978-3-540-79136-2_2>
- [3] EasyGUI *EasyGUI [Computer program]* [online]. Dostupné z URL: <<http://easygui.sourceforge.net/>>
- [4] GOBERMAN, A. M. *Correlation between acoustic speech characteristics and non-speech motor performance in Parkinson Disease* [online]. Med Sci Monit, ročník 11, č. 3, 2005: s. CR109–116 [cit. 23. 11. 2014]. Dostupné z URL: <<http://www.ncbi.nlm.nih.gov/pubmed/15735562>>
- [5] HARTELIUS L.; SVENSSON P. *Speech and swallowing symptoms associated with Parkinson's disease and multiple sclerosis: a survey*. Folia Phoniatr Logop, ročník 46, č. 1, 1994: s. 9–17
- [6] HILKER, R.; SCHWEITZER, K.; COBURGER, S.; GHAEMI, M.; WEISENBACH, S.; JACOBS, A. *Nonlinear progression of Parkinson disease as determined by serial positron emission tomographic imaging of striatal fluoro-dopa F 18 activity* Arch Neurol, ročník 62, č. 3, 2005: s.378–382.
- [7] KEFAUVER, A. P.; PATSCHKE, D. *Fundamentals of Digital Audio, New Edition* A-R Editions, 2007: s. 133. ISBN 9780895796110
- [8] MATTSON, S.; JAMES, S. *The pioneers of NMR and magnetic resonance in medicine: The story of MRI* Ramat Gan, Israel: Bar-Ilan University Press, 1996. ISBN 0-471-35128-8
- [9] NOISEREDUCE *Noisereduce [Computer program]* [online]. Dostupné z URL: <<https://pypi.org/project/noisereduce>>
- [10] NVIDIA Corporation *Nvidia RTX Voice [Computer program]* [online]. Version 0.5.12.6, retrieved 6 June 2016. Dostupné z URL: <<https://www.nvidia.com/en-us/geforce/guides/nvidia-rtx-voice-setup-guide/#RTX-Voice>>

- [11] PyInstaller Development Team *PyInstaller [Computer program]* [online]. Dostupné z URL: <<https://www.pyinstaller.org/>>
- [12] Cockos Incorporated *Reaper [Computer program]* [online]. Dostupné z URL: <<https://www.reaper.fm/>>
- [13] Steinberg *Virtual Studio Technology [Computer program]* [online]. Dostupné z URL: <<https://www.steinberg.net/en/company/technologies.html>>
- [14] Steinberg *Virtual Studio Technology 3 [Computer program]* [online]. Dostupné z URL: <<https://www.steinberg.net/en/company/technologies/vst3.html>>
- [15] TAAL, C. H. ; RICHARD, C. ; HEUSEDENS, R. ; JENSEN, J. *A short-time objective intelligibility measure for time-frequency weighted noisy speech* 14-19 March 2010 Delft University of Technology, Signal Information & Processing Lab, 2628 CD Delft, The Netherlands
- [16] TAYLOR, J. S. H.; RASTLE, K.; DAVIS, M. H. *Interpreting response time effects in functional imaging studies* Department of Psychology, Royal Holloway University of London Medical Research Council Cognition and Brain Sciences Unit, Cambridge Vydáno 27. Května 2014
- [17] TINTERA, J. *Klinické systémy magnetické rezonance: Vývoj za posledních 30 let a nové trendy* Základna radiodiagnostiky a intervenční radiologie, Institut klinické a experimentální medicíny, Praha, 15. 11. 2017.
- [18] VB Audio Software *VB-Cable [Computer program]* [online]. Version 1.0.3.3, retrieved 6 June 2016. Dostupné z URL: <<https://vb-audio.com/Cable/>>

