



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

Fakulta elektrotechnická  
Katedra radioelektroniky

**Studie možnosti potlačení akustické zpětné vazby**

**Study of Acoustic Feedback Suppression Possibility**

Diplomová práce

Studijní program: Elektronika a komunikace  
Studijní obor: Audiovizuální technika a zpracování signálů

Vedoucí práce: prof. Ing. Pavel Sovka, CSc.  
Druhý vedoucí práce: prof. Ing. Pavel Zahradník, CSc.

**Dominik Šmíd**

---

**Praha 2018**



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šmíd** Jméno: **Dominik** Osobní číslo: **420074**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra radioelektroniky**  
Studijní program: **Elektronika a komunikace**  
Studijní obor: **Audiovizuální technika a zpracování signálů**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Studie možnosti potlačení akustické zpětné vazby**

Název diplomové práce anglicky:

**Study of Acoustic Feedback Suppression Possibilities**

Pokyny pro vypracování:

1. Prostudujte a shrňte možnosti potlačení akustické zpětné vazby při hudební produkci.
2. Vyberte vhodný algoritmus algoritmus pro automatické potlačování akustické zpětné vazby.
3. Vytvořte v MATLABu systém pro simulaci vzniku tohoto nežádoucího jevu a pomocí něj algoritmus otestujte.
4. Ověřte schopnost algoritmu pracovat v reálném čase na číslicovém signálovém procesoru a reálné akustické soustavě.

Seznam doporučené literatury:

- [1] S. V. Vaseghi: Advanced Signal Processing and Digital Noise Reduction. Wiley Teubner, 1997
- [2] T. van Waterschoot, M. Moonen: 'Fifty years of acoustic feedback control: State of the art and future challenges', Proc. IEEE, vol. 99, no. 2, pp. 288-327, Feb. 2011
- [3] M. C. Green, J. Szymanski, M. Speed: 'Assessing the suitability of the magnitude slope deviation detection criterion for use in automatic acoustic feedback control', Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16), pp. 85-92, Sept. 2016

Jméno a pracoviště vedoucí(ho) diplomové práce:

**prof. Ing. Pavel Sovka, CSc., katedra teorie obvodů FEL**

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

**prof. Ing. Pavel Zahradník, CSc., katedra telekomunikační techniky FEL**

Datum zadání diplomové práce: **09.02.2018** Termín odevzdání diplomové práce: **25.05.2018**

Platnost zadání diplomové práce: **30.09.2019**

\_\_\_\_\_  
prof. Ing. Pavel Sovka, CSc.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Pavel Ripka, CSc.  
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





## **Bibliografický záznam**

ŠMÍD, Dominik. *Studie možnosti potlačení akustické zpětné vazby*. Praha, 2018. Diplomová práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra radioelektroniky. Vedoucí práce prof. Ing. Pavel Sovka, CSc. a prof. Ing. Pavel Zahradník, CSc.

## **Anotace**

Diplomová práce se zabývá automatickým potlačováním akustické zpětné vazby v ozvučovací technice. Věnuje se vývoji algoritmu schopného rozpoznat tento nežádoucí jev a automaticky ho omezit pomocí nastavitelných úzkopásmových filtrů vložených do signálové smyčky. Algoritmus je vyvíjen pomocí programu MATLAB, kde je vytvořena simulace s virtuálním prostředím elektroakustické zpětnovazební smyčky. Algoritmus je následně přepsán do jazyka C a dále je odladěn i testován v reálném čase a v reálných podmínkách pomocí vývojového přípravku s číslicovým signálovým procesorem. V textu jsou shrnuty dostupné teoretické informace k tématu, popsán navržený algoritmus i průběh vývoje a jsou provedena měření zkreslení i zhodnocení výsledků a kvality vytvořeného automatického omezovače akustické zpětné vazby.

## **Abstract**

This diploma thesis deals with automatic suppression of the acoustic feedback in sound system. It is concerned with the development of an algorithm for automatic suppression of this undesirable phenomenon using parametric narrow band notch filters inserted into the forward signal path of the feedback system. The algorithm is developed in the MATLAB software, where a simulation of electroacoustic feedback system is created. The algorithm is subsequently rewritten to C language and then challenged in real-time conditions using the OMAP-L138 development kit. The thesis summarizes the available theoretical information connected to the subject, describes the developed algorithm and its testing, presents the outcomes of the distortion measurements and finally evaluates the results and quality of the created automatic acoustic feedback suppressor.

## **Klíčová slova**

Akustická zpětná vazba, potlačení nežádoucího jevu, automatický omezovač zpětné vazby, ozvučení, reproduktor, mikrofon, zvukový signál, ekvalizace, adaptivní algoritmus, IIR notch filtr, OLS, rychlá Fourierova transformace, spektrum, C, MATLAB, číslicový signálový procesor, stabilita zpětnovazebního systému, MSG

## **Keywords**

Acoustic feedback, suppression of the undesirable phenomenon, automatic feedback suppressor, sound system, loudspeaker, microphone, audio signal, equalization, adaptive algorithm, IIR notch filter, OLS, Fast Fourier Transform, spectrum, C, MATLAB, Digital Signal Processor, feedback system stability, MSG



## **Prohlášení**

*„Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.“*

V Praze dne

.....



## **Poděkování**

Děkuji prof. Ing. Pavlu Sovkovi, CSc., vedoucímu práce, za cenné rady a připomínky při tvorbě diplomové práce a prof. Ing. Pavlu Zahradníkovi, CSc., druhému vedoucímu práce, za zapůjčení potřebného hardwaru a pomoci s implementací algoritmu na signálovém procesoru.



## Obsah

Úvod .....	13
1 Popis problematiky.....	15
1.1 Základní bloky zkoumané akustické soustavy.....	15
1.2 Nežádoucí jev akustické zpětné vazby .....	17
1.3 Identifikace jevu a rizika chybné identifikace .....	18
1.4 Známé metody omezování akustické zpětné vazby.....	19
1.4.1 Manuální možnosti předcházení zpětné vazbě.....	19
1.4.2 Frequency-Shifting a fázová modulace.....	20
1.4.3 Adaptivní filtrace .....	21
1.4.4 Auto-Notching.....	22
2 Navržený algoritmus automatického omezovače akustické zpětné vazby .....	23
2.1 Princip algoritmu .....	23
2.2 Ekvalizace pomocí kaskády přeladitelných IIR filtrů.....	27
2.3 Alternativní verze ekvalizace s filtrací ve spektru OLS .....	27
2.4 Použité matematické nástroje .....	28
2.4.1 FFT pro sledování časového průběhu amplitudy všech frekvencí.....	28
2.4.2 Kvocient geometrické posloupnosti .....	28
2.4.3 Relativní průměrná absolutní odchylka .....	28
2.4.4 Výpočet koeficientů IIR notch filtru 2. řádu.....	29
2.5 Simulace – implementace v MATLABu .....	29
2.5.1 Nastavitelné parametry algoritmu a parametry simulace.....	32
2.6 Testování v reálném čase – implementace v C na DSP.....	34
2.6.1 Optimalizace v C kódu algoritmu – využití prostředků DSP.....	37
2.6.2 Popis realizovaného systému .....	38
2.6.3 Aplikace a její uživatelské rozhraní .....	39
3 Zhodnocení a výsledky testování algoritmu .....	41
3.1 Kritéria pro zhodnocení kvality omezovače .....	41
3.2 Uspořádání simulace a dat pro testování .....	42
3.3 Výstupní data po skončení simulace.....	43
3.4 Výsledky testování a dosažené $\Delta$ MSG.....	47
3.5 Porovnání dvou verzí ekvalizace v algoritmu omezovače.....	49
3.6 Zkušenosti z testování v reálných podmínkách .....	51
Závěr.....	53
Zkratky a termíny .....	54

Použité zdroje .....	55
----------------------	----



## Úvod

Nejdříve bych rád uvedl motivaci a proč je tato práce věnována tématu potlačování akustické zpětné vazby se zaměřením na automatické metody omezování tohoto nežádoucího jevu v reálných situacích. Dělat zvukaře je moje volnočasová záliba, příležitostně ozvučuji koncerty přátel nebo moderátory programu na různých akcích. Jde o zajímavé propojení hudebního umění a elektroniky. Prostředí, do kterého se dostávám, je běžně pro ozvučení nepříznivé a aparatura je posbíraná z převážně starších zařízení. Řešení problémů s akustickou zpětnou vazbou je v takových podmínkách běžné a unavující. Na trhu existuje několik komerčních zařízení, které údajně dokážou pomoci při problémech s akustickou zpětnou vazbou i když ji nepotlačí zcela. Podle návodu na ovládání těchto zařízení lze předpokládat, že používají metodu automatické ekvalizace pomocí několika notch filtrů. Nikde ale nelze najít ani jedinou informaci o tom, jak funguje automatická detekce zpětné vazby v algoritmu těchto zařízení. Nikde nelze získat detaily potřebné k implementaci funkčního omezovače (natož najít zdrojový kód) k jednoduchým nebo složitým metodám popisovaným v mnoha vědeckých článcích o automatickém potlačování akustické zpětné vazby. Počáteční motivací pro zvolení tohoto tématu práce bylo vyzkoušet, jak složité je navrhnout funkční algoritmus, který by dokázal automaticky omezovat akustickou zpětnou vazbu, a co obnáší vytvořit a odladit zařízení, které by bylo použitelné v praxi.

V této práci jsou shrnuty dostupné informace o automatickém potlačování akustické zpětné vazby a zjištěné údaje byly následně použity jako výchozí bod při vývoji vlastního algoritmu. Hlavním cílem této práce je návrh algoritmu omezování akustické zpětné vazby (dále jen „omezovač“) a jeho praktická realizace pomocí vývojového přípravku s číslicovým signálovým procesorem (dále jen „DSP“). Dalším cílem je testování omezovače ve virtuálním prostředí simulace v MATLABu i testování omezovače v reálném čase, v reálném prostředí a reálné elektroakustické aparatuře (dále jen „ozvučení“). V první fázi je algoritmus omezovače vyvíjen v MATLABu a testován ve virtuálním prostředí simulace pomocí MATLABu. V druhé fázi je algoritmus omezovače implementován do DSP a je vytvořena aplikace zajišťující uživatelské rozhraní a fungování algoritmu jako zařízení připojené do ozvučení.

První kapitola se zabývá teorií okolo problematiky akustické zpětné vazby a shrnuje metody potlačování nežádoucího jevu akustické zpětné vazby.

Druhá kapitola popisuje navržený algoritmus omezování akustické zpětné vazby s pomocí blokového schématu. Podrobně jsou popsány jednotlivé bloky, ze kterých se algoritmus skládá, i jakým způsobem jsou naimplementovány v MATLABu nebo na DSP. V kapitole je popsán i způsob, jakým je simulována virtuální elektroakustická zpětnovazební smyčka. Rozebrány jsou všechny parametry, kterými lze nastavit algoritmus, a jakým způsobem ovlivňují jeho činnost nebo chování. Rozebrány jsou i parametry simulace a použité testovací signály a data.

V druhé části kapitoly jsou fotky s popisem realizované aparatury a prostředí, na kterém probíhá testování algoritmu v reálném čase a v reálných podmínkách. Je zde ukázka s popisem uživatelského rozhraní omezovače a dálkové ovládání přes chytrý telefon pro nastavování omezovače za běhu. Lze tím efektivně testovat omezovač a zkoumat vliv změny parametrů na chování omezovače, aniž by bylo třeba zastavit běh DSP a tím porušit podmínky běhu v reálném čase. Dále kapitola rozebírá optimalizace kódu pro DSP nutné pro běh algoritmu v reálném čase a využití prostředků DSP.

Třetí kapitola se snaží zhodnotit navržený algoritmus a dosažené výsledky během testování algoritmu, jak v simulacích, tak v reálném čase s DSP. Jsou zde uvedeny demonstrační výsledky

simulace pro konkrétní hodnoty parametrů a ukázka spektrogramu při vypnutém omezovači a spektrogramu při zapnutém omezovači. Jsou zde rozebrána kritéria pro hodnocení zkreslení vytvářeného omezovačem.

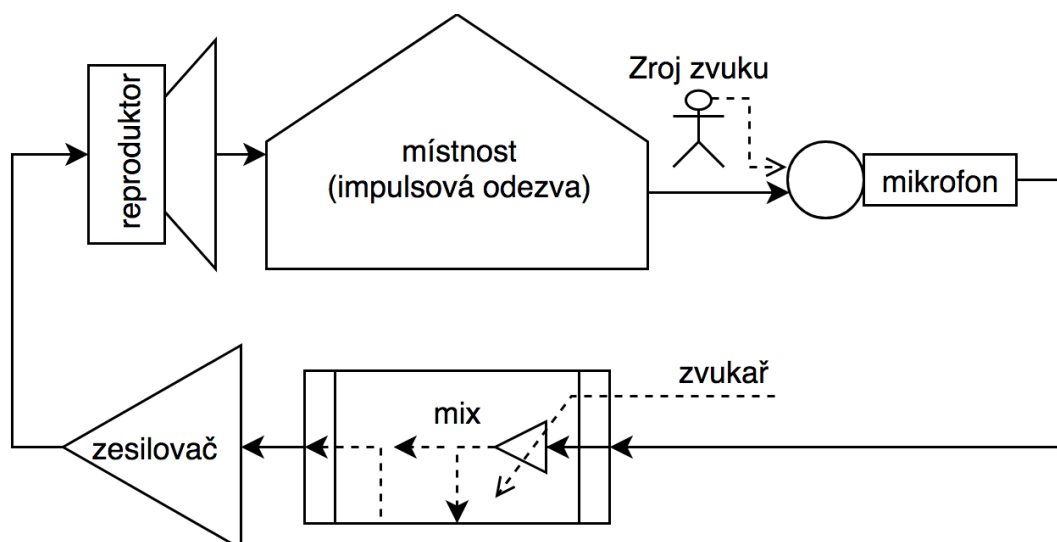
V závěru je posouzeno splnění úkolů ze zadání práce a zamyšlení nad tím, jak by bylo možné na práci dále navázat či vylepšit algoritmus omezovače.

# 1 Popis problematiky

Určitě si mnoho lidí vzpomene na nepříjemný, někdy až uši trhající zvuk přicházející z reproduktoru na koncertě, plese, na přednášce či v jiné chvíli, kdy se používají mikrofony i reproduktory zároveň. V tom momentu většina lidí bez dlouhého přemýšlení očima hledá, kde je zvukař, který ten nepříjemný zvuk způsobil. Do té doby vlastně dobrou práci zvukaře nikdo nevnímal a najednou, když je něco špatně, odnáší to zvukař. Ve skutečnosti ale za to často může někdo jiný – ten, kdo v ruce drží mikrofon a chodí s ním tam, kam nemá, nebo ho přímo odloží kousek pod reproduktor. Navíc zvukař musí v takových chvílích zajistit dva často protikladné požadavky, musí mikrofon stále udržet na zesílení dostačujícím ke srozumitelnosti a dostatečné hlasitosti reprodukce a zároveň mikrofon nezesílit příliš, aby nedošlo k překročení velmi tenké hranice, kdy se spustí ono nepříjemné pískání či hučení.

## 1.1 Základní bloky zkoumané akustické soustavy

Nejdříve je třeba popsat blokově elektroakustickou soustavu se zpětnou vazbou. Blokový popis vychází z běžně v praxi využívané sestavy ozvučení (aparatury). Pro jednoduchost stačí soustavu zobecnit na jeden mikrofon a jeden reproduktor. Pokud by v reálné soustavě bylo měničů více, lze je sloučit do jednoho měniče. S modelem soustavy na obr. 1 bude pracovat celá tato práce.



Obr. 1: Základní elektroakustická soustava se zpětnou vazbou - ozvučení

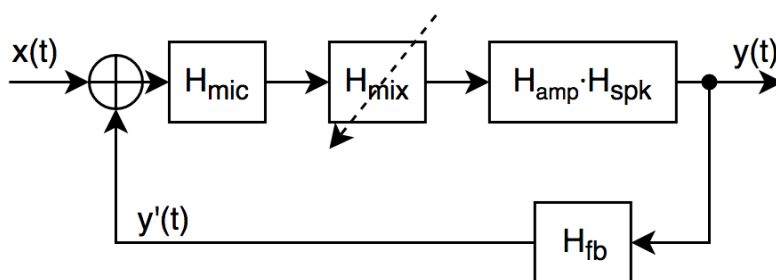
Schéma soustavy ozvučení dle obr. 1 je na obr. 2. Vstupem do soustavy je mikrofon. Mikrofon snímá zdroj akustického signálu  $x(t)$  (zpěv, mluvené slovo, některé snímané nástroje nebo ticho). Pro zjednodušení lze zdroj signálu umístit velmi blízko mikrofonu a odrazy v místnosti zanedbat, protože pro účely této práce nehrají roli. Mnohem důležitější roli hraje prostředí mezi reproduktorem a mikrofonem. Mikrofon přeměňuje akustický signál na elektrický s přenosem  $H_{mic}(\omega)$ . Dále signál pokračuje do mixážního pultu, kde lze nastavit zesílení a ekvalizaci  $H_{mix}(\omega)$ , případně propojení s dalšími vstupy či výstupy mixážního pultu (mixu). Tento blok je proměnlivý a zvukař v průběhu může měnit velikost zesílení a ekvalizaci, přesto tento blok lze považovat za LTI, protože změny jsou pomalé nebo málo časté. Z mixážního pultu vede signál do zesilovače s většinou neměnným zesílením  $H_{amp}(\omega)$ , který zesílí zvukový signál na potřebný výkon pro reproduktor. V reproduktoru se mění signál z elektrického na akustický s přenosem  $H_{spk}(\omega)$  a ten je pak vyslán do místnosti. Výstupní signál soustavy vyzářený reproduktorem je označen  $y(t)$ . Prostředí mezi reproduktorem a mikrofonem zavádí do soustavy akustickou zpětnou vazbu a lze ho charakterizovat impulsovou odezvou  $h_{rir}(t)$ . Blok

prostředí (místnost) je proměnný a jeho přenos  $H_{fb}(\omega)$  závisí na vzájemné poloze mikrofon-reproduktor i akustice místnosti. Pro účely potlačování akustické zpětné vazby postačí na prostředí nahlížet jako na LTI nebo alespoň v diskrétních okamžicích z pohledu algoritmu potlačení akustické zpětné vazby je prostředí LTI. Reproduktor má většinou pevnou polohu, zato mikrofon se často pohybuje a zásadně tím mění přenos (impulsovou odezvu). Z místnosti akustický signál dopadá zpět na mikrofon a zde se jako upravená a zpožděná kopie  $y'(t)$  přidává k signálu zdroje  $x(t)$ .

