

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra měření
Obor: Inteligentní budovy



Alternativní metoda stanovení STI pro vnitřní prostory budov

Alternative method of STI determination for indoor environments

DIPLOMOVÁ PRÁCE

Vypracoval: Bc. Tomáš Formánek
Vedoucí práce: Prof. Ing. Jan Holub, Ph.D.
Rok: 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Formánek** Jméno: **Tomáš** Osobní číslo: **483473**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Inteligentní budovy**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Alternativní metoda stanovení STI pro vnitřní prostory budov

Název diplomové práce anglicky:

Alternative method of STI determination for indoor environments

Pokyny pro vypracování:

Provedte rešerši existujících metod měření kvality ozvučení prostor se zaměřením na srozumitelnost řeči. Na jejím základě navrhnete vlastní metodu stanovení STI (Speech transmission index) pro vnitřní prostory budov, která bude oproti existujícím standardizovaným metodám snadněji proveditelná (např. bude využívat alternativní měřicí signál, významně zjednodušený výpočet apod.). Výslednou metodu otestujte a porovnejte s výsledky standardizovaného měření.

Seznam doporučené literatury:

- [1] ČSN EN ISO 3382-1:2009, Akustika - Měření parametrů prostorové akustiky - Část 1: Prostory pro přednes hudby a řeči
- [2] ČSN EN 60268-16 ed. 2:2012/Z1, Elektroakustická zařízení - Část 16: Objektivní hodnocení srozumitelnosti řeči indexem přenosu řeči
- [3] Everest, F. Alton, and Ken C. Pohlmann. 2022. Master Handbook of Acoustics. 7th ed. New York: McGraw Hill.

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Jan Holub, Ph.D. katedra měření FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **06.02.2024**

Termín odevzdání diplomové práce: **24.05.2024**

Platnost zadání diplomové práce:

do konce letního semestru 2024/2025

prof. Ing. Jan Holub, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne

.....
Bc. Tomáš Formánek

Poděkování

Tímto bych rád poděkoval Prof. Ing. Janu Holubovi, Ph.D. za ochotu, trpělivost a cenné rady během celého procesu vypracování této práce. Rád bych také poděkoval Ing. Tomášovi Drábkovi, Ph.D. za zprostředkování možnosti měřit katedrálu sv. Víta, Václava a Vojtěcha, knězi Pavlovi Kunešovi za možnost měřit kostel Nanebevzetí Panny Marie a v neposlední řadě Zuzaně Vithové, DiS. za výpomoc při provádění měření.

Bc. Tomáš Formánek

Název práce:

Alternativní metoda stanovení STI pro vnitřní prostory budov

Autor: Bc. Tomáš Formánek

Studijní program: Inteligentní budovy

Obor: Inteligentní budovy

Druh práce: Diplomová práce

Vedoucí práce: Prof. Ing. Jan Holub, Ph.D.

Abstrakt: Tato práce se zabývá návrhem alternativního výpočtu pro stanovení STI v interiérech budov. Teoretická část uvádí pojmy, které byly v praxi využity k návrhu výpočtu. Praktická část nejprve popisuje provedená měření reálných prostor a dále zkoumá souvislosti mezi parametry impulsní odezvy a stanoveným STI. Na základě pozorovaných souvislostí je pak navržen vlastní výpočet.

Klíčová slova: STI, STIPA, impulsní odezva, srozumitelnost řeči, NumPy

Title:

Alternative method of STI determination for indoor environments

Author: Bc. Tomáš Formánek

Abstract: This work deals with the design of an alternative calculation for determining STI in building interiors. The theoretical part introduces concepts that were used in practice for the calculation design. The practical part first describes the measurements of real spaces and then examines the relationships between the parameters of the impulse response and the determined STI. Based on the observed relationships, a custom calculation is then proposed.

Key words: STI, STIPA, impulse response, speech intelligibility, NumPy

Obsah

Úvod	1
1 Teoretická část	3
1.1 Zvuk a zvukový signál	3
1.1.1 Diskrétní signál	3
1.1.2 Decibel	4
1.1.3 Harmonické složky a Fourierova transformace	5
1.1.4 Amplitudová modulace	9
1.2 Lineární a v čase neměnné systémy - LTI	12
1.2.1 Impulsní odezva	12
1.3 Room Impulse Response - RIR	13
1.3.1 Metody měření RIR	15
1.4 Speech transmission index	18
1.4.1 Kroky výpočtu	18
2 Praktická část	21
2.1 Měření STI a IR v interiérech budov	21
2.1.1 Katedrála sv. Víta, Václava a Vojtěcha	22
2.1.2 Flight Arena FEL	23
2.1.3 Posluchárna 132 FEL	24
2.1.4 Posluchárna 337 FS	25
2.1.5 Kostel Nanebevzetí Panny Marie v Klecanech	26
2.1.6 Vyhodnocení měření	26
2.2 Vývoj alternativního výpočtu STI	27
2.2.1 Sběr dat	27
2.2.2 Nástroj pro zpracování impulsních odezev	29
2.2.3 Návrh výpočtu	34
2.3 Testování výpočtu	43
Závěr	45
Bibliografie	47
Přílohy	49
A Tabulka STI	49
B Převzaté impulsní odezvy z OpenAir	50

Úvod

Každý prostor vytváří určité podmínky pro šíření zvuku. Vlivem konstrukčního řešení lze tyto podmínky upravovat a ovlivňovat jimi například srozumitelnost řeči v řešené místnosti. Abychom byli schopni srozumitelnost objektivně posuzovat, používáme tzv. Speech Transmission Index (STI). STI se využívá k hodnocení nejen místností ale i jiných zvukových systémů, jako jsou například telekomunikační zařízení. Z tohoto důvodu existuje několik modelů výpočtu STI, které jsou optimalizované pro různé případy užití.

Základním modelem pro výpočet STI je Full STI, jehož časová náročnost je pro některé aplikace nežádoucí. Proto byl specificky pro účely hodnocení interiérů budov navržen rychlejší algoritmus RASTI (Room Acoustics STI). Dnes je již RASTI považován za obsolentní a je plně nahrazen algoritmem STIPA, který je rychlejší než Full STI a zároveň oproti RASTI univerzální.

Ačkoliv je algoritmus STIPA poměrně dobře optimalizovaný a umožňuje výpočet i nepřímo z impulsní odezvy, implementace samotného výpočtu je poměrně náročná. Cílem této práce je proto vyvinout zjednodušený algoritmus pro stanovení STI z impulsních odezev.

Kapitola 1

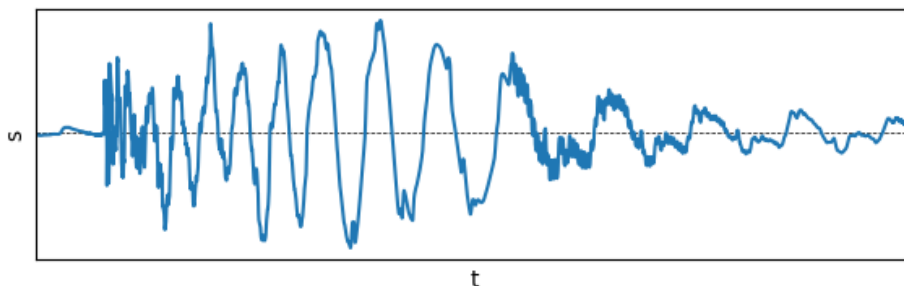
Teoretická část

Tato kapitola si klade za cíl uvést a velmi stručně vysvětlit základní pojmy, které je třeba znát pro pochopení měřicích metod a výpočtů užitých v této práci.

Text je pro názornost doprovázen příklady zdrojových kódů v jazyce Python 3 [1] s využitím knihoven NumPy [2], SciPy [3] a Matplotlib [4].

1.1 Zvuk a zvukový signál

Zvuk popisujeme jako drobné změny tlaku v prostředí, v našem případě vzduchu. Záznamem těchto změn získáváme zvukový signál. Ten lze popsat funkcí závislou na čase t , tedy $s(t)$. Viz obrázek 1.1. [5]

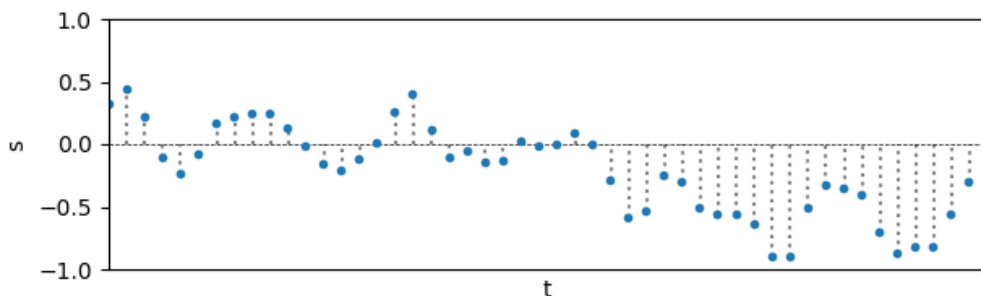


Obrázek 1.1: Funkce zvukového signálu $s(t)$

1.1.1 Diskrétní signál

Hovoříme-li o diskrétním neboli číslicovém signálu, průběh funkce $s(t)$ reprezentujeme seznamem hodnot, kterým říkáme vzorky. Sousedící vzorky jsou od sebe vzdáleny vždy stejný časový interval, jehož převrácená hodnota je tzv. vzorkovací frekvence. Obrázek 1.2 zobrazuje uspořádání vzorků v čase.

Vzorky lze zapisovat jako celá čísla v rozsahu -2^{b-1} až $2^{b-1} - 1$, kde b odpovídá počtu bitů jednoho vzorku, neboli bitové hloubce. Uvažujeme-li proto bitovou hloubku 16 bitů, rozsah v celočíselném tvaru odpovídá -32768 až 32767 . Vydělením všech vzorků maximální absolutní hodnotou záznamu získáme normalizovaný záznam s rozsahem -1 až 1 . [6]



Obrázek 1.2: Vzorčky normalizovaného diskrétního signálu v čase

1.1.2 Decibel

Působení fyzikální veličiny na člověka obecně popisují Weberův a Fechnerův zákon. Weberův zákon zavádí pojem Just Noticeable Difference (JND), tedy minimální velikost změny působení podnětu, kterou je možné lidským smyslem zaznamenat. Fechnerův zákon uvádí, že působení fyzikální veličiny je úměrné logaritmu intenzity vjemu. [7]

Z tohoto důvodu se v akustice i zpracování signálů setkáváme s logaritmickou jednotkou decibel (dB), jejíž celočíselný krok zhruba odpovídá JND. Význam této jednotky je závislý na kontextu a vztažené referenční hodnotě. V této práci se setkáme se dvěma typy decibelů, a to dB SPL a dB FS.

dB SPL

Sound Pressure Level (SPL), česky "úroveň akustického tlaku", je logaritmická míra akustického tlaku vztažená k prahu slyšitelnosti. Hodnotu úrovně akustického tlaku lze vypočítat pomocí vzorce 1.1, kde p_a reprezentuje měřený akustický tlak a p_0 odpovídá $2 \cdot 10^{-5}$ Pa, což je přibližně hodnota prahu slyšitelnosti. [5]

Decibel úrovně akustického tlaku zkracujeme buď normálně dB nebo $dB SPL$ podle přehlednosti kontextu.

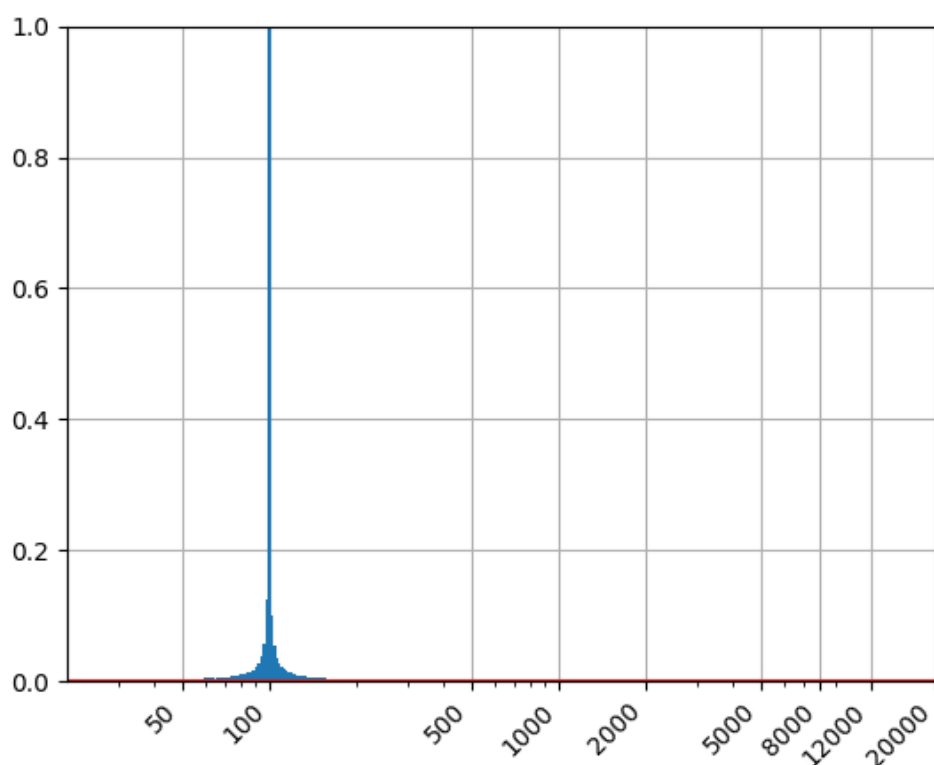
$$L_{SPL} = 20 \cdot \log \frac{p_a}{p_0} = 20 \cdot \log \frac{p_a}{2 \cdot 10^{-5}} \quad (1.1)$$

dB FS

Celý rozsah hodnot vzorků reprezentovatelný číslicovým systémem nazýváme Full Scale, zkráceně FS. Úroveň číslicového signálu je logaritmická míra vztažená k maximální možné hodnotě vzorku. Hodnotu úrovně číslicového signálu L_{FS} lze z lineární hodnoty A_{lin} a maximální možné hodnoty vzorku A_{max} vypočítat vzorcem 1.2. [5] [8]

Decibel úrovně číslicového signálu zkracujeme buď normálně dB nebo $dB FS$ podle přehlednosti kontextu.

$$L_{FS} = 20 \cdot \log \frac{A_{lin}}{A_{max}} \quad (1.2)$$



Obrázek 1.3: Frekvenční spektrum funkce sinus

1.1.3 Harmonické složky a Fourierova transformace

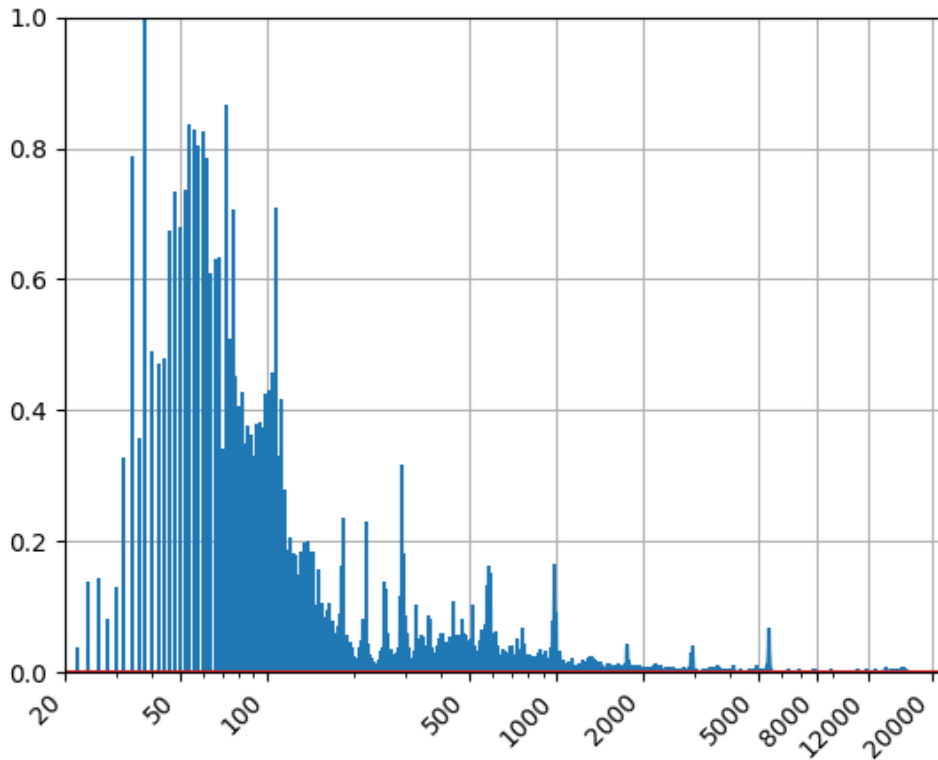
Průběh jakéhokoliv spojitého signálu lze rozložit na tzv. harmonické složky. Harmonickými složkami rozumíme řadu funkcí typu sinus o různých kmitočtech, fázích a amplitudách [5]. Zobrazením amplitud jednotlivých harmonických složek získáváme frekvenční spektrum signálu. Obrázek 1.3 zobrazuje frekvenční spektrum samotné funkce sinus o kmitočtu 100 Hz . Z povahy funkce vyplývá, že obsahuje jednu jedinou harmonickou složku, která se na obrázku projevuje zdaleka nejvýrazněji. Nenulové okolí 100 Hz je pak způsobeno konečnou délkou testovaného signálu.

