



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# ŘÍDKÉ APROXIMACE RELATIVNÍCH IMPULZNÍCH ODEZEV V ZÁVISLOSTI NA DOBĚ DOZVUKU

## Bakalářská práce

*Studijní program:* B2646 – Informační technologie  
*Studijní obor:* 1802R007 – Informační technologie  
*Autor práce:* **Tomáš Franěk**  
*Vedoucí práce:* doc. Ing. Zbyněk Koldovský, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# SPARSE APPROXIMATIONS OF RELATIVE IMPULSE RESPONSES DEPENDING ON THE REVERBERATION TIME

**Bachelor thesis**

*Study programme:* B2646 – Information Technology  
*Study branch:* 1802R007 – Information Technology  
*Author:* **Tomáš Franěk**  
*Supervisor:* doc. Ing. Zbyněk Koldovský, Ph.D.



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš Franěk**  
Osobní číslo: **M12000126**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Řídké aproximace relativních impulzních odezev v závislosti na době dozvuku**  
Zadávací katedra: **Ústav informačních technologií a elektroniky**

### Z á s a d y   p r o   v y p r a c o v á n í :

- [1] Seznamte se se způsoby výpočtu relativní impulzní odezvy mezi dvěma mikrofony a zdrojem akustického signálu pomocí metody nejmenších čtverců. Vytvořte si vhodnou databázi nahrávek v reálných prostředích s různou délkou dozvuku ( $T_{60}$ ).
- [2] Zvolte metody pro výpočet řídkých aproximací relativních impulzních odezev, zejména metody pro výpočet vážené úlohy LASSO.
- [3] Najděte vhodnou parametrizaci váhovací funkce, která je, mimo jiné, závislá na délce dozvuku.
- [4] Experimenty ověřte vhodnost (optimalitu) zvolené parametrizace váhovací funkce.

Rozsah grafických prací: Dle potřeby dokumentace

Rozsah pracovní zprávy: cca 30 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] I. Tashev, "Sound Capture and Processing" Practical Approaches", Wiley, 2008.
- [2] J. Málek and Z. Koldovský, "Sparse Target Cancellation Filters with Application to Semi-Blind Noise Extraction," Proc. of the 41st IEEE International Conference on Audio, Speech, and Signal Processing (ICASSP 2014), Florence, Italy, pp. 2128-2132, May 2014.

Vedoucí bakalářské práce: doc. Ing. Zbyněk Koldovský, Ph.D.  
Ústav informačních technologií a elektroniky

Datum zadání bakalářské práce: 12. září 2014

Termín odevzdání bakalářské práce: 15. května 2015

prof. Ing. Václav Kopecký, CSc.  
děkan



prof. Ing. Zdeněk Plíva, Ph.D.  
vedoucí ústavu

V Liberci dne 12. září 2014

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 15.5.2015

Podpis: *František*

## **Poděkování**

Tímto bych chtěl poděkovat doc. Ing. Zbyňkovi Koldovskému, Ph.D. za cenné rady, vstřícnost při konzultacích a za zapůjčení profesionálních mikrofonů.

## **Abstrakt**

Tato práce se zabývá hledáním vhodné parametrizace váhovací funkce pro výpočet řídké aproximace relativní impulsní odezvy pomocí vážené úlohy LASSO v závislosti na vlastnostech akustického prostředí místnosti.

Cílem práce je nalézt vzorec, který nalezne optimální parametry váhovací funkce. Optimální řídká relativní impulsní odezva získaná výpočtem pomocí těchto parametrů musí splňovat dvě podmínky. První podmínkou je, aby obsahovala co nejméně nenulových prvků. Druhou podmínkou je, aby se její schopnost potlačit vzájemně zvukové kanály příliš nelišila od schopnosti relativní impulsní odezvy spočtené pomocí metody nejmenších čtverců. V práci je řešen kompromis mezi těmito podmínkami.

Nakonec je pomocí experimentu ověřena optimalita řídké relativní impulsní odezvy spočtené pomocí vážené úlohy LASSO s parametry váhovací funkce získané pomocí nalezeného vzorce.

### **Klíčová slova:**

relativní impulsní odezva, řídká relativní impulsní odezva, vážená úloha LASSO, parametrizace váhovací funkce, doba dozvuku

## **Abstract**

In this thesis, an optimum parameterization of the weighting function for sparse approximation of relative impulse responses through weighted LASSO depending on the room acoustics is described.

The aim of the thesis is to find optimal parameterization of the weighting function. The optimal sparse relative impulse response approximation has to meet two conditions. The first condition is that the number of its nonzero coefficients is minimum. The second condition is that its capability to suppress a target signal should not be too different from that capability of relative impulse response computed by least mean squares. In this thesis, a trade-off between these two conditions is sought.

The optimality of sparse relative impulse response computed through weighted LASSO with parameters of the weighting function is verified by experiments.

### **Key words:**

relative impulse response, sparse relative impulse response, weighted LASSO, parameterization of weighting function, reverberation time

## Obsah

1 Úvod.....	11
2 Výpočet relativní impulsní odezvy.....	12
2.1 Výpočet relativní impulsní odezvy pomocí metody nejmenších čtverců (LMS). 12	
2.2 Výpočet řídké relativní impulsní odezvy pomocí vážené metody LASSO.....	13
3 Akustické vlastnosti místnosti.....	15
3.1 Doba dozvuku místnosti (T60).....	15
3.2 Výpočet early decay time (EDT).....	23
3.3 Výpočet C80.....	29
3.4 Výpočet D20 a D50.....	33
4 Hledání vhodné parametrizace.....	36
4.1 Vytvoření databáze nahrávek.....	36
4.2 Volba délky relativní impulsní odezvy.....	38
4.3 Výpočet závislosti parametrů $c_1$ , $c_2$ a $c_3$ na počtu nenulových prvků řídké relativní impulsní odezvy hrubou silou.....	41
4.4 Hledání kombinace parametrů $c_1$ , $c_2$ a $c_3$ , pro která je řídká relativní impulsní odezva nejřidší.....	45
4.5 Hledání závislostí parametrů váhovací funkce na akustických parametrech.....	53
5 Ověření optimality výsledného vzorce experimenty.....	58
6 Závěr.....	61
Seznam použité literatury.....	62
Přílohy.....	63
A Obsah přiloženého CD.....	63



## Seznam obrázků

1 Relativní impulsní odezva spočtená pomocí LMS.....	12
2 Příklad průběhu váhovací funkce.....	14
3 Příklad řídké relativní impulsní odezvy.....	14
4 Výpočet T60 pomocí měření poklesu o 60 dB.....	15
5 Výpočet T60 využitím linearitu poklesu hlasitosti.....	16
6 Graf průběhu hlasitosti dozvuku s červeně zvýrazněným EDT.....	23
7 Výpočet C80 z relativní impulsní odezvy.....	29
8 Výpočet D50 a D20.....	33
9 Graf závislosti součtu nenulových prvků řídké impulsní odezvy na parametru váhovací funkce c2.....	51
10 Graf závislosti součtu nenulových prvků řídké impulsní odezvy na parametru váhovací funkce c3.....	52
11 Graf závislosti parametru c2 váhovací funkce na akustických parametrech T60 a C20 .....	56
12 Graf závislosti parametru c3 váhovací funkce na akustických parametrech T60 a C20 .....	56

## Seznam tabulek

1 Naměřené akustické parametry zvukových záznamů.....	36
2 Volba délky relativní impulsní odezvy.....	38
3 Optimální hodnoty váhovací funkce pro měřené místnosti.....	52
4 Porovnání ideálních parametrů váhovací funkce s akustickými parametry.....	53
5 Volba délky relativní impulsní odezvy testovaných místností.....	59
6 Test optimality pomocí rozdílů hlasitostí.....	60

## Seznam zdrojových kódů

1 Testování, zda hodnota T60 je uložena.....	17
2 Nastavení parametrů pro výpočet T60.....	18
3 Výpočet průběhu hlasitosti zvukového záznamu.....	19
4 Hledání souřadnic bodů pro výpočet T60.....	20
5 Nalezení parametrů prokládající přímky a dobu T60.....	21
6 Uložení hodnoty T60 a zobrazení do grafu.....	22
7 Testování, zda hodnota EDT je uložena.....	24
8 Vytvoření jednokanálového záznamu z dvoukanálového.....	24
9 Výpočet hlasitosti ze zvukového záznamu.....	25
10 Nalezení doby EDT.....	26
11 Testování, zda byl EDT nalezen a vykreslení do grafu.....	27
12 Uložení hodnoty EDT do souboru.....	28
13 Hlavička funkce pro výpočet C80.....	30
14 Výpočet relativní impulsní odezvy pomocí LMS.....	31
15 Výpočet hodnoty C80.....	32
16 Hlavička funkce pro výpočet D20 a D50.....	34
17 Výpočet D20 a D50.....	35
18 Zaznamenání zvukového signálu.....	37
19 Hlavička funkce pro výpočet intenzity potlačení signálu v závislosti na délce relativní impulsní odezvy.....	39
20 Výpočet signálu, který vznikl rozdílem pravého kanálu a zfiltrovaného levého kanálu .....	39
21 Výpočet rozdílu hlasitostí mezi originálním zvukovým záznamem a zfiltrovaným zvukovým záznamem.....	40
22 Hlavička funkce pro hledání nejřidších odezev hrubou silou.....	42
23 Výpočet matice a vektoru pro výpočet řídké relativní impulsní odezvy.....	42
24 Cykly pro výpočet řídkých relativních impulsních odezev.....	42
25 Výpočet řídké relativní impulsní odezvy.....	43
26 Spočtení a uložení počtu nenulových prvků v řídké relativní impulsní odezvě.....	44
27 Hlavička funkce pro nalezení optimální hodnoty parametru váhovací funkce z předpočítaných dat.....	45
28 Vytvoření matice z vektoru.....	46

29 Výběr závislosti na vybraném parametru.....	46
30 Vykreslení průběhu závislosti hodnoty nenulových prvků relativní impulsní odezvy na vybraném parametru.....	46
31 Nalezení nejřidších relativních impulsních odezev.....	47
32 Interpolace mezi prázdnými indexy, kde se nenachází nejřidší hodnota relativní impulsní odezvy.....	48
33 Proložení bodů s nejřidšími relativními impulsními odezvami parabolou.....	49
34 Najít parametru $c_2$ nebo $c_3$ s nejřidší možnou relativní impulsní odezvou na parabole.....	50
35 Hlavička pro výpočet závislostní funkce parametrů $c_2$ a $c_3$ na akustických parametrech T60 a C80.....	54
36 Načtení optimálních parametrů $c_2$ a $c_3$ z naměřených místností a jejich akustických parametrů T60 a C80.....	54
37 Výpočet a zobrazení závislostní funkce parametrů $c_2$ a $c_3$ na akustických parametrech T60 a C80.....	55
38 Funkce pro výpočet optimálních parametrů váhovací funkce.....	57
39 Hlavička funkce pro ověření optimality spočteného parametru.....	58
40 Výpočet rozdílu potlačení signálu relativní impulsní odezvou spočtenou pomocí metody nejmenších čtverců a řídkou relativní impulsní odezvou.....	58

# 1 Úvod

Relativní impulsní odezva mezi dvěma mikrofony charakterizuje intenzitu odražených zvukových signálů v místnosti v závislosti na čase. Její tvar je závislý na umístění zvukového zdroje, rozmístění mikrofonů, velikosti místnosti, rozmístění různých předmětů a na jejich materiálu. Ideálně při konvoluci signálu z levého mikrofonu s relativní impulsní odezvou vznikne signál shodný se signálem z pravého mikrofonu. V praktických aplikacích je tvar relativní impulsní odezvy pouze přibližný. Rozdíl zfiltrovaného signálu z levého mikrofonu odhadnutou relativní impulsní odezvou a signálu z pravého mikrofonu již není nulový. Hlasitost tohoto signálu je závislá na tom, jak moc dobře je relativní impulsní odezva odhadnutá. Čím nižší je hlasitost tohoto signálu, tím se odhadnutá relativní impulsní odezva blíží jejímu ideálnímu tvaru. Rozdíl této hlasitosti od původní hlasitosti záznamu se tak stává kritériem optimality odhadnuté relativní impulsní odezvy.

Z praktického hlediska je relativní impulsní odezva spočtená pomocí metody nejmenších čtverců optimální, ale obsahuje velké množství nenulových prvků, a je tedy příliš složitá. Cílem této práce je tedy optimálně aproximovat relativní impulsní odezvu spočtenou pomocí metody nejmenších čtverců řídkou relativní impulsní odezvou tak, aby v ní zůstaly pouze nejvýznamnější koeficienty. Výpočet řídké relativní impulsní odezvy je prováděn pomocí vážené úlohy LASSO (Least Absolute Shrinkage and Selector Operator [2]), jejíž optimalita se od relativní impulsní odezvy spočtené pomocí metody nejmenších čtverců liší v závislosti na volbě parametrů váhovací funkce. Počet nenulových prvků relativní impulsní odezvy spočtené pomocí vážené úlohy LASSO lze měnit volbou vah k jednotlivým koeficientům relativní impulsní odezvy. V této práci je řešen kompromis mezi co nejmenším počtem nenulových prvků řídké relativní impulsní odezvy a mezi vzdáleností optimality od relativní impulsní odezvy spočtené pomocí metody nejmenších čtverců. Tento kompromis je určen již zmíněným kritériem, kde rozdíl v hlasitosti potlačeného zvukového signálu pomocí metody nejmenších čtverců a vážené úlohy LASSO činí maximálně 1 dB. Optimální parametry váhovací funkce pro výpočet řídké relativní impulsní odezvy jsou hledány v závislosti na akustických parametrech místnosti.

Výpočet řídké relativní impulsní odezvy se využívá například, máme-li neúplnou přenosovou funkci a chceme získat nejdříve relativní impulsní odezvu filtru, pro kterou budou prvky přenosové funkce nejlépe prokládat známé koeficienty (viz [1]).

## 2 Výpočet relativní impulsní odezvy

### 2.1 Výpočet relativní impulsní odezvy pomocí metody nejmenších čtverců (LMS)

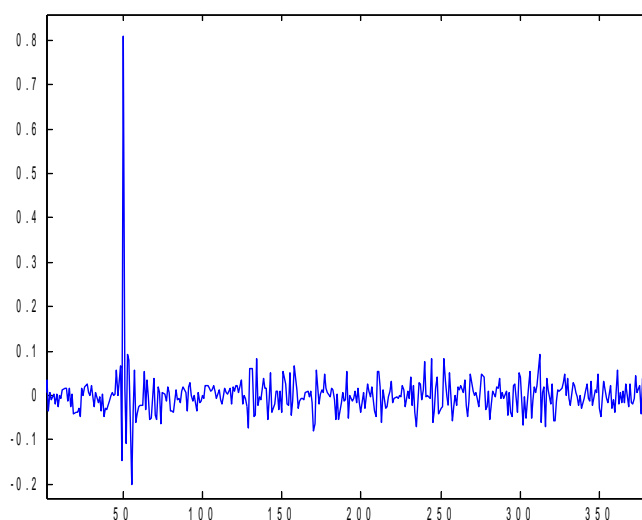
Výpočet relativní impulsní odezvy mezi dvěma mikrofony a zdrojem akustického signálu pomocí metody nejmenších čtverců se podle [2] provádí minimalizací, která je zobrazena ve vzorci 1.

$$\min_h \|X_L \mathbf{h} - \mathbf{x}_R\|_2^2 \quad (\text{Vzorec 1})$$

Sloupcový vektor  $\mathbf{h}$  značí výslednou relativní impulsní odezvu. Proměnná  $X_L$  značí toeplitzovskou matici, kde v každém sloupci je vektor zvukového záznamu z levého kanálu vždy o jeden řádek posunutý oproti předchozímu sloupci. Zbylé koeficienty jsou nahrazeny nulami. Toeplitzovská matice  $X_L$  má počet sloupců shodný s délkou vektoru  $\mathbf{h}$ . Proměnná  $\mathbf{x}_R$  značí sloupcový vektor pravého kanálu audio záznamu. Počet řádků matice  $X_L$  a délka vektoru  $\mathbf{x}_R$  jsou taktéž shodné.