Soustavu bez zpětné vazby (bloku místnost) lze matematicky popsat rovnicí, kde se přenosy jednotlivých bloků násobí. Přenos této soustavy bez zpětné vazby je:

$$\hat{H}(\omega) = \hat{H}_{mic}(\omega) \cdot \hat{H}_{mix}(\omega) \cdot \hat{H}_{amp}(\omega) \cdot \hat{H}_{spk}(\omega) \quad (1)$$

Schéma soustavy po zavedení zpětné vazby je na obr. 2.



Obr. 2: Schéma soustavy se zpětnou vazbou modelující sestavu z obr. 1:  $x(t)$  je zdroj zvuku,  $y(t)$  je výstup reproduktoru a  $y'(t)$  je zpětnovazební signál

Celkový přenos této soustavy s rozpojenou zpětnovazební smyčkou  $\hat{A}$  je:

$$\hat{A}(\omega) = \frac{FT\{y'(t)\}}{FT\{x(t)\}} = \hat{H}(\omega) \cdot \hat{H}_{fb}(\omega) = |\hat{A}(\omega)| \cdot e^{j\varphi(\omega)} \quad (2)$$

$$A(\omega) = |\hat{A}(\omega)| \quad (3)$$

kde  $A$  je absolutní hodnota (amplituda přenosu)  $\hat{A}$  a  $\varphi$  je úhel (fázový posun přenosu)  $\hat{A}$ .

Přenos soustavy s uzavřenou zpětnovazební smyčkou lze vyjádřit takto [2] [1]:

$$\hat{A}'(\omega) = \frac{\hat{H}(\omega)}{1 - \hat{H}(\omega)\hat{H}_{fb}(\omega)} = \frac{\hat{H}(\omega)}{1 - \hat{A}(\omega)} \quad (4)$$

Mnohačetné odrazy v místnosti (dále jen dozvuk) se vždy projeví při snímání či poslechu zdroje zvuku v této místnosti. Na zvuky bez žádných odrazů člověk není zvyklý a čistý zvuk bez dozvuku by působil nepřírozně. Ozvučení v místnosti vytváří další jevy, které mohou znít jako prodloužený dozvuk. Pokud elektroakustická soustava bude mít nastavené velké zesílení (přenos  $\hat{H}$ ), zpětnovazební přenos v soustavě  $\hat{H}_{fb}$  nebude dostatečně malý a jmenovatel v podílu se začne blížit nule, vznikne slyšitelné nadkriticky tlumené kmitání v systému. Tlumené kmitání na jedné či více frekvencích je čím dál více slyšet a při překročení určité meze se stane systém nestabilním a kmitání již není tlumeno vůbec. Tomuto riziku a jevu s tím spojenému je věnována následující kapitola.

Pro každou zpětnovazební soustavu existuje mez stability [2] a pro správnou funkci soustavy je třeba splnit podmínku stability:

$$\text{pro všechny } \omega, \text{ kde } \varphi = f(\omega) = 2k\pi; k \in \mathbb{Z} \quad (5)$$

$$\text{musí být } A < 1 \rightarrow |\hat{H} \cdot \hat{H}_{fb}| < 1 \quad (6)$$

Stačí tedy, aby modul  $H \cdot H_{fb}$  v žádné frekvenci nepřesáhl 1, a systém bude s jistotou stabilní.

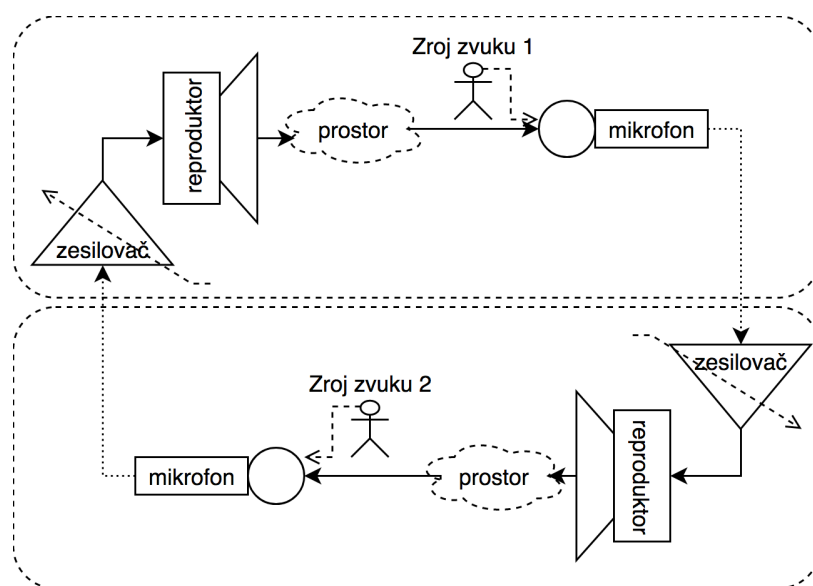
## 1.2 Nežádoucí jev akustické zpětné vazby

Když se vrátíme k myšlence textu na začátku první kapitoly, kde je popisováno pískání či hučení, které se může objevit během koncertů či reprodukci mluveného slova – odkud pochází tento nepříjemný zvuk? Jak to, že reproduktory produkují zvuk, který nemá nic společného se zdroji zvuku v prostoru snímaném mikrofonem, a jak to, že může vznikat i v čase, kdy nikdo do mikrofonu nemluví ani nehraje?

Jak název jevu prozrazuje, způsobuje to právě akustická zpětná vazba zavedená v elektroakustické soustavě popsané v předchozí kapitole. Pokud celkové zesílení nastavené v ozvučení i s přenosem akustické části, přesněji zpětné vazby zavedené přenosem při šíření zvuku místností, přesáhne kritickou mez ( $A=1$ ), soustava se stává nestabilní na frekvencích, kde  $A>1$  a fázový posun  $\Delta\varphi$  je roven  $k \cdot 2\pi$ . Zjednodušeně lze říci, že pokud reproduktorem v mikrofonu vybudíme signál větší, než vybudil původní zdroj, vyslaný signál se pak zesiluje ve smyčce víc a víc a tomu je třeba se vždy vyhnout při práci s takovou soustavou. Záleží na výsledném nastavení nejen zesilovače a mixážního pultu, ale i na impulsové odezvě místnosti a hlavně na jejích rezonančních frekvencích, na kterých nejdříve začne soustava „vazbit“. Bylo by vhodnější tento jev nazvat přebuzení akustickou zpětnou vazbou, v praxi se však ujalo pro jev označení akustická zpětná vazba. V některých pramenech lze najít i název Larsenův efekt.

Amplituda signálu na frekvencích, na kterých dojde k překročení meze stability podle podmínky (5) a (6), roste nade všechny meze a záleží zda dříve utečou všichni posluchači v místnosti nebo se zničí některý prvek soustavy kvůli přetížení. Také lze v takové chvíli pozorovat, jak při určité vysoké intenzitě signálu začne nejhorší prvek soustavy nelineárně zkreslovat a vytvářet vyšší harmonické frekvence k doposud čisté základní frekvenci.

Riziko tohoto nežádoucího jevu je kdekoliv, kde je pomocí mikrofonu snímán signál dále zpracován, zesilován a reprodukován reproduktory umístěnými ve stejném prostoru s tímto mikrofonem. To odpovídá soustavě popsané v předchozí kapitole, například všechny živé koncerty, přednášky, proslovy, moderování různých akcí a příležitostí. Zároveň velmi podobnou soustavu lze nalézt při telefonování a hovorech přes internet. Jsou to v podstatě dvě takové elektroakustické soustavy zapojené do smyčky, znázorněné na obr. 3, a lze je ošetřit proti zpětné vazbě spíše algoritmy AEC (Acoustic Echo Cancellation) [15], které využívají výhod nekorelovaných signálů různých dvou zdrojů na protějších stranách. Algoritmy AEC také využívají výhod menších vzdáleností a neměnné pozice mikrofonu a reproduktoru vůči sobě, proto by na soustavě popisované v této práci nefungovaly dost dobře.



Obr. 3: Blokové schéma dvou soustav zapojených do smyčky

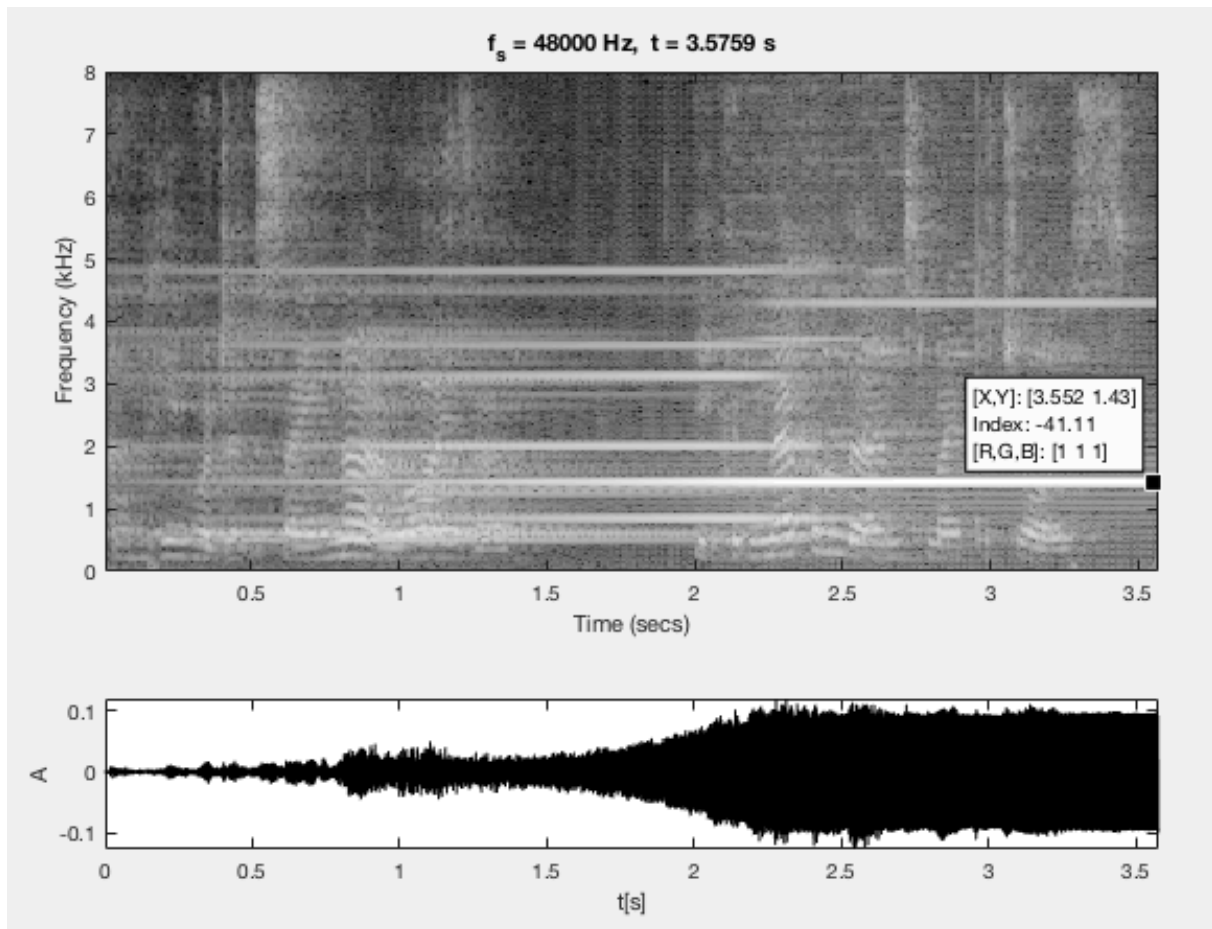
### 1.3 Identifikace jevu a rizika chybné identifikace

Hvízdání zpětné vazby by měl rozpoznat každý zvukař, ale někdy se vyskytnou situace, kdy si ani zkušený člověk není stoprocentně jistý, zda vysoký tón zahráný na elektrickou kytaru není náhodou nabíhající zpětná vazba, dokud její intenzita jasně nepřevyší ostatní užitečné zvukové signály. Proto může být složité vytvořit automatický algoritmus schopný takovou zpětnou vazbu vždy správně rozlišit. Nejlepší by bylo, kdyby byl automatický omezovač zpětné vazby schopný rozpoznat a zareagovat dřív, než to většina posluchačů vůbec postřehne. Náročnost identifikace nežádoucí zpětné vazby je velmi závislá na tom, jaký zvuk je zrovna snímán a reprodukován. Nejtěžší je včas identifikovat zpětnou vazbu v populární, rockové nebo metalové hudbě, kde často elektrická kytara vytváří zvuky velmi podobné zpětné vazbě. Na druhou stranu zesílení aparátu na koncertech takové hudby je často nastaveno na vysokou intenzitu a pokud ke zpětné vazbě dojde, intenzita nevystoupá příliš vysoko nad užitečný signál díky omezeným schopnostem reproduktorů. Naopak velmi rušivá může být zpětná vazba u klidné hudby, vážné hudby nebo mluveného slova. Tam pokud nabíhání zpětné vazby není včas zastaveno, může znamenat natolik rušivý zážitek, že to posluchačům zkazí dojem z celého koncertu. Reproduktory často dokáží vytvořit nežádoucí pískání o mnohem větší intenzitě, než má užitečný signál. Výhodou u tohoto typu hudby je, že nežádoucí zpětná vazba lze snáze rozpoznat mezi užitečným signálem. Na koncertech klidnější hudby se může objevit například klavír nebo smyčcové nástroje, které jsou trochu podobné zvuku zpětné vazby, není zde ale tolik podobnosti jako u rockové elektrické kytary.

Pokud se ozvučení skládá z více mikrofónů a nástrojů s vlastním snímačem, má automatické potlačování zpětné vazby oproti manuálnímu potlačování zvukařem výhodu v možnosti sledovat pouze některé kanály. Zvuky hudebních nástrojů, u nichž je jev zpětné vazby velmi nepravděpodobný, nemusí automatický algoritmus zbytečně rušit. Zvukař musí hlídat všechny kanály jako celek a správná identifikace nežádoucího pískání je těžší a reakce na identifikovaný jev je většinou pomalá. Automatická metoda potlačování se může soustředit pouze na některé rizikové mikrofony a eliminuje tak riziko chybné identifikace jevu zpětné vazby kvůli podobnosti jevu se zvuky některých hudebních nástrojů. Mikrofony, které mají zpěváci, sice snímají i okolní hudební nástroje, ale pouze slabě.

Na obr. 4 je ukázka reálné situace s jevem akustické zpětné vazby, ke kterému došlo během mluvení do mikrofónu. Dlouhé světlé čáry nejsou součástí řečového signálu, je to pískání nabíhající

zpětné vazby, které ve druhé sekundě vzroste natolik, že reproduktory začnou nelineárně signál zkreslovat a nejsilnější frekvence zpětné vazby saturuje ve smyčce, ostatní nabíhající frekvence ustanou. Od druhé sekundy zůstává pískání na  $f_0 = 1,43$  kHz a některé vyšší harmonické frekvence vzniklé limitujícím nelineárním zkreslením (např.  $4,29$  kHz =  $3 \cdot f_0$ ).



Obr. 4: Spektrogram a průběh řečového signálu - nahrávka reálné situace s jevem zpětné vazby

## 1.4 Známé metody omezování akustické zpětné vazby

Přestože se tato práce zabývá výhradně automatickým omezováním akustické zpětné vazby, a to hlavně jednou z těchto automatických metod, jsou zde krátce shrnuty možnosti manuálního omezování a tři základní metody automatického omezování akustické zpětné vazby.

### 1.4.1 Manuální možnosti předcházení zpětné vazbě

V této oblasti dlouho známých postupů, jak nazvučit prostor, aby byla akustická zpětná vazba co nejmenší, už není příliš prostoru ke zdokonalování. Tyto metody se používají od začátku doby, kdy se technika ozvučení používá.