Na obrázku 1.4 je ukázka frekvenčního spektra reálného signálu hudební nahrávky. Zde si lze všimnout výrazně vyššího počtu harmonických složek než v případě samotné funkce sinus.

Význam harmonických složek

Poměry harmonických složek ve zvukovém signálu určují z pohledu psychoakustiky barvu zvuku. Díky tomu jsme po sluchu schopni rozeznat například flétnu od houslí. Pokud by však flétna i housle hrály stejný tón a zároveň bychom např. v post-produkci odstranili všechny vyšší harmonické složky, zbyl by nám stejný fundament a došlo by tedy k nerozlišitelnosti.

Výskyt harmonických složek je významný také z hlediska akustických měření. Lidský sluch je v různých frekvenčních pásmech rozdílně citlivý. Proto se akustická měření často vyhodnocují v různých rovnoměrně rozmístěných frekvenčních pásmech zvlášť.



Obrázek 1.4: Frekvenční spektrum hudební nahrávky

Bílý a růžový šum

V měření akustiky se často setkáváme s tzv. bílým šumem. Bílý šum je náhodný stacionární signál, jehož frekvenční složky nabývají v celém frekvenčním spektru přibližně stejných amplitud. Vzhledem ke frekvenční závislosti citlivosti sluchu je nicméně v některých aplikacích vhodnější použít tzv. růžový šum, jehož frekvenční složky s rostoucí frekvencí amplitudově slábnou o 3dB na oktávu. [5]

Fourierova transformace

Jak uvádí zdroj [5], k převodu signálu z časové domény do domény frekvenční užíváme Fourierovu transformaci, kde $S(f)$ je fourierův obraz funkce $s(t)$, t reprezentuje čas a f je kmitočet zkoumané harmonické složky:

$$S(f) = \int s(t)e^{-2\pi ft} dt \quad (1.3)$$

Pro zpracování signálu v číslicové technice využíváme její diskrétní podoby, zkráceně DFT (Discrete fourier transform):

$$S(f) = \sum s(n)e^{-2\pi ft} dt \quad (1.4)$$

Pokud bychom si chtěli DFT naprogramovat, pravděpodobně narazíme na problém, že některé programovací jazyky nepodporují komplexní exponent. Je tedy potřeba pomocí eulerovy rovnosti rozepsat komplexní exponent na složky sinus a cosinus, jak uvádí zdroj [5]. Zdroj [9] podrobně rozebírá implementaci DFT v jazyce C. Rovnice 1.5 ukazuje rozepsaný tvar DFT:

$$S(f) = \sum s(n)[\cos(2\pi ft) + j \cdot \sin(2\pi ft)]dt \quad (1.5)$$

V praxi se pro výpočet DFT využívá optimalizovaných algoritmů souhrně zvaných Fast Fourier Transform (FFT), jejichž popis je však nad rámec rozsahu této práce. Postačí vědět, že vrací to samé co DFT akorát rychleji a s nižšími paměťovými nároky.

Interpretace výstupu FFT

Výstupem z FFT pro N vstupních vzorků je seznam N komplexních čísel z nichž amplitudu jednotlivých harmonických složek vypočteme jako jejich absolutní hodnotu. Výpočet amplitudy harmonické složky s indexem i_f ukazuje vzorec 1.6. [6]

$$A(i_f) = \sqrt{S_r(i_f)^2 + S_i(i_f)^2} \quad (1.6)$$

Z výstupního seznamu lze frekvenci harmonické složky f_i s indexem i vypočítat podle vztahu 1.7, kde f_s je vzorkovací frekvence a N je počet prvků výstupního seznamu. [6]

$$f_i = i \cdot \frac{f_s}{N} \quad (1.7)$$

Při vzorkování dochází k zrcadlení harmonických složek symetricky podle poloviny vzorkovací frekvence. Z toho vychází i Nyquistův teorém, který říká, že vzorkovací frekvence musí být alespoň dvojnásobná oproti nejvyšší frekvenci vzorkovaného signálu, jinak dojde k zrcadlení frekvenčních složek vyšších, než je polovina vzorkovací frekvence, do vzorkovaného signálu. Tento jev nazýváme aliasing. [5]

Ze vzorce 1.7 vyplývá, že druhá polovina výstupního seznamu obsahuje tutéž informaci co polovina první akorát zrcadlově převrácenou, tudíž ji v dalším zpracování můžeme ignorovat. Obrázek 1.5 zobrazuje jednotlivé fáze výpočtu frekvenčního spektra pro záznam s vzorkovací frekvencí 1000Hz funkce sinus o frekvenci 100Hz .

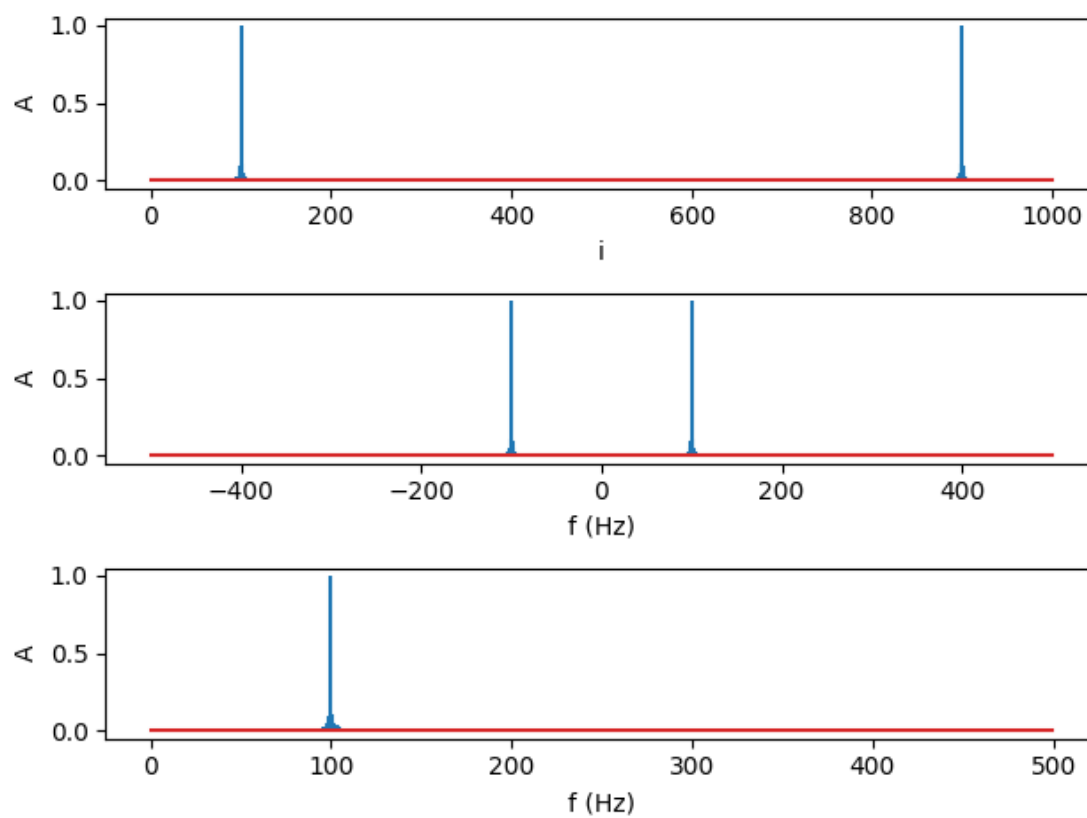
Kód 1.1 ukazuje výpočet frekvenčního spektra z reálného signálu pomocí knihovny NumPy.

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 from scipy.io import wavfile
4
5 fs , signal = wavfile.read('mono_example.wav')
6 spectrum = np.abs(np.fft.fft(signal))
7 frequencies = np.linspace(0, fs, len(spectrum))
8 half = int(len(spectrum)/2)
9
10 plt.stem(frequencies[:half], spectrum[:half], markerfmt=" ")
11 plt.show()

```

Kód 1.1: Výpočet frekvenčního spektra signálu s využitím knihovny NumPy



Obrázek 1.5: Od shora dolů: absolutní hodnoty výstupu z FFT, abs. hodnoty z FFT ve vztahu frekvencím harm. složek, data z první poloviny seznamu

1.1.4 Amplitudová modulace

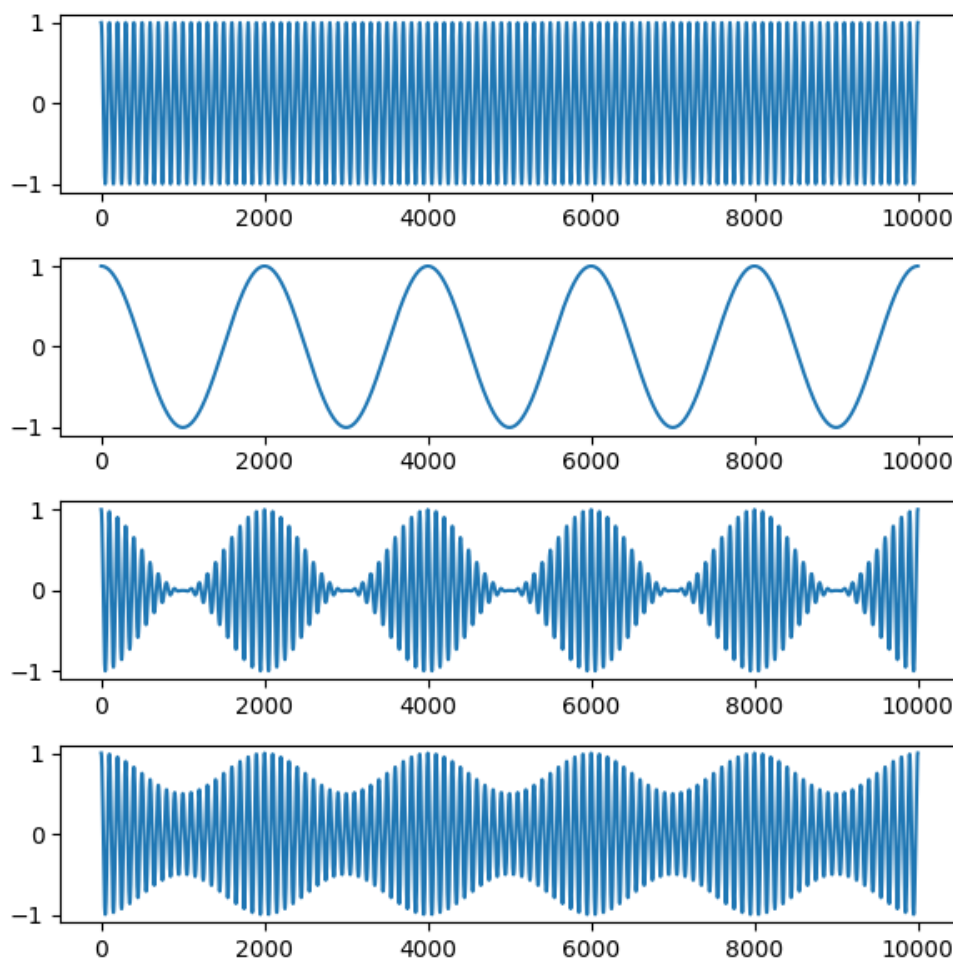
Amplitudová modulace je způsob modifikace amplitudy nosného signálu podle signálu modulačního. Jak uvádí zdroj [10], syntetizovat amplitudově modulovaný signál můžeme například podle rovnice 1.8, kde f_c , A_c jsou frekvence a amplituda nosného signálu a f_m , A_m jsou frekvence a amplituda modulačního signálu. Takto syntetizovaný signál pak dosahuje maximální amplitudy $A_{max} = A_c + A_m$ a minimální amplitudy $A_{min} = A_c - A_m$.

$$s(t) = \cos(2\pi f_c) \cdot \{A_m \cdot \cos(2\pi f_m) + A_c\} \quad (1.8)$$

Z hlediska hodnocení srozumitelnosti přenosu řeči hraje významnou roli tzv. **hloubka modulace**, která představuje poměr mezi amplitudou modulačního signálu a amplitudou nosného signálu. Viz vzorec 1.9. [10]

Na obrázku 1.6 jsou zobrazeny signály před modulací, modulovaný signál s hloubkou modulace 1 a modulovaný signál s hloubkou modulace 0.5.

$$m = \frac{A_m}{A_c} = \frac{A_{max} - A_{min}}{A_{max} + A_{min}} \quad (1.9)$$



Obrázek 1.6: Pořadí shora dolů: nosný signál ($f_c = 100Hz$), modulační signál ($f_m = 5Hz$), modulace o hloubce 1 ($A_c = 0.5$, $A_m = 0.5$), modulace o hloubce 0.5 ($A_c = 0.75$, $A_m = 0.25$)