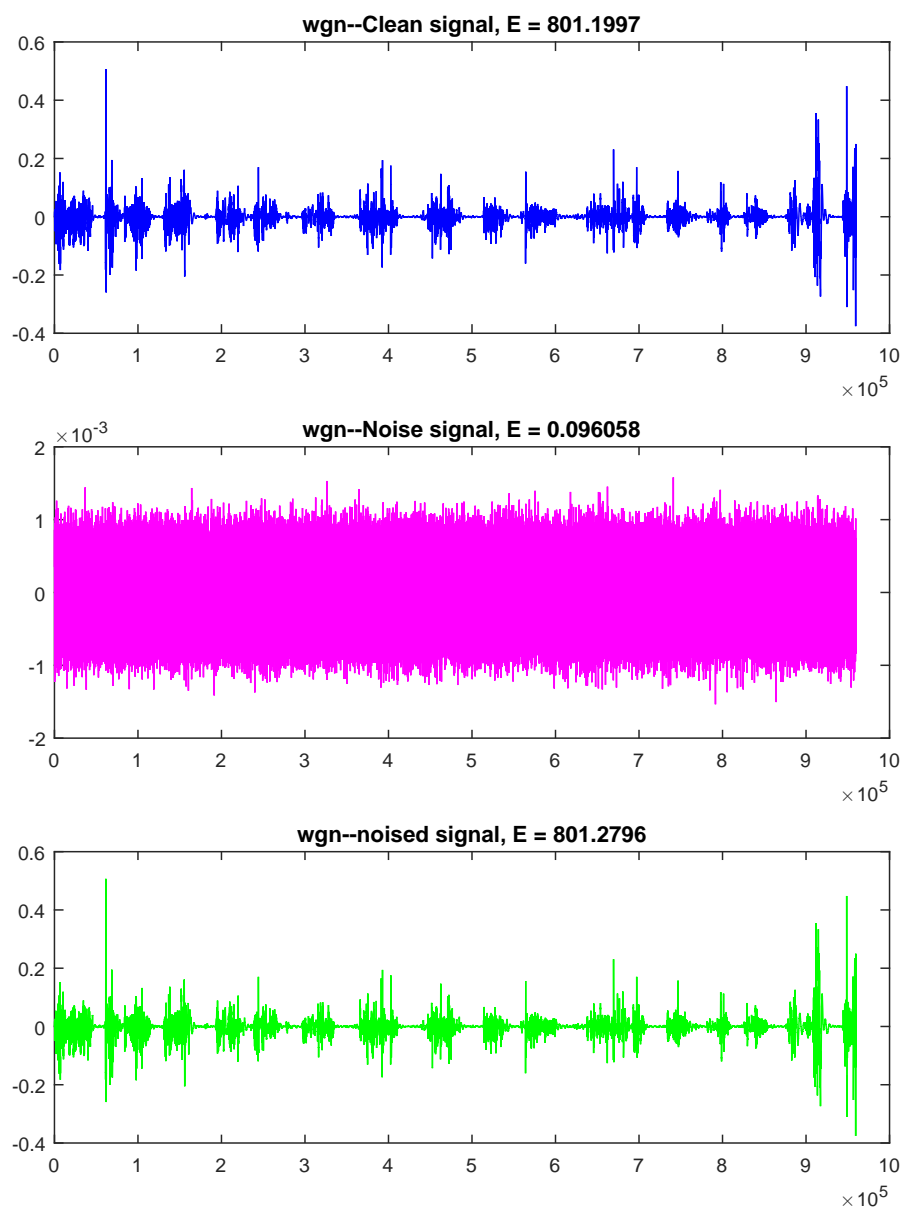
Seznam symbolů, veličin a zkratek

AI	umělá inteligence - artificial intelligence
<i>B</i>	magnetická indukce
<i>d</i>	vyhodnocení srozumitelnosti metriky STOI - d ratio
DAW	Digital Audio Workstation
DFT	diskrétní Fourierova transformace - Discrete Fourier Transform
D	rozptyl
FFT	rychlá Fourierova transformace - Fast Fourier Transform
GUI	grafické uživatelské rozhraní - Graphical User Interface
I/O	vstup a výstup - In / Out
IDE	integrované vývojové prostředí - Integrated Development Environment
IFFT	inverzní rychlá Fourierova transformace - Inverse Fast Fourier Transform
MR/MRI	magnetická rezonance - magnetic resonance imaging
PN	Parkinsonova nemoc
RT	čas odezvy - Response Time
STOI	Short-Time Objective Intelligibility measure
VB/VC	virtuální kabel - Virtual Cable
VST	Virtual Studio Technology