Základ všeho je v rozložení mikrofonu a reproduktoru v prostoru a jejich nasměrování. Je třeba mikrofon a reproduktor dát co nejdále od sebe a co nevíce využít směrovosti. U mikrofonů existuje různorodý výběr směrových charakteristik, u reproduktorů už taková pestrost není, většinou mají maximum vyzařovací charakteristiky v přímém směru, z boku se může relativně lišit a zezadu je minimum. Všesměrové mikrofony se ve scénářích odpovídajících soustavě popisované v této práci příliš nepoužívají, pokud k tomu není dobrý důvod. Je naopak vhodnější použít mikrofon co nejvíce

směrový a maximum charakteristiky nasměrovat pryč od jakéhokoliv reproduktoru. Pokud mikrofon není pevně na stojanu a zpěvák či mluvčí se s ním pohybuje v prostoru, je třeba zvolit spíše nějaký kompromis směrovosti. U příliš směrového mikrofonu hrozí, že člověk bude měnit úhel, pod kterým do mikrofonu mluví, a výsledkem budou časté větší změny hlasitosti. Pro snímání hudebních nástrojů nebo většího počtu lidí se také nedá použít příliš směrový mikrofon. Je třeba také vzít v úvahu, že směrová charakteristika se může s frekvencí trochu měnit a na okrajích slyšitelného pásma i více. Pracovat lze i s minimem směrové charakteristiky mikrofonů. Některé mikrofony mají zadní minimum téměř nulové, pokud se podaří do tohoto směru postavit reproduktor, přímá vazba může být opravdu nulová. Bohužel se většinou nedá vyhnout vazbě pomocí odrazů od stěn a jiných ploch v prostoru. Snížit vazbu v přímém směru a největší potencionální odraz často postačí nebo je dobré se vyvarovat tomu, aby hlavní lalok směrové charakteristiky mikrofonu mířil kolmo na velké odrazivé plochy. Celý systém s více reproduktory nebo mikrofony začne být nestabilní a pískat, pokud jediná ze všech možných kombinací vazeb reproduktor-mikrofon překročí mez. Správné rozložení elektroakustických měničů v systému a jejich nasměrování lze využít k předcházení zpětné vazbě, připravit vše předem. V průběhu koncertu pak už často není možnost do rozložení zasahovat.

Další možností, jak nežádoucí akustickou vazbu v systému zmenšit v průběhu i předem, je manuální ekvalizace. Před začátkem lze vyzkoušet, na kterých frekvencích hotová soustava nejdříve začne pískat při překročení stability a tyto frekvence vhodným nastavením ekvalizéru potlačit. Sofistikovanější možnost je předem změřit impulsovou odezvu místnosti (např. metodou MLS nebo sine-sweep [5]) a vypočítanou přenosovou charakteristiku vhodným nastavením ekvalizéru vyrovnat nebo alespoň omezit maxima. V průběhu produkce už není tak jednoduché ekvalizér nastavovat, protože je třeba se vyhnout pískání, které by jasně ukázalo, na které frekvenci ekvalizér utlumit. Je však možné sledovat spektrum signálů na jednotlivých mikrofonech a nastavit ekvalizér tak, aby omezil všechna pásma, v nichž se užitečný signál nevyskytuje. Při snímání nástrojů lze teoreticky odhadnout užitečné pásmo frekvencí a zbytek oříznout. Při manuální ekvalizaci je ale vždy riziko, že při omezování frekvencí zpětné vazby dojde i ke slyšitelným změnám užitečného signálu, tudíž je třeba nalézt správný kompromis.

Poslední možností je zpracování dynamiky signálu – pomocí vhodně nastaveného gate, kompresoru nebo jejich kombinace (gate je běžně používaný nástroj zpracování zvuku, který vypne signál, pokud jeho výkon klesne pod definovanou hranici). Gate může vypínat mikrofony, které nejsou používány, a vyvarovat se tak snímání hluku pozadí, ze kterého se někdy pískání zpětné vazby generuje. Kompresor zase pomůže ve chvílích, kdy ke zpětné vazbě už dojde. Pokud překročení meze stability bude malé, může vhodně nastavený kompresor zabránit zpětné vazbě vystoupat do velkých úrovní hlasitosti.

K manuálnímu omezování zpětné vazby samozřejmě patří i typická reakce zvukaře – stažení celkové hlasitosti zpět na stabilní úroveň.

## 1.4.2 Frequency-Shifting a fázová modulace

Pokud se zaměříme pouze na fázi v podmínce stability (5), je možné zásahem do signálu tuto podmínku „obejít“. Pokud v normálním případě úhlová frekvence na daném kmitočtu splňuje podmínku  $k \cdot 2\pi$ , lze případné kruhové zesilování ve smyčce (nestabilitu) narušit malým posunem frekvence či změnami fáze. Aplikací Frequency-Shiftingu dochází k vyrovnání amplitudové charakteristiky v přenosu zpětnovazebního systému [14]. Tento efekt může teoreticky fungovat maximálně do hranice, kdy průměrná hodnota přenosu dosáhne meze v podmínce stability (6). Tato hranice ovšem platí také pro všechny ostatní metody omezení zpětné vazby založené na ekvalizaci

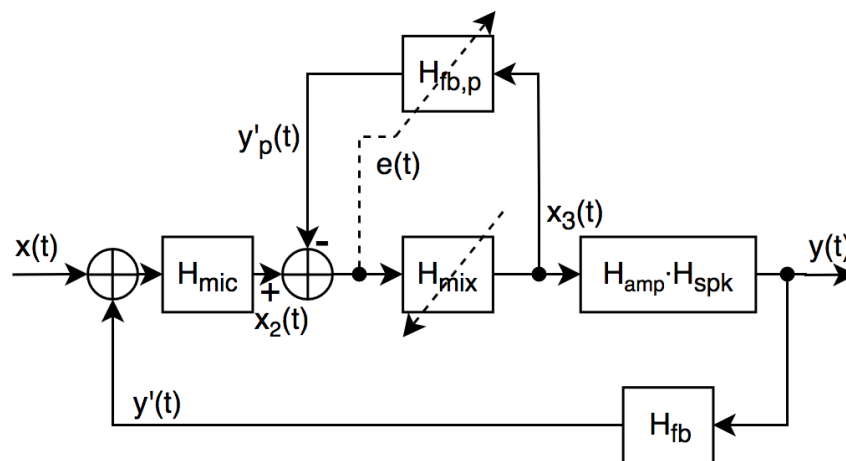


signálu. Posuny frekvence nemusí být pouze FS (Frequency-Shift) metodou, lze použít i periodicky měnící se (LTV) filtr s modulovanou fází, který opět zajistí, že pokud některá frekvence splní kritérium  $k \cdot 2\pi$ , v příštím okamžiku průchodu signálu smyčkou zpětnovazebního systému už kritérium nesplní.

Tento typ automatického omezování je vhodný spíše pro mluvené slovo, než pro zpěv a hudbu, protože při změnách frekvence přestávají platit harmonická hudební pravidla a původně čisté tóny přestávají správně ladit [14]. U mluveného slova lze použít trochu větší (ale i tak velmi malý) posun frekvence, ale vždy bude hranici tvořit stav, kdy se řečník či zpěvák začne cítit nekomfortně, protože z reproduktorů uslyší svůj hlas s jinou výškou tónu. Tato metoda omezování je poměrně triviální a proto výsledky nejsou tak dobré jako u následujících metod. Výhoda této metody je v nulové míře adaptivity, což znamená deterministické stabilní chování a tedy spolehlivost.

### 1.4.3 Adaptivní filtrace

Tato metoda (schéma obr. 5) se podobá adaptivním filtrům AEC, adaptivní algoritmus se snaží odhadnout signál  $y'(t)$ , který se vlivem akustické zpětné vazby  $H_{fb}$  přičítá ke vstupnímu signálu zdroje  $x(t)$  v mikrofonu, a vypočítanou estimaci  $y'_p(t)$  pak odečíst z přijatého signálu mikrofonu  $x_2(t)$ . Estimace  $y'_p(t)$  je vytvářena pomocí reproduktorového signálu  $x_3(t)$ , který je filtrován adaptivním filtrem  $H_{fb,p}(t)$ , jenž se snaží kopírovat přenos zbytku smyčky  $H_{amp} \cdot H_{spk} \cdot H_{fb} \cdot H_{mic}$ . Koeficienty adaptivního filtru jsou aktualizovány pomocí zbytkového signálu  $e(t)$ , který by měl co nejvíce odpovídat  $x(t)$ . Oproti AEC je zde situace složitější kvůli velké korelaci  $x(t)$  a  $y'(t)$ . Adaptivní algoritmus tak často odstraní i část užitečného signálu. U AEC filtru [15] pro řešení situace z obr. 3 lze předpokládat nízkou míru korelace mikrofonního a reproduktorového signálu na jedné i druhé straně.



Obr. 5: Adaptivní filtrace pro omezování akustické zpětné vazby

Teoreticky touto metodou lze dosáhnout i 100% kompenzace zpětné vazby, což by umožnilo zesílení v soustavě bez omezení a k jevu zpětné vazby by nikdy nemohlo dojít. Prakticky však model zpětné vazby není nikdy dokonalý [14] a navíc často není schopný dostatečně rychle se přizpůsobit rychlým změnám ve zpětné vazbě (např. při pohybu mikrofonem). Situaci navíc komplikuje přítomnost difuzního šumu při odrazech v místnosti a fakt, že je třeba do charakteristik zpětné vazby připočítat i často nelineární charakteristiku mikrofonu a reproduktoru, protože nejsme schopni do adaptivního algoritmu poslat akustický signál, ale pouze elektrický. Dokonce ani signál  $x(t)$  není čistý zvuk zdroje a často obsahuje dozvuk místnosti i s šumy pozadí. Dekorelací a dalšími kompenzacemi

rozšířený algoritmus adaptivní filtrace je popisován v [14]. Je třeba dobře kontrolovat konvergenci algoritmu, aby výsledek někdy nebyl horší než systém bez omezování akustické zpětné vazby.

V porovnání s jinými automatickými metodami omezování akustické zpětné vazby je tato metoda nejsložitější, výrazně výpočetně náročnější a tím nevhodná pro implementaci v reálném čase [14]. Na druhou stranu má velký potenciál předčít ostatní metody.

#### 1.4.4 Auto-Notching

Tato metoda původně vznikla z jednoduchého AGC (Automatic Gain Control), který automaticky snížil zesílení signálu v systému, pokud jeho hodnoty dosáhly určité hranice. Takovou hranicí může být střední kvadratická hodnota signálu nebo maximální hodnota signálu. Snížením zisku rovnoměrně v celém frekvenčním rozsahu sice AGC systém spolehlivě vrátil do stability, ale tím pádem nepřidal systému žádný decibel možného zesílení nad původní mez stability bez omezovače.

Dále vznikl AEQ (Automatic Equalizer), který dělal to samé, jen v několika samostatných subpásmech. Díky tomu bylo možné automaticky zeslabit jen ta pásma, která zavinila nestabilitu, a ve zbylých pásmech bylo možné větší zesílení při zachování stability systému než tomu bylo bez omezovače. Automatická aktivace AEQ pro udržení stability systému ale znamenala často slyšitelné změny užitečného signálu, proto vznikla metoda Auto-Notching, někdy označována též NHS (Notch-Filter-Type Howling Suppression).

Aby ekvalizace pro omezení frekvencí způsobujících nestabilitu co nejméně kazila užitečný signál, je třeba omezit co nejužší pásmo. K tomuto účelu nejlépe poslouží přeladitelné notch filtry.

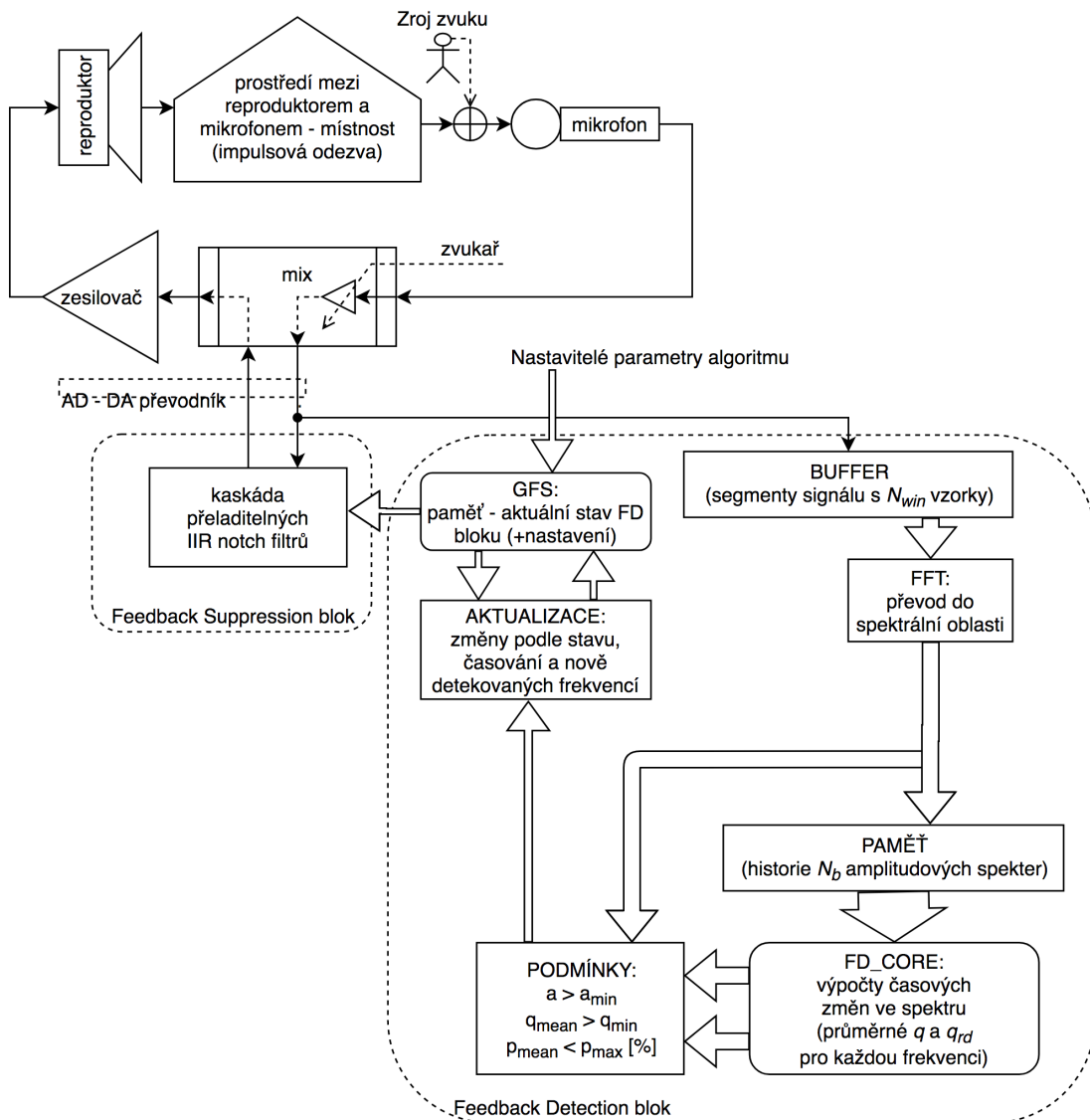
Auto-Notching metoda se skládá ze dvou bloků [14] – blok detekce (Feedback Detection, FD) a blok potlačování (Feedback Suppression, FS). FS blok je zapojen přímo do cesty signálu a obsahuje nastavitelnou filtraci pomocí notch filtrů. FD blok je poměrně složitější, snaží se v signálu hledat stopy nestabilní zpětné vazby a pokud nalezne frekvenci nabíhající zpětné vazby, nastaví FS blok tak, aby tuto frekvenci v signálu potlačoval úzkým notch filtrem. Svým způsobem lze tedy Auto-Notching považovat za speciální verzi adaptivní filtrace. Tato je oblíbená u komerčních výrobců zařízení pro potlačování zpětné vazby [14], protože výpočetní náročnost lze udržet v rozumných mezích pro práci v reálném čase, oproti adaptivní filtraci popisované v předchozí podkapitole.

## 2 Navržený algoritmus automatického omezovače akustické zpětné vazby

Článků a diskusí, které porovnávají všechny možné metody automatického potlačování akustické zpětné vazby, je více (např. [14], [3], [12]). Proto byla pro tuto práci zvolena jen jedna z nich a cílem je spíše navrhnout konkrétní algoritmus, zkusit metodu dále rozvíjet a diskutovat některé klady a zápory. Byl zvolen Auto-Notching (NHS), a to pro jeho předpokládanou volnost v rozvíjení či zpřesňování a menší složitost oproti čistě adaptivní metodě, která by teoreticky mohla mít i lepších výsledků za cenu příliš velké složitosti. Adaptivní filtraci by pravděpodobně nebylo možné implementovat na HW [14] a testovat na reálné soustavě s mikrofonom a reproduktorem, což je jeden z cílů této práce. Metoda Frequency-Shifting je zase příliš primitivní a její výsledky pravděpodobně budou horší [14] než u Auto-Notching metody.