Výsledná relativní impulsní odezva se spočte pomocí vzorce 2. Názorná ukázka relativní impulsní odezvy spočtená pomocí tohoto vzorce je zobrazena na obrázku 1. Tato relativní impulsní odezva byla spočtena z dvoukanálové nahrávky, která byla pořízena v místnosti s rozměry 2,8 m × 4,9 m a výškou stropu 2,7 m. Mikrofony byly umístěny 13 cm od sebe uprostřed místnosti a 1 m před nimi byl umístěn reproduktor reprodukcující záznam ženského hlasu.

$$\mathbf{h} = (\mathbf{x}_R X_L^T (X_L X_L)^{-1})^T \quad (\text{Vzorec 2})$$



Obr. 1: Relativní impulsní odezva spočtená pomocí LMS

## 2.2 Výpočet řídké relativní impulsní odezvy pomocí vážené metody LASSO

Výpočet relativní impulsní odezvy mezi dvěma mikrofony pomocí vážené úlohy LASSO se řeší minimalizací zobrazenou ve vzorci 3.

Tento vzorec oproti předchozímu vzorci navíc řeší minimalizaci L1 normy  $\tau \|h\|_1$ , která zajišťuje potlačení přebytečných koeficientů relativní impulsní odezvy.

$$\min_h \frac{1}{2} \|Ah - y\|_2^2 + \tau \|h\|_1 \quad (\text{Vzorec 3 [3]})$$

Proměnná  $h$  symbolizuje sloupcový vektor relativní impulsní odezvy. Matice  $A$  je maticový násobek toeplitzovské matice levého kanálu zvukového záznamu a její transpozice  $A = X_L' X_L$ . Přesný obsah matice  $X_L$  je popsán v předchozí kapitole. Vektor  $y$  je maticový násobek řádkového vektoru, který obsahuje záznam z pravého kanálu audio záznamu, a toeplitzovské matice  $X_L$ . Tento výsledek je následně transponován pro získání sloupcového vektoru. Viz tento vzorec:  $y = (x_R' X_L)'$ . Malé řecké  $\tau$  značí konstantu, která určuje intenzitu potlačení koeficientů řídké relativní impulsní odezvy. Její rozsah je v oboru reálných čísel větších nebo rovno nule. Obvykle se její hodnota volí blízká nule. Pokud je její hodnota nulová, je výsledná relativní impulsní odezva shodná, jako kdyby byla spočtena pomocí metody nejmenších čtverců.

Pro výpočet řídké relativní impulsní odezvy pomocí vážené metody LASSO je potřeba volit míru potlačení jednotlivých koeficientů individuálně. Tento problém řeší minimalizace zobrazená ve vzorci 4.

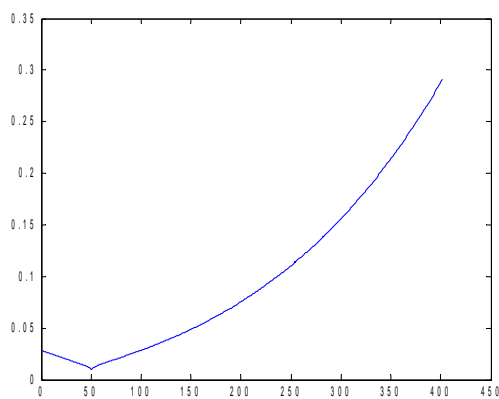
$$\min_h \|Wh\|_1 + \frac{1}{2} \|Ah - y\|_2^2 \quad (\text{Vzorec 4 [4]})$$

Rozdíl oproti vzorci 3 nastává nahrazením konstanty  $\tau$  váhovací diagonální maticí  $W$ , jejíž diagonála tvoří váhy pro jednotlivé koeficienty relativní impulsní odezvy. Vyšší váhovací koeficient na diagonále matice  $W$  značí vyšší tlak na úplné potlačení příslušného koeficientu relativní impulsní odezvy. Vyšší řídkostí relativní impulsní odezvy se optimalita vzdaluje od řešení pomocí metody nejmenších čtverců. Proto je řešení kompromis mezi vzdáleností optimality od relativní impulsní odezvy spočtené pomocí metody nejmenších čtverců a co nejmenším počtem nenulových prvků v řídké relativní impulsní odezvě. Pro výpočet diagonály váhovací matice  $W$  byla zvolena exponenciální funkce zobrazená ve vzorci 5.

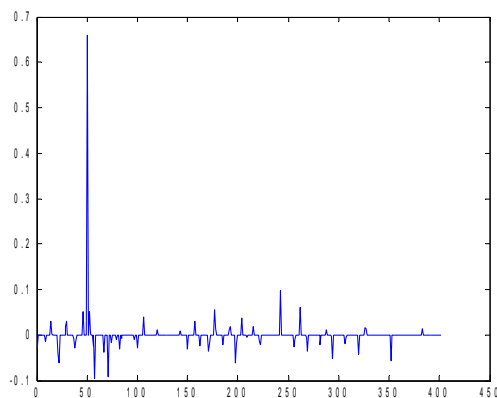
$$\mathbf{w}_i = c_1 e^{c_2 |i-d|^{c_3}}, i = 1, \dots, L \quad (\text{Vzorec 5})$$

Parametr  $i$  značí index váhovacího vektoru. Jeho hodnoty jsou od prvního indexu po délku vektoru relativní impulsní odezvy. Parametr  $d$  symbolizuje velikost zpoždění signálu mezi levým a pravým mikrofonem. Jeho velikost označuje počet indexů, o které je signál opožděn. Parametry  $c_1$ ,  $c_2$  a  $c_3$  jsou neznámé konstanty, které určují tvar průběhu váhovacího vektoru. Cílem této práce je nalézt vhodnou parametrizaci této váhovací funkce v závislosti na době dozvuku místnosti a zjistit možnost závislosti na dalších akustických vlastnostech, které jsou uvedeny v následujících kapitolách.

Jelikož výpočet řídké relativní impulsní odezvy pomocí minimalizace zobrazené ve vzorci 4 je algoritmicky náročnější než výpočet pomocí metody nejmenších čtverců, tak pro potřeby řešení této úlohy byl zvolen zásuvný modul do programovacího prostředí Matlab. Zásuvných modulů řešících výpočet relativní impulsní odezvy pomocí úlohy LASSO existuje větší množství. Nalezl jsem však pouze jednu, která využívá váhovací matici dle vzorce 4. Jeho název je L1 Homotopy a je dostupný z internetové adresy <http://users.ece.gatech.edu/~sasif/homotopy/>.



Obr. 2: Příklad průběhu váhovací funkce



Obr. 3: Příklad řídké relativní impulsní odezvy

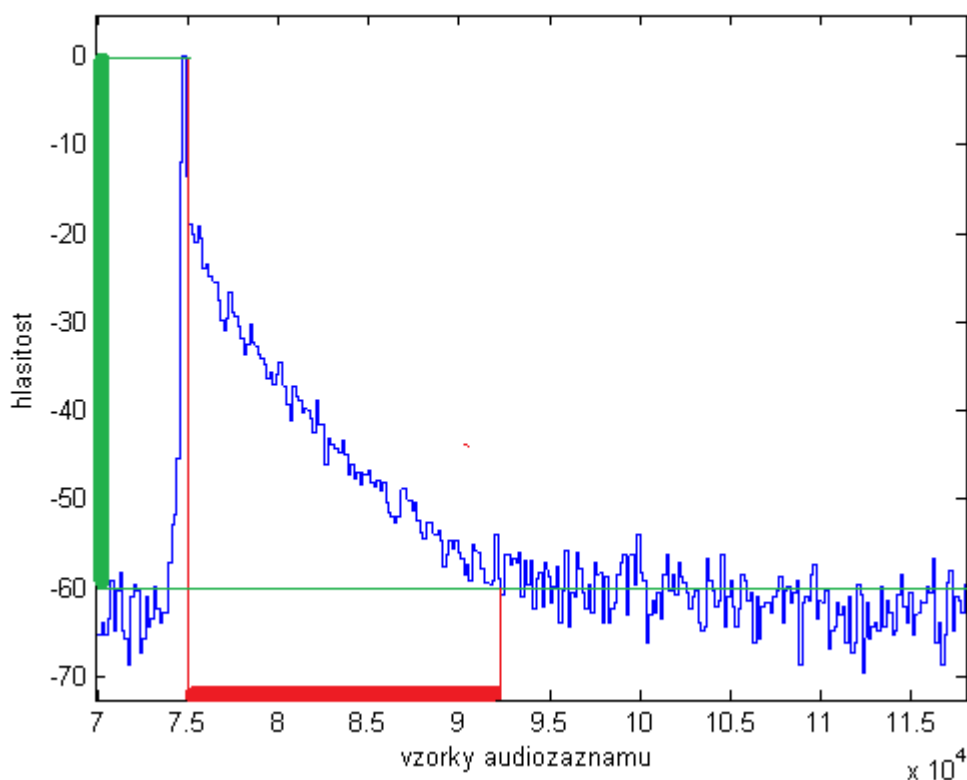
Názorný příklad tvaru váhovací funkce pro výpočet řídké relativní impulsní odezvy je na obrázku 2. Pro výpočet této funkce byly použity hodnoty parametrů  $c_1 = 0,01$ ;  $c_2 = 0,1$ ;  $c_3 = 0,6$ ;  $L = 402$  a  $d = 50$ . Následně tato funkce byla využita k výpočtu řídké relativní impulsní odezvy, která je zobrazena na obrázku číslo 3. Na obrázku číslo 1 je zobrazena relativní impulsní odezva stejné místnosti, která je vypočtená pomocí metody nejmenších čtverců. Oproti obrázku 1 si řídká relativní impulsní odezva na obrázku 3 kvalitativně pohoršila jen o 0,77 dB. Splňuje tedy zavedené kritérium z úvodu.

### 3 Akustické vlastnosti místnosti

#### 3.1 Doba dozvuku místnosti (T60)

Doba dozvuku místnosti určuje čas, po který trvá ozvěna v místnosti od okamžiku ukončení aktivity zdroje zvukového signálu až po okamžik poklesu intenzity hlasitosti dozvuku o 60 dB.

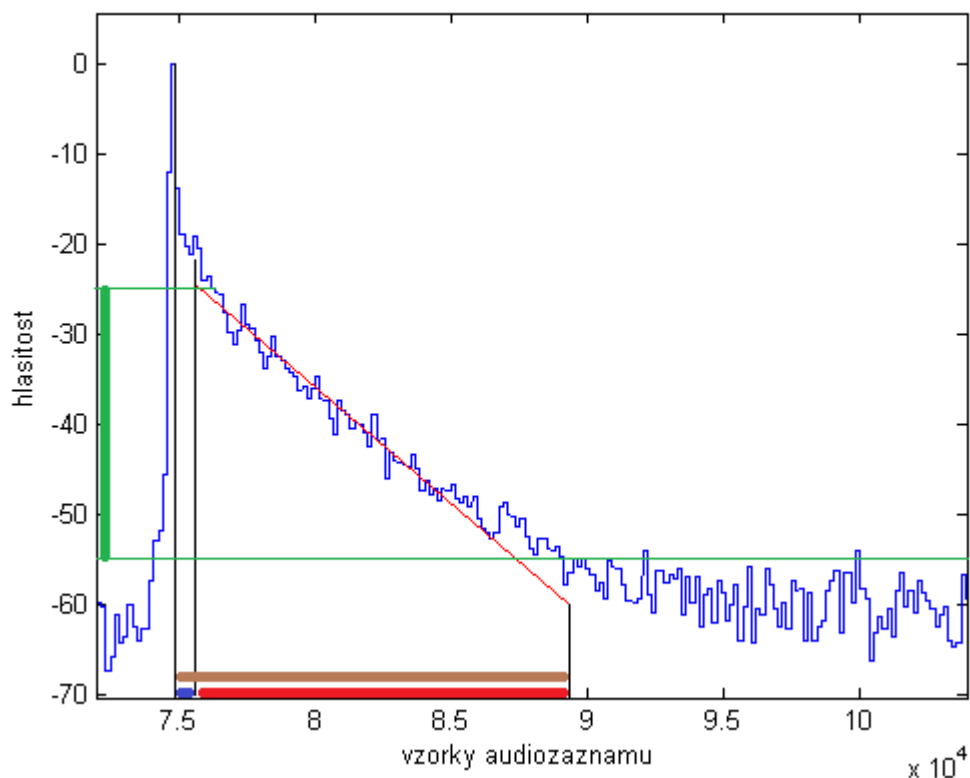
Ve čtvrtém obrázku je modře zobrazen průběh hlasitosti dozvuku v místnosti. Zvuk v místnosti byl vyvolán tlesknutím. V obrázku na ose  $y$  je zeleně vyznačen úsek, který vyjadřuje pokles hlasitosti dozvuku od maximální hlasitosti o 60 dB. Červeně na ose  $x$  je pak vyznačen časový úsek, který vyjadřuje dobu poklesu hlasitosti dozvuku o 60 dB. Nicméně pro výpočet doby dozvuku touto metodou je zapotřebí kvalitního mikrofonu a zesilovače, zaručující hodnotu odstupu signálu od šumu větší než je 60 dB. Z obrázku je zřetelně vidět, že konečná hodnota hlasitosti  $-60$  dB je už na hranici šumu, proto tato metoda není vhodná pro mikrofony s nižším dynamickým rozsahem.



Obr. 4: Výpočet T60 pomocí měření poklesu o 60 dB



Pro výpočet doby dozvuku pomocí mikrofону s nízkým dynamickým rozsahem je vhodná metoda zobrazená na pátém obrázku. Tato metoda využívá logaritmické linearity poklesu hlasitosti signálu v místnosti od 50 ms od konce reprodukce zvukového zdroje. Na obrázku je modře vyznačený počáteční 50ms úsek, kde je pokles nelineární. Dále zeleně vyznačený úsek na ose  $y$  označuje úroveň poklesu hlasitosti dozvuku o 30 dB. Hodnotu tohoto poklesu lze vybrat libovolně, pokud nezasahuje do úrovně, kde se už projevuje jen šum. Hodnota tohoto hlasitostního úseku je následně proložena přímkou, která je na obrázku zobrazena červeně. Konečný bod této přímky je protažen až do úrovně, kde rozdíl mezi tímto bodem a počáteční hlasitostní úrovní je 60 dB. Červený úsek na ose  $x$  tedy znázorňuje dobu od 50 ms po začátku dozvuku až po konečný pokles o 60 dB od počátku dozvuku. Výsledná doba dozvuku (hodnota T60) je zobrazena úsekem označeným hnědě, a je tedy součtem prvních 50 ms (modrý úsek) a doby, která je proložena přímkou (červený úsek).