```

1 import numpy as np
2 from scipy.signal import hilbert
3
4 def hilbert_envelope(signal):
5     return np.abs(hilbert(signal))
6
7 def moving_rms_envelope(signal, frame_size):
8     rms_envelope = []
9     number_of_frames = int(len(signal)/frame_size)
10
11     for i in range(number_of_frames):
12
13         #compute frame boundaries
14         start = frame_size * i
15         end = frame_size * (i + 1)
16         end = len(signal) if end >= len(signal) else end
17
18         #slice a frame
19         frame = signal[start:end]
20
21         #compute RMS
22         rms_envelope.extend(
23             [np.sqrt(np.mean(frame**2))] * frame_size
24         )
25
26     return rms_envelope

```

Kód 1.2: Výpočet obálky signálu

Amplitudovou modulaci lze provádět i jinými způsoby. Například v rámci přípravy měřicího signálu STI se využívá modulovaného signálu vypočteného násobením filtrovaného růžového šumu s modulační funkcí popsanou rovnicí 1.10. Výsledkem je sinusově modulovaný růžový šum s hloubkou modulace m a frekvence f_m . [11]

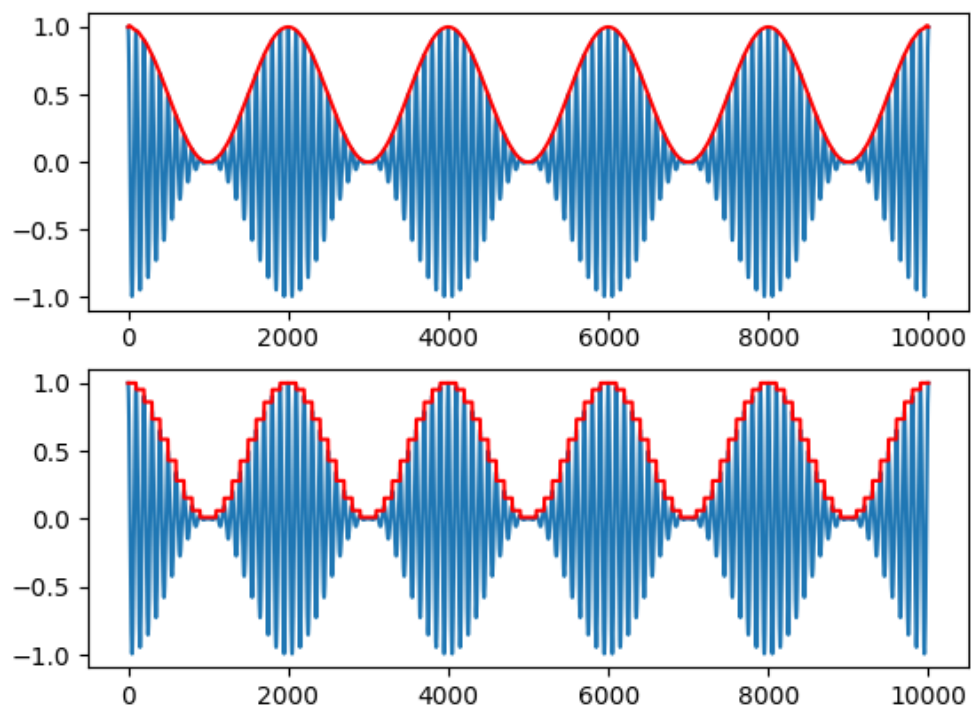
$$A_{f_m}(t) = \sqrt{0.5 \cdot (1 + m \cdot \cos(2\pi f_m t))} \quad (1.10)$$

Amplitudová obálka

Amplitudovou modulaci lze zároveň chápat jako generování periodické obálky. Obálkou rozumíme myšlenou křivku těsně ohraničující průběh signálu v časové doméně. [12]

Způsobů výpočtu amplitudové obálky existuje mnoho. Za zmínku stojí například výpočet absolutních hodnot z *hilbertovy transformace* nebo výpočet *RMS (Root Mean Square) po krátkých rámcích*.

Kód 1.2 ukazuje implementaci obou zmíněných metod. Na obrázku 1.7 je pak graficky znázorněn výpočet obálky modulované funkce cosinus s využitím kódu 1.2.



Obrázek 1.7: Ukázka vypočtené obálky (červená) z modulovaného signálu (modrá) hiltbertovou transformací (nahore) a moving RMS (dole).

1.2 Lineární a v čase neměnné systémy - LTI

Lineární a v čase neměnné systémy (Linear Time Invariant - LTI) jsou systémy splňující podmínky linearity a časové invariance [13]. LTI systémy mohou vstupní signál násobit konstantou, zpožďovat a sčítat. V případě, že dochází například k výraznému rušení cizím signálem, nebo ke vzniku harmonických složek signálu například vlivem clippingu, mluvíme již o systému nelineárním.

1.2.1 Impulsní odezva

Impulsní odezva $h(t)$ je reakce LTI systému na Diracův impuls. Diracův impuls je teoretický signál impulsního charakteru, který se vyznačuje délkou trvání blízkí se nule a maximální hodnotou amplitudy v nekonečnu. Frekvenčně lze považovat takový signál za všespektrální.

Pro každý LTI systém platí, že lze jeho chování zcela popsat vlastní impulsní odezvou. Výstup systému s impulsní odezvou $h(t)$ a vstupem $x(t)$ popisuje rovnice 1.11, kde symbol "*" představuje operaci zvanou **konvoluce**. [6]

$$y(t) = x(t) * h(t) \quad (1.11)$$

Konvoluci lze intuitivně chápat jako násobení dvou spojitých funkcí. Lze ji definovat rovnicí:

$$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau) \cdot f_2(t - \tau) d\tau \quad (1.12)$$

V číslicovém zpracování se konvoluce podobá násobení polynomů, reprezentujeme-li vzorky jako koeficienty polynomu. Obecně konvoluci číslicových signálů definujeme obdobně jako pro spojitý signál:

$$f_1(i) * f_2(i) = \sum_{m=0}^N f_1(m) \cdot f_2(i - m) \quad (1.13)$$

Inverzní zobrazení konvoluce nazýváme **dekonvoluce** a často jej uplatňujeme k získání impulsní odezvy z výstupního signálu LTI. Dekonvoluci lze provádět několika způsoby. Například z konvolučního teorému, vyplývá, že obraz fourierovy transformace impulsní odezvy se rovná podílu fourierových transformací vstupního a výstupního signálu LTI:

$$Y(f) = H(f) \cdot X(f) \rightarrow H(f) = \frac{Y(f)}{X(f)} \quad (1.14)$$

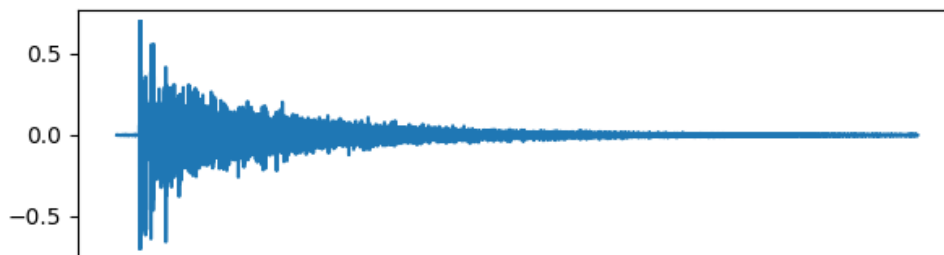
Tento přístup nicméně nelze využít v případech kdy $\exists f : Y(f) = 0$ a kdy Y obsahuje výrazný přidaný šum, jak uvádí zdroj [14]. Proto se v praxi používá jiných principů, které jsou specifické pro každou z měřicích metod.

Pokud impulsní odezvu podrobíme DFT, získáme **frekvenční odezvu** LTI. Jak uvádí zdroj [6], vztah mezi impulsní odezvou $h(t)$ a frekvenční odezvou $H(f)$ definujeme následovně:

$$H(f) = \sum_{n=-\infty}^{\infty} h(n) \cdot e^{-j2\pi f / f_s n} \quad (1.15)$$

1.3 Room Impulse Response - RIR

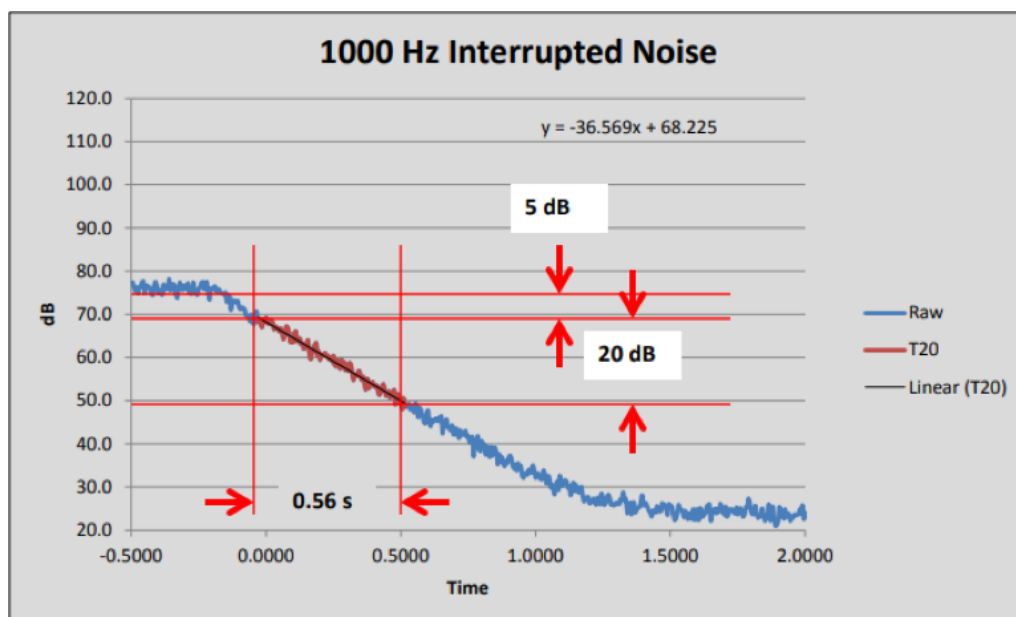
Tiché interiéry budov téměř splňují podmínky pro lineární a v čase neměnný systém. Proto je můžeme popisovat vlastními impulsními odezvami. Šířením zvuku v prostoru dochází přirozeně ke konvoluci vstupního signálu s impulsní odezvou místnosti. Díky tomu lze impulsní odezvu pořídit prostým záznamem impulsního zvuku v měřeném prostoru. Obrázek 1.8 zobrazuje průběh impulsní odezvy prostoru s dlouhým dozvukem.



Obrázek 1.8: Ukázka časového průběhu impulsní odezvy prostoru s dlouhým dozvukem

Zvukový impuls způsobí všesměrové šíření vln od zdroje. Při střetu s překážkou se část zvukové energie pohltí a část se odrazí zpět do místnosti. Odražené vlny se takto dále odrážejí a pohlcují, dokud zcela nezmizí a nenastane ticho. Zvuk, který vzniká vlivem tohoto jevu, nazýváme **dozvuk**, jak uvádí například [15].

Odražené vlny mohou mít negativní vliv na srozumitelnost řeči, a to zejména proto, že mohou maskovat zvuk vycházející z řečového aparátu. K odrazům zvukových vln nicméně nedochází ve všech částech frekvenčního spektra rovnoměrně. Zjednodušeně lze říci, že různé materiály překážek pohlcují různé vlnové délky, a tím dochází k frekvenční filtraci odraženého zvuku. To je třeba zohlednit nejen proto, že je citlivosti sluchu frekvenčně závislá, ale i proto, že různé skupiny hlásek vykazují různý obsah harmonických složek. [11] [16]



Obrázek 1.9: Ukázka grafu měření T20 přerušeným šumem. Zdroj: [17]

Doba dozvuku **RT60** je definována jako čas za který úroveň akustického tlaku klesne o 60 dB v difuzním zvukovém poli. Zpravidla se uvádějí dvě hodnoty **RT60**, a to **T20** a **T30**, které se zjednodušeně řečeno liší rozsahem hodnot využitým pro výpočet. Obrázek zobrazuje graf měření T20 přerušovaným šumem. Zdroj [18] uvádí podrobnosti o výpočtu těchto hodnot.

Střední doba dozvuku je číslo, které získáme výpočtem průměru hodnot doby dozvuku ve vybraných frekvenčních pásmech. Dle normy ISO 3382-1 [16] lze střední dobu dozvuku získat výpočtem průměru dob dozvuku na frekvencích 500 Hz a 1000 Hz.

Clarity je parametr, který se užívá k hodnocení srozumitelnosti impulsní odezvy. Jak uvádí zdroj [16], parametr vychází z pozorování, že pozdější dozvuk se na srozumitelnosti podepisuje negativněji než dozvuk brzký. Standardně se pro rozdělení brzkého a pozdního dozvuku užívá hranice 50 ms pro mluvené slovo a 80 ms pro hudbu. Z toho také vyplývá obvyklé označení **C50** a **C80**. Nás bude zajímat zejména hodnota **C50**, jejíž hodnota se spočítá podle zdroje [16] následovně:

$$C_{50} = 10 \times \log\left(\frac{\text{Energy}(0 - 50ms)}{\text{Energy}(50ms - end)}\right)dB \quad (1.16)$$

Definition je parametr podobný clarity. Jedná se o poměr energie brzkého dozvuku a energie celé impulsní odezvy. Značí se zpravidla D50 a udává se v procentech. Pro převod mezi C50 a D50 platí následující vztah:

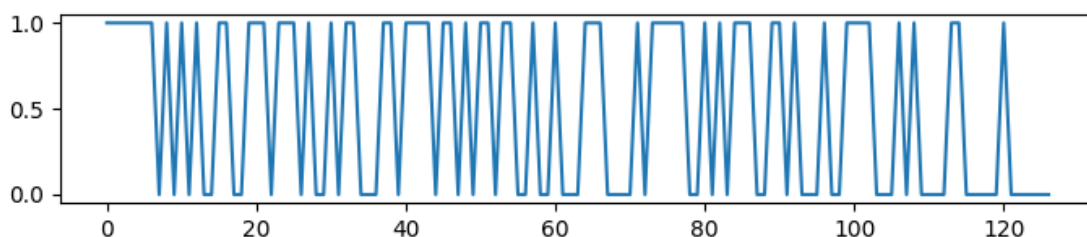
$$C_{50} = 10 \cdot \log\left(\frac{D_{50}}{1 - D_{50}}\right)dB \quad (1.17)$$

1.3.1 Metody měření RIR

Jednoduše lze impulzní odezvu místnosti získat tak, že pořídíme zvukový záznam vybuzení prostoru signálem, který se dostatečně podobá diracovu impulzu. K vybuzení se obvykle používá vystřel ze startovací pistole nebo prasknutí nafouklého balónku. Tato metoda je velmi výhodná v tom, že na rozdíl od jiných metod není třeba provádět žádné zpracování pořízeného záznamu, protože samotný záznam již impulzní odezvu obsahuje. Nevýhodou je, že tato metoda je poměrně náchylná na vnější rušení a budící zvuk lze jen velmi těžko kontrolovat. Proto se využívá spíše sofistikovanějších metod.

MLS

MLS (Maximum Length Sequence) je signál, který se na první poslech podobá bílému šumu. Vyznačuje se, stejně jako bílý šum, rovnoměrnou spektrální hustotou. Na rozdíl od bílého šumu jde ale o signál deterministický. Signál MLS je složen z hodnot 1 a 0 v pseudonáhodném pořadí. Příklad časového průběhu zobrazuje obrázek 1.10.



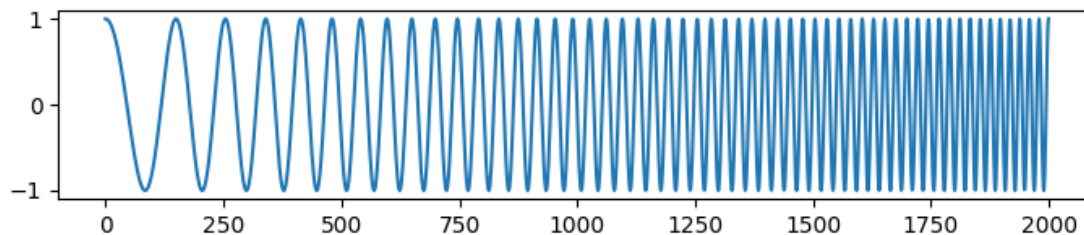
Obrázek 1.10: Ukázka průběhu MLS signálu vygenerovaného pomocí `scipy.signal.max_len_seq`

K měření impulzní odezvy pomocí MLS je potřeba vygenerovat budící signál, který se následně přehraje do měřeného prostoru. Poté se provede záznam odezvy prostoru na excitaci. Na závěr se provede dekonvoluce záznamu, čímž získáme impulzní odezvu prostoru.

Výhodou této metody je poměrně dobrá odolnost proti okolnímu rušení. K měření je nicméně potřeba více vybavení, které je často hůře přenosné. Pro měření touto metodou lze například využít softwarový nástroj MlsTool [19], který byl využíván i v rámci této práce. Open-source implementaci v jazyce python lze najít zde: [20]. Detailní popis algoritmu odkazu zde: [21].

Swept Sine

Swept Sine (přeladovaný tón) je sinusový signál, jehož kmitočet se v závislosti na čase zvyšuje buď lineárně nebo exponenciálně. Průběh lineárně přeladovaného tónu zobrazuje obrázek 1.11. Postup měření pomocí přeladovaného tónu je stejný jako v případě MLS. Nejprve je potřeba signál vygenerovat, poté přehrát a na závěr dekonvolvovat impulzní odezvu. Rozdíl je pouze v podobě budícího signálu a tedy i následném výpočtu dekonvoluce. K měření metodou přeladovaného tónu budeme využívat program Room Eq Wizard [22]. Metody přeladovaného tónu a MLS jsou součástí mezinárodního standardu ISO 18233.



Obrázek 1.11: Ukázka průběhu swept sine signálu vygenerovaného pomocí *scipy.signal.chirp*

Přístrojové vybavení

Požadavky na přístrojové vybavení pro měření impulsních odezev místností upravuje norma ISO 3382-1 [16].

Zdroj zvuku Pro měření impulsních odezev je nutné použít co nejvíce všesměrový zdroj. Obvykle se tohoto požadavku dosahuje použitím pasivního dvanáctistěnného reproduktoru. Příklad takového zařízení zobrazuje obrázek 1.12.

Zdroj by měl být schopný vybudit hladinu akustického tlaku alespoň takovou, aby nedošlo k narušení měření hlukem pozadí. V opačném případě lze vliv hluku pozadí potlačit synchronním průměrováním měření, což umožňuje jak metoda MLS tak metoda Sine Sweep. Pro případ užití metod, které synchronní průměrování neumožňují je nutné zajistit dosažení úrovně akustického tlaku alespoň 45 dB nad úroveň hluku pozadí. [16]



Obrázek 1.12: Ukázka dvanáctistěnného reproduktoru společnosti NTI. Zdroj: [23]

K záznamu zvuku by mělo být využito všesměrového kondenzátorového **mikrofonu** nejlépe s malou membránou pro eliminaci vlivu náhodného dopadu na plochu membrány. [16]

Typická podoba měřicího mikrofonu je vyobrazena na obrázku 1.13.



Obrázek 1.13: Ukázka měřicího mikrofonu společnosti Earthworks [24]

Záznamové zařízení by mělo disponovat dostatečným dynamickým rozsahem a doporučeným odstupem signál-šum minimálně 50 dB v každém kmitočtovém pásmu. Norma [16] uvádí jako možné záznamové médium magnetickou pásku nebo digitální paměť.

K digitálnímu záznamu prováděného počítačem je třeba dodatečného vybavení v podobě zvukové karty s integrovanými mikrofonními předzesilovači pro přizpůsobení vstupního signálu, zajištění A/D převodu a přenosu dat.

1.4 Speech transmission index

Jak uvádí norma [11], STI je číselné ohodnocení kvality přenosu řeči zvukovým systémem. Nabývá hodnot od 0 do 1, kde 0 představuje nejhorší a 1 nejlepší výsledek. STI počítá s modelem řečníka s dobrou artikulací, rychlostí řeči přibližně 3-4 slabiky za sekundu a předpokládá posluchače s normálním zdravým sluchem.

K záznamu zvukového signálu pro měření STI se využívá jednoho všesměrového mikrofону. Z toho mimo jiné vyplývá i fakt, že je model STI monoaurální. Jak uvádí zdroj [15], binaurální slyšení člověka má nezanedbatelný vliv na srozumitelnost přijímané zvukové informace. Jednou z nevýhod STI tedy je, že se tento vliv v neprospěch měřeného systému zanedbává. Za zmínku proto stojí práce [25], která se zabývá návrhem binaurální verze STI a porovnává výsledky se standardizovanou metodou.

V souvislé řeči se jednotlivé slabiky projevují jako rychlé změny v obálce zvukového signálu. Model STI vychází z předpokladu, že tyto změny mají zásadní vliv na srozumitelnost přenášené zvukové informace. Nejvíce tak hodnocení STI ovlivňují defekty způsobené dlouhým dozvukem, příměsí zpožděného signálu nebo příměsí cizího rušivého signálu.

Změny v obálce řečového signálu lze popsat jako sinusové modulace o různých frekvencích. Rozkladem funkce modulace na harmonické složky dostáváme spektrum obálky. Frekvenční složky modulačního signálu dobře artikulované řeči nabývají hodnot 0,5Hz po 16Hz s maximální modulací někde kolem 3Hz. Výpočet STI staví na předpokladu, že kvalitu přenosu řeči systémem lze popsat srovnáním hloubky modulace vstupního a výstupního signálu (tzv. m-hodnota). Funkce závislosti poměrů m-hodnot na modulační frekvenci se nazývá **Modulation Transfer Function (MTF)**. Viz obrázek 1.14.

V současnosti se ke stanovení STI využívá metod Full STI a STIPA. Rozdíl mezi nimi je v počtu analyzovaných modulačních frekvencí. Full STI měří pro každé ze 7 oktávových pásem modulační hloubku 14 modulačních frekvencí. Dohromady se provede 98 měření. STIPA naproti tomu zkoumá pro každé oktávové pásmo pouze dvě specifické modulační frekvence. [11]

1.4.1 Kroky výpočtu

Norma [11] uvádí následující kroky výpočtu STI:

1. **modulation transfer ratio (m-value/m-hodnota)** - poměr hloubky modulace vstupního a výstupního signálu pro specifickou modulační frekvenci a oktávové spektrum:

$$m_k(f_m) = \frac{m_o(f_m)}{m_i(f_m)} \quad (1.18)$$

2. **transmission index (TI)** - efektivní odstup signál/šum nabývající hodnot 0 až 1:

$$TI_{k,f_m} = \frac{10 \times \log \frac{m'_{k,f_m}}{1 - m'_{k,f_m}} + 15}{30} \quad (1.19)$$

, kde m'_{k,f_m} je korigovaná m-hodnota pro oktávové pásmo k a modulační frekvenci f_m .

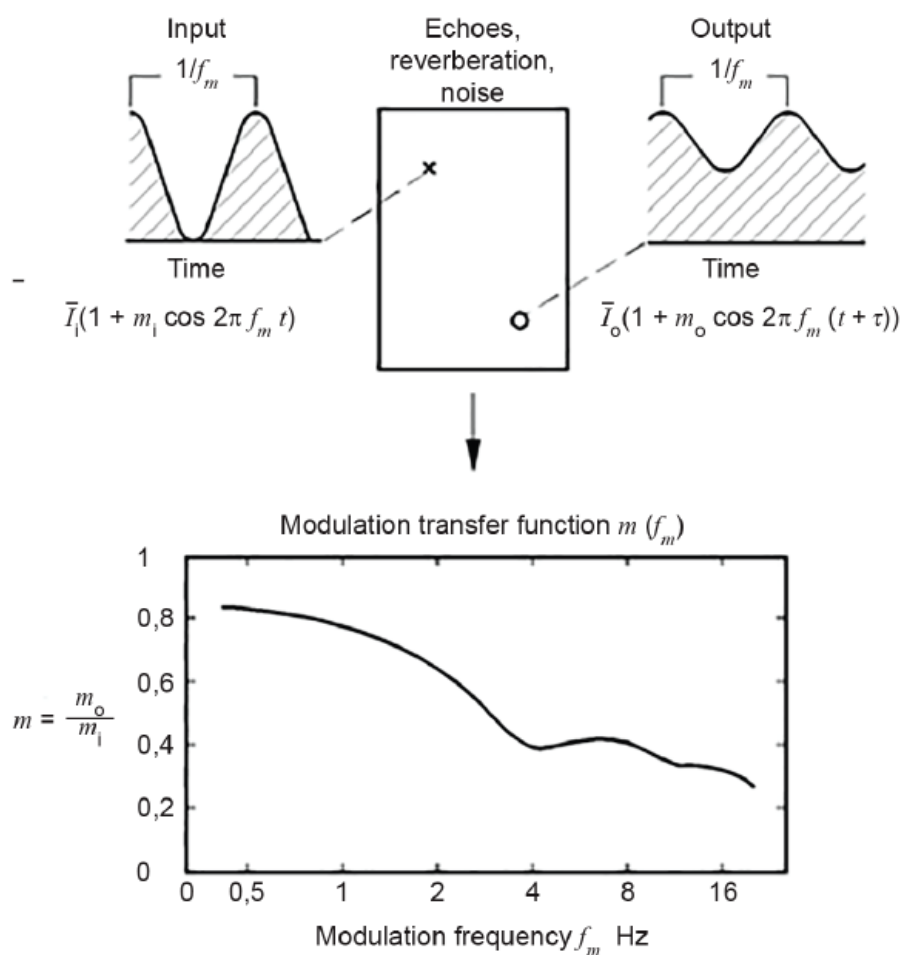
3. **modulation transfer index (MTI)** - neváhovaný průměr transmission indexů napříč všemi modulačními frekvencemi pro specifické oktávové pásmo nosného signálu:

$$MTI_k = \frac{1}{n} \sum_{m=1}^n TI_{k,f_m} \quad (1.20)$$

4. **speech transmission index (STI)**:

$$STI = \sum_{k=1}^7 \alpha_k \times MTI_k - \sum_{k=1}^6 \beta_k \times \sqrt{MTI_k + MTI_{k+1}} \quad (1.21)$$

, kde α_k je váhovací činitel a β_k je redundanční činitel oktávového pásmo k .



IEC

Obrázek 1.14: MTF z normy [11]

Kapitola 2

Praktická část

Tato část popisuje měření impulsních odezev a STI reálných prostorů, simulaci impulsních odezev, vývoj nástroje pro automatizované zpracování impulsních odezev a také návrh alternativního výpočtu STI.

2.1 Měření STI a IR v interiérech budov

K získání potřebných dat pro srovnání vlastností dozvuku s indexem srozumitelnosti řeči bylo provedeno měření pěti prostor. Prostory byly vybírány s ohledem na různorodost očekávaných vlastností dozvuku. K měření bylo využito následujícího vybavení:

Hlukoměr NTi Audio XL2: Umožňuje měřit STI pomocí algoritmu STIPA. Disponuje vlastním kalibrovaným měřicím mikrofonom.

Generátor signálu NTi Audio Minirator: Umožňuje přehrávat měřicí signál STIPA.

Notebook se zvukovou kartou Arturia MiniFuse 2: Měření impulsních odezev bylo realizováno pomocí softwarových nástrojů, k čemuž je potřeba dostatečně výkonného počítače s dobrými A/D, D/A převodníky, kterými disponuje USB zvuková karta MiniFuse 2.

Výkonový zesilovač Behringer EuroPower 2000 a dvanáctistěnný všesměrový reproduktor: Měřicí signál z Miniratoru a zvukové karty převádí na zvuk.

Neumann TLM 103: Všesměrový velkomembránový kondenzátorový mikrofón s dostatečně rovnou frekvenční odezvou, který byl v této práci využit k měření impulsních odezev.

Software: Měření impulsních odezev bylo nejprve realizováno pomocí softwaru MLS Tool od společnosti WaveArts <https://wavearts.com/products/plugins/mlstool>, později se zcela přešlo na Room Eq Wizard <https://www.roomeqwizard.com/>.

2.1.1 Katedrála sv. Víta, Václava a Vojtěcha

Jako první prostor se povedlo zajistit hlavní loď katedrály sv. Víta. Jedná se o rozměrný prostor s kamennými zdmi, což do jisté míry implikuje dlouhou dobu dozvuku. Ze subjektivního hlediska lze říci, že dozvuk v tomto prostoru není nijak nepříjemný a srozumitelnost nikterak výrazně neruší.

Ruchy, které byly způsobeny běžným provozem katedrály, mohly mít negativní vliv na přesnost měření. Například impulsní odezvy pořízené v rušném prostředí zpravidla trpí výrazným šumem. Vzhledem k velikosti prostoru se též pravděpodobně negativně projevil nedostatečný výkon zdroje měřicího zvuku. Například během měření STI nebylo možné ve všech měřicích bodech dosáhnout požadované úrovně akustického tlaku 80 dB. Výsledky mohou být proto lehce zkresleny.

Průměr STI ze všech měřicích bodů činí 0.56, což odpovídá zařazení do kategorie F podle tabulky STI v příloze. Parametry dozvuku vypočtené z impulsní odezvy jsou na obrázku 2.1.

Průměr hodnot D50 činí 66,65%. Abychom mohli STI a D50 porovnat, je žádoucí sjednotit rozsah obou parametrů a to jednoduše vydělením D50 konstantou 100 abychom si rozsah hodnot z 0% až 100% zmenšili na 0 až 1, jak je tomu u STI. Pokud takto upravenou hodnotu D50 porovnáme s naměřenými hodnotami STI, narazíme na velmi podobnou hodnotu, a to 0,68. Od průměru STI se průměr D50 lehce odchyluje. Střední doba dozvuku T20 je 5.893 sekundy.

	50	63	80	100	125	160	200	250	315	400	500	630
EDT (s)	1.259	2.466	2.694	4.922	6.088	5.845	6.492	5.640	5.174	4.727	4.897	4.845
T20 (s)		3.385	6.085	6.434	7.141	6.962	7.081	7.004	6.696	6.414	6.240	5.736
T30 (s)					7.078	7.181	7.086	7.246	6.901	6.872	6.379	6.155
Topt (s)	3.406	3.323	6.085	6.442	7.101	7.181	7.227	7.133	7.080	6.700	6.352	5.687
T60M (s)												
C50 (dB)	7.18	6.05	-1.34	-3.40	-6.34	-4.51	-5.32	0.20	2.36	2.83	1.20	3.38
C80 (dB)	6.87	6.11	-1.49	-2.79	-4.23	-3.60	-4.27	0.81	2.67	3.05	2.31	3.86
D50 (%)	83.9	80.1	42.3	31.4	18.8	26.2	22.7	51.2	63.3	65.7	56.9	68.5
TS (s)	0.115	0.130	0.216	0.303	0.406	0.351	0.401	0.264	0.190	0.165	0.196	0.146
	800	1,000	1,250	1,600	2,000	2,500	3,150	4,000	5,000	6,300	8,000	10,000
EDT (s)	4.574	4.026	4.381	3.305	3.530	3.401	2.165	1.954	0.381	0.041	0.364	0.005
T20 (s)	5.674	5.407	4.944	4.504	4.351	3.957	3.420	2.598	1.972	1.301	1.245	0.784
T30 (s)	6.050	5.851	5.132	4.804	4.507	4.068	3.515	2.786	2.165	1.499	1.325	0.887
Topt (s)	5.801	5.157	5.232	4.451	4.554	4.052	3.502	2.805	2.237	1.543	1.350	0.905
T60M (s)												
C50 (dB)	2.70	6.15	6.86	8.17	4.02	4.11	6.52	7.83	11.82	13.82	11.76	18.11
C80 (dB)	2.95	6.50	7.18	8.44	4.43	4.70	7.05	8.38	12.56	15.19	13.33	20.00
D50 (%)	65.1	80.5	82.9	86.8	71.6	72.0	81.8	85.8	93.8	96.0	93.8	98.5
TS (s)	0.158	0.090	0.075	0.056	0.105	0.099	0.060	0.041	0.015	0.009	0.013	0.003

Obrázek 2.1: Parametry impulsní odezvy Katedrály sv. Víta

2.1.2 Flight Arena FEL

Prostor Flight areny na ČVUT sice není prostorem určeným k poslechu mluveného slova, je nicméně zajímavý svým hutným dozvukem, který v kontextu srozumitelnosti řeči stojí za zkoumání. Jedná se o víceméně prázdnou místnost střední velikosti s omítnutými rovnoběžnými zdmi. Dozvuk v prostoru na poslech srozumitelnost znatelně degraduje, nikoliv však do míry, ve které by zabraňoval bezproblémové komunikaci.

Jelikož místnost byla tichá a prázdná, měření proběhlo bez problému. Průměrná hodnota STI činí 0.46, což odpovídá zařazení do kategorie H podle tabulky STI v příloze. Parametry dozvuku vypočtené z impulsní odezvy jsou na obrázku 2.2. Průměr hodnot D50 činí 46,54%, což se po převedení z procent na rozsah 0-1 opět velmi blíží průměrné hodnotě STI. Střední doba dozvuku T20 činí 2,18 sekundy.

	50	63	80	100	125	160	200	250	315	400	500	630
EDT (s)	0.579	2.188	2.565	1.059	1.858	2.086	1.859	1.832	1.893	2.050	1.634	2.111
T20 (s)		2.649	2.081	1.930	2.228	2.225	2.094	2.110	2.372	2.109	2.135	2.036
T30 (s)			2.210	1.970	2.049	2.081	1.972	2.186	2.276	2.107	2.134	2.107
Topt (s)	0.379	2.649	2.049	2.011	2.028	2.067	2.025	2.226	2.255	2.074	2.126	2.111
T60M (s)												
C50 (dB)	-1.04	2.87	-0.05	3.29	0.46	-2.58	-0.23	-2.84	-5.27	-0.61	-1.49	-0.84
C80 (dB)	0.01	-1.60	-0.09	6.28	3.49	-1.98	3.12	-0.88	-3.35	0.25	2.11	0.47
D50 (%)	44.1	66.0	49.7	68.1	52.6	35.5	48.7	34.2	22.9	46.5	41.5	45.2
TS (s)	0.064	0.158	0.178	0.090	0.127	0.168	0.112	0.135	0.165	0.139	0.110	0.125
	800	1,000	1,250	1,600	2,000	2,500	3,150	4,000	5,000	6,300	8,000	10,000
EDT (s)	1.993	2.042	1.836	2.184	2.285	2.132	1.917	1.627	1.447	1.082	0.995	0.687
T20 (s)	2.276	2.220	1.900	2.067	2.170	1.990	1.832	1.661	1.351	0.986	0.909	0.697
T30 (s)	2.210	2.210	1.925	2.092	2.178	2.055	1.871	1.646	1.365	1.042	0.930	0.723
Topt (s)	2.238	2.230	1.930	2.153	2.177	2.086	1.884	1.653	1.369	1.076	0.940	0.706
T60M (s)												
C50 (dB)	-2.86	-3.04	-0.62	-2.24	-2.32	-2.76	-0.53	-1.40	-1.97	2.25	4.65	4.55
C80 (dB)	-1.21	-0.35	0.90	-0.27	-1.12	-1.34	1.01	1.22	0.73	4.51	7.09	8.98
D50 (%)	34.1	33.2	46.4	37.4	37.0	34.6	47.0	42.0	38.8	62.7	74.5	74.0
TS (s)	0.158	0.138	0.111	0.134	0.146	0.143	0.108	0.103	0.103	0.055	0.036	0.033

Obrázek 2.2: Parametry impulsní odezvy Flight areny na ČVUT

2.1.3 Posluchárna 132 FEL

Tato místnost byla vybrána, jako příklad průměrně srozumitelné posluchárny. Dozvuk v tomto prostoru je lehce rušivý, ale mluvenému slovu je zde poměrně dobře rozumět.

Již při stavbě měřicí techniky se objevil problém. Pro akustická měření s všesměrovým reproduktorem se doporučuje reproduktor umístit doprostřed místnosti aby nedocházelo k parazitnímu odrazu o blízkou stěnu. V tomto případě ale nebylo možné trojnožku s všesměrovým reproduktorem stabilně umístit přímo doprostřed místnosti aniž by se poškodilo vybavení nebo nábytek místnosti. Reproduktor byl tedy umístěn na místo vyučujícího před tabulí, tedy relativně blízko ke stěně s určitým předpokladem nepřesnosti výsledku.

Průměrná hodnota STI činí 0,65, což odpovídá zařazení do kategorie D podle tabulky STI v příloze. Parametry dozvuku vypočtené z impulsní odezvy jsou na obrázku 2.3. Průměr hodnot D50 činí 53,19% což se po převedení z procent na rozsah 0-1 žádně z naměřených hodnot STI neblíží. Střední doba dozvuku činí 0,7 sekundy.

	50	63	80	100	125	160	200	250	315	400	500	630
EDT (s)	1.587	1.772	1.187	1.040	1.042	1.012	0.792	0.693	0.971	0.837	0.797	0.715
T20 (s)	1.272	1.761	1.450	1.093	0.936	0.799	0.715	0.567	0.666	0.546	0.637	0.657
T30 (s)			1.226	0.934	0.867	0.772	0.663	0.614	0.686	0.619	0.615	0.633
Topt (s)	1.235	1.540	1.228	0.993	0.861	0.767	0.632	0.621	0.666	0.666	0.648	0.639
T60M (s)												
C50 (dB)	-5.12	-5.87	0.14	-2.50	3.39	-0.16	2.81	3.09	1.41	1.53	0.59	0.69
C80 (dB)	-4.24	-4.35	-0.20	-0.67	4.23	5.16	5.35	4.33	4.43	4.58	3.97	5.60
D50 (%)	23.5	20.6	50.8	36.0	68.6	49.1	65.6	67.1	58.0	58.7	53.4	54.0
TS (s)	0.192	0.185	0.119	0.112	0.078	0.085	0.069	0.062	0.068	0.062	0.066	0.065
	800	1,000	1,250	1,600	2,000	2,500	3,150	4,000	5,000	6,300	8,000	10,000
EDT (s)	0.637	0.738	0.873	0.931	0.927	0.989	0.907	0.885	0.701	0.659	0.645	0.515
T20 (s)	0.716	0.754	0.910	0.953	0.934	0.879	0.909	0.808	0.769	0.658	0.632	0.572
T30 (s)	0.723	0.789	0.892	0.945	0.924	0.901	0.893	0.827	0.749	0.677	0.645	0.559
Topt (s)	0.705	0.813	0.915	0.953	0.926	0.913	0.943	0.822	0.758	0.686	0.655	0.561
T60M (s)												
C50 (dB)	0.25	1.94	-0.58	0.25	-1.13	-1.10	1.13	1.31	1.78	2.35	1.81	4.90
C80 (dB)	6.26	5.45	3.26	4.07	3.62	2.67	3.63	3.79	5.73	5.91	6.05	8.33
D50 (%)	51.5	61.0	46.7	51.5	43.6	43.7	56.5	57.5	60.1	63.2	60.3	75.6
TS (s)	0.065	0.062	0.074	0.070	0.077	0.077	0.067	0.062	0.052	0.051	0.054	0.040

Obrázek 2.3: Parametry impulsní odezvy posluchárny 132 na FEL

2.1.4 Posluchárna 337 FS

Tato posluchárna je oproti místnosti 132 ze subjektivního hlediska o poznání méně srozumitelná. Rozumět řeči přednášejícího vyžaduje dávku soustředění, ačkoliv doba dozvuku není na první poslech o tolik delší než v předchozím případě.

I v případě této posluchárny byl problém s umístěním reproduktoru doprostřed místnosti, proto se stejně jako v předchozím případě umístil na pozici před tabulí. Průměrná hodnota STI v tomto případě vyšla 0,49, což odpovídá zařazení do kategorie H podle tabulky STI v příloze. Parametry dozvuku vypočtené z impulsní odezvy jsou na obrázku 2.4.

Průměr hodnot D50 je v tomto případě 29,76, což se po převedení z procent na rozsah 0-1 od naměřených hodnot STI významně liší. Střední doba dozvuku T20 je 1,58 sekundy.

	50	63	80	100	125	160	200	250	315	400	500	630
EDT (s)	3.155	1.537	2.850	2.724	3.036	2.419	2.462	1.320	1.527	1.726	1.383	1.554
T20 (s)	4.097		2.550	2.555	2.359	1.738	2.276	2.324	1.984	1.585	1.618	1.486
T30 (s)			2.567		2.410	2.003	2.253	2.263	1.972	1.674	1.682	1.592
Topt (s)	4.042	0.153	2.567	2.656	2.405	1.990	2.264	2.249	1.913	1.704	1.664	1.486
T60M (s)												
C50 (dB)	-1.29	-5.14	-3.95	-8.27	-13.79	-6.33	-6.00	-3.58	-1.96	-1.43	-2.96	-4.03
C80 (dB)	-6.16	-2.83	-2.01	-7.39	-5.33	-4.63	-2.16	-0.42	2.07	-1.70	-0.77	-1.86
D50 (%)	42.7	23.5	28.7	13.0	4.0	18.9	20.1	30.5	38.9	41.9	33.6	28.4
TS (s)	0.297	0.133	0.241	0.285	0.254	0.210	0.209	0.136	0.121	0.131	0.119	0.126
	800	1,000	1,250	1,600	2,000	2,500	3,150	4,000	5,000	6,300	8,000	10,000
EDT (s)	1.390	1.572	1.533	1.654	1.570	1.493	1.542	1.390	1.178	0.997	0.933	0.742
T20 (s)	1.505	1.536	1.564	1.606	1.575	1.587	1.411	1.334	1.205	1.030	0.889	0.691
T30 (s)	1.568	1.561	1.617	1.608	1.564	1.553	1.421	1.329	1.216	1.031	0.917	0.718
Topt (s)	1.486	1.585	1.600	1.631	1.556	1.582	1.425	1.336	1.257	1.029	0.929	0.694
T60M (s)												
C50 (dB)	-7.20	-4.99	-4.21	-9.69	-6.02	-4.48	-4.26	-3.03	-1.94	0.66	0.06	2.28
C80 (dB)	-2.15	-2.41	-1.91	-3.82	-1.91	-1.85	-1.21	-0.14	1.13	3.07	3.49	5.41
D50 (%)	16.0	24.1	27.5	9.7	20.0	26.3	27.3	33.2	39.0	53.8	50.3	62.8
TS (s)	0.133	0.137	0.132	0.148	0.135	0.125	0.121	0.110	0.091	0.070	0.070	0.050

Obrázek 2.4: Parametry impulsní odezvy posluchárny 337 na FS

2.1.5 Kostel Nanebevzetí Panny Marie v Klecanech

Jako poslední prostor byl měřen kostel Nanebevzetí Panny Marie v Klecanech. Jedná se o kostel střední velikosti s omítnutými zdmi. I když je dozvuk v prostoru velmi zřetelný, srozumitelnost řeči je na poslech velmi dobrá.

Měření proběhlo bez větších komplikací a ruchu. Průměrná hodnota STI v tomto případě vyšla 0,53, což odpovídá zařazení do kategorie G podle tabulky STI v příloze. Parametry dozvuku vypočtené z impulsní odezvy jsou na obrázku 2.5. Průměr hodnot D50 je v tomto případě 59,7, což se po převedení na rozsah 0-1 naměřeným hodnotám STI blíží. Střední doba dozvuku T20 je 2,16 sekundy.

	50	63	80	100	125	160	200	250	315	400	500	630
EDT (s)	1.632	1.111	2.463	1.694	3.080	2.326	2.409	1.915	1.837	1.674	1.485	1.670
T20 (s)	2.882	2.385	2.612	3.076	2.725	2.296	2.740	2.383	2.863	2.264	2.486	2.398
T30 (s)	2.747	2.187			2.573	2.611	2.537	2.304	2.569	2.339	2.310	2.244
Topt (s)	2.747	2.242	2.776	3.100	2.587	2.181	2.551	2.324	2.635	2.286	2.226	2.240
T60M (s)												
C50 (dB)	-1.08	3.68	-1.06	-3.35	-0.39	0.22	1.67	0.56	0.11	0.87	-0.20	0.99
C80 (dB)	-0.38	3.33	0.28	0.25	0.21	0.93	2.09	1.06	0.60	3.60	3.65	2.78
D50 (%)	43.8	70.0	43.9	31.6	47.7	51.3	59.5	53.2	50.6	55.0	48.8	55.7
TS (s)	0.169	0.095	0.161	0.152	0.177	0.140	0.115	0.121	0.120	0.098	0.096	0.092
	800	1,000	1,250	1,600	2,000	2,500	3,150	4,000	5,000	6,300	8,000	10,000
EDT (s)	1.747	1.810	1.781	1.522	1.520	1.422	1.115	0.938	0.731	0.648	0.171	0.297
T20 (s)	2.057	1.836	1.645	1.523	1.477	1.255	1.146	0.971	0.783	0.686	0.582	0.523
T30 (s)	2.016	1.807	1.707	1.533	1.481	1.309	1.179	0.966	0.829	0.700	0.613	0.521
Topt (s)	2.034	1.807	1.747	1.554	1.499	1.339	1.178	0.966	0.838	0.705	0.634	0.536
T60M (s)												
C50 (dB)	0.60	1.12	3.31	1.37	1.09	0.36	3.13	3.98	2.68	8.01	11.90	11.51
C80 (dB)	3.02	3.25	4.61	2.95	3.23	2.46	5.10	6.36	7.11	12.39	15.47	14.81
D50 (%)	53.4	56.4	68.2	57.8	56.3	52.1	67.3	71.4	65.0	86.3	93.9	93.4
TS (s)	0.090	0.085	0.067	0.077	0.076	0.078	0.051	0.045	0.041	0.017	0.011	0.009

Obrázek 2.5: Parametry impulsní odezvy kostela Nanebevzetí Panny Marie v Klecanech

2.1.6 Vyhodnocení měření

Z naměřených hodnot je na první pohled vidět, že střední doba dozvuku nemá na index srozumitelnosti řeči sama o sobě pozorovatelný vliv. Například podle STI je katedrála sv. Víta lépe hodnocena než Flight aréna, ačkoliv má dobu dozvuku znatelně delší, zatímco u poslucháren je tomu naopak.

Naproti tomu parametr D50 se s STI ve většině případů mění podobně. Ve dvou případech průměr D50 dokonce odpovídá některé z naměřených hodnot a v jednom případě se tomu blíží. Ve zbylých dvou případech však došlo ke stanovení hodnot STI zcela odlišných od průměru D50. Nutno podotknout, že v obou případech mohlo dojít ke zkreslení naměřených hodnot vlivem postavení reproduktoru příliš blízko ke stěně. Je tedy zřejmé, že další průzkum vztahu D50 a STI by mohl přispět k nalezení hledané alternativní metody pro odhad STI.

2.2 Vývoj alternativního výpočtu STI

Měření ukázalo, že mezi parametry D50 a STI pravděpodobně existuje nějaký vztah. V rámci této podkapitoly se pokusíme tuto skutečnost ověřit a případně výpočet D50 přizpůsobit tak, aby se hodnoty obou výpočtů co nejvíce přibližovaly. Správnost výsledného výpočtu bude ověřena na testovacích datech.

2.2.1 Sběr dat

Pro vývoj a testování výpočtu je potřeba sada dat, která byla v rámci práce složena z naměřených impulsních odezev v předchozí části, odezev získaných z veřejné banky *OpenAir* [26] a vygenerovaných impulsních odezev s využitím simulace. Sada musí být rozdělena na část určenou pro vývoj a část určenou pro výsledné ověření správnosti.

Simulované impulsní odezvy

K simulaci impulsních odezev byla využita knihovna *PyRoomAcoustics* [27], která umožňuje vytvořit model místnosti se stěnami sepsifikovaných materiálů, do prostoru virtuálně umístit mikrofon i zdroj zvuku a na základě vstupů vygenerovat impulsní odezvu místnosti.

Zde je uveden příklad kódu, který byl využit pro vygenerování jedné z odezev:

```
1 import pyroomacoustics as pra
2 import matplotlib.pyplot as plt
3 from scipy.io import wavfile
4 import numpy as np
5
6
7 room_dim = [5, 5, 3] # meters
8
9 m = pra.Material(energy_absorption="carpet_hairy")
10 pol = 1 * np.array([[0,0], [0,4], [5,2], [5,0]]) .T
11
12 room = pra.Room.from_corners(pol, fs=44100, max_order=15, materials
    =m, ray_tracing=True)
13
14 room.extrude(2.5, materials=m)
15 room.set_ray_tracing()
16 room.add_source([1, 1, 1.76])
17 room.add_microphone(loc=[1, 3, 1.76])
18
19 room.compute_rir()
20
21 wavfile.write('ir.wav', 44100, np.array(room.rir[0][0]))
```

Vygenerovaná odezva disponuje poměrně vysokým STI díky pohltivému materiálu "chlupatého koberece", jehož bylo využito pro všechny stěny místnosti. Vzhledem k tomu, že taková místnost je poměrně atypická bylo nejpraktičtější si ji nasimulovat.

Stejná místnost akorát s velmi odrazivým materiálem byla naopak využita k vygenerování impulsní odezvy s nízkým STI.

Výpočet STI z impulsních odezev

K výpočtu STI z nasbíraných impulsních odezev bylo využito STIPA algoritmu implementovaného v prostředí Matlab dostupného ze zdroje [28]. Výpočet probíhal automatizovaně tak, že se pro každou impulsní odezvu vypočetla konvoluce s měřícím signálem a výsledný signál se pak podrobil vyhodnocovací funkci, jejíž výsledek se následně uložil do JSON souboru pro snadné další zpracování.

```
1 duration = 25;
2
3 path='../..../dataset/';
4 sti = [];
5 dirfindstirng = strcat(path, '*.wav');
6 filelist = dir(dirfindstirng);
7
8 for i = 1 : length(filelist)
9
10     filename = strcat(path, filelist(i).name);
11     [ir, fs] = audioread(filename);
12     stipaSignal = generateStipaSignal(duration, fs)*0.1;
13     ir = ir * 0.1;
14     measured = conv(stipaSignal, ir);
15
16
17     sti = [sti stipa(measured, fs)];
18
19 end
20
21 json_text = jsonencode(sti);
22
23 fid = fopen(strcat(path, 'sti.json'),'w');
24 fprintf(fid, '%s', json_text);
25 fclose(fid);
```

2.2.2 Nástroj pro zpracování impulsních odezev

Ke zpracování impulsních odezev byl vytvořen vlastní skript, který posloužil k návrhu alternativní metody stanovení STI. Skript je koncipován nejen jako samostatně spustitelný program ale i jako integrovatelný modul. Tento modul s názvem *IRProcessor.py* obsahuje stejnojmennou třídu zapouzdřující veškerou integrovatelnou funkcionalitu.

Třída *IRProcessor* je navržena tak, aby každá její instance prováděla operace nad jednou impulsní odezvou.

Funkční požadavky:

- normalizace signálu impulsní odezvy
- filtrace oktávových pásem
- ořez signálu nad stanovenou hranicí ticha
- výpočet parametru definition - D50

Konstruktor

Konstruktor třídy přebírá instanční data prostřednictvím parametrů a připravuje instanci ke správnému chodu. Třída *IRProcessor* přebírá následující parametry:

- *signal* - obsahuje vzorkovaný signál (NumPy array)
- *sampling_frequency* - uvádí vzorkovací frekvenci, vnitřně *self.fs*
- *frame_size* - určuje šířku okna výpočtu obálky pomocí moving RMS
- *gate_threshold_dbfs* - určuje úroveň pro časový ořez signálu, vnitřně *self.gate_threshold*
- *custom_useful_duration_ms* - defaultně *None*, jinak nastavuje vlastní délku trvání oříznuté impulsní odezvy v milisekundách

Dále konstruktor provede výpočet maximální absolutní hodnoty *self.max_value* a počtu rámců *self.number_of_frames*:

```
1 self.max_value = np.max(np.abs(signal))
2 self.number_of_frames = -(signal.size // -self.frame_size)
```

Na závěr se provede volání procedur pro přípravu dat k dalšímu zpracování. Podrobný popis těchto procedur je uveden dále v této sekci:

```
1 self._compute_rms_frames()
2 self._compute_useful_part()
3 self._compute_filtered_signals()
```

Normalizace

Abychom mohli impulsní odezvy porovnávat mezi sebou, je nutné provést normalizaci podle nejvyšší hodnoty vzorku signálu. Upravíme tak rozsah hodnot vzorků na -1 až 1:

```
1 def normalize(self):
2     self.signal = self.signal / self.max_value
```

Filtrace oktávových pásem

Aby bylo možné signál analyzovat v jednotlivých oktávových pásmech, je třeba provést jejich filtraci. K tomu využijeme knihovny PyOctaveBand [29], která je pro tento účel navržena. Funkci knihovny zapouzdřuje privátní metoda `_compute_filtered_signals`:

```

1 from filtering.PyOctaveBand import octavefilter
2
3 def _compute_filtered_signals(self):
4     _, self.filter_frequencies, self.filtered_signals =
5         octavefilter(self.signal,
6                     fs = self.fs,
7                     fraction = FILTER_FRACTION,
8                     order = FILTER_ORDER,
9                     limits = FILTER_LIMITS,
10                    show = 0,
11                    sigbands = 1)

```

Konstanta `FILTER_FRACTION = 1` nastavuje šířku jednoho filtru na jednu oktávu, `FILTER_ORDER = 6` nastavuje řád filtru a `FILTER_LIMITS = [125, 8000]` nastavuje hranice, mezi kterými se mají oktávové filtry vypočítat. Tyto konstanty jsou nastaveny na začátku kódu modulu.

Výsledek se uloží jako 2D pole do proměnné `self.filtered_signals` a středové frekvence jednotlivých filtru uchová `self.filter_frequencies`.

Ořez signálu nad stanovenou hranicí ticha

Impulsní odezva nemusí začínat ani končit stejně s nahrávkou. Předpokládáme proto, že na začátku i na konci nahrávky je nepotřebné ticho, respektive šum, kterého je potřeba se zbavit. K tomu je nutné nejprve zhruba spočítat obálku signálu v decibelech a poté podle stanovené úrovně signál oříznout. Výpočet obálky v tomto případě zajistí *moving RMS* s převodem na dBFS:

```

1 def _to_dbFS(self, data):
2     return 20 * np.log10(np.abs(data) / self.max_value)
3
4 def _compute_rms_frames(self):
5     #compute RMS values
6     for i in range(self.number_of_frames):
7         #compute boundaries to slice frame
8         start = self.frame_size * i
9         end = self.frame_size * (i + 1)
10        end = self.signal.size if end >= self.signal.size else end
11        #slice frame from signal and convert to int64
12        frame = self.signal[start:end]
13        frame = frame.astype("float64")
14        #compute RMS
15        self.rms_frames[i] = np.sqrt(np.mean(frame**2))
16        self.rms_frames = self._to_dbFS(self.rms_frames)

```

Začátek ořezu se detekuje nárůstem RMS nad stanovenou hranici a konec poklesem pod ní. Výsledkem jsou vypočtené indexy `self.start_idx` a `self.end_idx`, které jsou při dalším zpracování využity ke *slicingu* původního signálu:

```

1 def _compute_useful_part(self):
2     gate_open = False
3     for i in range(self.rms_frames.size):

```

```

4         if not gate_open and self.rms_frames[i] > self.
gate_treshold:
5             cut_start_idx = (i + 1) * self.frame_size
6             gate_open = True
7
8         if gate_open and self.rms_frames[i] < self.gate_treshold:
9             cut_end_idx = i * self.frame_size
10            break
11    self.start_idx = cut_start_idx
12    self.end_idx = cut_end_idx

```

Výpočet parametru definition - D50 a DX

Jak se ukázalo ve fázi měření, podíl energie prvních 50 milisekund impulsní odezvy ku celkové energii pravděpodobně koreluje s STI. Proto se jevílo jako užitečné implementovat metodu, která by umožňovala provádět tento výpočet s několika modifikacemi. Implementace metody nese název *compute_DX*.

Z označení D50 bylo odvozeno označení DX, kde X symbolizuje proměnnou délku porovnávané úvodní energie. Parametr *custom_time_span* umožňuje výpočet provádět nad specifikovanou délkou odezvy a tím ignorovat automatický stříh vypočtený metodou *_compute_useful_part*. Parametr *octave_band_idx* umožňuje vybrat specifické oktávové pásmo, na kterém se má výpočet provést.

```

1 def compute_DX(self, x, custom_time_span = None, octave_band_idx =
None):
2     start = self.start_idx
3     end = self.end_idx if custom_time_span == None else self.
millisecods_to_sapmles(custom_time_span)
4     if octave_band_idx:
5         sig = self.filtered_signals[octave_band_idx]
6     else:
7         sig = self.signal
8     sig = sig[start:end]**2
9     t = self.millisecods_to_sapmles(x)
10    DX = np.sum(sig[:t]) / np.sum(sig)
11    return DX

```

Main

Jak je zmíněno v úvodu této podkapitoly, skript lze spouštět i jako samostatný program. V takovém režimu se provedou výše uvedené operace nad zvolenou impulsní odezvou a výsledky se zobrazí v grafech. Tento režim se hodí zejména pro ladění parametrů dalšího zpracování konkrétní sady impulsních odezev. Parametry běhu se pro jednoduchost nastavují jako konstanty přímo v kódu programu. Viz kód 2.1.

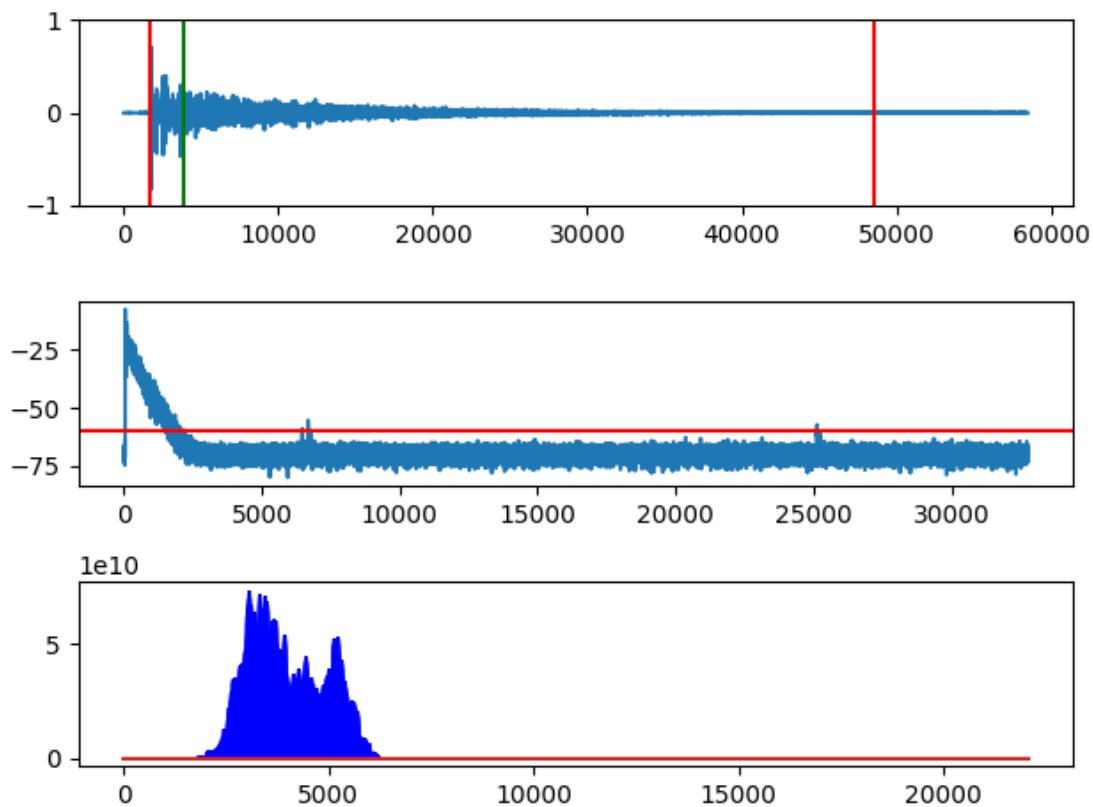
Od shora první graf na obrázku 2.6 zobrazuje normalizovaný časový průběh nahrávky (modře), hranice automatického stříhu (červeně) a hranici prvních X milisekund odezvy (zeleně). Druhý graf pak zobrazuje hodnoty RMS rámců (modře) a hranici pro automatický stříh (červeně). Poslední, dolní graf zobrazuje frekvenční spektrum vybraného oktávového pásma.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.io import wavfile
4 from scipy import fft
5
6 IR_FILE_PATH = 'dataset/flight arena mls.wav'
7 USEFUL_TRESHOLD_DBFS = -60
8 BAND_IDX = 5
9 X = 50
10 TAIL = 10000
11 FRAME_SIZE = 32
12
13 #ZDE SE NACHAZI KOD IRProcessoru
14
15 if __name__ == "__main__":
16
17     fs, ir = wavfile.read(IR_FILE_PATH)
18     sp = IRProcessor(ir, fs, FRAME_SIZE, USEFUL_TRESHOLD_DBFS)
19     sp.normalize()
20
21     plt.subplot(311)
22     ax = plt.gca()
23     ax.set_ylim([-1, 1])
24     plt.plot(sp.signal[:sp.end_idx + TAIL])
25
26     plt.axvline(x = sp.start_idx, color = 'r', label = 'cut start')
27     plt.axvline(x = sp.end_idx, color = 'r', label = 'cut end')
28     plt.axvline(x = sp.start_idx + sp.milliseconds_to_samples(X),
29                 color = 'g', label = f'{X} ms')
30
31     plt.subplot(312)
32
33     plt.plot(sp.rms_frames)
34     plt.axhline(y = USEFUL_TRESHOLD_DBFS, color = 'r', label = '
35     gate threshold')
36
37     print(f"D{X} = {sp.compute_DX(X)}")
38     plt.subplot(313)
39
40     plt.stem(np.linspace(0, int(fs/2), int(sp.signal.size/2)),
41              np.abs(fft.fft(sp.filtered_signals[BAND_IDX]))[:int(sp
42              .signal.size/2)]),
43              'b-', markerfmt=" ")
44
45     plt.tight_layout()
46     plt.show()

```

Kód 2.1: IRProcessor main



Obrázek 2.6: Výstup modulu IRProcessor.py spuštěného jako samostatný program

2.2.3 Návrh výpočtu

Nejprve byl implementován skript pro automatizovaný výpočet D50. Metoda *direct_autocut_D50_alternative_sti* zajišťuje samotný výpočet D50 s využitím dříve vytvořeného modulu *IRProcessor*.

Kód hlavního běhu programu provede nejprve načtení předvypočtených hodnot STI z JSON souboru. Poté ve smyčce postupně načte všechny impulsní odezvy a pro každou z nich provede výpočet D50, jehož výstup uloží do listu *results*. Po skončení smyčky se vypočtou odchylky odpovídajících hodnot STI (proměnná *sti*) a D50 (proměnná *results*), ze kterých následně vypočítá maximální a průměrnou odchylku. Vypočtené hodnoty vypíše a nakonec data zobrazí v grafech.

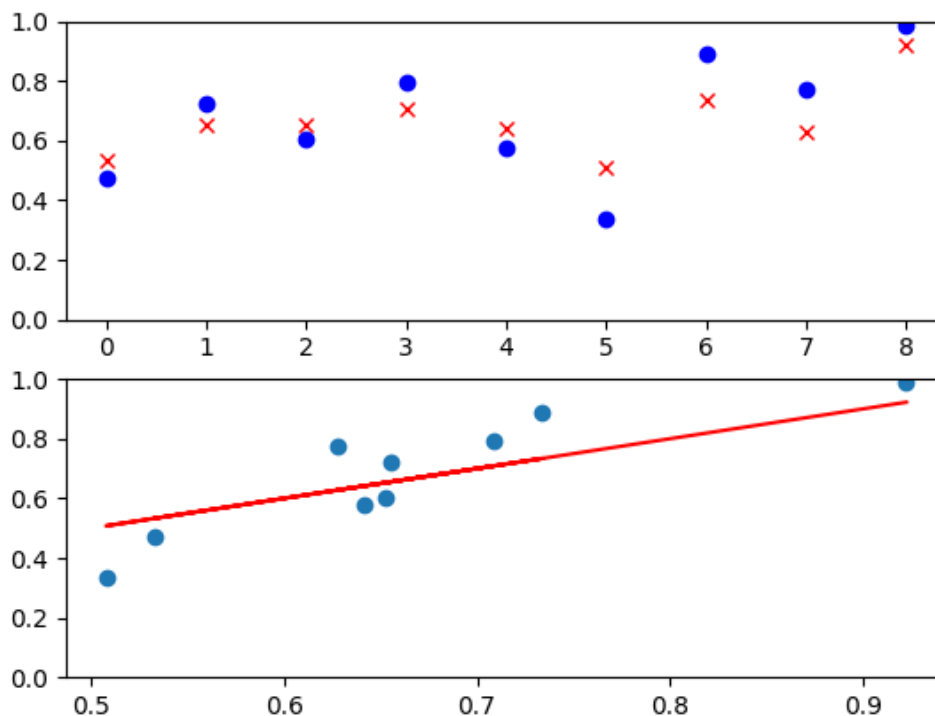
Horní graf ukazuje hodnoty STI a hodnoty D50 v závislosti na pořadí impulsní odezvy a dolní graf ukazuje hodnoty D50 v závislosti na STI.

```

1 from IRProcessor import IRProcessor
2 from scipy.io import wavfile
3 import glob
4 import numpy as np
5 import json
6 import matplotlib.pyplot as plt
7
8 USEFUL_TRESHOLD_DBFS = -60
9
10 def direct_autocut_D50_alternative_sti(ir_file_name):
11     fs, ir = wavfile.read(ir_file_name)
12     sp = IRProcessor(ir, fs, 32, USEFUL_TRESHOLD_DBFS)
13     sp.normalize()
14     D50 = sp.compute_DX(x=50)
15     return D50
16
17 if __name__ == "__main__":
18     results = []
19     dataset = json.load(open('./dataset/sti.json'))
20     sti = dataset
21
22     for ir_file_name in glob.glob('./dataset/*.wav'):
23         D50 = direct_autocut_D50_alternative_sti(ir_file_name)
24         results.append(D50)
25
26     sti = np.array(sti)
27     results = np.array(results)
28
29     d = np.abs(sti - results)
30     print(d)
31     print(np.average(d))
32     print(np.max(d))
33
34     plt.subplot(211)
35     ax = plt.gca()
36     ax.set_ylim([0,1])
37     plt.plot(r, 'bo')
38     plt.plot(sti, 'rx')
39
40     plt.subplot(212)
41     ax = plt.gca()
42     ax.set_ylim([0,1])
43     plt.scatter(sti, r)
44     plt.scatter(sti, sti)

```

Po spuštění výpočtu nad sadou dat určenou pro vývoj získáme výstup viditelný na obrázku 2.7 s průměrnou odchylkou 9.5% a maximální odchylkou 17%. Z výstupu je vidět, že D50 určitou informaci o STI pravděpodobně nese, ale pro zpřesnění je vhodné vypracovat sofistikovanější výpočet.



Obrázek 2.7: Porovnání hodnot D50 a STI

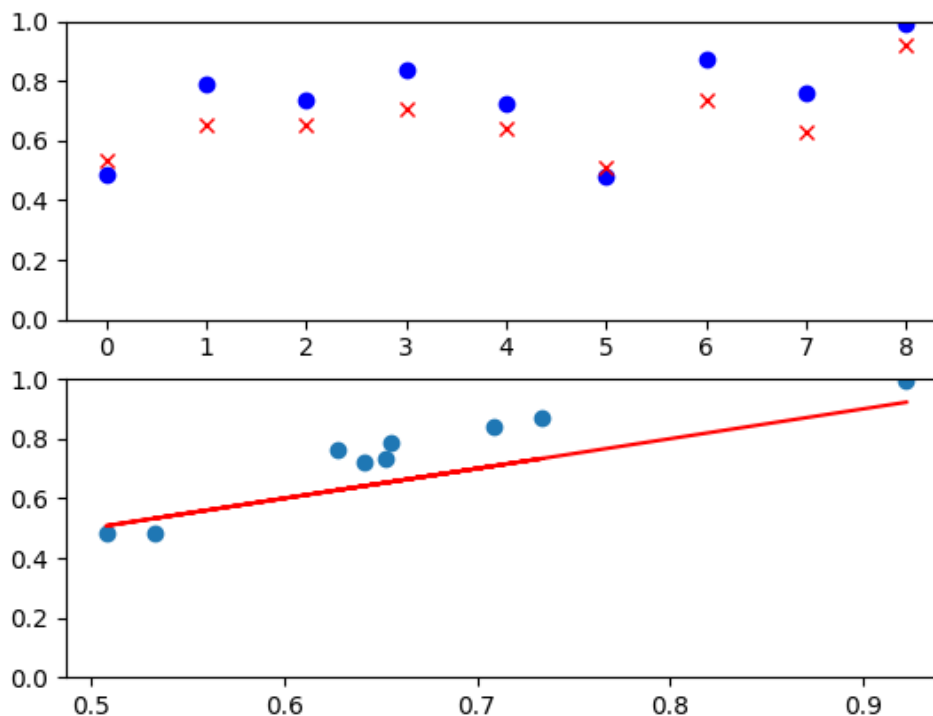
Další vývoj výpočtu se řídil následující úvahou. Nejprve připomeňme, že STIPA vyhodnocuje ztrátu hloubky modulace na dvou modulačních frekvencích pro každé ze sedmi oktávových pásem. Z principu výpočtu konvoluce můžeme usuzovat, že distribuce energie impulsní odezvy má jistý vliv na ztrátu modulace.