### 2.1 Princip algoritmu

Princip metody Auto-Notching byl již krátce shrnut v kapitole 1.4.4. Na obr. 6 je blokový popis soustavy i s prvky potlačování akustické zpětné vazby, jehož návrhem a testováním se zabývá tato práce.



Obr. 6: Diagram algoritmu Auto-Notching navrženého v této práci

Blok FS (Feedback Suppression) je složen z parametricky nastavitelné kaskády IIR notch filtrů, zjednodušeně řečeno, ekvalizér „vyřezávající“ ze signálu v soustavě některé frekvence pomocí velmi úzké pásmové zádrže. Každý filtr v kaskádě lze nastavit na určitou frekvenci (střed pásma) a nastavit Q. Z těchto parametrů se vypočítají koeficienty IIR filtru druhého řádu. Notch filtr FIR by v této situaci vyšel s velmi vysokým řádem, proto je použit IIR filtr, který bude mnohem efektivnější ve výpočetní náročnosti implementace. Kvůli IIR návrhu je třeba navíc ošetřit, aby vypočítané koeficienty nevytvořily nestabilní filtr. Každý filtr v kaskádě lze zároveň vypnout nebo zapnout, případně přepnout celý FS do režimu „bypass“, aby bylo možné signál propouštět beze změny.

Blok FD (Feedback Detection) slouží pouze ke sledování zvukového signálu soustavy, aniž by do něj přímo zasahoval. Obsahuje většinu výpočtů algoritmu potlačování akustické zpětné vazby a zajišťuje veškerou jeho adaptivitu. Výstupem tohoto bloku jsou povely pro FS blok, které adaptivně mění zpracování signálu pomocí jeho filtrů. Změny filtrů FS bloku se po malém zpoždění ve smyčce elektroakustické soustavy projeví i na vstupu "sledujícího" FD bloku, lze tak zjistit, zda nastavení FS bloku bylo účinné nebo ne.

FD blok se skládá z různých dílčích bloků, které je třeba dále popsat. Vzorčky vstupního signálu jsou rozděleny do segmentů délky  $N_{win}$  s volitelným posunem  $N_{sh}$ , který zároveň udává interval převodu obsahu bufferu do frekvenční oblasti pomocí FFT (rychlé Fourierovy transformace). Zbytek bloků provádí výpočty právě v tomto intervalu, díky tomu je k dispozici historie spektra v diskretních krocích:

$$f_{FD} = \frac{f_s}{N_{sh}} \quad (7)$$

kde  $f_s$  je vzorkovací frekvence AD převodníku a  $N_{sh}$  počet vzorků posunu jednotlivých segmentů. Segmentace může být s překryvem ( $N_{sh} < N_{win}$ ), bez překryvu ( $N_{sh} = N_{win}$ ) nebo dokonce s vynechanými částmi ( $N_{sh} > N_{win}$ ). Výstupem FFT je komplexní spektrum, je třeba vypočítat amplitudové spektrum, informace o fázi je irrelevantní:

$$|\mathbf{x}| = \sqrt{\Re\{\mathbf{x}\}^2 + \Im\{\mathbf{x}\}^2} \quad (8)$$

Rovněž lze zahodit pravou půlku amplitudového spektra, která je zrcadlovou kopií levé půlky, a pracovat pouze s  $N_{fft}/2$  vzorky. V diagramu je znázorňován paralelní přenos  $N_{fft}/2$  hodnot středně tlustými šipkami. Nové aktuální vypočítané amplitudové spektrum je vždy ukládáno do paměti tak, aby v každém cyklu (myšleno interval udávaný  $f_{FD}$ ) mohly být v bloku „FD\_CORE“ vypočítány určité časové parametry z historie  $N_b$  amplitudových spekter. Výpočet probíhá pro každou frekvenci následovně:

$$q_i[k] = a_i[k-1] + a_i[k], \quad (9)$$

průměrný kvocient  $q_i$ :

$$\bar{q}_i = \frac{1}{N_b-1} \sum_{k=1}^{N_b-1} q_i[k], \quad (10)$$

z průměrného  $q_i$  statistický parametr  $p_i$  pro každou frekvenci:

$$p_i = \frac{100}{N_b-1} \sum_{k=1}^{N_b-1} |q_i[k] - \bar{q}_i| \quad [\%] \quad (11)$$

kde  $i$  je index dané frekvence spektra a  $N_b$  je parametr algoritmu udávající délku historie, ze které algoritmus tyto statistiky počítá. Proměnné  $q$  a  $p$  jsou veličiny vysvětlené v kapitole 2.4, které udávají míru podobnosti časového průběhu amplitudy na dané frekvenci s exponenciálně rostoucí amplitudou ideálního modelu jevu akustické zpětné vazby. Výsledkem těchto výpočtů je jsou tedy dvě čísla pro každou frekvenci spektra. V dalším bloku jsou přidány hodnoty amplitudového spektra jako třetí číslo, a následně blok podmínek pošle dál pouze množinu indexů frekvencí, které splnily všechny 3 tyto podmínky:

$$\begin{aligned} a_i &> a_{min} \\ \bar{q}_i &> q_{min} \\ p_i &< p_{max} [\%] \end{aligned} \quad (12)$$

V diagramu je paralelní přenos  $N_b \cdot N_{fft}/2$  vzorků značen nejtlustší šipkou a přenos menšího počtu různých hodnot málo tlustou šipkou. Jednotlivé vzorky či spojitý signál pak znázorňuje tenká šipka.

Vypočítané indexy frekvencí lze přepočítat na přesnou frekvenci jednoduše:

$$f_{res} = \frac{f_s}{N_{fft}} \rightarrow f_i = i \cdot f_{res} \quad (13)$$

Právě na těchto frekvencích se podařilo algoritmu rozpoznat pískání akustické zpětné vazby a je tedy následně vybaveno z paměti „GFS“, zda pro tyto konkrétní frekvence již je nastavený nějaký filtr, a pokud není, tak filtr nastaví. Pokud již na frekvenci nebo v její blízkosti je zařazen filtr, je třeba ho rozšířit nebo mírně doladit.

Paměť „GFS“ uchovává mnoho hodnot. Jsou zde uloženy globální parametry algoritmu i data o aktuálním nastavení FS bloku, tj. parametry jednotlivých IIR notch filtrů. Globální parametry určují chování algoritmu a lze je nastavit externě. Zvukař má tak možnost naladit chování algoritmu pro konkrétní situaci. Některé parametry mohou zlepšit výsledky algoritmu, ale zároveň také zvětšit zpoždění, se kterým FD blok zareaguje na nabíhající zpětnou vazbu. Také je třeba nastavit správně práh amplitudy signálu, aby algoritmus nebral v úvahu příliš slabé signály a zařadil chybné filtry vyvozené ze šumu v ozvučovací technice. Ke každému filtru ve FS bloku je třeba uchovávat v paměti „GFS“ informaci o střední frekvenci, šířce pásma, úrovni, času od zapnutí filtru nebo jeho poslední aktualizace a zda je zapnutý nebo vypnutý.

Další úlohou bloku aktualizace je časování. Každý zapnutý IIR notch filtr v kaskádě má vlastní počítadlo času, je to údaj o stáří tohoto filtru, případně doba od poslední úpravy parametrů tohoto filtru. Pokud by algoritmus neměl žádnou dobu životnosti již zapnutých filtrů, mohlo by po relativně krátké době dojít k potlačení příliš mnoho frekvencí a přenosová charakteristika FS bloku by se spíše podobala útlumu ve větší části spektra než „vyřezání“ úzkých vazbících frekvencí. Zvláště pokud by docházelo k častým změnám přenosu (akustické zpětné vazby) místnosti kvůli přesunujícímu se mikrofonu či jeho natočení, nebo dokonce kvůli příchodu/odchodu zpěváka od mikrofonu. Změny v akustice místnosti může způsobit příchod většího počtu diváků do původně prázdné místnosti. Malé posuny rezonančních frekvencí místnosti, hrající velkou roli při akustické zpětné vazbě, může vyvolat i změna teploty místnosti či vlhkosti v místnosti a tím způsobená změna rychlosti šíření zvuku [3]. Aktuálně potřebné notch filtry pro potlačení zpětné vazby se mohou v čase často měnit, hlavně pokud zvukař mění zesílení v mixu. Je třeba nepotřebné filtry vypnout, aby zbytečně neničily užitečný signál, a zároveň není možné implementovat ve FS bloku neomezený počet IIR filtrů, proto algoritmus musí umět „recyklovat“ již nepotřebné filtry v kaskádě, aby je mohl využít v příštích okamžicích pro jiné frekvence.

Při vývoji algoritmu bylo třeba co nejlépe vyřešit problém, kdy nelze správně zhodnotit, zda některý ze zapnutých filtrů je nebo není v danou chvíli potřeba. Částečným řešením tohoto problému se ukázalo krátkodobé vypínání určitého filtru, aby bylo možné po chvíli zjistit, zda dříve potlačená pískající frekvence se objeví znovu nebo ne. Testovací krátkodobá deaktivace filtru probíhá při stavu, kdy algoritmus věnuje zvýšenou pozornost této konkrétní frekvenci, a pokud se objeví náznak pískání, velmi rychle tento filtr opět aktivuje a zvýší jeho parametr úrovně. Parametr úrovně filtru udává, kolikrát u tohoto filtru byla provedena krátkodobá deaktivace pro zjištění potřeby filtru. Zároveň při každém zvýšení úrovně filtru se zvýší i časový interval, po jehož uplynutí znovu proběhne krátkodobá deaktivace a sledování potřeby filtru. Díky tomuto řešení filtry, které už byly ověřeny, algoritmus zkouší čím dál méně častěji a jejich parametr úrovně udává potřebnost filtru. Toto řešení ve výsledku však způsobuje oproti trvalým filtrům opakované písknutí akustické zpětné vazby na dané frekvenci, protože vypnutím filtru, který potlačuje zpětnou vazbu, k tomuto často slyšitelnému náběhu pískání dochází vícekrát, postupně s rostoucím intervalem. Je tedy třeba najít správný kompromis mezi narážením na chyby při nedostatku volných filtrů a častěji se opakujícím jevům potencionálně slyšitelného náběhu pískání akustické zpětné vazby.

Volba časových parametrů pro algoritmus se ukázala jako velmi důležitá ve chvílích, kdy pískání vzroste na velmi slyšitelnou úroveň, než jej algoritmus potlačí správných nastavením FS bloku. Jedním z hlavních časových parametrů určujících dobu reakce je  $N_b$ , počet uložených amplitudových spekter v historii, a logicky také  $N_{sh}$ , počet vzorků udávajících interval mezi jednotlivými spektry v této historii. Pokud je  $N_b$  větší, lze dosáhnout spolehlivější identifikace akustické zpětné vazby, zato doba reakce algoritmu se prodlouží a nabíhající pískání může dojít k větší hlasitosti, než je potlačeno přidáním filtru.

Další otázkou bylo, jak správně v algoritmu rozlišit, zda pískání či zvuk na dané frekvenci je nebo není zpětná vazba. Navržený algoritmus se snaží vypočítat míru podobnosti časového průběhu amplitudy na dané frekvenci s průběhem dle modelu ideální akustické zpětné vazby v nestabilní soustavě. Podle [14] lze vypočítat mnoho dalších statistických ukazatelů, se kterými lze pracovat, často jsou však příliš výpočetně náročné (např. poměr výkonu ku výkonu vyšších harmonických frekvencí) nebo jejich použití při testování algoritmu neukázalo pozorovatelné zlepšení výsledků omezovače (např. poměr výkonu ku průměrnému výkonu nebo ku výkonu sousedních frekvencí). Princip rozlišování opravdového pískání zpětné vazby od velmi podobných pískání je ve stanovení určité meze (threshold), se kterým jsou pak porovnány aktuální výsledné hodnoty spočítané pro každou frekvenci ve spektru. Zmíněné hodnoty reprezentují podobnost signálu s jevem akustické zpětné vazby a konkrétně se ukázala vhodná kombinace tří různých kritérií. Hodnota aktuální amplitudy  $a$  vytváří ochranu proti příliš slabému signálu a tím proti chybám způsobeným velkým šumem ve chvílích, kdy zdroj zvuku je zticha. Hodnota kvocientu  $q$  udává rychlost náběhu pískání zpětné vazby a tím teoretický údaj, jak moc je na dané frekvenci překročena mez stability. A poslední statistická hodnota  $p$  udává procento relativní průměrné absolutní odchylky  $q$  při jeho průměrování. Amplituda  $a$  je použita z aktuálního vzorku spektra a hodnoty  $q$  a  $p$  jsou vypočítány z posledních několika vzorků v historii aktuálního  $q$  (viz kapitoly 2.4.2 a 2.4.3). Tyto meze pro rozpoznávání jevu akustické zpětné vazby slouží jako další nastavitelné parametry algoritmu, kterými lze měnit chování a spolehlivost algoritmu.

Je možné definovat různé hodnoty parametrů algoritmu pro různé aplikace omezovače, například pro koncerty populární hudby mohou být rozpoznávací meze více přísné, než pro přednášky a jiné čistě hlasové produkce, protože dobře fungující meze při mluveném slově mohou u populárních koncertů vytvářet mnoho chybných identifikací jevu. Na druhou stranu přísnější meze v algoritmu mohou méně výrazné případy jevu nerozpoznat a v mluveném slově by to znamenalo vznik velmi

rušivého zvuku. Stejný rušivý zvuk by však v populární hudbě nebyl tolik rušivý a bylo by možné ho do určité míry tolerovat.

## 2.2 Ekvalizace pomocí kaskády přeladitelných IIR filtrů

Důležitou součástí algoritmu je FS blok (Feedback Suppression), který zajišťuje nastavitelný parametrický ekvalizér. Tento ekvalizér nahrazuje manuální práci zvukaře, který by ekvalizérem vyrovnával amplitudovou přenosovou charakteristiku prostředí a elektroakustických měničů, aby mohl nastavit co největší zesílení, aniž by systém uvedl do nestability. Algoritmus automaticky ovládá tento ekvalizér a snaží se reagovat mnohem rychleji a přesněji, než by reagoval v takové situaci zvukař. Ekvalizér FS bloku je speciálně navržen pro potlačování akustické zpětné vazby. Manuální ekvalizér zabudovaný v analogových mixážních pultech bývá příliš jednoduchý třípásmový ekvalizér s pevným  $Q$  i pevnou střední frekvencí. Pomocí takového ekvalizéru má zvukař možnost potlačovat akustickou zpětnou vazbu jen velmi málo. V digitálních mixážních pultech lze najít grafický ekvalizér, který obsahuje tři až pět plně nastavitelných filtrů, a lze takovým filtrem po několika minutách dobře zpětnou vazbu omezit. Ekvalizér používaný ve FS bloku tohoto algoritmu je specializován na velký útlum desítek velmi úzkých pásem pomocí IIR notch filtrů. IIR notch ekvalizérem lze potlačovat konkrétní frekvence, které nesplňují podmínky stability (5) a (6), aniž by ekvalizér zbytečně utlumil užitečný signál. O hledání správných frekvencí se v automatickém algoritmu stará FD blok, případně lze pro účely testování algoritmu nastavit FD blok s ekvalizérem ručně, popřípadě jeho stav „zmrazit“ nebo uložit. Pokud by byl omezovač použit v situaci, kdy je pozice a nasměrování mikrofonů a reproduktorů pevná, lze nechat předem algoritmus omezovače najít kritické frekvence při maximálním zesílení, které chceme teoreticky používat, potom FD blok „zmrazit“ a opět zesílení snížit na běžnou úroveň. Tento postup (nazývaný „ring-out room“) se někdy v praxi používá při přípravě ozvučení sálů, například v divadlech a přednáškových místnostech.

Pro realizaci ekvalizéru FS bloku byla zvolena kaskáda IIR notch filtrů, namísto jediného IIR filtru vysokého řádu. Každý IIR filtr v kaskádě je 2. řádu a koeficient  $a_0$  je vždy roven 1. Všech zbylých 5 koeficientů je možné vypočítat ze střední frekvence a  $Q$  nebo šířky pásma.

## 2.3 Alternativní verze ekvalizace s filtrací ve spektru OLS

Při vývoji algoritmu omezovače byla navržena a testována i alternativní verze FS bloku, která používá filtrování ve spektru, které může nahradit kaskádu IIR filtrů. Filtrace ve spektru spočívá v převedení signálu do frekvenční oblasti pomocí FFT (Fourierovy transformace), následné manipulaci s vzorky spektra a potom převedení upraveného signálu zpět do časové oblasti. Spektrální úpravy signálu lze provést metodou OLS (Overlap and Save) [7], která spočívá v segmentaci signálu a pronásobení spektra jednotlivých segmentů s navrženou amplitudovou charakteristikou požadovaného filtru.

Všechny dílčí notch filtry s parametry generovanými FD blokem jsou sloučeny do jedné amplitudové přenosové charakteristiky a ta je násobena vzorek po vzorku s každým segmentem signálu převedeného do frekvenční oblasti. Vytvarování jednotlivých „notchů“ probíhá interpolací několika bodů základního modelu do požadovaného počtu vzorků (závislé na  $Q$ ) levé půlky a druhá půlka je vždy symetrická. Po vytvarování je „notch“ umístěn na požadované místo ve spektru (závislé na střední frekvenci filtru). V algoritmu je možno nastavit různé tvary „notche“ a zvolit lineární nebo kubickou interpolaci. Nejprimitivnější tvar je úzký obdélník, dále je na výběr tvar  $V$  nebo další tvary podobné tvaru IIR notch filtru 2. řádu.