Obr. 5: Výpočet T60 využitím linearity poklesu hlasitosti

V následujících zdrojových kódech je popsána realizace výpočtu doby dozvuku (T60) pomocí metody využívající linearitu poklesu hlasitosti.

```
1  function [ T_60 ] = T_60( nahravka, Fs, soubor )
2      if nargin == 3
3          if exist(soubor,'file') == 2
4              saved = open(soubor);
5              if isfield(saved, 'T_60')
6                  T_60 = saved.T_60;
7                  return
8              end
9          else
10             disp('vybrany soubor neexistuje, vypocet
11                 nebude ulozen, pouze vracen')
12         end
13     end
```

*Zdrojový kód 1: Testování, zda hodnota T60 je uložena*

V zobrazeném zdrojovém kódu se na prvním řádku nachází hlavička funkce. Jejími parametry jsou proměnná `nahravka`, která obsahuje monofonní, nebo stereofonní zvukový záznam. Dalším povinným parametrem je proměnná `Fs`, která uchovává hodnotu vzorkovací frekvence audiozáznamu. Poslední parametr `soubor` je nepovinný a obsahuje textový řetězec názvu souboru \*.mat, včetně přípony, ze kterého je již naměřená hodnota T60 přečtena, případně se spočtená hodnota T60 do tohoto souboru uloží. Při ponechání prázdného parametru `soubor` pouze tato funkce vrátí naměřenou hodnotu T60.

Podmínka na druhém řádku testuje, zda je vyplněn parametr `soubor`. Pokud není vyplněn, tak se další testování přeskočí. Pokud vyplněn je, tak následuje podmínka na třetím řádku, kde se testuje, zda zvolený soubor existuje. Pokud soubor neexistuje, je zobrazena upozorňující hláška a program pokračuje dále, jakoby název souboru nebyl vyplněn. Pokud soubor existuje, tak je na čtvrtém řádku otevřen a na pátém řádku otestován, zda obsahuje uloženou hodnotu T60. Pokud je nalezena uložená hodnota T60, funkce uloží její hodnotu do proměnné `T_60`, okamžitě vrátí její hodnotu a skončí.

```

13  presnost = 0.01;
14  konecnyPokles = 60;
15  pomocnyPokles = konecnyPokles/2;
16  if size(nahravka,2) == 2
17      nahravka = (nahravka(:,1)+nahravka(:,2))/2;
18  end
19  krok = Fs * presnost;
20  hlasitost = NaN * ones(size(nahravka,1),1);

```

*Zdrojový kód 2: Nastavení parametrů pro výpočet T60*

Na třináctém řádku je nastavená hodnota proměnné `presnost` na 0,01 a označuje velikost úseku průběhu hlasitosti zvukového záznamu ve vteřinách, ve kterém bude nalezeno hlasitostní maximum. Toto nalezené maximum bude nastaveno všem prvkům v uvedeném úseku.

Proměnná `konecnyPokles` na čtrnáctém řádku uchovává hodnotu rozdílu poklesu hlasitosti dozvuku od konce činnosti akustického zdroje. Pro výpočet hodnoty T60 je nastavena na hodnotu 60, tedy 60 dB. Pro výpočet hodnoty T40 je nutno nastavit tuto hodnotu na 40 atd. V této práci je počítáno pouze s hodnotou T60.

Hodnota proměnné  `pomocnyPokles` uchovává hodnotu poklesu hlasitosti dozvuku, který se měří až po uplynutí 50 ms od konce činnosti akustického zdroje. V tomto případě je zvolen na polovinu celkového poklesu hlasitosti dozvuku.

Podmínka na šestnáctém řádku zjišťuje, zda se jedná o zvukový záznam jednokanálový nebo dvoukanálový. V případě dvoukanálového zvukového záznamu se na sedmnáctém řádku vytvoří záznam jednokanálový. Jednokanálového zvukového záznamu je dosaženo průměrováním jednotlivých vzorků na stejné pozici.

Proměnná `krok` na devatenáctém řádku uchovává hodnotu počtu vzorků zvukového záznamu, které odpovídají časovému údaji uloženém v proměnné `presnost`.

Na dvacátém řádku je inicializovaná proměnná `hlasitost`, která bude uchovávat průběh hlasitosti zvukového záznamu v dB. Tato proměnná je inicializovaná na hodnoty NaN (Not a Number) o délce shodné s délkou zvukového záznamu.

```
21 for i = 1:krok:size(nahravka,1)-krok
22     hlasitost(i:i+krok) =
23         10*log10(max((nahravka(i:i+krok)).^2));
24 end
```

*Zdrojový kód 3: Výpočet průběhu hlasitosti zvukového záznamu*

Ve třetím zdrojovém kódu je napsán výpočet průběhu hlasitosti zvukového záznamu. V proměnné `krok` je časový údaj ve vzorcích záznamu. Na tomto časovém úseku je nalezeno maximum hodnoty záznamu na druhou. Mohl by být spočten i průměr tohoto úseku. Jelikož záznam pro změření T60 obsahuje tlesknutí, je hlasitostní maximum velmi úzké a průměrná hodnota by se vyskytovala mnohem níže než je skutečný průběh hlasitosti. Z této maximální hodnoty je spočten desetinásobek logaritmu se základem deset. Výsledná hodnota je dosazena do vektoru `hlasitost` na indexech odpovídajících prvnímu časovému oknu o délce uložené v proměnné `krok`. Takto se cyklus na řádcích dvacet jedna až dvacet tři opakuje, dokud nejsou naplněna veškerá časová okna nahrávky.

```

24 maxX = find(hlasitost == max(hlasitost),1,'last')
      -(krok/2);
25 maxY = hlasitost(maxX);
26 T60_Y = maxY-konecnyPokles;
27 startX = maxX + (Fs * 0.05);
28 startY = hlasitost(startX);
29 konecX = find(hlasitost(startX:size(hlasitost,1)) <
      (startY-pomocnyPokles), 1, 'first') + (krok / 2) + startX;

```

*Zdrojový kód 4: Hledání souřadnic bodů pro výpočet T60*

Na dvacátém čtvrtém řádku je nalezena a do proměnné `maxX` uložena hodnota indexu, na kterém se nachází maximální hlasitost. Index je vždy vybrán zprostřed časového okna. Na následujícím řádku je do proměnné `maxY` uložena míra hlasitosti v místě s nalezenou maximální hlasitostí.

Do proměnné `T60_Y` je uložena hodnota hlasitosti, která je o 60 dB nižší než maximální nalezená hlasitost v proměnné `maxY`.

Na dvacátém sedmém řádku je do proměnné `startX` uložen nalezený index, jež odpovídá místu 50 ms za indexem s nejvyšší hlasitostí. Do proměnné `startY` je následně uložena hlasitost, jež se vyskytuje na indexu `startY`.

Do proměnné `konecX` je uložen index, který odpovídá místu poklesu hlasitosti o 30 dB od místa, které je 50 ms za nejvyšší hlasitostí.

```

30  if(isempty(konecX) || (T60_Y > startY)
31      plot (hlasitost);
32      T_60 = false;
33      return
34  end
35  Y = hlasitost(startX:konecX)';
36  X = (startX:konecX);
37  xmean = mean(X);
38  ymean = mean(Y);
39  sumx2 = (X-xmean)*(X-xmean)';
40  sumxy = (Y-ymean)*(X-xmean)';
41  a = sumxy/sumx2;
42  b = ymean-a*xmean;
43  T60_X = (T60_Y-b)/a;
44  T_60 = (T60_X-maxX)/Fs;

```

*Zdrojový kód 5: Nalezení parametrů prokládající přímky a dobu T60*

V podmínce na třicátém až třicátém čtvrtém řádku je zjišťováno, zda bylo nalezeno místo poklesu o 30 dB od místa 50 ms za největší hlasitosti. Pokud toto místo nebylo nalezeno, nebo je dozvuk místnosti tak krátký, že hlasitost dozvuku poklesla o 60 dB do 50 ms, funkce vykreslí průběh hlasitosti, vrátí hodnotu `false` a skončí.

Pokud podmínka neprojde, tak se do vektoru `Y` uloží průběh hlasitosti od indexu `startX` do indexu `konecX` a do vektoru `X` se uloží rozsah těchto indexů. Na třicátém sedmém a třicátém osmém řádku se do proměnných `xmean` a `ymean` uloží průměrné hodnoty vektorů `X` a `Y`.

Dále se do proměnné `sumx2` uloží skalární výsledek maticového násobku dvou vektorů, které obsahují prvky vektoru `X`, od kterých byla odečtena jejich průměrná hodnota. Proměnná `sumxy` obsahuje skalární výsledek součinu vektorů `X` a `Y`, od jejichž hodnot byla odečtena jejich průměrná hodnota.

Do proměnných `a` a `b` jsou spočteny parametry lineární funkce, jejíž grafem je přímka prokládající lineární část poklesu hlasitosti dozvuku. Do proměnné `T60_X` je spočten index přímky, jehož hodnota odpovídá úrovni poklesu hlasitosti o 60 dB. Nakonec je do proměnné `T_60` uložena výsledná doba dozvuku ve vteřinách.

```

45     if nargin == 3
46         if exist(soubor, 'file') == 2
47             save(soubor, 'T_60', '-append')
48         end
49     end
50     figure('name',strcat('T_60 = ', num2str(T_60),
        ' s'),'NumberTitle','off');
51     plot(hlasitost)
52     hold on
53     plot([startX T60_X],[a*startX+b T60_Y],'r')
54     xlabel('vzorky audiozaznamu');
55     ylabel('hlasitost (dB)');
56 end

```

*Zdrojový kód 6: Uložení hodnoty T60 a zobrazení do grafu*

Podmínky na čtyřicátém pátém a čtyřicátém šestém řádku programu rozhodují, zda je zadán název souboru, či zda soubor existuje. Pokud soubor existuje, je do něj uložena výsledná doba T60.

Na padesátém řádku je příkaz pro zobrazení okna s popiskem, který zobrazuje hodnotu výsledné T60. Na dalším řádku je zobrazen do grafu průběh hlasitosti zvukového záznamu a na padesátém třetím řádku je do stejného grafu zobrazena přímka, která prokládá lineární část poklesu hlasitosti dozvuku.

Příkazy `xlabel` a `ylabel` zobrazují v grafu popisky obou os  $x$  a  $y$ , kde na souřadnici  $x$  se nacházejí jednotlivé vzorky zvukového záznamu a na souřadnici  $y$  je vyjádřena hodnota hlasitosti zvukového záznamu v dB.

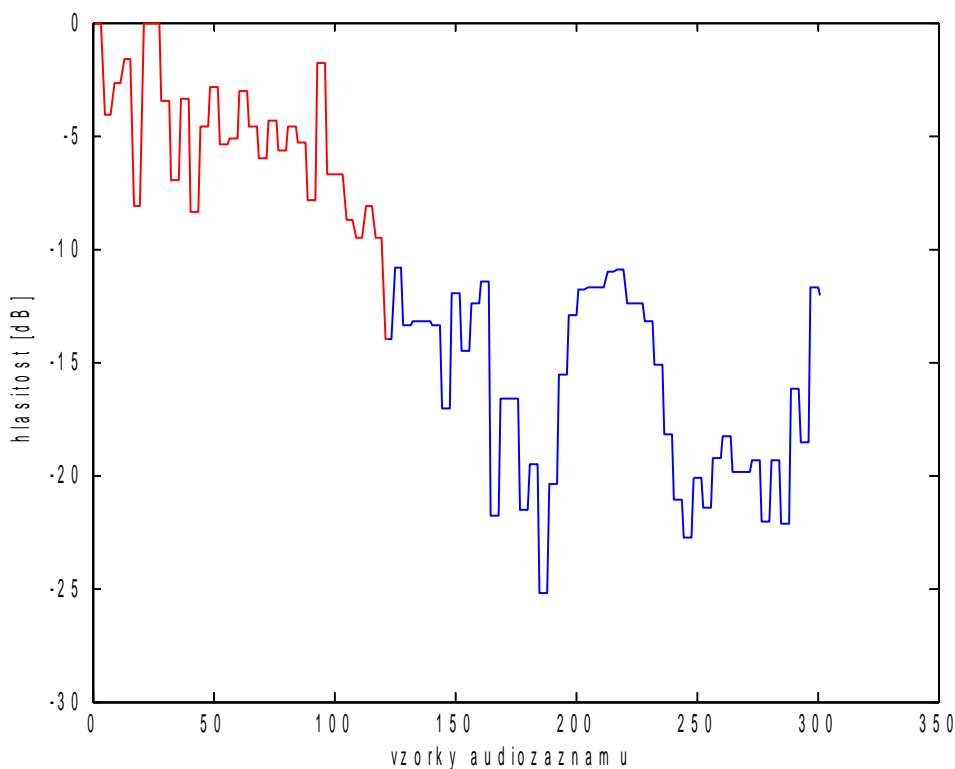
Po skončení úspěšného běhu funkce vrátí hodnotu T60 případně hodnotu `false`, pokud se nepovedlo hodnotu T60 nalézt.

## 3.2 Výpočet early decay time (EDT)

Early decay time neboli EDT je akustický parametr, který měří dobu dozvuku místnosti pomocí nejdřívějších odrazů. Tato metoda změří parametr T10, to znamená dobu dozvuku od konce činnosti akustického zdroje dokud nedosáhne poklesu hlasitosti o 10 dB. Výsledná časová hodnota je vynásobena šesti, kvůli možnosti srovnání s parametrem T60. [5]

Na obrázku číslo šest je zobrazen průběh hlasitosti zvukového signálu těsně po přerušení akustického zdroje. Na ose x jsou vyznačeny jednotlivé vzorky zvukového signálu a na ose y jeho hlasitost naměřená v dB. Červeně vyznačená část grafu vyznačuje průběh hlasitosti, která odpovídá parametru T10. Z grafu lze vyčíst, že dozvuk audiosignálu dosáhl poklesu o 10 dB za zhruba 120 vzorků. Jestliže byl zvukový záznam nahrán se vzorkovací frekvencí 16 000 Hz, lze spočítat, že doba dozvuku T10 je 7,5 ms. Parametr EDT však tuto hodnotu počítá jako šestinásobek. Výsledná hodnota EDT je tedy 45 ms.

Na následujících stranách je zobrazena realizace výpočtu parametru EDT v programovacím prostředí Matlab.



Obr. 6: Graf průběhu hlasitosti dozvuku s červeně zvýrazněným EDT



```

1  function [ EDT ] = EDT( nahravka, Fs, soubor )
2      if nargin == 3
3          if exist(soubor,'file') == 2
4              saved = open(soubor);
5              if isfield(saved, 'EDT')
6                  EDT = saved.EDT;
7              return
8          end
9      else
10         disp('vybrany soubor neexistuje, vypocet
              nebude ulozen, pouze vracen')
11     end
12 end

```

*Zdrojový kód 7: Testování, zda hodnota EDT je uložena*

Stejně jako pro výpočet T60 funkce EDT vyžaduje zvukový záznam, který je v proměnné *nahravka*, vzorkovací frekvenci v proměnné *Fs* a nepovinně název \*.mat souboru v proměnné *soubor* pro uložení naměřené hodnoty EDT.

V následujících podmínkách je testováno, zda zvolený soubor existuje a zda už naměřenou hodnotu EDT neobsahuje. Pokud soubor naměřenou hodnotu EDT obsahuje, tak funkce její hodnotu vrátí a běh funkce ukončí. Pokud soubor nebyl nalezen, je vy-psána hláška o tomto stavu a program dál pokračuje v počítání EDT, ale výsledek už do souboru nebude uložen.

```

13  if size(nahravka,2) == 2
14      nahravka = (nahravka(:,1)+nahravka(:,2))/2;
15  end

```

*Zdrojový kód 8: Vytvoření jednokanálového záznamu z dvoukanálového*

Dále pokud je na třináctém řádku zjištěno, že v proměnné *nahravka* se nachází dvoukanálový záznam, je tento záznam převeden do záznamu jednokanálového. Monofonní záznam je vytvořen zprůměrováním vzorků v jednotlivých kanálech.

```

16  krok = 4;
17  hlasitost = NaN * ones(size(nahravka,1),1);
18  for i = 1:krok:size(nahravka,1)-krok
19      hlasitost(i:i+krok) = 10 * log10
        (max((nahravka(i:i + krok)).^2));
20  end

```

*Zdrojový kód 9: Výpočet hlasitosti ze zvukového záznamu*

Na šestnáctém řádku je do proměnné `krok` uložen počet vzorků zvukového záznamu, které určují velikost časového okna, po kterých bude nalezeno hlasitostní maximum. Ostatní prvky z tohoto okna budou nahrazeny nalezenou maximální hodnotou. Toto hledání maxima po časových rámcích zajišťuje relativní plynulost změn hlasitosti ve zvukovém signálu.