Výpočet D50 sleduje energii v prvních 50. milisekundách, což odpovídá půlperiodě frekvence 10 Hz. To je druhá modulační frekvence testovaná na oktávovém pásmu se středovou frekvencí 1000 Hz. Zdálo se tedy vhodné zjistit, jaké budou výsledky, pokud na impulsní odezvě provedeme filtraci oktávových pásem, na každém z nich vypočteme poměr počáteční energie vymezené časem půlperiody příslušné modulační frekvence ku celkové energii a z těchto poměrů vypočteme aritmetický průměr. K vymezení počáteční energie využijeme druhých modulačních frekvencí, neboť periody prvních modulačních frekvencí jsou pro výběr počáteční energie moc dlouhé.

Této metodě výpočtu říkáme *D05* jako *Definition z 0.5 periody modulační frekvence na oktávovém pásmu*. Výpočet implementuje následující kód:

```
1 from IRProcessor import IRProcessor
2 from scipy.io import wavfile
3 import glob
4 import numpy as np
5 import json
6 import matplotlib.pyplot as plt
7
8 STIPA_MOD_FREQUENCIES = [
9 [125, 1.6, 8 ],
10 [250, 1, 5 ],
11 [500, 0.63, 3.15],
12 [1000, 2, 10 ],
13 [2000, 1.25, 6.25],
14 [4000, 0.8, 4 ],
15 [8000, 2.5, 12.5]
16 ]
17
18 def average_DT05_of_STIPA_modfreqs_and_octavebands_alternative_sti(
19     ir_file_name):
20     fs, ir = wavfile.read(ir_file_name)
21     sp = IRProcessor(ir, fs, 32, USEFUL_TRESHOLD_DBFS)
22     sp.normalize()
23
24     dx_results = []
25
26     for band in range(7):
27         for i in range(1):
28             mod_frequency = STIPA_MOD_FREQUENCIES[band][i + 2]
29             T = (1/mod_frequency)*1000
30             X = round(T/2)
31             DX = sp.compute_DX(x=X, octave_band_idx=band)
32             dx_results.append(DX)
33
34     dx_results = np.array(dx_results)
35     print(dx_results)
36     dx_avg = np.average(dx_results)
37     return dx_avg
38
39 if __name__ == "__main__":
40     r = []
41     dataset = json.load(open('./dataset/sti.json'))
42     sti = dataset
43
44     for ir_file_name in glob.glob('./dataset/*.wav'):
45         DT05_avg =
46         average_DT05_of_STIPA_modfreqs_and_octavebands_alternative_sti(
47             ir_file_name
48         )
49         results.append(DT05_avg)
50 # KOD PROKRACUJE STEJNE JAKO V PREDCHOZI UKAZCE
```

Spuštěním skriptu na datech pro vývoj získáme výstup na obrázku 2.8, s průměrnou odchylkou 9% a maximální odchylkou 14%.



Obrázek 2.8: Porovnání hodnot D05 a STI

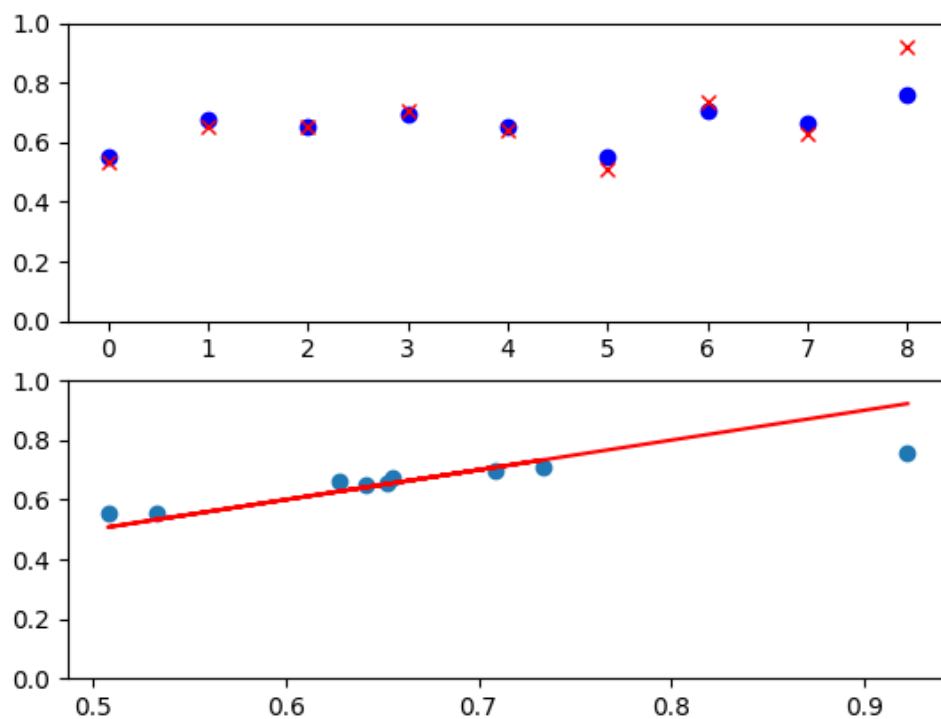
Vypočtená data ukazují horší výsledek co se odchylek týče. Z grafů je nicméně vidět, že vypočtené hodnoty lépe kopírují křivku hodnot STI. Další vývoj byl proto směřován k nalezení funkce, která by odchylku D05 od naměřených hodnot STI kompenzovala.

Návrh kompenzační funkce

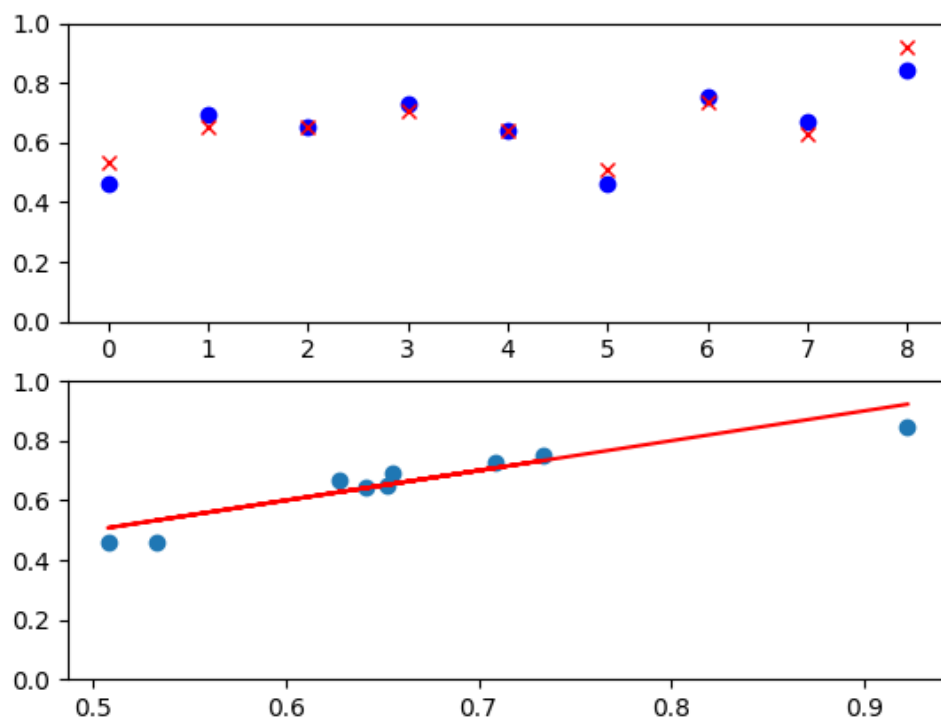
Alternativní výpočet by měl umožnit alespoň odhad kategorie měřeného prostoru. Z tabulky STI je vidět, že do kategorií zařazujeme hodnoty v rozsahu 0.36 až 0.76. Je proto vhodné, aby obor hodnot kompenzační funkce tomuto rozsahu odpovídal na definičním oboru 0 až 1. Nabízelo se proto nejprve vyzkoušet lineární funkci přesně s tímto rozsahem. Tu lze popsat následujícím polynomem:

$$y(x) = 0.36 + 0.4x \quad (2.1)$$

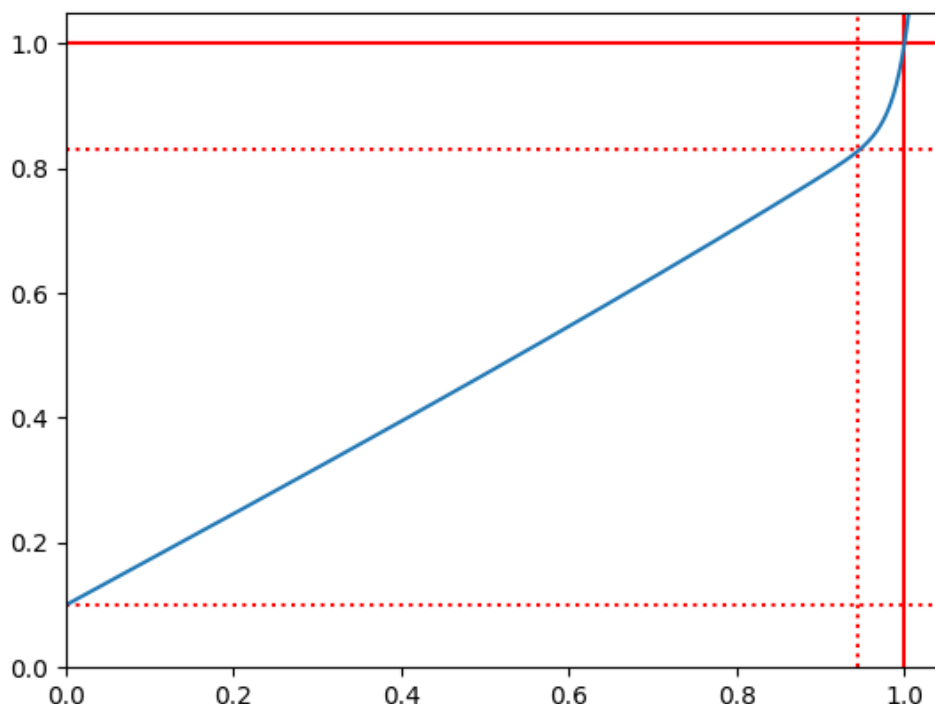
Podrobíme-li hodnoty D05 této funkci, získáme výsledek zobrazený na obrázku 2.9, kde průměrná odchylka činí 4% a maximální 16%.



Obrázek 2.9: Výsledky s využitím lineární kompenzační funkce



Obrázek 2.10: Výsledky s využitím lineární kompenzační funkce s větším rozsahem



Obrázek 2.11: Graf kompenzační funkce popsané polynomem vyššího řádu

Ačkoliv se podařilo získat výsledky s poměrně nízkou průměrnou odchylkou, maximální odchylka je stále neuspokojivě veliká. To je způsobeno zejména vlivem malého rozsahu použité kompenzační funkce. Iterativní úpravou koeficientů uvedeného polynomu s cílem rozšířit rozsah výpočtu se podařilo najít následující funkci:

$$y(x) = 0.1 + 0.75x \quad (2.2)$$

S využitím této funkce dosáhneme průměrné odchylky cca 4% a maximální odchylky 8%. Výsledky zobrazuje obrázek 2.10. Rozsah byl tímto rozšířen na 0.1 až 0.85. Výpočet dosahuje stále vysoké maximální odchylky na impulsních odezvách s vyšším STI. Dalším krokem proto bylo najít polynom vyššího řádu, který by odchylku pro tyto odezvy lépe kompenzoval. Vzhledem k úspěšnosti kompenzace odchylky u ostatních impulsních odezav byla snaha tvar předchozí funkce co nejvíce zachovat. Manuálním experimentováním s různými řády a koeficienty s využitím vizuální zpětné vazby a představy výsledného tvaru byl namodelován následující polynom, jemuž odpovídá průběh na obrázku 2.11:

$$y = 0.01 + 0.73x + 0.04x^3 + 0.13x^{60} \quad (2.3)$$

S využitím tohoto polynomu získáme výsledky zobrazené na obrázku 2.12, kde průměrná odchylka činí 3% a maximální 7,5%. Na první pohled vypadá polynom poněkud divoce kvůli vysokému řádu 60. Proto se jevílo jako vhodné jej nahradit polynomem nižšího řádu. Toho bylo dosaženo polynomiální regresí s využitím následujícího skriptu:

```

1 x = np.linspace(0, 1.01, 10000)
2 values = 0.1 + 0.73 * x + 0.04*x**3 + 0.13*x**60
3 alt_poly = np.polyfit(x, values, 13)
4 print(alt_poly)
5
6 alt_values = np.polyval(alt_poly, x)
7
8 ax = plt.gca()
9 ax.set_ylim([0,1])
10 ax.set_xlim([0,1])
11
12 plt.plot(x, values)
13 plt.plot(x, alt_values, '--')
14 plt.show()

```

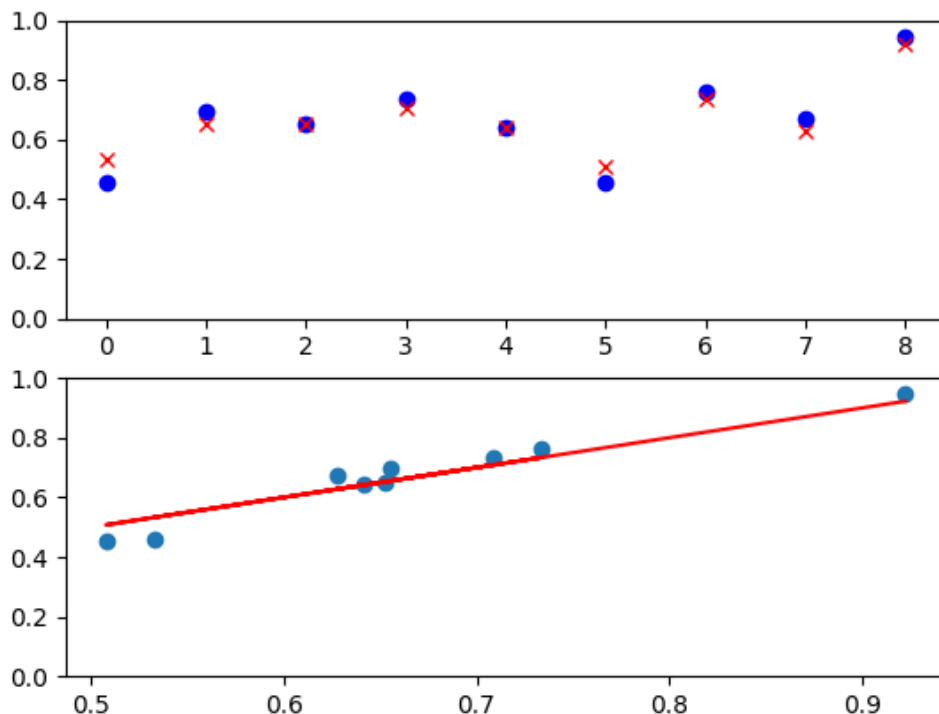
Polynom řádu 12 již nenabýval dostatečné přesnosti, proto byl pro regresi zvolen polynom řádu 13. Výstup ukazuje vypočtené koeficienty:

```

1 [ 1.77123845e+04 -1.00940123e+05  2.51854001e+05 -3.61417161e+05
2   3.29518352e+05 -1.99016494e+05  8.05494457e+04 -2.16171712e+04
3   3.72194494e+03 -3.86032152e+02  2.15412263e+01  2.12221436e-01
4   1.03047965e-01]

```

Náhradní polynom je na první pohled o poznání komplikovanější vlivem vyššího počtu členů. Z důvodu lepší přehlednosti kódu budeme nadále využívat původního polynomu.