Porovnání obou verzí FS bloku se nachází v kapitole 3.5.

## 2.4 Použité matematické nástroje

V této podkapitole jsou podrobněji vysvětleny matematické operace, které algoritmus využívá.

### 2.4.1 FFT pro sledování časového průběhu amplitudy všech frekvencí

Jedním z klíčových nástrojů algoritmu je výpočet Diskrétní Fourierovy transformace, konkrétně její rychlá varianta FFT. Tato matematická operace slouží v algoritmu k převedení signálu z časové oblasti do frekvenční oblasti. Pro většinu bloků algoritmu je vhodnější zpracovávat signál ve frekvenční oblasti.

Simulace a implementace algoritmu v MATLABu používá knihovnu FFTW, která zajišťuje nejrychlejší adaptivní výpočet FFT výběrem (či kombinací) vhodného algoritmu pro dané  $N_{fft}$  z různých algoritmů. Implementace algoritmu na DSP používá knihovnu [9] obsahující Cooley-Tukey verzi FFT s předem vygenerovanou tabulkou „twiddle factors“, která usnadňuje výpočet FT, ale je třeba ji přepočítat, pokud se změní počet vzorků  $N_{fft}$ .

V algoritmu je výstup FFT každého segmentu uchovávan v historii, díky tomu je možné sledovat časový průběh amplitudy každé frekvence a z něj pak vyhodnocovat důležité hodnoty pro FD blok algoritmu. Další výpočty popsané níže už probíhají jednotlivě pro časový průběh amplitudy určité frekvence. Vzorkovací frekvence pro vzorky amplitudy odpovídá  $f_{FD}$ , která závislá na vzorkovací frekvenci AD převodníku  $f_s$  a na délce posunu segmentů signálu  $N_{sh}$  podle vzorce (7). Časový průběh amplitudy pro určitou frekvenci s indexem  $k$  lze vyjádřit posloupností  $a_{k,n}$ ,  $a_{k,n-1}$ ,  $a_{k,n-2}$ ,... kde index  $n$  značí pořadí v čase.

### 2.4.2 Kvocient geometrické posloupnosti

Exponenciální průběh amplitudy signálu lze popsat jako geometrickou posloupnost [6] vzorků s kvocientem  $q$ , který udává rychlost růstu nebo klesání této exponenciály. Pokud očekáváme exponenciální průběh, lze pro dva sousední vzorky vypočítat tento kvocient:

$$a_n = q_n \cdot a_{n-1} \rightarrow q_n = \frac{a_n}{a_{n-1}} \quad (14)$$

Pokud zkoumaná posloupnost vzorků amplitudy  $q_n$  bude stále stejné nebo velmi podobné číslo, znamená to, že zkoumaný signál je podobný exponenciálnímu průběhu amplitudy. Bude-li toto číslo větší než 1, znamená to rostoucí exponenciální průběh. Ideální model vzniku jevu nestabilní akustické zpětné vazby má přesně exponenciální průběh amplitudy s kvocientem větším než 1, vyplývá to ze schématu na obr. 2. Zároveň tento fakt potvrzuje i spodní graf na obr. 4 z reálné situace (samozřejmě se exponenciální růst zastaví, když soustava začne kvůli přebuzení nelineárně zkreslovat).

### 2.4.3 Relativní průměrná absolutní odchylka

Jako určitý ukazatel míry, jak moc se naměřený průběh historie jedné frekvence ve spektru podobá ideálnímu exponenciálnímu růstu, lze použít  $p_{rd}$  relativní průměrnou absolutní odchylku spočítaných  $q_n$  (kvocientů geometrické posloupnosti):

$$p_{rd} = \frac{100}{q_p} \cdot \frac{1}{N} \sum_{n=1}^N |q_n - q_p| \quad [\%] \quad (15)$$

kde  $N$  je počet uchovávané historie spektra a  $q_p$  je průměr všech  $q_n$ :



$$q_p = \frac{1}{N} \sum_{n=1}^N q_n \quad (16)$$

#### 2.4.4 Výpočet koeficientů IIR notch filtru 2. řádu

Kód pro výpočet koeficientů IIR filtru pochází z výpočtu použitého v DSP System Toolboxu v MATLABu. Algoritmus potřebuje navrhnout  $b_n$  a  $a_n$  koeficienty IIR notch filtru 2. řádu z údajů o střední frekvenci a šířce pásma. Zde je výpočet řešení, nejdříve je třeba převést frekvenční hodnoty na normalizovaný tvar, protože  $f_0$  je v jednotkách indexu frekvence (znamená pořadí vzorku ve vypočítaném spektru o  $N_{fft}$  vzorcích):

$$\omega_{0n} = \frac{f_0}{N_{fft}} 2\pi \quad [\text{rad/s}] \quad (17)$$

$$\omega_{bn} = \frac{\omega_{0n}}{Q} \quad [\text{rad/s}] \quad (18)$$

kde  $f_0$  je index střední frekvence požadovaného IIR notch filtru a  $Q$  je ve činitel jakosti, daný globálním parametrem algoritmu.

Požadovaný útlum filtru na krajích pásma je potřeba převést z dB hodnot na amplitudové hodnoty:

$$d = 10^{\frac{dep\_db}{20}} \quad (19)$$

V dalším kroku je výpočet zisku  $g$ :

$$\beta = \frac{\sqrt{1-2^d}}{d} \cdot \text{tg}\left(\frac{\omega_{bn}}{2}\right) \quad (20)$$

$$g = \frac{1}{1+\beta} \quad (21)$$

A nakonec výpočet samotných  $b$  a  $a$  koeficientů:

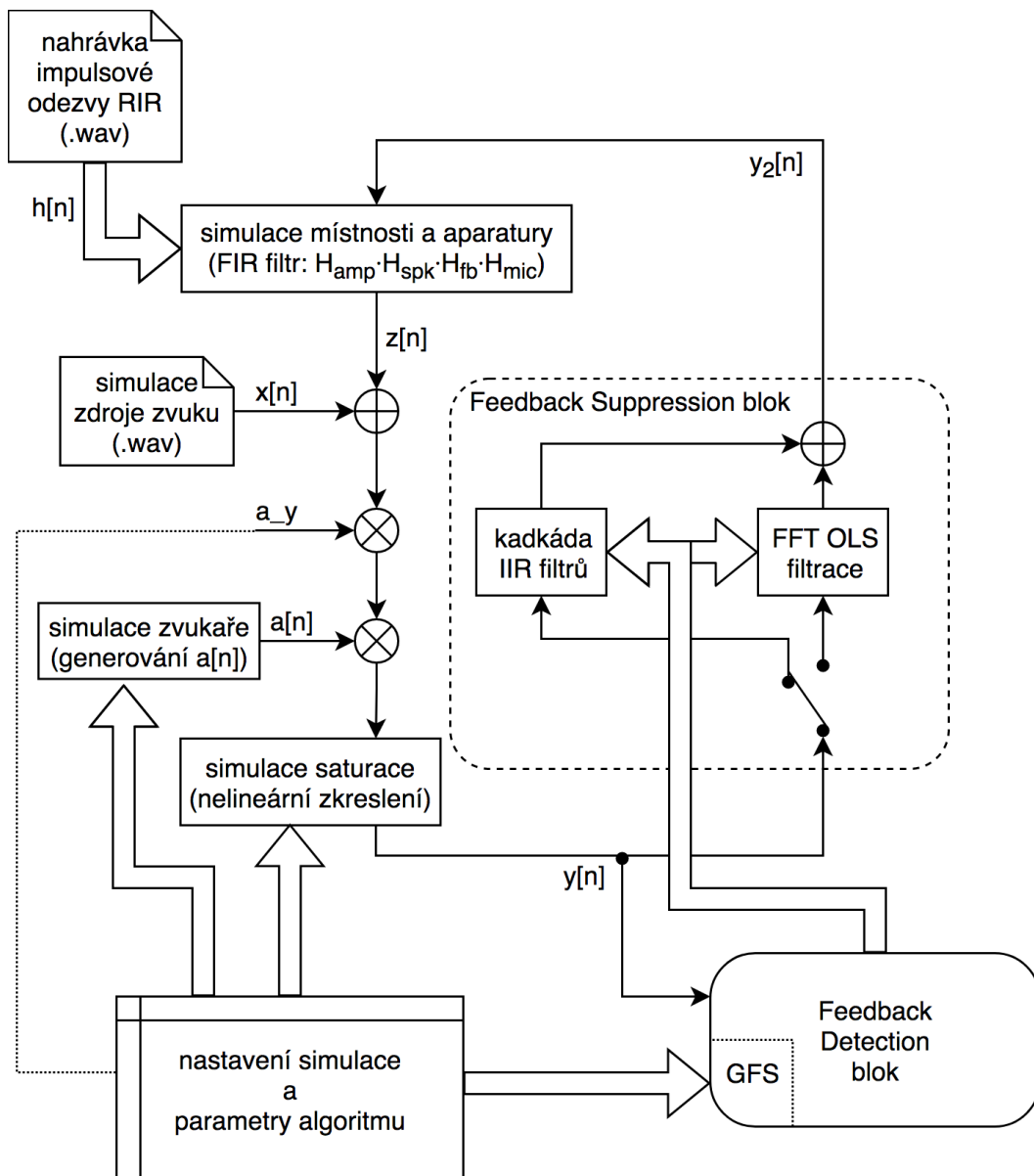
$$\begin{aligned} b_0 &= g, \quad b_1 = -2g \cos(\omega_{0n}), \quad b_2 = g \\ a_0 &= 1, \quad a_1 = -2g \cos(\omega_{0n}), \quad a_2 = 2g - 1 \end{aligned} \quad (22)$$

### 2.5 Simulace – implementace v MATLABu

Vývojové prostředí MATLABu nabízí mnoho šikovných nástrojů ke grafickému zobrazení dat a programovací jazyk vyšší úrovně usnadňující programování složitých algoritmů. Oproti tomu vývoj algoritmu přímo na číslicovém signálovém procesoru (DSP) v programovacím jazyce C by v této fázi vývoje nebyl tak efektivní. Implementace výše popisovaného algoritmu potlačování akustické zpětné vazby v MATLABu by nebyla schopná pracovat v reálném čase, jde tedy pouze o simulaci, která veškeré fyzické prvky elektroakustické soustavy popisované v první kapitole simuluje softwarově a poslouží tak velmi dobře pro základní vývoj algoritmu a testování dílčích částí. V druhé části vývoje algoritmu je kód původně napsaný v MATLABu přepsán do jazyka C a algoritmus je tak implementován na desce vývojového přípravku s DSP. Následuje odladění a zkoušení algoritmu v reálných podmínkách s reálným připojeným ozvučením. Některé věci by se v MATLABu velmi těžko simulovaly, hlavně plynule se měnící prostředí – impulsová odezva místnosti, plynulé nastavování mixu zvukařem, různé šумы v aparatuře i difuzní šum. Čas simulace v MATLABu navíc

oproti reálnému času probíhá mnohem pomaleji (řádově 40-100x), a i kdyby byla implementace v MATLABu propojena přes zvukovou kartu s reálnou aparaturou, algoritmus by nestihl vše vypočítat dost rychle.

Implementaci algoritmu v MATLABu odpovídá schéma na obr. 7. Je zde navíc několik bloků, které nepatří do algoritmu, ale starají se o simulaci prostředí a aparatury podle elektroakustické soustavy popisované v první kapitole. Dva základní bloky algoritmu FD a FS zde fungují přesně, jak popisuje předchozí kapitola.



Obr. 7: Schéma simulace v MATLABu a v ní implementovaného algoritmu potlačení akustické zpětné vazby

Nejdůležitější blok simulace je FIR filtr, který konvoluuje signál  $x_2[n]$  se speciální impulsovou odezvou. Těto impulsově odezvě se říká RIR (room impulse response) a lze pomocí takovéto impulsově odezvy simulovat akustické prostředí (místnost). Zároveň je potřeba simulovat přenosy všech dílů elektroakustické soustavy  $H_{amp} \cdot H_{spk} \cdot H_{fb} \cdot H_{mic}$ , proto lze do této RIR pro účely simulace zahrnout celkovou impulsovou odezvu všech těchto částí dohromady, kde přenos  $H_{fb}$  odpovídá čisté

impulsové odezvě prostředí  $h_{rir}[n]$ . Zároveň se tím vyřeší složité získávání  $h_{rir}$  a mnohem jednodušeji lze metodou MLS [5] získat celkovou  $h[n] = h_{amp} \cdot h_{spk} \cdot h_{rir} \cdot h_{mic}$ . RIR použitá v simulaci dokonce zahrnuje i charakteristiku a zpoždění AD a DA převodníku, protože byla získána prostřednictvím stejné desky s DSP, jako je použita při implementaci algoritmu v kapitole 2.6. Přímou v desce s DSP byl nahrán MLS signál vyslaný z výstupu desky, prošlý zkoumanou soustavou (mikrofon, prostředí ak.z.v., reproduktor) až zpět do DSP. MLS metodou byla v MATLABu ze zdrojového MLS signálu a přijatého MLS signálu vypočítána impulsová odezva RIR a uložena do \*.wav souboru, který je následně použit v simulaci. V simulaci lze samozřejmě použít i jiné z internetu stažené impulsové odezvy místností, které však většinou nezahrnují i aparaturu.

Zdroj zvuku  $x[n]$  simuluje nahrávka načtená z externího souboru \*.wav. Lze použít různé typy nahrávek reprezentující různé situace použití omezovače akustické zpětné vazby. Jako testovací vzorky poslouží tyto nahrávky – anglicky mluvící žena v anechoické místnosti, koncert vážné hudby, rocková skladba a několik nahrávek samostatných hudebních nástrojů v anechoické místnosti.

Signál  $x[n]$  je sečtený se signálem  $z[n]$ , který přichází zpětnou vazbou a prochází simulovanou aparaturou. Pomocí předzesilovače  $a_y$  je pevně nastaveno celkové zesílení smyčky s rozpojenou zpětnou vazbou tak, aby při  $a[n] = 0$  byla celá smyčka jen několik dB pod mezí stability. Hodnota  $a_y$  se v průběhu simulace nemění, změny v zesílení ovládané zvukařem se simulují pomocí zisku  $a[n]$ , nastaveném na virtuálním mixážním pultu. Časová závislost zisku  $a[n]$  v průběhu simulace je modelována několika parametry popsány v následující kapitole.

Následující blok simuluje nelineární zkreslení reálné aparatury, ke kterému dochází při přebuzení některého zesilovače, mikrofonu nebo reproduktoru. Nezáleží příliš na tom, který z prvků aparatury při velké intenzitě začne zkreslovat jako první, důležité je to, že k saturování (ořezávání) signálu někde dochází, proto je tento blok umístěn virtuálně na výstup mixážního pultu. Parametrem  $x_m$  lze nastavit tuto maximální hodnotu a všechny vzorky signálu větší než  $+x_m$  nebo menší než  $-x_m$  jsou nahrazeny mezní hodnotou. Pokud by tento blok v simulaci chyběl, pískání zpětné vazby by rostlo nade všechny meze do nekonečna až na hranici použité aritmetiky, což u čísel typu „double“ je tak vysoká hodnota, že by průběh simulace velmi málo odpovídal reálné situaci.

Celý průběh simulace je implementován metodou výpočtu vzorek po vzorku, protože blokový výpočet u tak rozsáhlé struktury není možný. Doba výpočtu celé simulace se tím výrazně prodlužuje. Pouze u FIR filtru simulujícího RIR je možné výstupní vzorek vypočítat skalárním součinem vektoru  $h$  s vektorem posledních vzorků signálu  $z$  o stejné délce jako  $h$ . Tím lze docílit mnohem efektivnějšího výpočtu operace, která se provádí pro každý jednotlivý vzorek signálu procházející simulovanou smyčkou. Některé dílčí části FD bloku jsou také implementovány vektorovými nebo maticovými operacemi.

Na konci simulace jsou graficky zobrazeny výsledky a přehrán signál  $z[n]$  pomocí reproduktorů PC. Případně mohou být všechny výsledky exportovány do několika souborů na disku, které mají v názvu zapsáno několik základních hodnot parametrů algoritmu i parametry simulace. Exportované soubory mohou sloužit k porovnávání různých proběhlých simulací. Názorná ukázka výsledků simulace je v kapitole 3.3.

Pro testování zkreslení samotným FS blokem (kaskády IIR notch filtrů nebo OLS spektrálního filtru) byly vytvořeny další 2 samostatné programy MATLABu iir\_benchmark.m a ols\_benchmark.m. Do nich byl zkopírován pouze kód jedné nebo druhé verze FS bloku a následně testován vliv parametrů  $Nwin$ ,  $Nsh$ ,  $Niir$  a  $q_m$  (popis lze nalézt v tab. 1 níže). Pro testování v těchto oddělených skriptech je třeba vytvořit několik různých neměnicích se filtrů, protože zde neexistuje žádný FD blok,

kteřý by je nastavil. Lze testovat i zkreslení signálu v závislosti na těchto různých sadách filtrů, ať už IIR notch filtrů v kaskádě nebo OLS spektrálního filtru.