Na sedmnáctém řádku je inicializovaná proměnná `hlasitost` na vektor o délce zvukového záznamu s hodnotami NaN (Not a Number).

V cyklu typu `for` na devatenáctém řádku probíhá naplnění proměnné `hlasitost` hodnotami po časových rámcích o délce uložené v proměnné `krok`. V jednotlivých časových úsecích je nalezena maximální hodnota proměnné `nahravka` na druhou. Z této maximální hodnoty je spočtená hlasitost v dB jako desetinásobek logaritmu se základem deset. Výsledná hlasitostní hodnota je přiřazena všem prvkům z časového okna do vektoru `hlasitost`. Celý proces v cyklu se opakuje, dokud není naplněný celý vektor `hlasitost`.

```

21 maxX = find(hlasitost == max(hlasitost),1,'first');
22 maxY = hlasitost(maxX);
23 max_delka_T10 = 300;
24 velikost_poklesu = 10;
25 pocet_vzorku_doby_dozvuku =
        find(hlasitost(maxX:maxX+max_delka_T10) <
        (maxY - velikost_poklesu),1,'first');
26 EDT = (pocet_vzorku_doby_dozvuku / Fs) * 6;

```

*Zdrojový kód 10: Nalezení doby EDT*

Do proměnné `maxX` je na dvacátém prvním řádku uložen index vektoru `hlasitost`, na kterém se nachází nejvyšší hodnota hlasitosti. Na následujícím řádku je do proměnné `maxY` uložena hodnota nejvyšší hlasitosti.

Konstantní proměnná `max_delka_T10` na řádku dvacet tři uchovává maximální počet vzorků, na kterých bude hledán pokles o 10 dB od nalezení maxima. Následující proměnná `velikost_poklesu` obsahuje hodnotu poklesu hlasitosti v dB. Pro výpočet T10 je tato hodnota naplněna hodnotou deset.

Na řádku dvacet pět je nalezen počet vzorků, který odpovídá době, po kterou trvá pokles hlasitosti dozvuku o 10 dB. Výsledná hodnota je uložena do proměnné `pocet_vzorku_doby_dozvuku`. Na následujícím řádku je spočtena výsledná doba EDT jako šestinásobek doby, po kterou trvá dozvuk, než poklesne o 10 dB. Výsledná hodnota je spočtena ve vteřinách.

```

27  if isempty(EDT)
28      EDT = false;
29      figure('name',strcat('EDT nebylo nalezeno'),
              'NumberTitle','off');
30      plot(hlasitost, 'b');
31  else
32      figure('name',strcat('EDT = ', num2str(EDT),
              ' s'),'NumberTitle','off');
33      plot(hlasitost(maxX : maxX +
                    pocet_vzorku_doby_dozvuku), 'r')
34      hold on;
35  plot(size(maxX : maxX + pocet_vzorku_doby_dozvuku, 2)
        : (size(maxX + pocet_vzorku_doby_dozvuku : maxX +
max_delka_T10, 2) - 1 + size(maxX : maxX +
pocet_vzorku_doby_dozvuku, 2)), hlasitost(maxX +
pocet_vzorku_doby_dozvuku : maxX +
max_delka_T10), 'b');
36  hold off

```

*Zdrojový kód 11: Testování, zda byl EDT nalezen a vykreslení do grafu*

V jedenáctém zdrojovém kódu podmínka na dvacátém sedmém až třicátém prvním řádku testuje, zda byla hodnota EDT nalezena. Pokud nalezena nebyla, je její hodnota nastavena na `false` a tuto hodnotu funkce také vrátí. Dále také vypíše hlášku, že hodnota EDT nebyla nalezena a vykreslí graf průběhu hlasitosti.

Pokud hodnota EDT nalezena byla, také je vykreslen graf průběhu hlasitosti zvukového záznamu v místě, kde EDT bylo naměřeno. Na třicátém druhém řádku je příkaz pro nadepsání okna s grafem, kde se vypíše naměřená hodnota EDT. Na následujícím řádku je vykreslena část grafu, která odpovídá průběhu po dobu EDT. Tato část je zvýrazněna červeně. Na třicátém pátém řádku je vykreslena část průběhu hlasitosti, která je mezi koncem doby EDT a maximální dobou, kde je EDT ještě hledáno. Tato část je v grafu vykreslena modře. Maximální počet vzorků, které může dosahovat doba EDT, je uložena v proměnné `max_delka_T10`.

```
37         if nargin == 3
38             if exist(soubor, 'file') == 2
39                 save(soubor, 'EDT', '-append')
40             end
41         end
42     end
43     xlabel('vzorky audiozaznamu');
44     ylabel('hlasitost [dB]');
45 end
```

*Zdrojový kód 12: Uložení hodnoty EDT do souboru*

V podmínce na třicátém sedmém řádku je otestováno, zda byl zadán parametr, obsahující textový řetězec s názvem souboru \*.mat, do kterého má být výsledná hodnota EDT uložena. Pokud podmínka projde, je v následující podmínce testováno, zda soubor existuje. Pokud soubor existuje, je do něj uložena výsledná hodnota EDT. Pokud soubor neexistuje, tak je pouze vrácena spočtená hodnota EDT.

Na řádcích čtyřicet tři a čtyřicet čtyři jsou příkazy, které nadepisují jednotlivé osy grafu, který byl vykreslen v předcházejících zdrojových kódech.

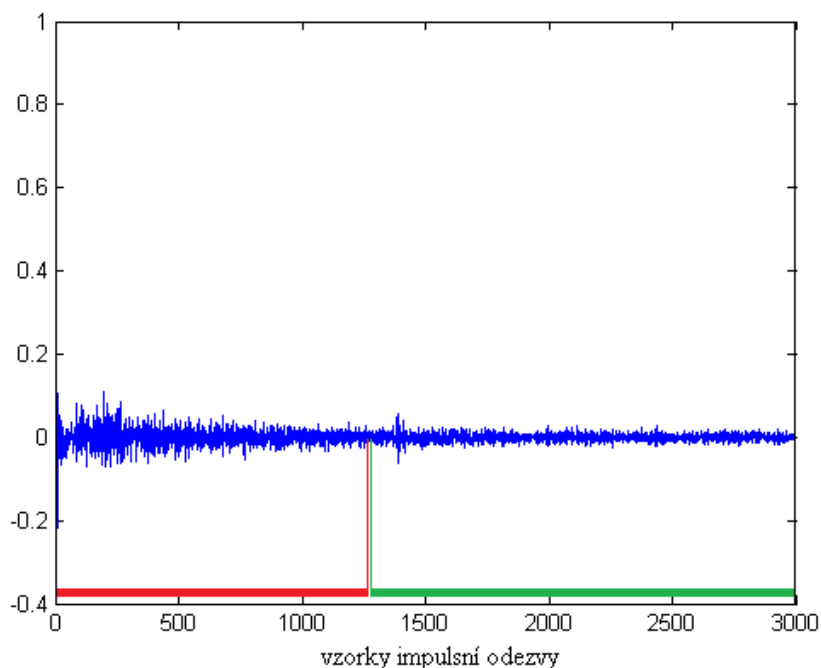
### 3.3 Výpočet C80

Výpočet C80 je akustický parametr spočtený z relativní impulsní odezvy. Využití nachází při zjišťování charakteristiky ozvěn v koncertních síních. Hodnota parametru C80 je vhodnější k využití pro prostory, které jsou určeny pro reprodukci hudby než k verbální komunikaci[5].

Hodnota parametru C80 se spočítá podle vzorce 6 jako desetinásobek logaritmu poměru obsahu hodnot relativní impulsní odezvy na druhou v úseku relativní impulsní odezvy odpovídající prvním 80 ms (červeně vyznačená část na obrázku 7) a úseku od 80 ms do nekonečna (zeleně vyznačená část). Vektor  $h$  značí relativní impulsní odezvu. V tomto konkrétním případě spočtenou pomocí metody nejmenších čtverců. Výsledná hodnota je spočtena v dB. Při vzorkování zvukového signálu kmitočtem 16 000 Hz vychází konkrétně hodnota indexu v obrázku pro 80 ms 1 280. Čím vyšší je výsledná hodnota C80, tím více ozvěn se nachází v prvních 80 ms. S rostoucí hodnotou tedy klesá doba dozvuku místnosti.

$$C_{80} = 10 \log_{10} \left( \frac{\int_0^{80 \text{ ms}} h^2(t) dt}{\int_{80 \text{ ms}}^{\infty} h^2(t) dt} \right) \quad (\text{Vzorec 6 [5]})$$

Prakticky nelze spočítat nekonečně dlouhou relativní impulsní odezvu, proto se její délka volí nejdelší možná. Minimálně však dvakrát delší než je počet vzorků reprezentující 80 ms. Na následujících stránkách je zobrazena realizace výpočtu hodnoty C80 v programovacím prostředí Matlab.



Obr. 7: Výpočet C80 z relativní impulsní odezvy

```

1  function [ C_80 ] = C80( nahravka, L, Fs, soubor )
2      if nargin == 4
3          if exist(soubor,'file') == 2
4              saved = open(soubor);
5              if isfield(saved, 'C_80')
6                  C_80 = saved.C_80;
7              return
8          end
9      else
10         disp('vybrany soubor neexistuje, vypocet
              nebude ulozen, pouze vracen')
11     end
12 end

```

*Zdrojový kód 13: Hlavička funkce pro výpočet C80*

Hlavička funkce pro výpočet hodnoty C80 na prvním řádku vyžaduje celkem čtyři parametry. První parametr s názvem *nahravka* je dvoukanálový zvukový záznam. Druhý parametr s názvem *L* je číslo, určující délku relativní impulsní odezvy, ze které je spočtena hodnota C80. Třetí parametr *Fs* je hodnota vzorkovací frekvence zvukového záznamu a čtvrtý nepovinný parametr je textový řetězec s názvem *\*.mat souboru* pro uložení nebo načtení již spočtené hodnoty C80.

Následující podmínky testují, zda byl zadán název souboru. Pokud ano, tak je testováno, zda soubor existuje. Pokud soubor existuje, tak poslední podmínka na pátém řádku testuje, zda se v souboru nachází již uložená hodnota C80 z předešlého výpočtu. Pokud je hodnota v souboru nalezena, funkce vrátí tuto hodnotu a ve výpočtu dále nepokračuje.

```

13 posun = 5;
14 left = nahravka(:,1);
15 right = nahravka(:,2);
16 nasobek = sqrt(5/(mean(left.^2)+mean(right.^2)));
17 left = left.*nasobek;
18 right = right.*nasobek;
19 Xr = [zeros(1,posun),right'];
20 Xl = toeplitz(zeros(1,L),[left',zeros(1,posun)]);
21 hLMS=((Xr * Xl') * (inv(Xl * Xl')))' ;

```

*Zdrojový kód 14: Výpočet relativní impulsní odezvy pomocí LMS*

Ve čtrnáctém zdrojovém kódu je uveden výpočet relativní impulsní odezvy pomocí metody nejmenších čtverců pro výpočet hodnoty C80 dle postupu uvedeného ve druhém vzorci.

Na třináctém řádku je do proměnné `posun` uložena hodnota, která určuje o kolik vzorků bude relativní impulsní odezva posunuta. Posunutí relativní impulsní odezvy je důležité z důvodu prevence pohybu hlavní špičky impulsní odezvy mimo počítaný rozsah, který je způsoben nekolmostí zvukového zdroje vůči mikrofonom.

Na čtrnáctém a patnáctém řádku je rozdělen dvoukanálový zvukový záznam do samostatných proměnných. Na následujících třech řádkách je provedena normalizace hlasitosti v obou kanálech a to na zvolenou průměrnou hodnotu vzorků na druhou, jejíž hodnota byla zvolena na hodnotu pět.

Do proměnné `Xr` je uložen vektor pravého kanálu, který je posunutý o počet prvků v proměnné `posun`. Prázdné koeficienty jsou nastavené na nulovou hodnotu. Následující proměnná `Xl` ukládá toeplitzovskou matici vektorů z levého kanálu, jejíž tvar je popsán v druhé kapitole.

Dvacátý první řádek vypočítává do proměnné `hLMS` výslednou relativní impulsní odezvu pomocí metody nejmenších čtverců dle druhého vzorce uvedeného ve druhé kapitole.



```

22     peak = find(hLMS == max(hLMS),1,'first');
23     ms_80 = Fs * 0.08;
24     konec = peak + ms_80 + (L - (ms_80 + posun+10));
25     C_80 = 10 * log10((trapez(hLMS(peak:peak +
ms_80).^2)) / (trapez(hLMS(peak + ms_80:konec).^2)));
26     if nargin == 4
27         if exist(soubor,'file') == 2
28             save(soubor, 'C_80', '-append')
29         end
30     end
31 end

```

*Zdrojový kód 15: Výpočet hodnoty C80*

Do proměnné `peak` na dvacátém druhém řádku je nalezen index, na jehož pozici je maximální hodnota relativní impulsní odezvy. Na následujícím řádku je do proměnné `ms_80` spočten počet vzorků zvukového záznamu, který odpovídá 80 ms.

Na dvacátém čtvrtém řádku je do proměnné `konec` spočten poslední index, který představuje poslední pozici, ze které se vypočítá hodnota C80. Jeho hodnota je počítána tak, aby celková délka pozic, ze kterých se počítá C80, byla vždy stejně dlouhá při stejné celkové délce relativní impulsní odezvy a při posunu pozice hlavní špičky odezvy maximálně o pět pozic na každou stranu.

Na následujícím řádku je do proměnné `C_80` uložena výsledná hodnota parametru C80. Výpočet odpovídá šestému vzorci pro výpočet hodnoty C80. Příkaz `trapez` počítá obsah pod průběhem relativní impulsní odezvy a nahrazuje tím výpočet integrálu.

Ve zbývajících podmínkách je uložena hodnota C80 do souboru, pokud byl zvolen jeho název.