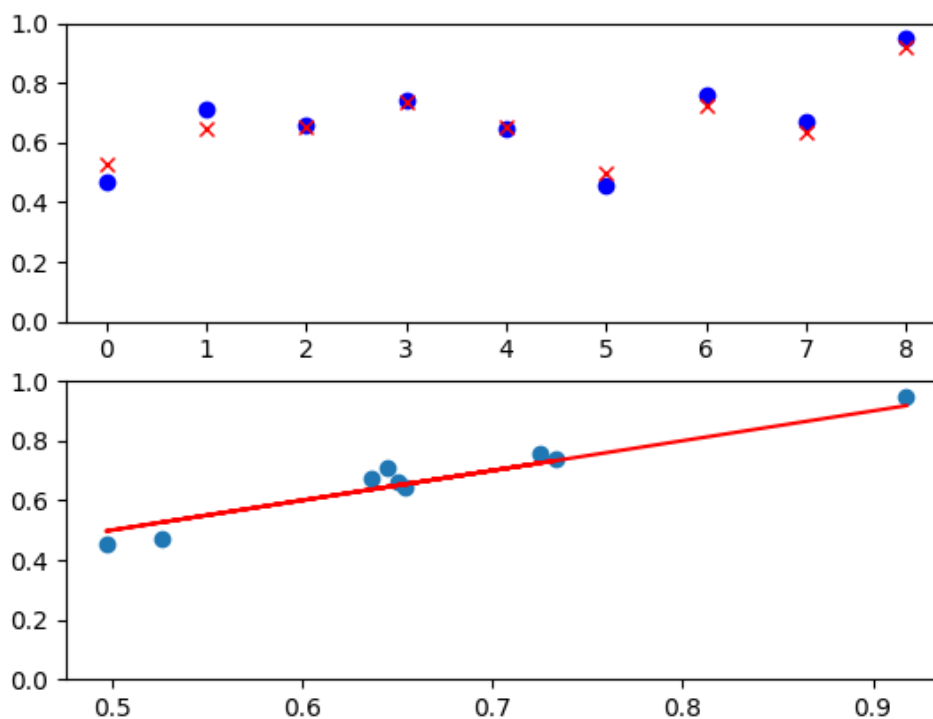
Obrázek 2.12: Výsledky s užitím polynomu vyššího řádu

Váhování

V této fázi se již výpočet jevil býti poměrně přesný. Bylo nicméně vhodné ověřit, zda se výpočet ještě nezpřesní váhováním jednotlivých pásem. K tomu bylo využito váhování převzaté z normy [11] pro výpočet STIPA.

```
1 WEIGHTS = np.array([0.085, 0.127, 0.23, 0.23, 0.309, 0.224, 0.173])
```

Výsledek zobrazuje obrázek 2.13. Průměrná odchylka v tomto případě činí 3% a maximální 6.5%. Vliv váhování je tedy minimální, došlo pouze ke zmírnění maximální odchylky.



Obrázek 2.13: Výsledky výpočtu s váhováním

Výsledný kód výpočtu

```

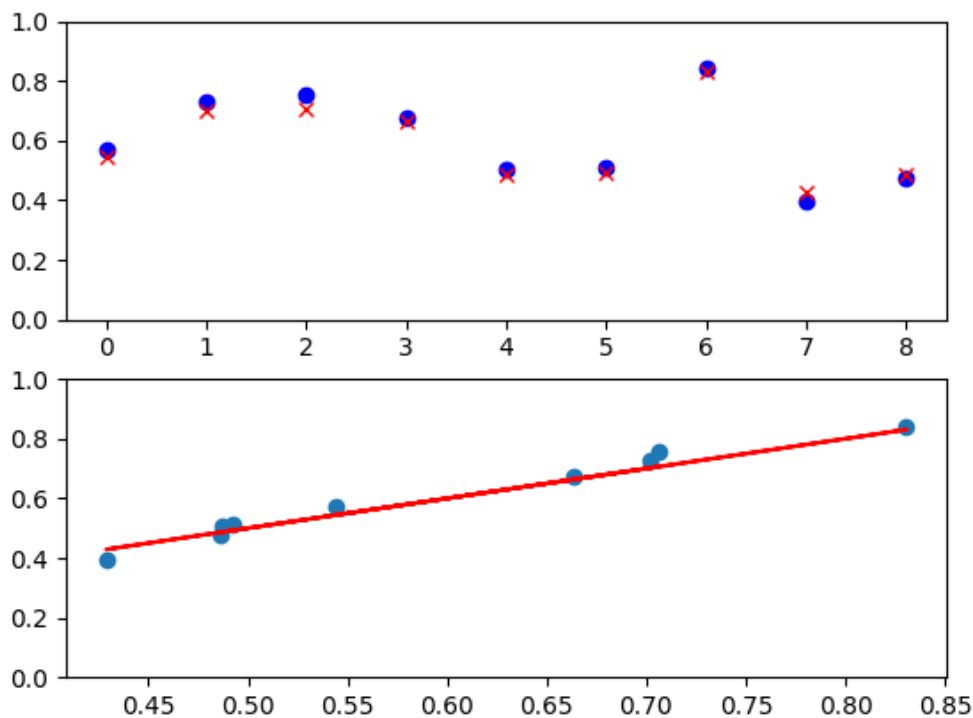
1 from IRProcessor import IRProcessor
2 from scipy.io import wavfile
3 import glob
4 import numpy as np
5 import json
6 import matplotlib.pyplot as plt
7
8 USEFUL_TRESHOLD_DBFS = -60
9 STIPA_MOD_FREQUENCIES = [
10 [125, 1.6, 8 ],
11 [250, 1, 5 ],
12 [500, 0.63, 3.15],
13 [1000, 2, 10 ],
14 [2000, 1.25, 6.25],
15 [4000, 0.8, 4 ],
16 [8000, 2.5, 12.5]
17 ]
18
19 WEIGHTS = np.array([0.085, 0.127, 0.23, 0.23, 0.309, 0.224, 0.173])
20
21 def average_DT05_of_STIPA_modfreqs_and_octavebands_alternative_sti(
22     ir_file_name):
23     fs, ir = wavfile.read(ir_file_name)
24     sp = IRProcessor(ir, fs, 32, USEFUL_TRESHOLD_DBFS)
25     sp.normalize()
26
27     dx_results = []
28
29     for band in range(7):
30         for i in range(1):
31             mod_frequency = STIPA_MOD_FREQUENCIES[band][i + 2]
32             T = (1/mod_frequency)*1000
33             X = round(T/2)
34             DX = sp.compute_DX(x=X, octave_band_idx=band)
35             dx_results.append(DX)
36
37     dx_results = np.array(dx_results)
38     print(dx_results)
39     dx_avg = np.average(dx_results, weights=WEIGHTS)
40     return dx_avg
41
42 if __name__ == "__main__":
43     results = []
44     dataset = json.load(open('./dataset/sti.json'))
45     sti = dataset
46
47     for ir_file_name in glob.glob('./dataset/*.wav'):
48         print(ir_file_name)
49         DT05_avg =
50         average_DT05_of_STIPA_modfreqs_and_octavebands_alternative_sti(
51             ir_file_name)
52         r5 = 0.1 + 0.73 * DT05_avg + 0.04*DT05_avg**3 + 0.13*
53         DT05_avg**60
54         print(r5)
55         results.append(r5)
56
57     sti = np.array(sti)

```

```
54 results = np.array(results)
55
56 d = np.abs(sti - results)
57 print(d)
58 print(np.average(d))
59 print(np.max(d))
60
61 plt.subplot(211)
62 ax = plt.gca()
63 ax.set_ylim([0,1])
64 plt.plot(results, 'bo')
65 plt.plot(sti, 'rx')
66
67 plt.subplot(212)
68 ax = plt.gca()
69 ax.set_ylim([0,1])
70 plt.plot(sti, sti, 'r-')
71 plt.scatter(sti, results)
72
73 plt.show()
```

2.3 Testování výpočtu

Správnost výpočtu byla následně ověřena na testovací sadě dat. Jak ukazuje obrázek 2.14, výpočet obstál dokonce s lepšími výsledky než na vývojové sadě. Průměrná odchylka činí 2% a maximální odchylka činí 5%. Tímto je správnost výpočtu ověřena.



Obrázek 2.14: Výsledky výpočtu z testovacích dat

Závěr

Cílem práce bylo navrhnout alternativní výpočet STI z impulsních odezev, který by byl jednodušší než výpočty stávající. Za účelem toho bylo provedeno měření impulsních odezev a STI různých interiérů budov. Během vyhodnocení došlo ke srovnání různých vlastností impulsních odezev s naměřenými hodnotami, na základě čehož se podařilo identifikovat souvislost mezi parametrem D50 a STI.

Naměřené impulsní odezvy posloužily jako základ pro vývojovou a testovací datovou sadu. Pro potlačení vlivů vlastního měření ve vývoji výpočtu bylo žádoucí datovou sadu vytvořit z různých zdrojů, tedy některé impulsní odezvy převzít a některé vygenerovat s využitím simulace. Ke zpracování impulsních odezev byl vytvořen vlastní nástroj, který zajišťoval automatizovaný ořez, normalizaci a filtraci oktávových pásem.

Samotný návrh výpočtu vycházel z podobnosti hodnot D50 a STI identifikované ve fázi měření. Z úvahy o vlivu distribuce energie impulsní odezvy na ztrátu modulace a souvislosti 50 ms s jednou z modulačních frekvencí STIPA byl navržen výpočet D05, jehož výstup vykazoval podobný trend jako výstupní hodnoty STIPA. Ke zpřesnění výpočtu bylo nicméně potřeba navrhnout funkci, která by zmenšila odchylky odpovídajících hodnot D05 a STI. Tuto funkci se podařilo najít experimentováním s různými lineárními funkcemi a následně zpřesněním polynomem vyššího řádu. Ve výpočtu se nakonec využilo i váhování oktávových pásem převzaté z normy [11]. Na závěr byla správnost výpočtu ověřena na testovacích datech.

Cíl práce se podařilo naplnit. Vlastní výpočet je v porovnání se STIPA jednodušší a na testovaných datech podává srovnatelné výsledky. Nutno ovšem podotknout, že vývoj a testování výpočtu, který by byl spolehlivostí a přesností srovnatelný se STIPA, vyžaduje užití sofistikovanějších metod a daleko větší datové sady, než je v rámci práce této povahy možné. Výstupy lze proto vnímat spíše jako podnět pro další výzkum.

Bibliografie

1. *Python website*. Dostupné také z: <https://www.python.org/>.
2. *NumPy website*. Dostupné také z: <https://numpy.org/>.
3. *SciPy website*. Dostupné také z: <https://scipy.org/>.
4. *Matplotlib website*. Dostupné také z: <https://matplotlib.org/>.
5. ING. ROMAN BERKA, Ph.D.; ING. FRANTIŠEK RUND, Ph.D.; HUSNÍK, Dr. Ing. Libor; ING. ADAM SPORKA, Ph.D. *Multimédia 1*. České Vysoké Učení Technické v Praze, 2018. ISBN 978-80-01-05859-6.
6. ZOLZER, Udo. *DAFX: Digital Audio Effects*. 2nd. Wiley Publishing, 2011. ISBN 0470665998.
7. BRINTZENHOFE, Kurt T. *Investigating Human Interaction through Mathematical Analysis: The Queue Transform*. Taylor & Francis Group, 2023. Dostupné také z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=7244898>.
8. TARR, Eric. *Hack Audio : An Introduction to Computer Programming and Digital Signal Processing in MATLAB*. Taylor & Francis Group, 2018. Dostupné také z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=5439065>.
9. *Simple DFT in C*. Dostupné také z: <https://batchloaf.wordpress.com/2013/12/07/simple-dft-in-c/>.
10. *Tutorialspoint - Amplitude modulation*. Dostupné také z: https://www.tutorialspoint.com/analog_communication/analog_communication_amplitude_modulation.htm.
11. ČSN EN IEC 60268-16 ED.3 (368305) *Elektroakustická zařízení - Část 16: Objektivní hodnocení srozumitelnosti řeči indexem přenosu řeči*. [B.r.].
12. *Envelopepaper*. Dostupné také z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7891054>.
13. PARK, Tae Hong. *Introduction to Digital Signal Processing: Computer Musically Speaking*. World Scientific Publishing Co., 2009. ISBN 9789812790279. Dostupné z DOI: 10.1142/6705.
14. *Fourier deconvolution*. Dostupné také z: <https://terpconnect.umd.edu/~toh/spectrum/Deconvolution.html>.
15. SÝKORA, B. *Stereofonie v praxi*. SNTL - Státní nakladatelství technické literatury, 1980. Polytechnická knihovna. Řada II.
16. ČSN EN ISO 3382-1 - *Akustika - Měření parametrů prostorové akustiky - Část 1: Prostory pro přednes hudby a řeči*. 2009.

17. DAVIDS, Larson. *Obrázek měření T20 přerušovaným šumem*. Dostupné také z: <https://www.larsondavis.com/learn/building-acoustics/Reverberation-Time-in-Room-Acoustics>.
18. *REW Help*. Dostupné také z: https://www.roomeqwizard.com/help/help_en-GB/html/graph_rt60.html.
19. *MLS Tool*. Dostupné také z: <https://wavearts.com/products/plugins/mlstool>.
20. *MLS AAT8*. Dostupné také z: https://github.com/ignaciodesimon/MLS_AAT8.
21. HEE, Jens. *Impulse response measurements using MLS*. 2003. Dostupné také z: <https://jenshee.dk/signalprocessing/mls.pdf>.
22. *Room Eq Wizard*. Dostupné také z: <https://www.roomeqwizard.com/>.
23. AUDIO, NTi. *Obrázek dvanáctistěnného reproduktoru*. Dostupné také z: <https://www.nti-audio.com/en/products/noise-sources/ds3-dodecahedron-speaker>.
24. EARTHWORKS. *Obrázek měřicího mikrofonu*. Dostupné také z: <https://earthworksaudio.com/measurement-microphones/m50/>.
25. WIJNGAARDEN SJ, van; R, Drullman. *Binaural intelligibility prediction based on the speech transmission index*. 2008. Dostupné z DOI: 10.1121/1.2905245.
26. *Open Air*. Dostupné také z: <https://www.openair.hosted.york.ac.uk/>.
27. SCHEIBLER, Robin; BEZZAM, Eric; DOKMANIC, Ivan. *Pyroomacoustics: A Python package for audio room simulations and array processing algorithms*. *CoRR*. 2017, roč. abs/1710.04196. Dostupné z arXiv: 1710.04196.
28. ZÁVIŠKA, Pavel. *STIPA - Implementace v prostředí MATLAB*. 2023. Dostupné také z: <https://github.com/zawi01/stipa>.
29. PLENS, Jose M. Requena. *PyOctaveBand repo*. 2020. Dostupné také z: <https://github.com/jmrplens/PyOctaveBand/blob/master/>.

Přílohy

A Tabulka STI

Category	Nominal STI value	Type of message information	Examples of typical uses (for natural or reproduced voice)	Comment
A+	> 0,76		Recording studios	Excellent intelligibility but rarely achievable in most environments
A	0,74	Complex messages, unfamiliar words	Theatres, speech auditoria, parliaments, courts, Assistive Hearing Systems (AHS)	High speech intelligibility
B	0,7	Complex messages, unfamiliar words		
C	0,66	Complex messages, unfamiliar words	Theatres, speech auditoria, teleconferencing, parliaments, courts	High speech intelligibility
D	0,62	Complex messages, familiar words	Lecture theatres, classrooms, concert halls	Good speech intelligibility
E	0,58	Complex messages, familiar context	Concert halls, modern churches	High quality PA systems
F	0,54	Complex messages, familiar context	PA systems in shopping malls, public buildings offices, VA systems, cathedrals	Good quality PA systems
G	0,5	Complex messages, familiar context	Shopping malls, public buildings offices, VA systems	Target value for VA systems
H	0,46	Simple messages, familiar words	VA and PA systems in difficult acoustic environments	Normal lower limit for VA systems
I	0,42	Simple messages, familiar context	VA and PA systems in very difficult spaces	Limited intelligibility
J	0,38		Not suitable for PA systems	
U	< 0,36		Not suitable for PA systems	

These values should be regarded as minimum target values.

NOTE 1 Perceived intelligibility relating to each category also depend on the frequency response at each listening position.

NOTE 2 The STI values refer to measured values in sample listening positions or as required by specific application standards.

Obrázek 15: Převzato z normy [11]

B Převzaté impulsní odezvy z OpenAir

- **Falkland Palace Royal Tennis Court:**
 - autor: Dr. Damian T. Murphy
 - odkaz: https://www.openair.hosted.york.ac.uk/?page_id=476
- **Innocent Railway Tunnel**
 - autoři: Andrew Chadwick, Simon Shelley
 - odkaz: https://www.openair.hosted.york.ac.uk/?page_id=525
- **Lady Chapel, St Albans Cathedral**
 - autoři: Marcin Gorzel, Gavin Kearney, Aglaia Foteinou, Sorrel Hoare, Simon Shelley
 - odkaz: https://www.openair.hosted.york.ac.uk/?page_id=595
- **Spokane Woman's Club**
 - autor: James Cadwallader
 - odkaz: https://www.openair.hosted.york.ac.uk/?page_id=659
- **Stairway, University of York**
 - autor: Simon Shelley
 - odkaz: https://www.openair.hosted.york.ac.uk/?page_id=678
- **Tvisöngur Sound Sculpture, Iceland (Model)**
 - autoři: Oliver McIntyre
 - odkaz: https://www.openair.hosted.york.ac.uk/?page_id=752