## 2.5.1 Nastavitelné parametry algoritmu a parametry simulace

Před spuštěním simulace je možné nastavit mnoho parametrů samotného simulovaného prostředí i mnoho parametrů algoritmu omezovače. V této kapitole je seznam s popisem všech parametrů a zjištěným vlivem na simulaci nebo na algoritmus omezovače.

název	popis parametru algoritmu
$N_{win}$	Počet vzorků v segmentu signálu. Větší hodnota znamená větší frekvenční rozlišení spektra, zároveň však dochází ke zvýšení výpočetní náročnosti. Tento parametr zároveň udává zpoždění přidané do signálu vlivem vstupně-výstupních bufferů u AD a DA převodníku. MATLAB sice nemá AD a DA převodník, ale simuluje toto zpoždění uměle. Hodnota tohoto parametru tedy přímo úměrně udává do signálu latenci vzniklou průchodem FS bloku. Limit $N_{win}$ je přibližně 1000 (při $f_s = 48$ kHz), kdy vzniká latence 20.8 ms. Větší latence už by byla slyšet a vznikal by velmi nepřírozený echo efekt. Běžná latence vstup-výstup u digitálních zvukových efektů bývá do 10ms.
$N_{sh}$	Posun – počet vzorků rozestupu dvou po sobě jdoucích segmentů. Pokud je menší než $N_{win}$ dochází k překryvu segmentů. Hodnota ovlivňuje frekvenci FD bloku $f_{FD}$ , podle vzorce (7). Pro většinu situací je vhodná hodnota $N_{win}/2$ .
$N_{fft}$	Počet vzorků při výpočtu spektra pomocí FFT. Nesmí být menší než $N_{win}$ a pokud je větší než $N_{win}$ dochází k interpolaci ve spektru (doplněním vstupních vzorků nulami [15]). Ideální je ponechat $N_{fft}=N_{win}$ a zvětšovat přímo hodnotu $N_{win}$ .
$N_b$	Počet vzorků uchovávaných v historii amplitudy (historii spekter). Větší číslo znamená výrazně menší chybovost algoritmu, ale zároveň přímo úměrně delší dobu reakce algoritmu a zvýšení výpočetní náročnosti. Dobrým kompromisem se osvědčila hodnota 16, pro mluvené slovo postačí hodnota 8 a více, pro rockovou hudbu je potřeba větší, přibližně 32.
$N_{iir}$	Počet slotů na IIR filtry – maximální počet IIR notch filtrů v kaskádě FS bloku. Obecně platí, že větší počet je lepší. Pokud algoritmus potřebuje potlačit více než $N_{iir}$ frekvencí, algoritmus nemá dostatek filtrů a nezvládne udržet zpětnovazební smyčku stabilní. Větší počet znamená vyšší výpočetní náročnost. Ve většině testovaných případech se ukázal jako dostačující počet 16 až 32. Pokud to výpočetní výkon dovolí, lze zvolit i číslo větší.
$db\_min$	Minimální amplituda – jedna z podmínek pro detekci nežádoucí zpětné vazby na určité frekvenci v dB ( $a_{min}=10^{db\_min/20}$ ). Pokud je hodnota příliš malá může docházet k častému chybnému identifikování jevu v místech příliš slabého signálu. Ideální hodnota se hledá těžce a vhodné nastavení spočívá v nalezení správné kombinace čísel $db\_min$ , $fd\_p\_max$ a $fd\_a\_min$ . Během testování se ukázalo jednodušší hledat tuto kombinaci pomocí implementace algoritmu na DSP, kde lze operativně měnit tyto tři hodnoty a sledovat chování algoritmu pohybem mikrofonu či plynulým ovládním faderu na mixu. Velmi velké hodnoty tohoto parametru způsobí výrazněji slyšitelné náběhy nežádoucího jevu, protože intenzita jevu vystoupá do příliš velkých hodnot, než splní podmínku danou touto mezí.

<i>fd_p_max</i>	Maximální procento relativní průměrné absolutní odchylky $p_{rd}$ kvocientu $q$ – jedna z podmínek pro detekci nežádoucí z. v. na určité frekvenci. Ideální hodnota se hledá těžce a vhodné nastavení spočívá v nalezení správné kombinace čísel <i>db_min</i> , <i>fd_p_max</i> a <i>fd_a_min</i> . Z testování MATLAB implementace i DSP implementace algoritmu vyplývá vhodná hodnota přibližně 5%. Tento parametr je klíčový v rozeznání nežádoucího jevu akustické zpětné vazby od zbytku užitečného signálu, který v časovém průběhu amplitudy může být v některých úsecích podobný.
<i>fd_a_min</i>	Minimální velikost průměrného kvocientu $\bar{q}$ – jedna z podmínek pro detekci nežádoucí z. v. na určité frekvenci. Ideální hodnota se hledá těžce a vhodné nastavení spočívá v nalezení správné kombinace čísel <i>db_min</i> , <i>fd_p_max</i> a <i>fd_a_min</i> . Z testování MATLAB implementace vyplývá vhodná hodnota 1.01 a z testování DSP implementace vyplývá vhodná hodnota 1.005 až 1.008. Tento parametr zajišťuje, aby aproximovaná exponenciála časového průběhu amplitudy byla kladná, proto musí být parametr vždy větší než 1. Je třeba nastavit tak malou hodnotu, aby i velmi pomalu vzrůstající amplituda nestabilní frekvence smyčky splnila tuto podmínku, jinak by ji algoritmus ignoroval a pískání by sice pomalu, ale jistě došlo krajní velké intenzity.
<i>q_m</i>	Činitel jakosti $Q$ použitý pro všechny vytvářené filtry. Parametr ovlivňuje, jak úzké budou všechny algoritmem navrhované notch filtry. Účinnost změn tohoto parametru je složité sledovat, protože se projevují velmi odlišně pro různé situace nasazení omezovače. Širší potlačená pásma, způsobené malým $q_m$ , přinášejí větší odolnost proti změnám prostředí, vznikajících pohybem mikrofonu vůči reproduktoru a způsobujících malý posun potenciálně nestabilních frekvencí smyčky. Na druhou stranu se tím zvětšuje zkreslení vzniklé útlumem většího procenta užitečného signálu. Poměrně dobré výsledky jsou dosaženy při nastavení $q_m = 4$ .
<i>fs_enable</i>	Tímto boolean parametrem lze vypnout nebo zapnout FS blok. Pokud je hodnota false, vstup FS bloku je přepojen přímo do výstupu a žádným způsobem nemění procházející signál. Lze tím testovat a porovnat rozdíl zapnutého nebo vypnutého omezovače ak. z. v.
<i>iir_fft_enable</i>	Tento boolean parametr přepíná verzi FS bloku. Hodnota true zapne alternativní verzi s FFT OLS filtrací.

Tab. 1: Seznam nastavitelných parametrů algoritmu omezovače v MATLABu

<b>název</b>	<b>popis parametru simulace</b>
<i>t<sub>sim</sub></i>	Délka simulace v sekundách (zbytek vstupního signálu $x$ se nepoužije).
<i>f<sub>s</sub></i>	Vzorkovací kmitočet. Jiné hodnoty než 48 000 Hz nebyly testovány, protože je nežádoucí, aby se kvůli průchodu omezovače snižoval frekvenční rozsah signálu a tím i jeho kvalita.
<i>rir</i>	RIR – simulovaná impulsová odezva prostředí i s aparaturou. Načítá se ze souboru *.wav a většinou je zkrácena na vhodný počet vzorků. Různé RIR mohou simulovat různé sestavy ozvučení a akustické prostředí.
<i>x</i>	Vstupní signál – simulovaný zdroj zvuku. Načítá se ze souboru *.wav a je zkrácen na $N_{sim}$ vzorků ( $N_{sim} = f_s \cdot t_{sim}$ ). Různé signály mohou různě ovlivňovat chybovost algoritmu i efekt nežádoucího jevu.
<i>x<sub>m</sub></i>	Mezní hodnota signálu. Signál je saturován na tuto hodnotu – simulace nelineárního zkreslení v reálné přebuzené elektroakustické soustavě. Je vhodné nastavit takovou hodnotu, která způsobí zkreslení signálu až ve chvílích kdy by ji zkreslovalo reálné ozvučení (aparatura).

$a\_ydb$	Zisk předzesilovače v dB ( $a\_y=10^{a\_ydb/20}$ ). Role předzesilovače v simulaci je pouze pomocná.
$a1\_db$	Počáteční zisk simulovaného faderu mixážního pultu v dB (simulovaný zásah zvukaře), většinou 0. Parametry $a1\_db$ , $a2\_db$ , $t_1$ a $t_g$ lze simulovat změnu stabilní smyčky na nestabilní uprostřed časového průběhu simulace.
$a2\_db$	O kolik dB se zvýší zisk simulovaného faderu mix. pultu, když uběhne doba $t_1$ .
$t_1$	Doba, po které dojde v simulaci ke zvýšení zisku faderu mix. pultu, v sekundách.
$t_g$	Doba lineárně se měnícího zisku z $a1\_db$ na $a2\_db$ .

Tab. 2: Seznam nastavitelných parametrů simulace v MATLABu

## 2.6 Testování v reálném čase – implementace v C na DSP

Po vyvinutí základního algoritmu v MATLABu a základním testování v simulovaném prostředí se ukázalo, že pro další vylepšování a ladění algoritmu omezovače bude lepší využít reálné prostředí a implementaci algoritmu na DSP. Problém simulace spočívá ve velmi malé operativnosti. Simulace se připraví, nastaví se parametry a signály a po spuštění už není možné operativně do průběhu zasahovat. Není možné plynule měnit parametry algoritmu nebo plynule měnit simulované prostředí a sledovat, jak na to algoritmus reaguje. Navíc bylo třeba ověřit, že algoritmus schopný fungovat v ideálním prostředí simulace bude fungovat i v reálném prostředí. Simulovat změny zesílení ve smyčce vytvořené zvukařem není tak velký problém, jako simulovat změny prostředí, které v reálné smyčce do jisté míry probíhají neustále. Navíc simulovat skokové změny RIR by zdaleka neodpovídalo plynulým změnám probíhajícím v reálném prostředí.

Na druhou stranu pokud je algoritmus implementován na DSP, je velmi složité výsledky algoritmu posuzovat objektivními metodami, jako to bylo možné v simulaci v MATLABu. Dalším problémem je práce s hardwarem DSP. Všechno, co bylo v MATLABu jednoduché naprogramovat, je v C složité. Navíc v MATLABu není třeba se zabývat programováním kódu pro AD/DA převodník, externí RAM, časovačů a dalších periférií DSP, které jsou nutné, aby vůbec DSP fungovalo tak jak má. Dalším omezením implementace na DSP je LCD displej. V MATLABu všechny rozmanité grafické prvky stačilo zobrazit jedním příkazem, na DSP je třeba všechno programovat „pixel po pixelu“. Pěkné grafy a jiná chytrá zobrazení tak u sledování algoritmu na DSP nepřipadají v úvahu. Některé vnitřní části algoritmu omezovače je tak třeba vyvíjet v MATLABu a ostatní detaily algoritmu zas vyvíjet testováním v realitě a prací s DSP.

Kvůli náročnosti implementace na DSP i potížím s výpočetním výkonem vyžadující optimalizaci přidaného kódu, nezbyl čas dokončit a testovat FFT OLS verzi FS bloku. Proto není možné v kapitole 3 porovnat obě verze FS bloku na DSP, ale pouze tyto verze porovnat v simulaci v MATLABu.

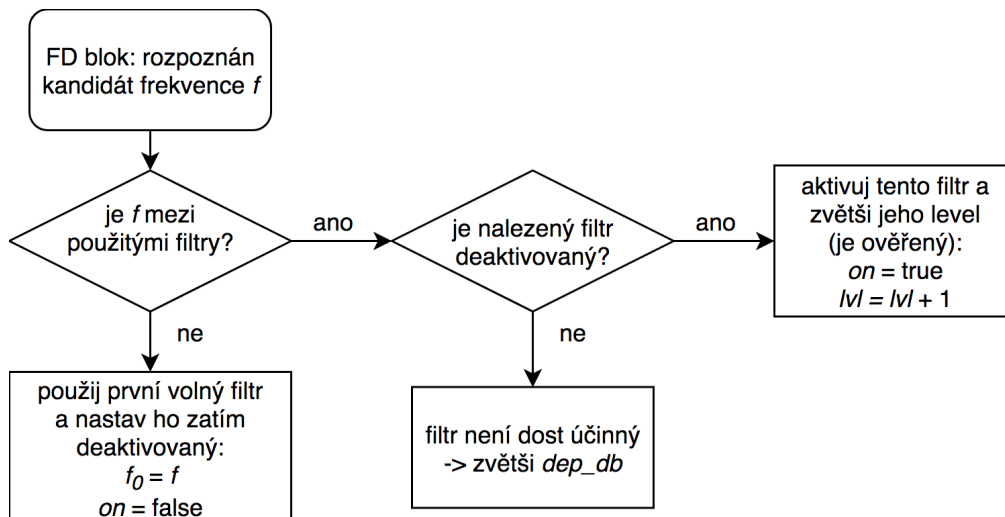
Ještě je třeba popsat strukturu parametrů a vysvětlit časování a doby životnosti IIR notch filtrů v kaskádě. Tato část algoritmu byla dovyvinuta až při implementaci a testování na DSP, protože nutnost omezit životnost filtrů v kaskádě se objevila až při testování v reálném prostředí a reálném čase. Simulace v MATLABu probíhají na tak krátkých úsecích času, že zapnuté IIR filtry v kaskádě není potřeba po čase vypnout nebo měnit. Simulace v této oblasti nestačí k testování algoritmu, aby ho nutila reagovat na dynamicky se měnící prostředí.

<b>GFS (nastavení a paměť FD bloku)</b>
<i>Nwin, Nsh, Nfft, Nb, Niir, fs, tstep(=1/FFD), q_m, db_min, fd_p_max, fd_a_min, fs_enable, iir_fft_enable</i> last_used = index posledního použitého filtru v kaskádě
kaskáda IIR notch filtrů
f0 = index střední frekvence <0;Nfft> (vektor Niir) fb = šířka pásma <0;Nfft> (vektor Niir) dep_db = útlum na krajích pásma [dB] (vektor Niir) lvl = level (vektor Niir) t = čas filtru [tstep] (vektor Niir) on = aktivace filtru (vektor Niir)
<i>(pouze pro IIR notch verzi FS bloku)</i>
bb = b koeficienty IIR filtru (matice Niir x 3) aa = a koeficienty IIR filtru (matice Niir x 3)
<i>(pouze pro FFT OLS verzi FS bloku)</i>
H = aktuální amplitudová přenosová funkce filtru (vektor Nfft)
nastavení časování filtrů
lvl_tmout = časy aktivace pro různé levely filtru (vektor Nlvl) permdis_tmout = timeout pro permanentní deaktivaci filtru
gain = nastavení interního zesilovače na výstupu FS bloku mute = vypne signál na výstupu FS bloku fd_enable = může zastavit FD blok a "zmrazit" nastavení FS b.