### 3.4 Výpočet D20 a D50

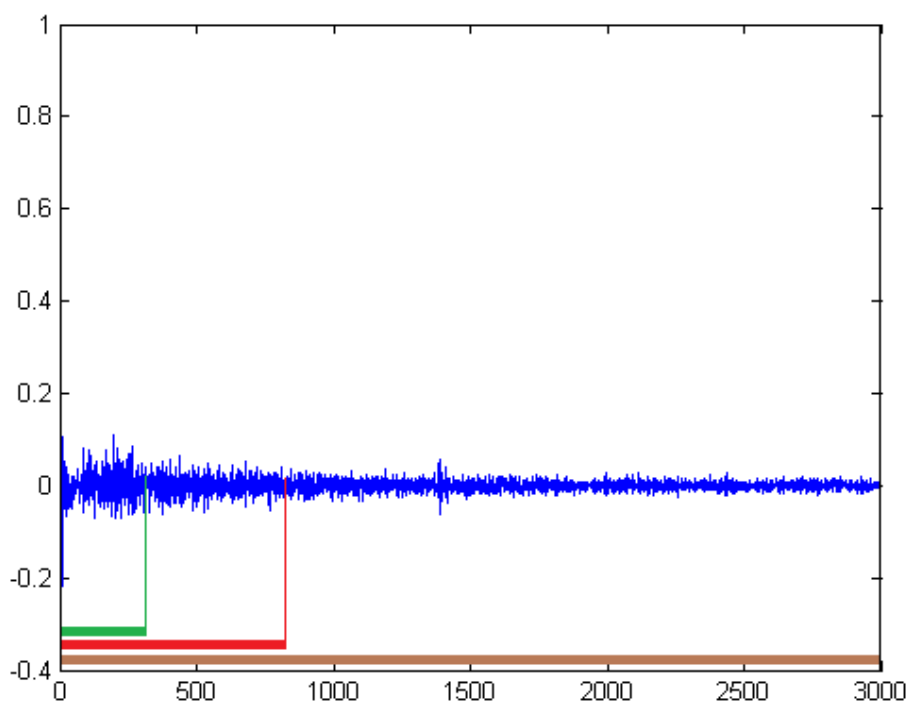
Akustické parametry D20 a D50, známé jako „Deutlichkeit“ popisují srozumitelnost řeči v prostředí s různým dozvukem jako vztah mezi počáteční a koncovou částí relativní impulsní odezvy. [5]

Hodnota D50 se vypočte jako poměr obsahu prvních hodnot relativní impulsní odezvy na druhou odpovídající 50 ms ku obsahu všech hodnot relativní impulsní odezvy na druhou. Hodnota D20 je obdobná s tím rozdílem, že počáteční úsek relativní impulsní odezvy odpovídá 20 ms. Čím vyšší hodnota vyjde, tím víc ozvěn se nachází v prvních 50 ms respektive 20 ms záznamu. S rostoucí hodnotou tedy klesá doba dozvuku.

Hodnota D50 je popsána v sedmém vzorci a na osmém obrázku prvních 50 ms odpovídá červená výseč o délce 800 vzorků, je-li vzorkovací frekvence zvukového záznamu 16 000 Hz. Vzorec pro hodnotu D20 je zobrazen v osmém vzorci a na osmém obrázku prvním 20 ms odpovídá zelená výseč o délce 320 vzorků. Hnědá výseč na osmém obrázku označuje délku celé relativní impulsní odezvy.

$$D_{50} = \frac{\int_0^{50\text{ms}} h^2(t) dt}{\int_0^{\infty} h^2(t) dt} \quad (\text{Vzorec 7 [5]})$$

$$D_{20} = \frac{\int_0^{20\text{ms}} h^2(t) dt}{\int_0^{\infty} h^2(t) dt} \quad (\text{Vzorec 8 [5]})$$



Obr. 8: Výpočet D50 a D20

```
1 function [ D_X0 ] = D50( nahravka, L, Fs, parametr,  
                          soubor )
```

*Zdrojový kód 16: Hlavička funkce pro výpočet D20 a D50*

Prvním parametrem funkce pro výpočet hodnoty D20 a D50 je dvoukanálový zvukový záznam libovolné délky, který se uloží do proměnné `nahravka`. Druhým parametrem je délka relativní impulsní odezvy, ze které se D20 a D50 vypočítá. Třetím parametrem je vzorkovací frekvence zvukového záznamu v hertzech. Čtvrtým parametrem je textový řetězec označující volbu výpočtu D20 nebo D50. Pro výpočet D20 má řetězec tvar „D\_20“ a pro výpočet D50 má tvar „D\_50“. Poslední nepovinný parametr `soubor` obsahuje textový řetězec s názvem souboru pro uložení hodnoty D20 nebo D50.

Dále v kódu následuje testování, zda zvolený soubor existuje, případně je-li v něm uložená spočtená hodnota D20 nebo D50. V případě, že je v souboru nalezena příslušná hodnota, tak jí funkce vrátí a běh programu skončí. Zdrojový kód je obdobný jako u předchozích funkcí.

Před výpočtem relativní impulsní odezvy je u zvukového záznamu normalizována hlasitost. Pro výpočet hodnot D20 nebo D50 je dále spočtená relativní impulsní odezva pomocí metody nejmenších čtverců. Kód zajišťující tyto úkony je uveden ve zdrojovém kódu čtrnáct.

```

2   peak = find(hLMS == max(hLMS), 1, 'first');
3   if strcmp('D_50', parametr)
4       ms_50 = Fs * 0.05;
5       konec = peak + (L - (posun + 5));
6       D_X0 = (trapz(hLMS(peak:peak + ms_50).^2)) /
              (trapz(hLMS(peak:konec).^2));
7   end
8   if strcmp('D_20', parametr)
9       ms_20 = Fs * 0.02;
10      konec = peak + (L - (posun + 5));
11      D_X0 = (trapz(hLMS(peak:peak + ms_20).^2)) /
              (trapz(hLMS(peak:konec).^2));
12  end

```

*Zdrojový kód 17: Výpočet D20 a D50*

Na druhém řádku je do proměnné `peak` uložen index relativní impulsní odezvy, na kterém se nachází maximum.

V následujících podmínkách je zjišťováno, zda se má vypočítat hodnota D20 nebo hodnota D50. Uvnitř podmínek je do proměnných `ms_50` a `ms_20` uložen počet indexů odpovídající 50 ms nebo 20 ms. Do proměnné `konec` je uložen konečný index, pro který se výsledná hodnota bude ještě počítat. Tato hodnota je zvolena tak, aby vzdálenost mezi indexem `peak` a `konec` byla u stejných délek relativní impulsní odezvy vždy stejná bez ohledu na umístění indexu `peak` z důvodu možného posunu vlivem nekolmosti zvukového zdroje vůči mikrofonům.

Do následující proměnné `D_X0` je uložena výsledná hodnota D20 nebo D50 udávaná v procentech. Pokud byl zadán název souboru, je do něj následně hodnota D20 nebo D50 uložena a vrácena. Pokud nebyl zadán název souboru, vypočtená hodnota je pouze vrácena.

## 4 Hledání vhodné parametrizace

### 4.1 Vytvoření databáze nahrávek

Pro potřeby této práce bylo nahráno celkem pět zvukových záznamů v místnostech s různými akustickými vlastnostmi, viz tabulku 1. Tyto akustické vlastnosti byly vypočítány pomocí skriptů v předchozí kapitole. Záznamy byly pořízeny dvojicí mikrofonů, které od sebe byly vzdáleny 13 cm a zdroj zvuku byl reproduktor reprodukcující ženský hlas, který byl vzdálen 1 m kolmo od mikrofonů. Pro výpočet hodnot T60 a EDT bylo místo reprodukce ženského hlasu použito jako zdroj zvuku tlesknutí 1 m před mikrofony.

Rozměry první místnosti jsou 4,4 m × 4,1 m, rozměry druhé místnosti jsou 2,8 m × 2,4 m, rozměry třetí místnosti jsou 1,8 m × 1,7 m a rozměry čtvrté místnosti jsou 2,8 m × 4,9 m. Výška všech místností je 2,7 m. U poslední místnosti se jedná o schodiště panelového domu, které má rozměry 4,1 m × 6 m. Výška celého schodiště je pak 30 m.

Na následující stránce je popsán skript v programovacím prostředí Matlab, pomocí kterého byly veškeré záznamy pořízeny.

*Tabulka 1: Naměřené akustické parametry zvukových záznamů*

Místnost	Hodnoty akustických parametrů				
	T60 [s]	EDT [s]	C80 [dB]	D20 [%]	D50 [%]
1	0,307	0,0169	8,77	61	80
2	0,315	0,0124	6,19	45	68
3	0,379	0,0079	5,78	39	66
4	0,313	0,0454	8,37	65	81
5	0,910	0,0109	5,54	52	67

```
1  Fs = 16000;
2  NBITS = 16;
3  channels = 2;
4  zaznam = audiorecorder(Fs, NBITS, channels);
5  record(zaznam);
6  pause
7  stop(zaznam);
8  nahravka = zaznam.getaudiodata();
9  nahravka(:,1) = nahravka(:,1) - mean(nahravka(:,1));
10 nahravka(:,2) = nahravka(:,2) - mean(nahravka(:,2));
```

*Zdrojový kód 18: Zaznamenání zvukového signálu*

Na prvním řádku je do proměnné `Fs` uložena hodnota vzorkovací frekvence v hertzech (horizontální rozlišení). Na druhém řádku je do proměnné `NBITS` uložen počet bitů využitých ke kvantování záznamu (vertikální rozlišení). Dále do proměnné `channels` je uložen počet kanálu zvukového záznamu. Na čtvrtém řádku je nastavena proměnná `zaznam` pro nahrávání.

Na pátém až sedmém řádku probíhá záznam zvukového signálu, dokud není stisknuta libovolná klávesa.

Na osmém řádku je do proměnné `nahravka` uložen nahraný zvukový záznam, ze kterého je na devátém a desátém řádku odstraněna stejnosměrná složka odečtením průměrné hodnoty celého kanálu od jednotlivých vzorků.

## 4.2 Volba délky relativní impulsní odezvy

Pro výpočet průběhu závislosti parametrů  $c_1$ ,  $c_2$  a  $c_3$  z pátého vzorce na řídkosti relativní impulsní odezvy je třeba pro každou místnost zvolit optimální délku relativní impulsní odezvy tak, aby relativní impulsní odezva spočtená pomocí metody nejmenších čtverců dokázala rozdílem zfiltrovaného levého kanálu touto odezvou a pravého kanálu potlačit signál stejně, jako relativní impulsní odezvy pro ostatní místnosti. Pro stejné potlačení signálu je třeba zvolit délku relativní impulsní odezvy pro každou místnost jinak dlouhou.

Účinnost potlačení závisí také na době dozvuku ( $T_{60}$ ), tzn. čím delší je doba dozvuku, tím delší je potřeba relativní impulsní odezva, aby dokázala potlačit hlasitost stejně. Z tohoto důvodu byla zvolena nejdelší možná relativní impulsní odezva pro místnost s nejdelší dobou dozvuku. Podle první tabulky má nejdelší dobu dozvuku místnost 5, a to 0,91 s. Byla jí tedy přidělena nejdelší možná délka relativní impulsní odezvy, pro kterou byl schopen Matlab vypočítat veškeré výpočty. Z paměťových důvodů pro záznam dlouhý 1 s byla zvolena délka relativní impulsní odezvy pro pátou místnost 3 000. Velikost vzájemného potlačení kanálů relativní impulsní odezvou je tedy 7,947 dB.

Pomocí programu uvedeného na následujících stránkách byly dopočítány délky relativních impulsních odezev i pro ostatní místnosti tak, aby se co nejvíce blížily stejné intenzitě potlačení. Výsledné spočítané hodnoty jsou uvedeny v tabulce 2.

Tabulka 2: Volba délky relativní impulsní odezvy

Místnost	Míra potlačení [dB]	Výsledná délka relativní impulsní odezvy
1	7,9438	720
2	7,9469	1 115
3	7,9430	2 152
4	7,9385	402
5	7,9470	3 000

```
1 function [ pokles ] = LMS_pokles( nahravka, L )
```

*Zdrojový kód 19: Hlavička funkce pro výpočet intenzity potlačení signálu v závislosti na délce relativní impulsní odezvy*

V devatenáctém zdrojovém kódu je uvedena hlavička funkce, která v parametrech vyžaduje dvoukanálový zvukový signál a délku relativní impulsní odezvy. Tato funkce vrací rozdíl v hlasitosti mezi původním signálem a rozdílem mezi zfiltrovaným levým a pravým kanálem.

Po hlavičce následuje výpočet relativní impulsní odezvy pomocí metody nejmenších čtverců, jejíž zdrojový kód je uveden ve třináctém zdrojovém kódu. Výsledná relativní impulsní odezva je uložena do proměnné hLMS.

```
2     konvoluce = conv(left, hLMS);  
3     konvoluce = konvoluce(posun + 1:size(right,1) +  
                           posun, :);  
4     zbytkovy_signal = right - konvoluce;
```

*Zdrojový kód 20: Výpočet signálu, který vznikl rozdílem pravého kanálu a zfiltrovaného levého kanálu*

Na druhém řádku je do proměnné konvoluce uložen výsledný signál, který vznikl konvolucí mezi levým signálem left a relativní impulsní odezvou spočtenou pomocí metody nejmenších čtverců hLMS. Na následujícím řádku jsou odříznuty počáteční a koncové části signálu, které po konvoluci přesahují délku originálního levého kanálu. Bez odstranění těchto částí by délka signálu neodpovídala pravému kanálu. Tyto dva vektory by nebylo možno vzájemně odečíst.

Na následujícím čtvrtém řádku je do proměnné zbytkovy\_signal uložen vektor, který vznikl rozdílem originálního pravého signálu right a zfiltrovaného levého signálu konvoluce.



```

5     puvodni_hlasitost =
           10*log10(mean(((left+right)/2).^2));
6     LMS_hlasitost =
           10*log10(mean(zbytkovy_signal.^2));
7     pokles = puvodni_hlasitost - LMS_hlasitost;
8     end

```

*Zdrojový kód 21: Výpočet rozdílu hlasitostí mezi originálním zvukovým záznamem a zfiltrovaným zvukovým záznamem*

Na pátém řádku je do proměnné `puvodni_hlasitost` spočtená hlasitost originálního zvukového záznamu jako desetinásobek logaritmu o základu deset průměrné hodnoty obou kanálů na druhou. Na následujícím řádku je obdobně spočtena hlasitost signálu, která vznikla rozdílem zfiltrovaného levého kanálu relativní impulsní odezvou a pravého kanálu. Funkce nakonec tuto hodnotu vrátí a program skončí.

Na sedmém řádku je do proměnné `pokles` spočten výsledný rozdíl obou hlasitostí. Všechny hlasitosti v uvedeném zdrojovém kódu jsou vypočteny v dB.

### 4.3 Výpočet závislosti parametrů $c_1$ , $c_2$ a $c_3$ na počtu nenulových prvků řídké relativní impulsní odezvy hrubou silou

Pro získání závislosti parametrů  $c_1$ ,  $c_2$  a  $c_3$  váhovací funkce (viz vzorec 5) na počtu nenulových prvků relativní impulsní odezvy byly vypočteny řídké relativní impulsní odezvy v určitém rozsahu těchto parametrů. Rozsahy jednotlivých parametrů  $c_1$  až  $c_3$  byly zvoleny v okolí pravděpodobných optimálních hodnot. Optimální hodnota parametrů  $c_1$  až  $c_3$  je:

- Pokud neklesne schopnost řídké relativní impulsní odezvy potlačit vzájemně kanály zvukového záznamu o více než 1 dB oproti schopnosti relativní impulsní odezvy spočtené pomocí metody nejmenších čtverců.
- Když řídká relativní impulsní odezva obsahuje co nejméně nenulových prvků.

Před samotným výpočtem je záznam normalizován na průměrnou hodnotu 5 všech vzorků na druhou. Důvodem je možnost zkreslení výsledků v případě, že v jiné místnosti měl reprodukováný zvuk jinou hlasitost.

Jelikož je nežádoucí potlačovat hlavní špičku relativní impulsní odezvy, tak hodnota parametru  $c_1$  byla zvolena na konstantních 0,01, protože tento parametr pouze vertikálně posouvá váhovací funkci a minimum této funkce určuje míru potlačení hlavní špičky relativní impulsní odezvy. Pro ostatní parametry byl zvolen interval hodnot od 0,01 až do 1 s krokem 0,01.