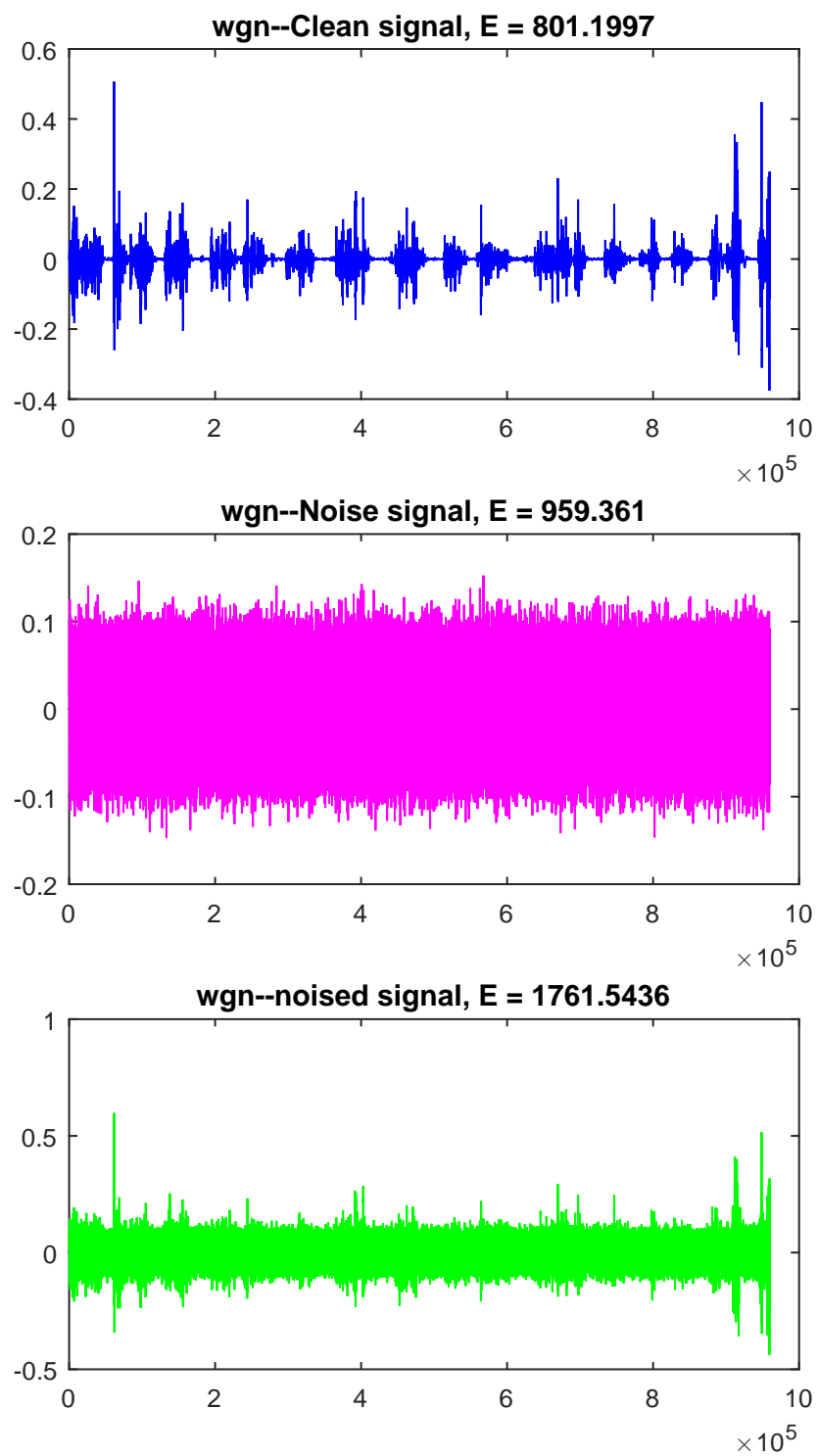
Seznam příloh

A	Testování metriky STOI	61
B	Příklady zdrojových kódů	64
B.1	Zdrojový kód metody <code>speaker_output</code> funkce <code>rtx_input</code>	64
B.2	Zdrojový kód metody <code>choose_rtx_output</code> funkce <code>rtx_record</code> . . .	65
B.3	Zdrojový kód funkce funkce <code>rtx_denoise</code>	66
C	Obsah přílohy	67