**Tab. 3: GFS - globální parametry a paměť stavu filtrů**

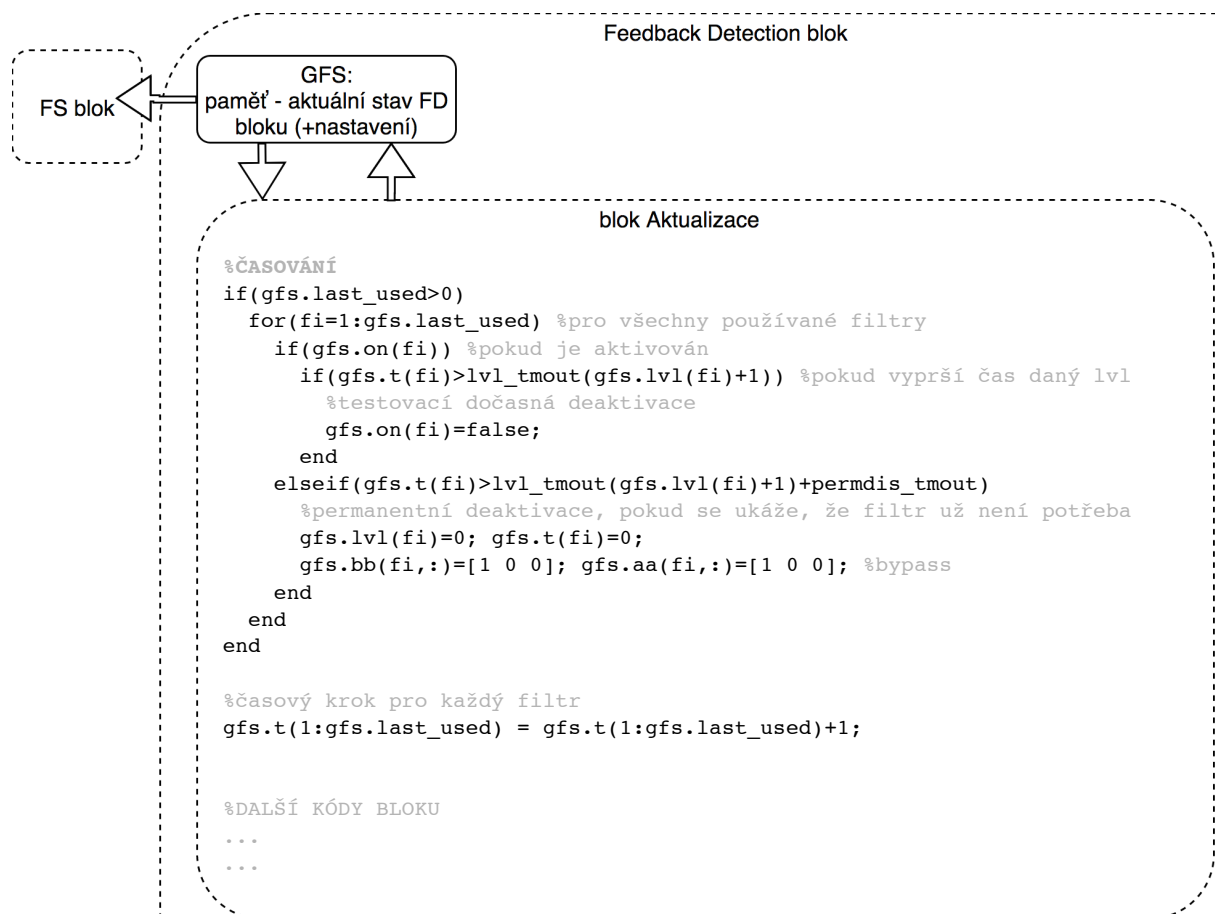
V tab. 3 je kompletní obsah GFS paměti, kde již dříve vysvětlené globální parametry algoritmu jsou v prvním řádku, a pod tím jsou parametry pro  $N_{iir}$  počet filtrů. Algoritmus se snaží pomocí *last\_used* pohlídat, aby alespoň poslední slot na IIR filtr byl k dispozici. Každý filtr může být ve stavu „volný“, „deaktivovaný“ nebo „aktivovaný“. Používané filtry jsou na začátku a volné filtry na konci, hodnota *last\_used* udává hranici mezi použitými a volnými filtry. Každý použitý filtr má parametr *on*, který pomocí true/false udává, zda je filtr aktivovaný. Deaktivovaný filtr uchovává všechny své parametry, pouze koeficienty IIR filtru ve FS bloku se nastaví na  $bb=\{1,0,0\}$  a  $aa=\{1,0,0\}$ , což způsobí, že daný filtr nijak nemění procházející signál.

Algoritmus omezovače je při implementaci na DSP rozšířen o fázi ověřování filtrů. Každý FS blokem vyhodnocený kandidát frekvence, označené jako nestabilní frekvence jevu akustické zpětné vazby, musí projít fází ověřování, než je příslušný filtr opravdu aktivován. Ověřování probíhá podle vývojového diagramu na obr. 8.



Obr. 8: Vývojový diagram ověřování kandidáta na potlačení frekvence v bloku Aktualizace

Pro popis časování filtrů v kaskádě je použit komentovaný zdrojový kód v obr. 9. Pro tento blok je vhodnější než slovní popis. Místo kódu v C, který je syntakticky složitější, je použit ekvivalentní kód z implementace algoritmu v MATLABu. Tento kus kódu se v algoritmu provádí vždy v intervalu  $t_{step} = 1/f_{FD}$ .



Obr. 9: Popis časování v bloku Aktualizace s pomocí komentovaného zdrojového kódu z MATLABu

Implementace algoritmu omezovače na DSP je ekvivalentem implementace v MATLABu, kód pro DSP byl vytvořen podle předlohy algoritmu z MATLABu. I pozdější úpravy a vylepšení algoritmu



jsou principiálně provedeny v kódu pro DSP i pro MATLAB. Obě implementace se liší pouze v detailech souvisejících s rozdílnou platformou a nutnými optimalizacemi v kódu pro DSP. Pouze kód zajišťující simulaci a běh algoritmu v MATLABu v kódu pro DSP není. Místo toho byla pro algoritmus omezovače na DSP naprogramována aplikace, která se stará o správný běh desky vývojového přípravku s DSP, všech používaných periférií a poskytuje rozhraní pro nastavení parametrů algoritmu uživateli. Algoritmus omezovače lze považovat za „jádro“ této aplikace. Pro účely testování implementace na DSP přidává do GFS paměti FD bloku (v tab. 3) proměnné *gain*, *mute* a *fd\_enable* a na výstupu FS bloku je přidán virtuální zesilovač, umožňující zesílení, zeslabení nebo úplně vypnutí signálu procházejícím FS blokem. Tento rozšiřující zesilovač lze ovládat skrze aplikaci, ve které algoritmus běží. Proměnnou *fd\_enable* lze vypnout FD blok a tím „zmrazit“ stav FS bloku, aniž by se zastavilo zpracování signálu. Lze toho využít pro účely testování algoritmu.

## 2.6.1 Optimalizace v C kódu algoritmu – využití prostředků DSP

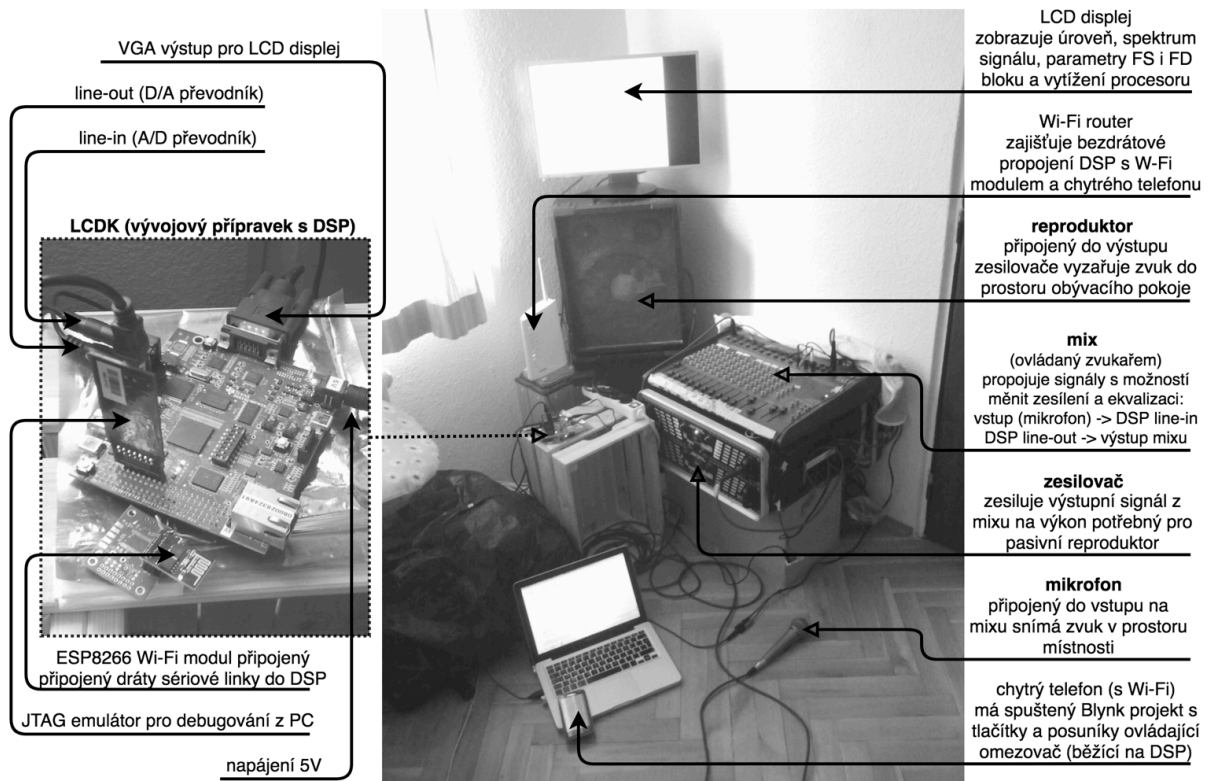
Po dokončení přepisu algoritmu z jazyku MATLAB do C algoritmus nestíhal provést všechny výpočty tak rychle, aby stihl zpracovávat signál v reálném čase. Aby nemuselo dojít ke zmenšení některých parametrů ovlivňujících výpočetní náročnost algoritmu, což by znemožnilo správně porovnat běh algoritmu v simulaci MATLABu s reálným prostředím s DSP, bylo třeba přepsat některé části kódu na efektivnější formu. Velkou úsporu výpočtu v algoritmu přineslo přeorganizování paměti tak, aby proměnné a pole využívané v kritických částech kódu byly umístěny v interní paměti. Interní paměť má velmi omezenou kapacitu a bylo třeba zkoušet, které proměnné zrychlí výpočet a které ne. Kritické se ukázalo dělení v operaci výpočtu  $q$ , které se počítá v každém cyklu FD bloku, kde pro každou frekvenci spektra je počítáno  $N_b-1$  kvocientů  $q$ , tj.  $f_{FD} \cdot (N_{fft}/2) \cdot (N_b-1)$  krát za sekundu. Pomocí uchovávání mezivýsledků pro každou frekvenci spektra lze počet dělení snížit na  $f_{FD} \cdot (N_{fft}/2)$ . Mnoho dalších detailů v kódu bylo dále zoptimalizováno a povedlo se rozběhat algoritmus v reálném čase i s téměř plným nastavením parametrů algoritmu. Zde je výstup modulu aplikace, který počítá vytížení procesoru jednotlivými částmi programu:

```
ReSupp vR0.3beta fs=48.0kHz FFT_resolution=46.88Hz
---- Nwin=1000,Nsh=500(=10ms),Nfft=1024 ----
---- memory: L2 194kB(73%), DDR 4829kB(3%) ----
Profiling statistics (300 cycles): 16.696% free CPU time
mixing signals: 0.058%
ReSuppIIRFilter() - FS block: 28.016%
real -> complex signal: 2.381%
CPU meter + level meter: 3.110%
DSPF_sp_fftSPxSP() FFT: 14.630%
complex -> power spectrum: 0.525%
filters, scaling: 0.396%
power -> ampl.spectr.: 2.945%
ReSuppActualizeIIRBank():
- feedback detection: 10.792%
- sort fd_ar[]: 0.164%
- IIR cascade actualize: 0.022%
- timeouts: 0.017%
- recyclation: 0.013%
ReSuppDrawInfo() - screen output: 20.093%
==== cycle_total:4735908c(456MHz)->10ms ====
```

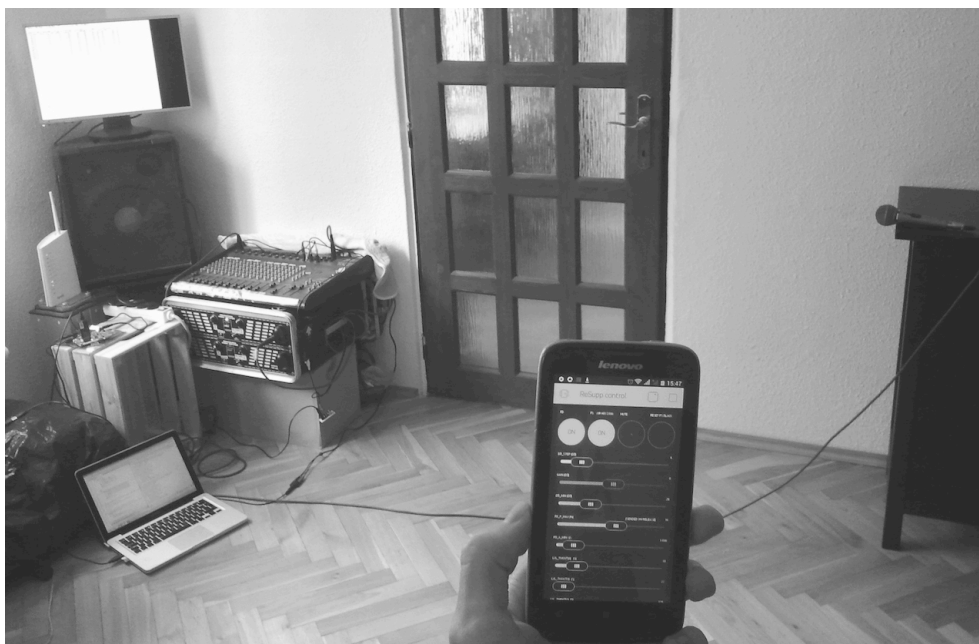
Pro běh algoritmu v reálném čase s běžným nastavením parametrů na vývojovém přípravku LCDK [8] s DSP jádrem C6748 [11] od firmy Texas Instruments je potřeba pro FS blok 28 % a FD blok 32 % a podpůrné aplikace 23 %, celkem 83 % výkonu procesoru při taktování 456 MHz. Paměť, kterou program zabírá, je 194 kB interní paměti DSP a 4,72 MB externí paměti DDR na desce, z toho celkem 2% je program (91 kB) umístěn celý do interní paměti a zbytek 98% data, jejichž velikost závisí na hodnotách některých parametrů algoritmu ( $N_{win}$ ,  $N_{fft}$ , aj.).

## 2.6.2 Popis realizovaného systému

Pro odladění algoritmu a pro experimenty v reálných podmínkách byla sestavena aparatura popsaná a vyfocena na obr. 10 a 11.



Obr. 10: Sestava reálné testovací aparatury pro algoritmus omezovače implementovaný na DSP. Sestava přesně odpovídá schématu z obr. 1 a 2.



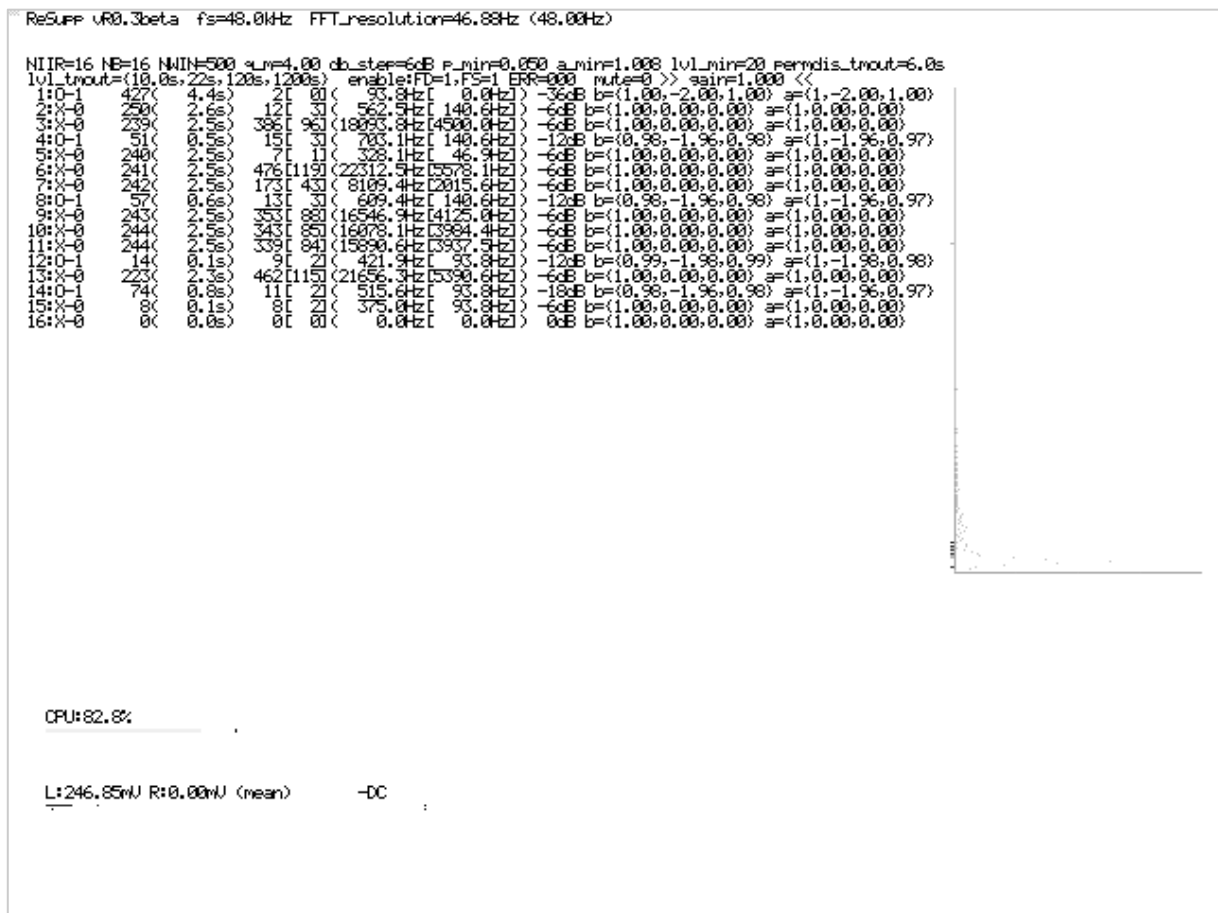
Obr. 11: Sestava reálné testovací aparatury vyfocena v průběhu testování. Vzdálenost mezi mikrofonom a reproduktorem 2 m. Objem místnosti 72 m<sup>2</sup>.