Výsledné úrovně počtu nenulových prvků jsou pro pozdější zpracování uloženy do souboru. Pokud řídká relativní impulsní odezva nesplní pravidlo horšího potlačení maximálně o 1 dB je místo počtu nenulových prvků této odezvy uložena hodnota NaN. Spolu s touto hodnotou jsou ukládány i kombinace hodnot parametrů  $c_1$  až  $c_3$ .

V případě, že dojde při výpočtu k přerušení běhu programu, jsou do souboru uloženy i parametry určující délku relativní impulsní odezvy, rozsahy jednotlivých parametrů  $c_1$  až  $c_3$  a velikost posunu hlavní špičky relativní impulsní odezvy. Po znovu spuštění funkce běh programu pokračuje ze stejného místa, jelikož pro velké rozsahy parametrů  $c_1$  až  $c_3$  a velké délky relativní impulsní odezvy trvá běh funkce v řádu několika hodin až dnů.

V následujících zdrojových kódech je zobrazena realizace výpočtu množství nenulových prvků řídké relativní impulsní odezvy v závislosti na parametrech váhovací funkce.

```
1 function [] = najdi_hrubou_silou( nahravka, L, c1,
                                c2, c3, uloz_jako )
```

*Zdrojový kód 22: Hlavička funkce pro hledání nejřidších odezev hrubou silou*

Funkce pro výpočet množství nenulových koeficientů řídké relativní impulsní odezvy v závislosti na parametrizaci váhovací funkce vyžaduje celkem šest parametrů. První parametr `nahravka` obsahuje dvoukanálový zvukový záznam z měřeného prostředí. Druhý parametr `L` obsahuje délku relativní impulsní odezvy. Třetí až pátý parametr `c1`, `c2` a `c3` obsahují intervaly hodnot jednotlivých parametrů váhovací funkce pro výpočet řídké relativní impulsní odezvy. A pátý parametr `uloz_jako` obsahuje textový řetězec s názvem souboru `*.mat`, do kterého budou ukládány spočtené hodnoty.

Dále následuje výpočet relativní impulsní odezvy pomocí metody nejmenších čtverců, který je uveden ve čtrnáctém zdrojovém kódu a výpočet rozdílu hlasitostí mezi originálním záznamem a zbylým signálem po odečtení pravého kanálu od zfiltrovaného levého kanálu, jak je uvedeno ve dvacátém prvním zdrojovém kódu.

```
2 A = Xl * Xl' / size(nahravka, 1);
3 y = (Xr * Xl')' / size(nahravka, 1);
```

*Zdrojový kód 23: Výpočet matice a vektoru pro výpočet řídké relativní impulsní odezvy*

Pro výpočet řídké relativní impulsní odezvy je potřeba spočítat matici `A`, která vznikne maticovým násobením matic `Xl` a její transpozice. Veškeré prvky matice `A` jsou poté vyděleny počtem vzorků zvukového záznamu v jednom kanále. Vektor `y` vznikne transpozicí výsledku vzniklého maticovým násobením vektoru `Xr` a transponované matice `Xl`. Obsah matice `Xl` a vektoru `Xr` je popsán v kapitole 2.1.

```
4 for a = c1
5     for b = c2
6         for c = c3
```

*Zdrojový kód 24: Cykly pro výpočet řídkých relativních impulsních odezev*

Dále v programu následuje třikrát vnořený cyklus, který zajistí výpočet řídké relativní impulsní odezvy pro všechny kombinace parametrů `c1` až `c3` v daných rozsazích. Aktuální hodnota parametru `c1` je uložena do proměnné `a`, parametr `c2` je uložen do proměnné `b` a parametr `c3` je uložen v proměnné `c`.

```

7  W=a*exp(b*(abs((1:L) '-peak).^c));
8  opts.W = W;
9  vysledek = l1homotopy(A,y,opts);
10 hL1 = vysledek.x_out;
11 konvoluce = conv(left,hL1);
12 konvoluce = konvoluce(posun+1:size(right,1)+posun,:);
13 zbytkovy_signal = right-konvoluce;
14 L1_hlasitost = 10*log10(mean(zbytkovy_signal.^2));
15 L1decibel = puvodni_hlasitost - L1_hlasitost;

```

*Zdrojový kód 25: Výpočet řídké relativní impulsní odezvy*

Na sedmém řádku je do proměnné  $W$  spočítán průběh hodnot váhovací funkce, která je vyjádřena ve vzorci 5. Příkaz  $(1:L)'$  značí vektor s indexy od 1 do délky relativní impulsní odezvy. Ve vzorci 5 je tato část označena  $i$ . Proměnná  $peak$  značí index nejvyšší hodnoty relativní impulsní odezvy spočtenou pomocí metody nejmenších čtverců. Výpočet její hodnoty je zobrazen ve zdrojovém kódu 15 na 22. řádku.

Na osmém řádku je výsledný průběh váhovací funkce uložen do pomocné proměnné  $opts.W$ .

Na devátém řádku je do proměnné  $vysledek$  uložen výstup z  $L_1$  homotopního algoritmu. Na následujícím řádku je z předchozího výstupu získána řídká relativní impulsní odezva, která je následně uložena do proměnné  $hL1$ .

Jedenáctý řádek vypočítá konvoluci mezi originálním levým zvukovým kanálem a řídkou relativní impulsní odezvou. Na následujícím řádku jsou z výsledné konvoluce odříznuty hodnoty na začátku a na konci signálu, které vlivem konvoluce přesahují rozměr originálního signálu. Do proměnné  $zbytkovy\_signal$  je uložen signál, který vznikl rozdílem zfiltrovaného levého zvukového signálu a originálního pravého zvukového signálu.

Na čtrnáctém řádku je ze signálu uloženém v proměnné  $zbytkovy\_signal$  spočtena hlasitost v dB. Následně je do proměnné  $L1decibel$  uložen rozdíl hlasitostí původního zvukového záznamu a předchozího potlačeného signálu.

```

16  if (abs(LMSdecibel - L1decibel) <= 1)
17      h0 = sum(hL1 ~= 0);
18  else
19      h0 = NaN;
20  end
21  h0vect(i,1) = h0;
22  parametry(i,:) = [a,b,c];
23  save(uloz_jako, 'parametry', 'h0vect', '-append');
24  i = i+1;

```

*Zdrojový kód 26: Spočtení a uložení počtu nenulových prvků v řídké relativní impulsní odezvě*

Na šestnáctém řádku se nachází podmínka, která zjišťuje, zda je rozdíl v potlačených signálech relativní impulsní odezvou spočtenou pomocí metody nejmenších čtverců a řídkou relativní impulsní odezvou menší než 1 dB. Pokud je podmínka splněna, je do proměnné `h0` uložen počet nenulových prvků řídké relativní impulsní odezvy. Pokud podmínka splněna není, je do proměnné `h0` uložena hodnota `NaN`.

Dále na dvacátém prvním řádku je do vektoru `h0vect` postupně ukládána hodnota `h0`. Na následujícím řádku jsou do vektoru `parametry` přidávány aktuální kombinace parametrů  $c_1$ ,  $c_2$  a  $c_3$ . Nakonec na dvacátém třetím řádku jsou tyto vektory s hodnotami po každém průběhu cyklem uloženy do souboru.

Po dokončení ukládání je proměnná `i` inkrementovaná o jednotku a obsah funkce uvedený ve zdrojových kódech 25 a 26 se cyklicky vykonává, dokud nejsou vyčerpány veškeré kombinace hodnot parametrů  $c_1$ ,  $c_2$  a  $c_3$ .

## 4.4 Hledání kombinace parametrů $c_1$ , $c_2$ a $c_3$ , pro která je řídká relativní impulsní odezva nejřidší

Z předešlé funkce byl vygenerován soubor s hodnotami nenulových prvků řídkých relativních impulsních odezev, které byly vygenerovány v určitých rozsazích parametrů váhovací funkce  $c_1$ ,  $c_2$  a  $c_3$ . V následující funkci je popsáno zobrazení průběhu závislosti počtu nenulových prvků řídké relativní impulsní odezvy na parametru  $c_2$  a  $c_3$ , jelikož parametr  $c_1$  byl zvolen na konstantní hodnotu. Dále je zvýrazněn průběh nejřidších hodnot relativních impulsních odezev. Tento průběh je proložen parabolou pomocí metody nejmenších čtverců a na této parabole je nalezeno minimum, ze kterého je odečtena hodnota parametru  $c_2$  nebo  $c_3$ , jako optimální pro konkrétní místnost.

```
1 function [ graf, min_val ] = vykresli_graf
    (nazev_souboru, barva, parametr)
```

*Zdrojový kód 27: Hlavička funkce pro nalezení optimální hodnoty parametru váhovací funkce z předpočítaných dat*

Ve zdrojovém kódu 27 je zobrazena hlavička funkce, která vykreslí graf průběhu počtu nenulových prvků řídké relativní impulsní odezvy v závislosti na parametrech  $c_2$  a  $c_3$ . Zároveň vrátí hodnotu vybraného parametru, která byla nalezena jako neoptimálnější.

Funkce vyžaduje celkem tři parametry. První parametr `nazev_souboru` obsahuje textový řetězec s názvem souboru `*.mat`, ve kterém se nachází vypočítané součty nenulových prvků řídkých relativních impulsních odezev ve zvoleném rozsahu. Druhý parametr `barva` vyžaduje pole, které obsahuje tři hodnoty reprezentující intenzity barev RGB v rozsahu 0 až 1. Touto barvou bude vykreslen průběh nejřidších relativních impulsních odezev. Třetí parametr `parametr` vyžaduje textový řetězec „C2“ nebo „C3“, který vybírá parametr podle kterého má být minimum nalezeno a zobrazen průběh.

Funkce vrátí referenci na vykreslený graf v proměnné `graf` a hodnotu zvoleného parametru, který byl nalezen jako neoptimálnější v proměnné `min_val`.

```

2     j=1;
3     for i=1:sizeC2
4         matice_zavislosti(i,:)=saved.h0vect(j:j+sizeC3-1);
5         j=j+sizeC3;
6     end

```

*Zdrojový kód 28: Vytvoření matice z vektoru*

Dále je z načteného vektoru `saved.h0vect`, který obsahuje počet nenulových prvků řádké relativní impulsní odezvy v závislosti na parametrech  $c_2$  a  $c_3$ , vytvořena matice `matice_zavislosti`, kde v řádcích jsou součty nenulových prvků závislé na parametru  $c_3$  a ve sloupcích na parametru  $c_2$ .

```

7     if(strcmp(parametr, 'C3'))
8         matice_zavislosti = matice_zavislosti';
9         osaX = saved.c2;
10    else
11        osaX = saved.c3;
12    end

```

*Zdrojový kód 29: Výběr závislosti na vybraném parametru*

V dvacátém devátém zdrojovém kódu je uvedena podmínka, která testuje, zda byl zvolen výpočet pro parametr  $c_3$ . Pokud ano, tak se matice `matice_zavislosti` transponuje a proměnná `osaX` se naplní rozsahem hodnot parametru  $c_2$ , který byl uložen do souboru spolu s výpočty. Pokud podmínka neprošla, tak do proměnné `osaX` je uložen pouze rozsah hodnot parametru  $c_3$ .

```

13    for i = 1 : size(matice_zavislosti,1)
14        plot(osaX,matice_zavislosti(i,:), 'Color', [.8 .8 .8]);
15        hold on
16    end

```

*Zdrojový kód 30: Vykreslení průběhu závislosti hodnoty nenulových prvků relativní impulsní odezvy na vybraném parametru*

Ve třicátém zdrojovém kódu je napsán kód pro vykreslení průběhu závislosti součtu nenulových prvků řádké relativní impulsní odezvy na vybraném parametru. Jednotlivé průběhy jsou přečteny z matice `matice_zavislosti` a jsou vykresleny šedivou barvou. Proměnná `osaX` obsahuje rozsah hodnot souřadnice  $x$  z vybraného parametru.

```

17 nejridsi_graf = NaN * ones(size(osaX,2),1);
18 for i = 1 : size(matice_zavislosti,1)
19     index = find(matice_zavislosti(i,:) ==
                min(matice_zavislosti(i,:),1,'last'));
20     if ~isempty(index) &&
        isnan(matice_zavislosti(i,size(osaX,2)))
21         if ((isnan(nejridsi_graf(index,1))) ||
            (nejridsi_graf(index,1) >
             min(matice_zavislosti(i,:))))
22             nejridsi_graf(index,1) =
                min(matice_zavislosti(i,:));
23         end
24     end
25 end

```

*Zdrojový kód 31: Nalezení nejřidších relativních impulsních odezev*

Na sedmnáctém řádku je proměnná `nejridsi_graf` inicializovaná na vektor hodnot `NaN` o délce rozsahu zvoleného parametru. Za tímto řádkem se nachází cyklus, který projde všechny řádky matice `matice_zavislosti`.

Na devatenáctém řádku je podmínka, zda bylo minimum na daném řádku nalezeno a zda poslední index na daném řádku je `NaN`. Kdyby bylo minimum nalezeno, ale poslední prvek by obsahoval hodnotu, která neodpovídá hodnotě `NaN`, tak je toto minimum ignorováno, jelikož se může minimum teoreticky nacházet mimo rozsah uložených hodnot.

Pokud byla podmínka splněna, tak se na dvacátém prvním řádku ještě otestuje, zda je pozice nalezeného minima ve vektoru `nejridsi_graf` už obsazená, a pokud ano, zda nové minimum je menší než to dříve nalezené. Pokud je tedy pozice prázdná, či je nová hodnota nižší, je na tuto pozici do vektoru `nejridsi_graf` uložena nejnižší hodnota nalezená na aktuálním řádku matice `matice_zavislosti`.



```

26 nans = isnan(nejridsi_graf);
27 ind = 1:size(osaX,2);
28 nejridsi_graf(nans) =
    interp1(ind(~nans),nejridsi_graf(~nans),ind(nans));
29 graf = plot
    (osaX,nejridsi_graf,'Color',barva,'LineWidth',2);

```

*Zdrojový kód 32: Interpolace mezi prázdnými indexy, kde se nenachází nejřidší hodnota relativní impulsní odezvy*

Jelikož proměnná `nejridsi_graf` nemusí mít všechny indexy obsazeny nalezenou minimální hodnotou, jsou místa bez nalezené hodnoty mezi dvěma nalezenými lineárně proloženy. Takto je provedeno ve zdrojovém kódu 32.

Do proměnné `nans` jsou uloženy indexy proměnné `nejridsi_graf`, na jejichž pozicích je obsažena hodnota NaN. Do proměnné `ind` jsou uloženy veškeré indexy proměnné `osaX`.

Na dvacátém osmém řádku jsou proloženy prvky NaN v proměnné `nejridsi_graf` pomocí vestavěné funkce `interp1`. Výsledný proložený průběh nejmenších součtů nenulových prvků řídkých relativních impulsních odezvy je na dvacátém devátém řádku vykreslen do grafu a reference na tento graf je uložena do proměnné `graf`.

```

30 souvisly_prubeh = ~isnan(nejridsi_graf);
31 prvni = find(souvisly_prubeh == 1,1,'first');
32 posledni = find(souvisly_prubeh == 1,1,'last');
33 h = polyfit(osaX(prvni:posledni),
              nejridsi_graf(prvni:posledni)',2);
34 parabola = h(1).*osaX(1:size(osaX,2)).^2 +
              h(2).*osaX(1:size(osaX,2)) + h(3);
35 plot(osaX(1:size(osaX,2)), parabola, 'Color',
       [1 .6 0]);

```

*Zdrojový kód 33: Proložení bodů s nejřidšími relativními impulsními odezvami parabolou*

Ve třicátém třetím zdrojovém kódu je zobrazen kód, který zajišťuje proložení průběhu v proměnné `nejridsi_graf` parabolou.