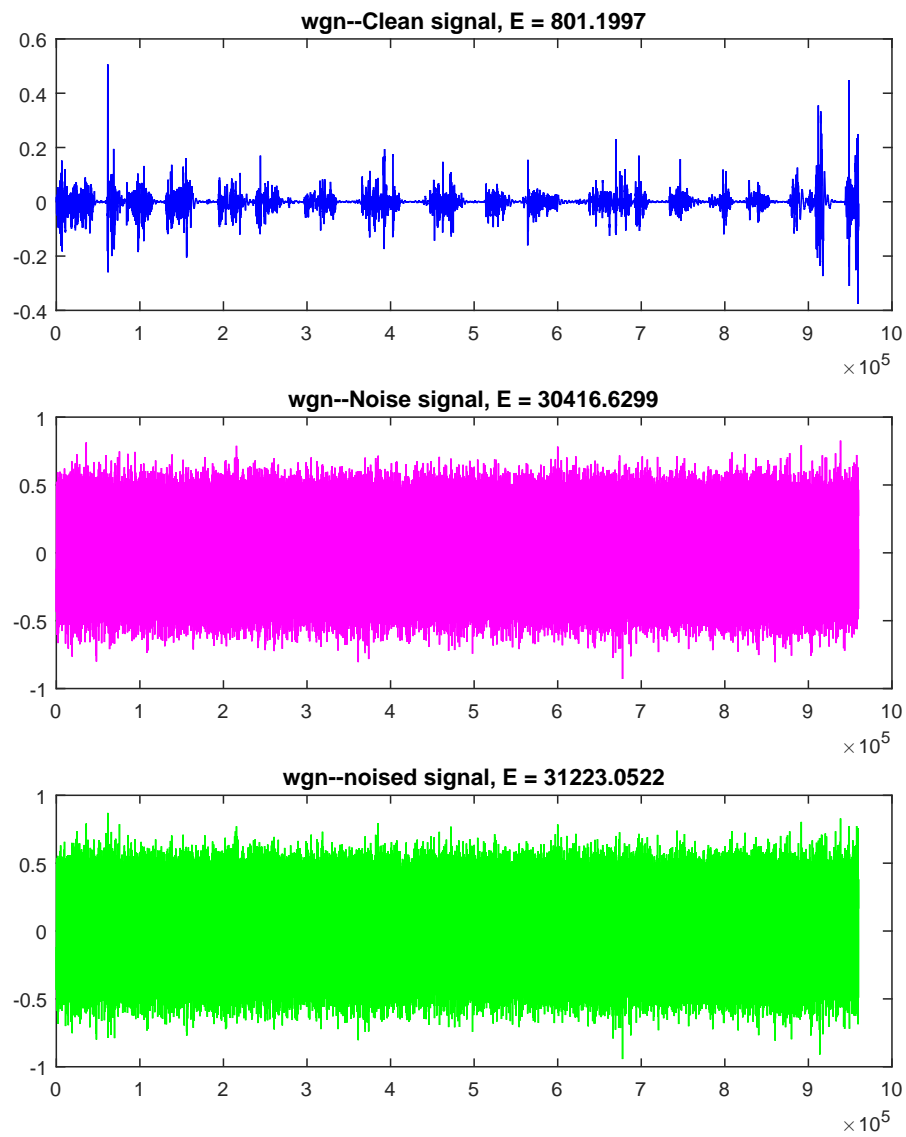
A Testování metriky STOI



Obr. A.1: Signál zatížený šumem o úrovni -70dbW , $d=0,949$



Obr. A.2: Signál zatížený šumem o úrovni -30dbW , $d=0,473$



Obr. A.3: Signál zatížený šumem o úrovni -15dbW , $d=0,334$

B Příklady zdrojových kódů

B.1 Zdrojový kód metody speaker_output funkce rtx_input

Výpis B.1: Zdrojový kód metody speaker_output funkce rtx_input

```
def select_speaker_output():
    if not check_cfg_exists():
        cfg_create()
    print("—— SPEAKER OUTPUT ——")
    mixer.init()
    speakerDevices = [
        get_audio_device_name(x, 0).decode() for x in range(
            get_num_audio_devices(0))
    ]
    mixer.quit()
    for i in speakerDevices:
        if ("AUX" or "Stereo Mix") in i:
            print("\nUsing " + i)
            valid = input("Is this correct (y/n)? ")
            if valid.lower() == "y":
                return speakerDevices[speakerDevices.index(i)]
    print("\nCould not find VB Audio AUX")

    loop_count = 0
    print("\nAudio Options: ")
    for i in speakerDevices:
        print(str(loop_count) + ". ", i)
        loop_count += 1
    print("\nIf VB Audio is not found here, please check it is not
        disabled.")
    speaker_choice = input("\nPlease select the VB Audio speaker input:
        ")
    cfg_write("speaker_output", speakerDevices[int(speaker_choice)])
    return speakerDevices[int(speaker_choice)]
```