Na třicátém řádku jsou do proměnné `souvisly_graf` uloženy indexy proměnné `nejridsi_graf`, které neobsahují hodnotu NaN. Na dalším řádku je do proměnné `prvni` uložen první index proměnné `souvisly_prubeh`, který neobsahuje hodnotu NaN a do proměnné `posledni` je uložen naopak poslední index, který neobsahuje hodnotu NaN.

Na třicátém třetím řádku jsou do proměnné `h` spočteny parametry paraboly  $a, b, c$  ze vzorce  $y = ax^2 + bx + c$ , která prokládá souvislý průběh hodnot v proměnné `souvisly_prubeh`. Na dalším řádku jsou pro jednotlivé indexy osy  $x$  spočteny hodnoty paraboly, které jsou uloženy do proměnné `parabola`. Následně je tato parabola vykreslena do grafu oranžovou barvou.

```
36 rozsahC = osaX(1:size(osaX,2));  
37 min_val = rozsahC(find(parabola == min(parabola), 1,  
                        'first'));
```

*Zdrojový kód 34: Najítí parametru c2 nebo c3 s nejdřívší možnou relativní impulsní odezvou na parabole*

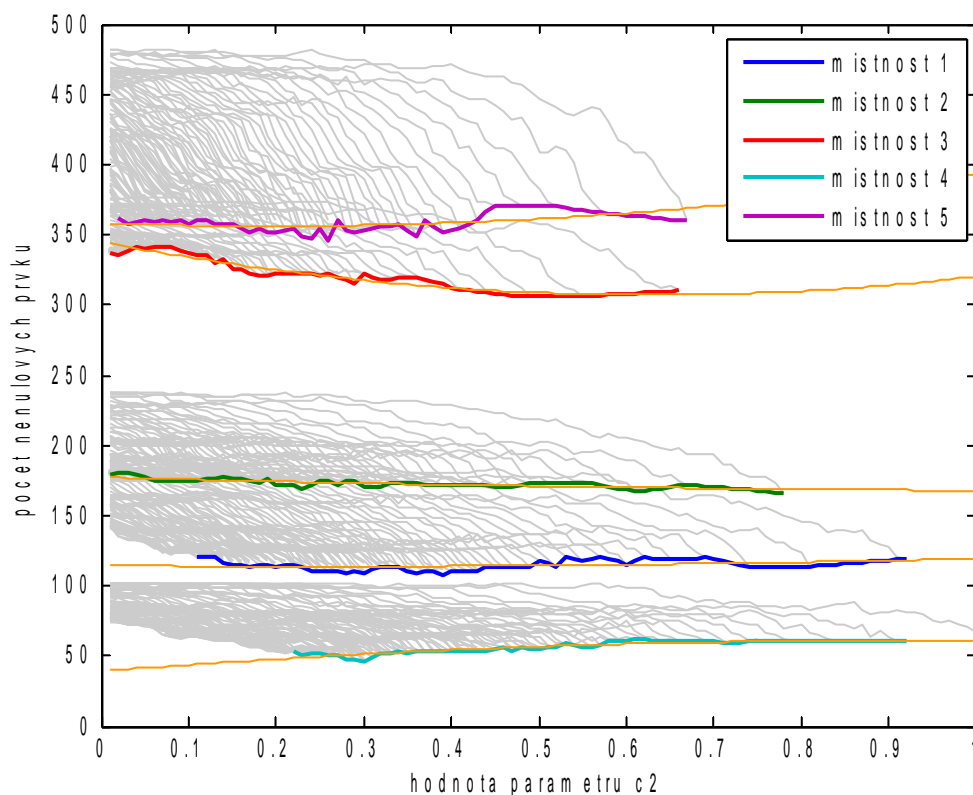
Na třicátém šestém řádku jsou do proměnné `rozsahC` zkopírovány veškeré hodnoty v proměnné `osaX`. Následně na dalším řádku je do proměnné `min_val` nalezena hodnota zvoleného parametru váhovací funkce, na které se podle proložené paraboly nachází nejmenší počet nenulových prvků řídké relativní impulsní odezvy. Tato hodnota je nakonec funkcí vrácena.

Na obrázcích devět a deset jsou zobrazeny výsledné průběhy součtů nenulových prvků řídké relativní impulsní odezvy v závislosti na parametrech váhovací funkce  $c_2$  a  $c_3$ . Tyto průběhy jsou na obrázcích zobrazeny šedou barvou a tuto část vykreslila část programu uvedená ve třicátém zdrojovém kódu.

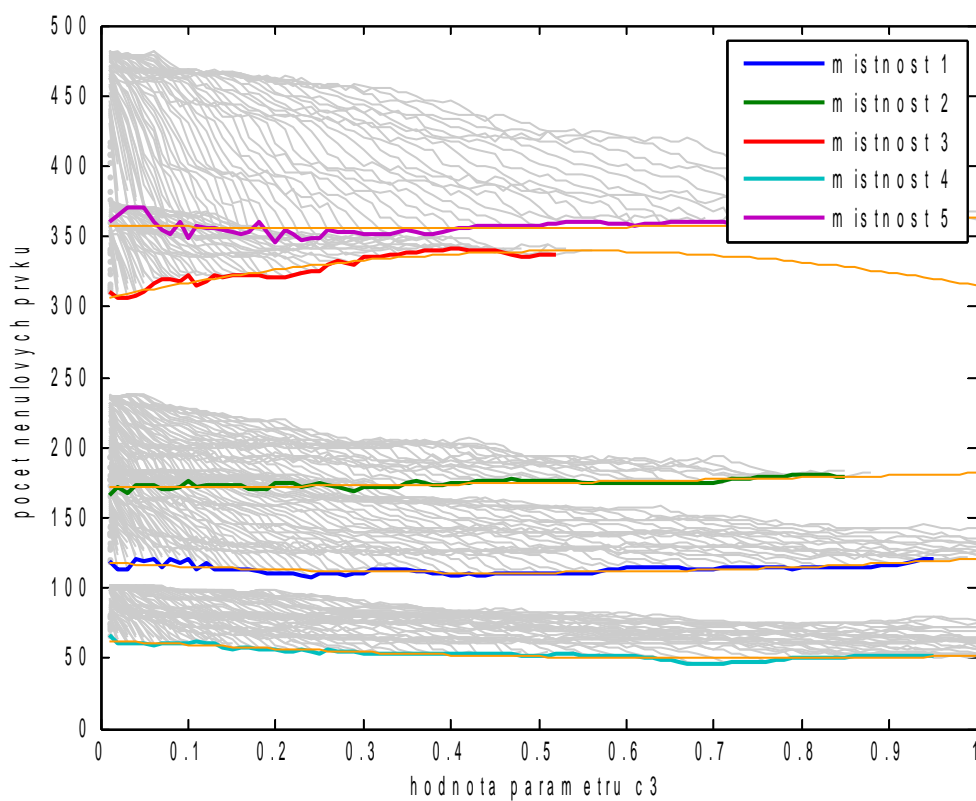
Různými barvami jsou zvýrazněny průběhy, kde jsou pro danou místnost nejřidší relativní impulsní odezvy. Nalezení nejřidších relativních impulsních odezev je realizováno v třicátém prvním zdrojovém kódu a jeho lineární proložení je provedeno ve třicátém druhém zdrojovém kódu.

Proložení nejřidšího průběhu oranžovou parabolou je realizováno ve třicátém třetím zdrojovém kódu a hodnota parametru  $c_2$  nebo  $c_3$  v místě, kde se nachází na parabole minimum, je popsáno ve třicátém čtvrtém zdrojovém kódu.

Výsledné optimální hodnoty parametrů váhovací funkce  $c_1$ ,  $c_2$  a  $c_3$  pro naměřené místnosti jsou uvedeny ve třetí tabulce.



Obr. 9: Graf závislosti součtu nenulových prvků řídké impulsní odezvy na parametru váhovací funkce  $c_2$



Obr. 10: Graf závislosti součtu nenulových prvků řídké impulsní odezvy na parametru váhovací funkce  $c_3$

Tabulka 3: Optimální hodnoty váhovací funkce pro měřené místnosti

Místnost	Hodnoty optimálních parametrů váhovací funkce		
	$c_1$	$c_2$	$c_3$
1	0,01	0,24	0,45
2	0,01	1	0,01
3	0,01	0,63	0,01
4	0,01	0,01	0,72
5	0,01	0,2	0,36

## 4.5 Hledání závislosti parametrů váhovací funkce na akustických parametrech

V následující tabulce jsou zobrazeny ideální parametry  $c_2$  a  $c_3$  váhovací funkce pro jednotlivé naměřené místnosti, které byly změřeny předchozí funkcí. K těmto parametrům jsou vypsány i jednotlivé naměřené akustické vlastnosti jednotlivých místností.

Jelikož vhodná parametrizace váhovací funkce pro výpočet řídké relativní impulsní odezvy je závislá zejména na době dozvuku, je pro vhodnou parametrizaci vybrán parametr T60. Jako další parametr, na kterém podle tabulky 4 nejvíce závisí parametry  $c_2$  a  $c_3$  byl vybrán akustický parametr C80. Parametry T60 a C80 jsou tedy nejideálnější pro výpočet parametrů váhovací funkce, protože nejlépe charakterizují akustické prostředí.

V následujících zdrojových kódech je uveden výpočet funkce, která vyjadřuje závislost vhodné parametrizace váhovací funkce na akustických parametrech T60 a C80. Pro parametrizaci váhovací funkce v závislosti na T60 a C80 byla zvolena lineární funkce.

Tabulka 4: Porovnání ideálních parametrů váhovací funkce s akustickými parametry

Místnost	Optimální parametry váhovací funkce		Akustické parametry				
	$c_2$	$c_3$	C80 [dB]	D20 [%]	D50 [%]	T60 [s]	EDT [s]
1	0,24	0,45	8,77	61	80	0,307	0,0169
2	1	0,01	6,19	45	68	0,315	0,0124
3	0,63	0,01	5,78	39	66	0,379	0,0079
4	0,01	0,72	8,37	65	81	0,313	0,0454
5	0,2	0,36	5,54	52	67	0,910	0,0109

```

1   function [ parametryC2, parametryC3 ] =
        najdi_zavislosti( varargin )

```

*Zdrojový kód 35: Hlavička pro výpočet závislostní funkce parametrů c2 a c3 na akustických parametrech T60 a C80*

Funkce pro výpočet závislosti optimálních parametrů váhovací funkce na akustických parametrech T60 vyžaduje pouze libovolný počet parametrů, které obsahují textový řetězec s názvem souboru \*.mat, ve kterém jsou uloženy spočítané hodnoty T60, C80 a hodnoty součtů nenulových prvků v závislosti na parametrech  $c_2$  a  $c_3$ , které jsou vypočítány hrubou silou ve zvoleném rozsahu.

Funkce vrátí parametry  $a_1$ ,  $b_1$  a  $c_1$  pro výpočet parametru váhovací funkce  $c_2$  a parametry  $a_2$ ,  $b_2$  a  $c_2$  pro výpočet parametru váhovací funkce  $c_3$  ve tvaru  $c_2 = a_1 + b_1 \cdot T_{60} + c_1 \cdot c_{80}$  a  $c_3 = a_2 + b_2 \cdot T_{60} + c_2 \cdot c_{80}$ .

```

2   for i = 1:nargin
3       [grafC2(i), minC2(i)] =
            vykresli_graf(char(varargin(i)),
            barvy(mod(i, size(barvy,1)), :), 'C2');
4   end
5   for i = 1:nargin
6       [grafC3(i), minC3(i)] = vykresli_graf
(char(varargin(i)), barvy(mod(i, size(barvy,1)), :), 'C3');
7       C80(i) = saved.C_80;
8       T_60(i) = saved.T_60;
9   end

```

*Zdrojový kód 36: Načtení optimálních parametrů c2 a c3 z naměřených místností a jejich akustických parametrů T60 a C80*

V cyklech uvedených ve třicátém šestém zdrojovém kódu jsou, pro každý soubor uvedený v parametrech funkce, do proměnné minC2 a minC3 uloženy optimální hodnoty parametrů váhovací funkce  $c_2$  a  $c_3$ , které jsou spočteny pomocí funkce z předchozí kapitoly.

Ve druhém for cyklu jsou navíc načteny do proměnné C80 a T\_60 hodnoty C80 a T60 z každé měřené místnosti.

```

10 figure();
11 parametryC2 = fit([T_60', C80'], minC2', 'poly11');
12 plot(parametryC2, [T_60', C80'], minC2');
13 xlabel('T_6_0');
14 ylabel('C_8_0');
15 zlabel('C2');
16 figure();
17 parametryC3 = fit([T_60', C80'], minC3', 'poly11');
18 plot(parametryC3, [T_60', C80'], minC3');
19 xlabel('T_6_0');
20 ylabel('C_8_0');
21 zlabel('C3');

```

*Zdrojový kód 37: Výpočet a zobrazení závislostní funkce parametrů  $c_2$  a  $c_3$  na akustických parametrech  $T_{60}$  a  $C_{80}$*

Ve třicátém sedmém zdrojovém kódu na jedenáctém řádku jsou naměřené závislosti optimálních parametrů  $c_2$  na akustických parametrech  $T_{60}$  a  $C_{80}$  proloženy plochou pomocí vestavěné funkce `fit`. Výsledné parametry funkce pro výpočet  $c_2$  jsou uloženy do proměnné `parametryC2`.

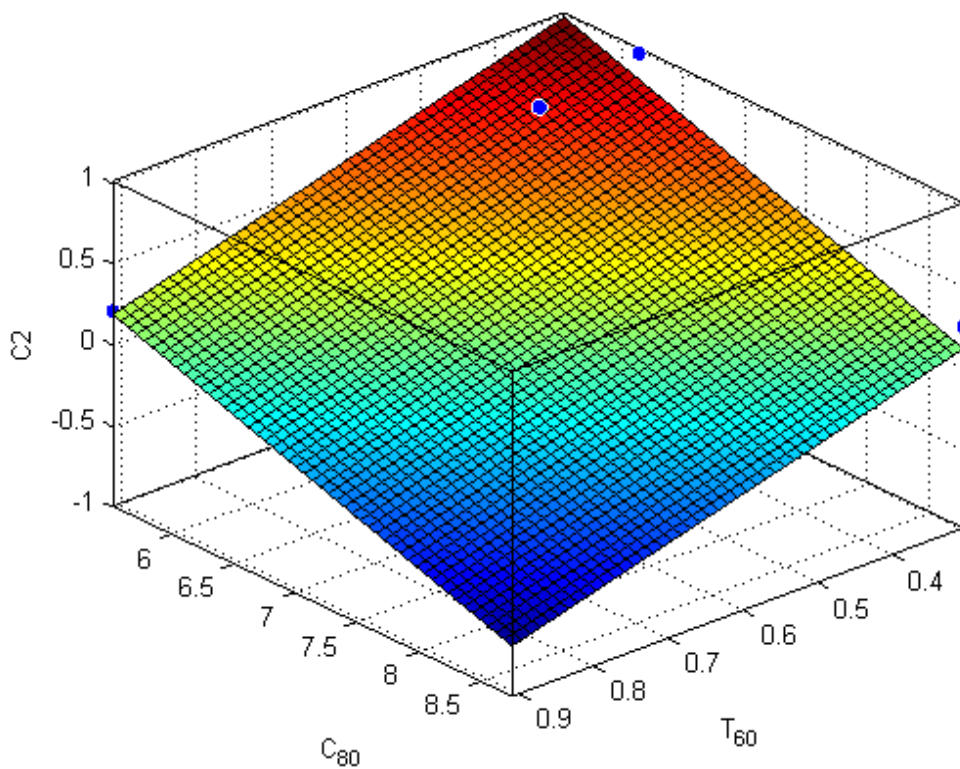
Na dvanáctém řádku jsou tyto body a proložená plocha vykresleny do grafu. Na třináctém až čtrnáctém řádku jsou popsány osy vykresleného grafu.