B.2 Zdrojový kód metody choose_rtx_output funkce rtx_record

Výpis B.2: Zdrojový kód metody choose_rtx_output funkce rtx_record

```
def choose_rtx_output():
    print(" — RTX INPUT —")
    pyaudio_rtx_instance = pyaudio.PyAudio()
    info = pyaudio_rtx_instance.get_host_api_info_by_index(0)
    numdevices = info.get("deviceCount")
    for i in range(0, numdevices):
        if (
            pyaudio_rtx_instance.
                get_device_info_by_host_api_device_index(0, i).get(
                    "maxInputChannels") > 0:
            device = pyaudio_rtx_instance.
                get_device_info_by_host_api_device_index(
                    0, i
                ).get("name")

            if "NVIDIA RTX Voice" in device:
                print("\nUsing " + device)
                pyaudio_rtx_instance.terminate()
                cfg_write("mic_input", str(i))
                return i

    print("\n Could not find NVIDIA RTX Microphone.\n")
    print("Audio Options: ")
    for i in range(0, numdevices):
        if (
            pyaudio_rtx_instance.
                get_device_info_by_host_api_device_index(0, i).get(
                    "maxInputChannels"
                )
        ) > 0:
            device = pyaudio_rtx_instance.
                get_device_info_by_host_api_device_index(0, i).get("
                name")
            print( str(i)+ ". ", device)

    print("\nIf RTX Microphone is not found here, please check it is
        not disabled and is installed correctly.")
    microphone_choice = input("\nPlease select the RTX Microphone input
        to record: ")
    return int(microphone_choice)
```

B.3 Zdrojový kód funkce funkce rtx_denoise

Výpis B.3: Zdrojový kód funkce rtx_denoise

```
def rtx_denoise(input_song_path, output_directory):

    export_song_path = output_directory + "rtx_" + os.path.basename(
        input_song_path)
    if check_cfg_exists() == False:
        print("Config file not found... creating")
        mixer.init(devicename=select_speaker_output())
        mic_input = choose_rtx_output()
    else:
        mixer.init(devicename=cfg_read("speaker_output"))
        mic_input = int(cfg_read("mic_input"))
    mixer.music.load(input_song_path)
    input_track = wave.open(input_song_path)

    frames_of_input_song = input_track.getnframes()
    framerate_of_input_song = input_track.getframerate()
    input_track_length = (frames_of_input_song /
        framerate_of_input_song)

    length_of_input_song = float(input_track_length)
    hours, mins, seconds = convert(length_of_input_song)

    print("\nNow Processing: " + input_song_path + "\n")
    print("Input file framerate: " + str(framerate_of_input_song))
    print('Duration: %d:%d:%d' % (hours, mins, seconds))
    mixer.music.play()
    record(length_of_input_song, export_song_path, mic_input,
        framerate_of_input_song)
```

C Obsah přílohy

Příložený soubor ve formátu .rar obsahuje veškeré důležité soubory pro funkci programu a také složku obsahující testovací soubory. Soubor „test.wav“ se nachází ve složce „recordings/noised“ a obsahuje krátkou řečovou nahrávku s hlukem v pozadí. Již zpracované verze této nahrávky „rtx_test.wav“, „nr_test.wav“ apod. vhodné pro rychlé odzkoušení programu či jeho demonstraci se dále nacházejí ve složce „recordings/denoised“. Pro testování metody SUBTRACTION je ve složce „recordings/noise files“ připravena odpovídající šumová nahrávka potřebná ke správné funkci a testování této metody.

```
/ ..... kořenový adresář příloženého CD
├── recordings ..... složka s testovacími soubory
│   ├── noised
│   │   └── test.wav
│   ├── denoised
│   │   ├── rtx_test.wav
│   │   ├── daw_test.wav
│   │   ├── sub_test.wav
│   │   └── nr_test.wav
│   └── noise files
│       └── test.wav
├── main.py ..... hlavní soubor programu sloužící ke spuštění
├── load_denoise.py
├── rtx_denoise.py
├── rtx_input.py
├── rtx_record.py
├── noisereduce_denoise.py
├── subtraction_denoise.py
├── stoi_module.py
├── config.cfg ..... soubor pro I/O nastavení programu
├── requirements.txt ..... soubor pro instalaci požadovaných Python knihoven
├── README.txt ..... soubor s doporučeným nastavením
├── stoi_results.xlsx ..... tabulkový soubor s výsledky metriky STOI
├── DENOISER.exe ..... samostatně spustitelný .exe soubor s navrženým programem
└── DP_VRBA.pdf ..... text samotné práce ve formátu .pdf
```