Na sedmnáctém až dvacátém prvním řádku je shodný výpočet funkce, která určuje závislost parametru  $c_3$  na akustických parametrech  $T_{60}$  a  $C_{80}$ . Výsledný graf s naměřenými body a proloženou plochou je zobrazen na obrázcích 11 a 12.

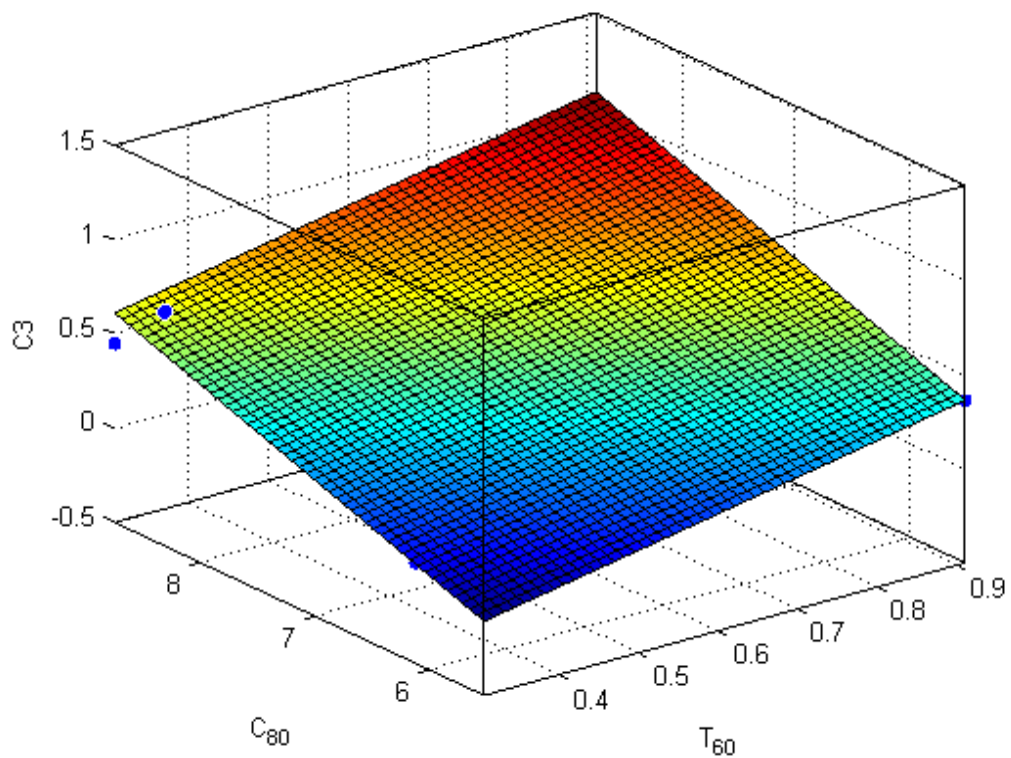
Výsledná závislost parametrů  $c_1$ ,  $c_2$  a  $c_3$  váhovací funkce na akustických parametrech  $T_{60}$  a  $C_{80}$  je zobrazena v devátém vzorci. Výsledky nalezených vzorců platí pouze pro výsledky  $c_2 \geq 0 \wedge c_3 \geq 0$ . Záporné hodnoty parametrů váhovací funkce by způsobily obrácení váhovací funkce, která by naopak potlačovala hlavní špičku relativní impulsní odezvy a skoro by nepotlačovala postranní hodnoty.

$$\begin{aligned}
c_1 &= 0,01 \\
c_2 &= 2,871 - 1,316 T_{60} - 0,2698 C_{80} \\
c_3 &= -1,581 + 0,779 T_{60} + 0,2229 C_{80}
\end{aligned}
\tag{Vzorec 9}$$





Obr. 11: Graf závislosti parametru  $c_2$  váhovací funkce na akustických parametrech  $T_{60}$  a  $C_{20}$



Obr. 12: Graf závislosti parametru  $c_3$  váhovací funkce na akustických parametrech  $T_{60}$  a  $C_{20}$

```

1  function [ c1, c2, c3 ] = idealParams( C_80, T_60 )
2      c1 = 0.01;
3      c2 = max(2.871 + (-1.316 * T_60) +
4              (-0.2698 * C_80), 0);
5      c3 = max(-1.581 + (0.779 * T_60) +
6              (0.2229 * C_80), 0);
7  end

```

*Zdrojový kód 38: Funkce pro výpočet optimálních parametrů váhovací funkce*

Pro rychlý výpočet optimálních parametrů váhovací funkce byla vytvořena funkce, která je zobrazená ve třicátém osmém zdrojovém kódu. Funkce pouze vyžaduje spočtené hodnoty akustických parametrů C80 a T60. Tato funkce vrátí spočtené ideální parametry  $c_1$ ,  $c_2$  a  $c_3$  váhovací funkce.

Na druhém řádku je do proměnné  $c_1$  uložena pouze konstantní hodnota parametru  $c_1$ . Na třetím a čtvrtém řádku je do proměnné  $c_2$  a  $c_3$  spočítána ideální hodnota z akustických parametrů  $c_{80}$  a  $T_{60}$  podle zjištěného devátého vzorce.

## 5 Ověření optimality výsledného vzorce experimenty

```
1 function [ dBrozdil ] =  
    over_spravnost_parametru(nahravka, c1, c2, c3, L )
```

*Zdrojový kód 39: Hlavička funkce pro ověření optimality spočteného parametru*

Pro ověření optimality vypočtených parametrů váhovací funkce byla vytvořena funkce `over_spravnost_parametru`, která vyžaduje celkem pět parametrů. První parametr `nahravka` obsahuje dvoukanálový záznam zkoumané místnosti. Druhý až čtvrtý parametr `c1`, `c2` a `c3` obsahují vypočtené parametry váhovací funkce pomocí získaného devátého vzorce. Poslední pátý parametr `L` vyžaduje délku relativní impulsní odezvy, pro kterou má být optimalita parametrů váhovací funkce vypočítána.

Dále následuje normalizace hlasitosti a výpočet relativní impulsní odezvy pomocí metody nejmenších čtverců podle zdrojového kódu 14. Následně je vypočítán signál, který vznikne vzájemným odečtením levého a pravého zvukového kanálu, kde levý kanál je zfiltrován spočtenou relativní impulsní odezvou podle zdrojového kódu 20. Dále je spočten rozdíl hlasitostí původního normalizovaného signálu a potlačeného signálu dle zdrojového kódu 21. Výsledná hlasitost je uložena do proměnné `LMSdecibel`.

Podle zdrojového kódu 23 a 25 je spočtena hlasitost rozdílu obou kanálů, kde je levý kanál zfiltrován relativní impulsní odezvou, která vznikne vytvořením řídké relativní impulsní odezvy pomocí váhovací funkce, jejíž parametry `c1`, `c2` a `c3` jsou získány z parametrů funkce. Tato hodnota hlasitosti je uložena do proměnné `L1decibel`.

```
2 dBrozdil = LMSdecibel - L1decibel;  
3 end
```

*Zdrojový kód 40: Výpočet rozdílu potlačení signálu relativní impulsní odezvou spočtenou pomocí metody nejmenších čtverců a řídkou relativní impulsní odezvou*

Z výsledných hlasitostí potlačených signálů pomocí metody nejmenších čtverců a vážené úlohy LASSO je do proměnné `dBrozdil` uložen hlasitostní rozdíl, o kolik dB má řídká relativní impulsní odezva horší schopnost potlačit vzájemně zvukové kanály oproti relativní impulsní odezvě pomocí metody nejmenších čtverců.

Pro ověření správnosti nalezeného vzorce pro výpočet parametrů  $c_1$ ,  $c_2$ ,  $c_3$  váhovací funkce byla použita stejná pětice nahrávek místností jako pro zjištění závislosti na akustických parametrech. Navíc byly nahrány dvě nahrávky šest a sedm v jiných prostředích s nahrávkou mužského hlasu. Šestá místnost má rozměry  $3 \times 1,5$  m a sedmá místnost má rozměry  $1,3 \times 1,8$  m. Obě mají výšku stropu 2,7 m.

Délky relativních impulsních odezev u místností jedna až pět byly zvoleny shodně, jako v podkapitole 4.2. Délky relativních impulsních odezev šest a sedm byly zvoleny tak, aby odečtení zfiltrovaného levého kanálu relativní impulsní odezvou spočtenou pomocí metody nejmenších čtverců od pravého kanálu vznikl signál se stejnou hlasitostí, jako u ostatních místností. Podrobněji je tato metoda volby délky relativní impulsní odezvy popsána v kapitole 4.2. Výsledné délky relativní impulsní odezvy místností šest a sedm jsou zobrazeny v páté tabulce.

*Tabulka 5: Volba délky relativní impulsní odezvy testovaných místností*

<b>Místnost</b>	<b>Míra potlačení [dB]</b>	<b>Výsledná délka relativní impulsní odezvy</b>
6	8,01	84
7	7,94	86

Pomocí funkce `LMS_pokles` byly vybrány délky relativních impulsních odezev tak, aby hlasitost signálu po odečtení pravého kanálu od zfiltrovaného levého kanálu relativní impulsní odezvou spočtenou pomocí metody nejmenších čtverců byla u všech místností stejná. Pomocí funkce `T_60` byla u každé místnosti spočtena délka dozvuku a pomocí funkce `C80` byl spočítán akustický parametr C80. Na základě spočtené doby dozvuku a parametru C80 byly spočteny pomocí funkce `idealParams` optimální parametry  $c_1$ ,  $c_2$  a  $c_3$  váhovací funkce pro výpočet řídké relativní impulsní odezvy.

Následně byla spočtena hlasitost, která vznikne rozdílem výsledných hlasitostí odečtením pravého zvukového kanálu od levého, který byl zfiltrován relativní impulsní odezvou spočtenou pomocí metody nejmenších čtverců a řídkou relativní impulsní odezvou.

Tato hlasitost by měla v ideálním případě mít hodnotu 1 dB. Z vypočtených hodnot v šesté tabulce je největší hlasitostní rozdíl 1,7175 dB u první místnosti. Nejnižší hlasitostní rozdíl 0,4599 dB má šestá místnost. Odchylka naměřená pomocí experimentu činí +0,7175 dB a -0,5401 dB.

*Tabulka 6: Test optimality pomocí rozdílů hlasitostí*

Místnost	Délka relativní impulsní odezvy	Akustické parametry		Optimální parametry váhovací funkce			Rozdíl hlasitostí [dB]
		T60 [s]	C80 [dB]	$c_1$	$c_2$	$c_3$	
1	720	0,3069	8,7659	0,01	0,1021	0,6120	1,7175
2	1115	0,3148	6,1935	0,01	0,7857	0,0448	1,0012
3	2152	0,3793	5,7826	0,01	0,8117	0,0034	1,0762
4	402	0,3133	8,3736	0,01	0,1995	0,5295	1,1092
5	3000	0,9097	5,5416	0,01	0,1787	0,3629	1,4034
6	84	0,2396	6,9126	0,01	0,6906	0,1465	0,4599
7	86	0,1938	7,5756	0,01	0,5721	0,2585	0,6372

## 6 Závěr

V první části práce bylo seznámeno s výpočtem relativní impulsní odezvy pomocí metody nejmenších čtverců. V následující podkapitole byla vybrána L1 homotopní metoda pro výpočet řídké relativní impulsní odezvy jelikož jako jediná z metod, které řeší váženou úlohu LASSO pomocí diagonální váhovací matice.

V druhé teoretické části bylo seznámeno s výpočtem akustických parametrů pro zjištění doby dozvuku T60 a EDT ze zvukového záznamu. Dále bylo seznámeno s akustickými parametry C80, D20 a D50, které určují míru srozumitelnosti lidské řeči v závislosti na době dozvuku místnosti.

V následující praktické části bylo pořízeno pět nahrávek se záznamem ženského hlasu pomocí dvou mikrofonů v místnostech s různými hodnotami akustických parametrů. Zejména hodnoty T60. Dále byla spočtena hrubou silou závislost parametrů váhovací funkce na počtu nenulových prvků řídké relativní impulsní odezvy spočtené váženou úlohou LASSO. Dále byly ze spočítaných průběhů pro každou nahrávku nalezeny optimální parametry váhovací funkce, pro které byla řídká relativní impulsní odezva nejřidší, ale zároveň se její schopnost potlačit pravý zvukový signál zfiltrovaným levým signálem nezhoršila o více než 1 dB, než má relativní impulsní odezva spočtená pomocí metody nejmenších čtverců. Nakonec byla nalezena závislost mezi optimálními parametry váhovací funkce a akustickými parametry T60 a C80. Z této nalezené závislosti byl vytvořen vzorec, který umožňuje rovnou nalézt optimální parametry váhovací funkce, aniž by musely být hledány hrubou silou na určitém rozsahu.

V poslední kapitole je provedeno ověření optimality spočtených parametrů váhovací funkce pomocí získaného vzorce na již získaných zvukových záznamech a dvou nových záznamech s nahrávkou mužského hlasu. Výsledná odchylka od optimálních hodnot je minimální a nad očekávání velmi uspokojivá.

## Seznam použité literatury

- [1] KOLDOVSKÝ, Zbyněk, Jiří MÁLEK a Sharon GANNOT. 2015. Spatial Source Subtraction Based on Incomplete Measurements of Relative Transfer Function. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. : 1-1. DOI: 10.1109/taslp.2015.2425213.
- [2] MÁLEK, Jiří a Zbyněk KOLDOVSKÝ. 2014. Sparse target cancellation filters with application to semi-blind noise extraction. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. DOI: 10.1109/icassp.2014.6853971.
- [3] WRIGHT, S.J., R.D. NOWAK a M.A.T. FIGUEIREDO. 2009. Sparse Reconstruction by Separable Approximation. *IEEE Transactions on Signal Processing*. **57**(7): 2479-2493. DOI: 10.1109/TSP.2009.2016892. ISSN 1053-587x. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4799134>
- [4] L1 Homotopy: A MATLAB Toolbox for Homotopy Algorithms in L1 Norm Minimization Problems. 2013. *School of Electrical and Computer Engineering at the Georgia Institute of Technology* [online]. [cit. 2015-05-05]. Dostupné z: <http://users.ece.gatech.edu/~sasif/homotopy/>
- [5] PEER, Itai, Boaz RAFAELY a Yaniv ZIGEL. 2008. Room Acoustics Parameters Affecting Speaker Recognition Degradation Under Reverberation. *2008 Hands-Free Speech Communication and Microphone Arrays*. DOI: 10.3403/30081124.

## **A Obsah příloženého CD**

- text bakalářské práce
  - bakalarska\_prace\_2015\_Tomas\_Franek.pdf
  - bakalarska\_prace\_2015\_Tomas\_franek.odt
  - kopie\_zadani\_bakalarske\_prace\_2015\_Tomas\_franek.pdf
- zdrojové kódy programu
  - funkce
  - L1\_homotopy\_v2.0
- použité zvukové záznamy
  - záznamy pro T60
  - záznamy pro zpracování
  - zdrojové záznamy s hlasy
- zpracovaná data