Přesný popis použité elektronické aparatury: dynamický mikrofon Shure 588SD připojený XLR symetrickým kabelem, mixážní pult Peavey Unity Series 1002, zesilovač McTaato Nightline Pro 400 připojený jack kabelem, velký pasivní reproduktor 8  $\Omega$  připojený nestíněnou dvoulinkou, deska vývojového přípravku OMAP-L138 LCDK od Texas Instruments obsahující číslicový signálový procesor s jádrem C6748 připojený dvěma jack 3,5 mm kabely do konektorů AIC31 kodeku na desce. Další použitá elektronika nijak nepracuje se zvukem ani signálem, ale podílí se na fungování sestavy jiným způsobem. Pro nahrání a debugování kódu omezovače na DSP slouží počítač připojený USB kabelem k JTAG emulátoru XDS100 zasunutému do desky vývojového přípravku. Elektronika potřebná k dálkovému nastavování omezovače se skládá z chytrého telefonu s operačním systémem Android, ESP8266 Wi-Fi modulu a šikovného Blynk software.

### 2.6.3 Aplikace a její uživatelské rozhraní

Pro efektivní zkoušení a ladění algoritmu omezování akustické zpětné vazby byla vytvořena uživatelská aplikace ReSupp. Díky této aplikaci je možné sledovat na LCD obrazovce různé údaje a měnit některé parametry algoritmu v reálném čase. Aplikace umožňuje deaktivovat některé části algoritmu nebo přepínat mezi variantami některých částí, aniž by bylo třeba upravit kód, zkompilovat algoritmus a program znovu nahrát do desky s DSP. Aplikace umožňuje interakci s uživatelem pomocí výstupu LCD obrazovky, 4 LED, 2 tlačítek a 4 přepínačů na desce vývojového přípravku s DSP a Wi-Fi rozhraní umožňující vytvořit různorodé ovládací prvky na displeji mobilního telefonu.

Na obr. 12 je zachycena ukázka LCD obrazovky s grafickými výstupy aplikace. Vlevo nahoře je nápis s názvem aplikace a její verzí, který se zobrazí ihned po zapnutí desky s DSP. Následuje údaj o vzorkovací frekvenci AD/DA převodníku a rozlišení algoritmu dosažené při výpočtu interpolovaného frekvenčního spektra, v závorce neinterpolovaného. V prostřední části jsou údaje o aktuálním stavu kaskády IIR notch filtrů a v prvních dvou řádcích jsou uvedeny aktuální hodnoty všech nastavitelných parametrů algoritmu. Každý filtr v kaskádě má jeden řádek, na kterém je písmeno X, pokud je filtr deaktivován, a za pomlčkou je aktuální level filtru, čas od poslední změny filtru v počtu segmentů  $i$  v sekundách, indexy střední frekvence a šířky pásma  $i$  s převodem na Hz, útlum filtru a aktuální koeficienty IIR filtru. Vpravo je zobrazeno aktuální amplitudové spektrum vstupního signálu FD bloku, kde lineární frekvenční osa je svisle s rostoucí frekvencí směrem nahoru a lineární amplitudová osa je vodorovně. Pod svislou frekvenční osou lze najít čárky odpovídající středním frekvencím jednotlivých filtrů v kaskádě, kde deaktivované filtry mají světlejší barvu. Ve spodní části obrazovky lze najít údaj o aktuálním vytížení procesoru, které záleží na nastavených parametrech algoritmu. Pokud by výpočet byl příliš náročný a vytížení by dosahovalo 100%, může docházet k vynechávání segmentů signálu a zasekávání zvuku procházejícího FS blokem. Takovému stavu je třeba se vždy vyhnout, protože aplikace neběží v žádném operačním systému. Řešením by bylo vytvořit pro FS blok a FD blok oddělená programová vlákna, ale cílem této práce není dokonalý kód programu, ale spíše vyvinout a implementovat algoritmus potlačení akustické zpětné vazby. Poslední údaj na obrazovce indikuje aktuální úroveň signálu na vstupu desky s DSP, přesněji na vstupu AD převodníku. Tento údaj je velmi užitečný pro správné nastavení aparatury, do které je deska s DSP připojena. Pro algoritmus omezovače byl použit pouze levý kanál, pravý kanál prochází na výstup desky s DSP nezměněn. Úroveň signálu je počítána průběžně jako aritmetický průměr vzorků každého segmentu signálu a tečka pohybující se nad vodorovným sloupcem indikátoru značí maximální hodnotu. Pokud maximální hodnota signálu dosáhne maximální hodnoty AD převodníku, na konci ukazatele úrovně se rozsvítí výstražný indikátor přebuzení vstupu (a zároveň výstražná LED na desce s DSP) a zůstane svítit. AD převodník na desce nemá žádné ochrany proti přebuzení, a proto je třeba dbát na to, aby se tento indikátor během práce s vývojovým přípravkem nikdy nerozsvítil.

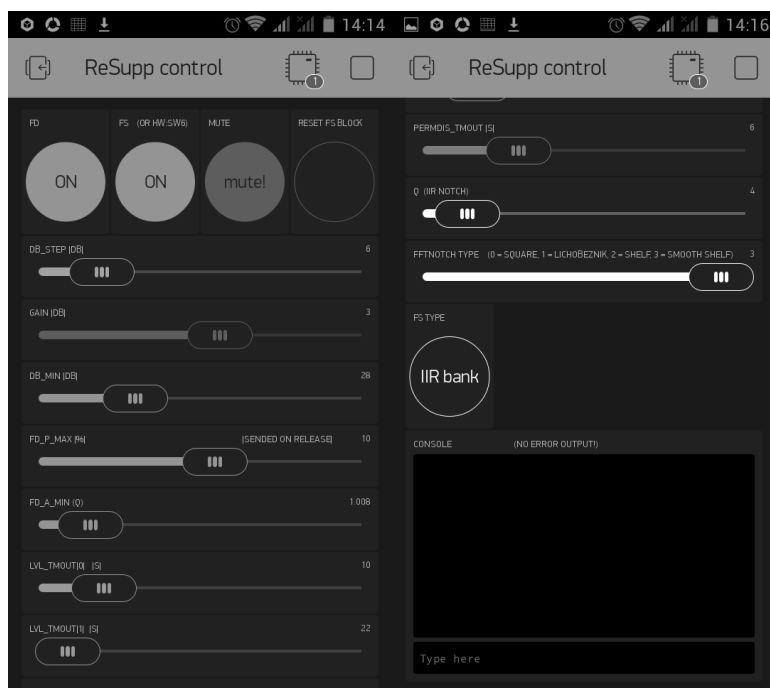


Obr. 12: Výstupní obrazovka aplikace vyfocená za běhu

## Ovládání a nastavení parametrů přes chytrý telefon

Během testování DSP implementace omezovače se velmi hodilo uživatelské rozhraní, které umí přes Wi-Fi měnit parametry algoritmu v reálném čase a tím efektivně testovat a sledovat chování algoritmu, aniž by musel debugger zastavit běh programu v DSP. Celý systém ovládání začíná Android aplikací Blynk na displeji chytrého telefonu, kde je vytvořen projekt s tlačítky. Projekt Blynk komunikuje přes Wi-Fi bezdrátové spojení s ESP8266 Wi-Fi modulem. Wi-Fi modul obsahuje firmware NodeMCU, podporující skriptovací jazyk Lua. V Lua skriptu nahraném na Wi-Fi modulu je kód, který používá knihovnu Blynk a převádí povely z chytrého telefonu na piny sériové linky UART. Wi-Fi modul je připojen přes UART do desky s DSP [8] [10] [11], kde kód aplikace s algoritmem omezovače přijímá povely a nastavuje odpovídající parametry do GFS paměti algoritmu.

Na dotykovém displeji chytrého telefonu lze během testování algoritmu měnit i sledovat aktuální hodnotu většiny parametrů algoritmu omezovače, viz obr. 13. Jednoduše lze testovat chování algoritmu při různé kombinaci všech parametrů a subjektivně porovnávat chování algoritmu při různém nastavení. Jednoduše lze aktivovat nebo deaktivovat FS blok i FD blok algoritmu, pomocí tlačítka „mute“ vypnout signál a rozpojit tím případně elektroakustickou smyčku, která se dostala mimo kontrolu. Pomocí softwarového tlačítka „reset“ je možné vypnout a vynulovat všechny filtry vytvořené algoritmem a tím kdykoliv v průběhu testování dostat algoritmus do původního stavu (před spuštěním) při současném uchování změn v parametrech algoritmu.



Obr. 13: Dva screenshoty Android aplikace Blynk na chytrém telefonu, která slouží jako bezdrátové (Wi-Fi) uživatelské rozhraní pro aplikaci ReSupp a algoritmus implementovaný na DSP

### 3 Zhodnocení a výsledky testování algoritmu

Pro testování algoritmu v MATLABu lze v simulaci vytvořit několikrát po sobě naprosto stejné situace výskytu jevu akustické zpětné vazby. Implementace v MATLABu proto nabízí větší možnosti objektivního posouzení algoritmu. Na druhou stranu simulace nezahrnuje plynulé změny akustického prostředí zpětné vazby a testování na dynamické změny ve zpětnovazební smyčce je lepší provádět s mikrofonom v ruce, tedy využít implementaci na DSP, ale zhodnocení takového testování bude spíše subjektivní.

Akustická zpětná vazba je jev, u kterého se poměrně těžko objektivně hodnotí snahy o jeho potlačení. Jediné jasné kritérium je, zda pískání vzroste do maximální úrovně nebo nevzroste. Protože metoda Auto-Notching jev potlačuje až ve chvíli, kdy se objeví, nemá příliš velký smysl hodnotit kritérium zda se jev objevil nebo neobjevil, případně zda byl slyšet nebo nebyl. Poměrně hůře definovatelná kritéria můžeme najít v oblasti zkreslení signálu způsobené omezovačem, doby reakce na změny ve zpětnovazební smyčce, času mezi vznikem slyšitelného pískání a jeho úplného potlačení, hladině intenzity rušivého jevu oproti intenzitě užitečného signálu, schopnosti fungovat v různých odlišných prostředích, míry nastavitelnosti uživatelem, spolehlivosti, chybovosti nebo výpočetní náročnosti.

#### 3.1 Kritéria pro zhodnocení kvality omezovače

Základní vývoj pomocí MATLABu poskytl algoritmus omezovače, který je schopný v simulaci nestabilní smyčku vrátit zpět do stabilního stavu ve většině případů, kdy zisk převyšuje mez stability o řádově jednotky dB.

Jednoduché kritérium hodnocení úspěšnosti algoritmu spočívá v počtu dB – kolik můžeme zvětšit zisk v elektroakustické smyčce nad původní mez stability bez omezovače, aniž by smyčka

začala být nestabilní. V odborné literatuře se objevuje pojem MSG (Maximum Stable Gain) [14], který označuje kolik dB zvukařem přidaného zisku zbývá do meze stability smyčky. Úspěšnost algoritmu tedy odpovídá zvýšení hodnoty MSG. Pokud bychom však chtěli porovnat i vedlejší efekty, které algoritmus vytvoří, aby tohoto zvýšení MSG (označíme  $\Delta\text{MSG}$ ) dosáhl, je třeba zapojit do hodnocení i poslechovou kvalitu zvuku a jeho zkreslení, nejlépe využít objektivní i subjektivní hodnocení. Zkreslení signálu vzniklá FS blokem při menším  $\Delta\text{MSG}$  nejsou slyšitelná, při  $\Delta\text{MSG} > 4$  dB už zkreslení slyšet je. Ve srovnání s rušivými efekty samotné zpětnovazební smyčky a občasného zapískání pozdě utlumené zpětné vazby je zkreslení filtrem omezovače zanedbatelné.

Pro zhodnocení samotného zkreslení vytvářeného omezovačem poslouží poměr SNR [13]:

$$\text{SNR}_{dB} = 10 \log \left( \frac{P_x}{P_n} \right) = 10 \log \left( \frac{P_x}{P_{|y-x|}} \right) \quad (23)$$

kde  $P_x$  je výkon užitečného signálu,  $P_n$  je výkon šumu, který odpovídá výkonu zbytkového signálu po odečtení vstupního užitečného signálu z celkového výstupního signálu. Pro lepší zhodnocení u řečových signálů poslouží kepstrální vzdálenost [13]:

$$d_2 = 4,34 \sqrt{2 \cdot \sum_{k=1}^L (c_x[k] - c_y[k])^2} \quad [dB] \quad (24)$$

kde  $c_x$  a  $c_y$  jsou kepstrální koeficienty signálu vstupního a výstupního. Melfrekvenční kepstrální vzdálenost se počítá stejně, pouze je při výpočtu kepstra využito přepočítání koeficientů FFT spektra na koeficienty melovské banky filtrů [13]. Jako třetí zhodnocení poslouží PEAQ [4], pro který nemá smysl uvádět složitý vzorec napodobující lidské vnímání zvuku a tato práce se jeho výpočtem nijak nezabývá, pouze je použita knihovna<sup>1</sup> MATLAB a její výsledek pro porovnání rozdílu původního a zkresleného signálu. Knihovna PEAQ vyhodnotí rozdíl signálů podle stupnice ODG (Objective Difference Grade), která napodobuje stupnice používané pro subjektivní hodnocení:

- 0 = nevnímateľné
- 1 = vnímateľné, ale nerušivé
- 2 = mírně rušivé
- 3 = rušivé
- 4 = velmi rušivé

Tato rozdílová kritéria lze použít pouze pro měření zkreslení signálu při průchodu FS blokem. Aby dávalo měření zkreslení smysl, musí být prováděno na samostatném FS bloku, testovaném mimo celý algoritmus a mimo virtuální signálovou smyčku. Porovnání zdrojového signálu  $x[n]$  s jakýmkoliv signálem v simulované smyčce neměří zkreslení vytvořené algoritmem omezovače, ale zkreslení vzniklé zavedenou akustickou zpětnou vazbou. Použití SNR, kepstrální vzdálenosti nebo PEAQ na porovnání signálu  $y[n]$  nasimulovaného bez omezovače a  $y[n]$  nasimulovaného s omezovačem není vhodné. Takové posouzení lze provést spíše subjektivním hodnocením při poslechu výsledků. Hlavním kritériem pro zhodnocení kvality celého omezovače je zvýšení MSG vytvořené omezovačem.

## 3.2 Uspořádání simulace a dat pro testování

Při testování MATLAB verze algoritmu byly pro simulaci prostředí RIR použity tyto impulsní odezvy  $h_{rir}[n]$  (vzdálenost mezi mikrofonem a reproduktorem je uvedena v závorce):

<sup>1</sup> volně dostupné z: <http://www-mmsp.ece.mcgill.ca/Documents/Software/>

<sup>2</sup> volně dostupné z: <http://www.openairlib.net/auralizationdb>

1. Obývací pokoj – vlastní RIR (3m)
2. York Guildhall Council Chamber (12m)<sup>2</sup>
3. Sport Centre, University of York (10m)<sup>2</sup>

A tyto testovací zvuky  $x[n]$ :

1. Anglicky mluvící žena v anechoické místnosti<sup>3</sup>
2. Koncert vážné hudby nahrávaný po jednotlivých nástrojích v anechoické místnosti (W. A. Mozart: An aria of Donna Elvira from the opera Don Giovanni)<sup>4</sup>
3. Rocková skladba (Grant Duncan - Won't be leaving town)<sup>5</sup>
4. Několik nahrávek samostatných hudebních nástrojů v anechoické místnosti<sup>3</sup>

Pro výpočet hodnoty MSG simulovaného prostředí bez vloženého omezovače byla vytvořena funkce, která dokáže z RIR předem vypočítat teoretickou hodnotu MSG, aby bylo možné porovnávat simulace s různými RIR. Ukázka z vyhodnocování MSG je v následující kapitole na obr. 15.

Základní test v simulaci (podle obr. 7) byl proveden pomocí všech kombinací RIR, zdrojových signálů  $x[t]$  a parametru  $a2\_db$ . Simulovaný průběh zesílení  $a[n]$  byl nastaven pomocí parametrů  $a1\_db$ ,  $t_1$  a  $t_g$  (vysvětleny v tab. 2), čímž byl vytvořen průběh podobný průběhu na třetím grafu na obr. 14, pouze se schodem posunutým na prvních 200 ms. Ve výsledných zvukových signálech  $z[n]$  vzniklých během simulace bylo zhodnoceno, zda vzniklý jev akustické zpětné vazby byl po nějaké přechodné době potlačen nebo alespoň omezen na přijatelnou mez. Pro každou RIR byla vysledována hranice  $a2\_db$ , kdy algoritmus omezovače přestává fungovat, způsobem postupného opakování simulace se zvyšováním  $a2\_db$ .  $a1\_db$  bylo vždy nastaveno tak, aby počáteční MSG bylo rovno 1dB. Během simulace vždy dojde k zvýšení zisku na  $a2\_db$  a tím dojde k nestabilitě. Zisk nad mez stability ( $a2\_db - a1\_db - 1\text{dB}$ ) se rovná  $\Delta\text{MSG}$ , které vytvoří omezovač akustické zpětné vazby, pokud signálovou smyčkou dokáže vrátit zpět do stability.

### 3.3 Výstupní data po skončení simulace

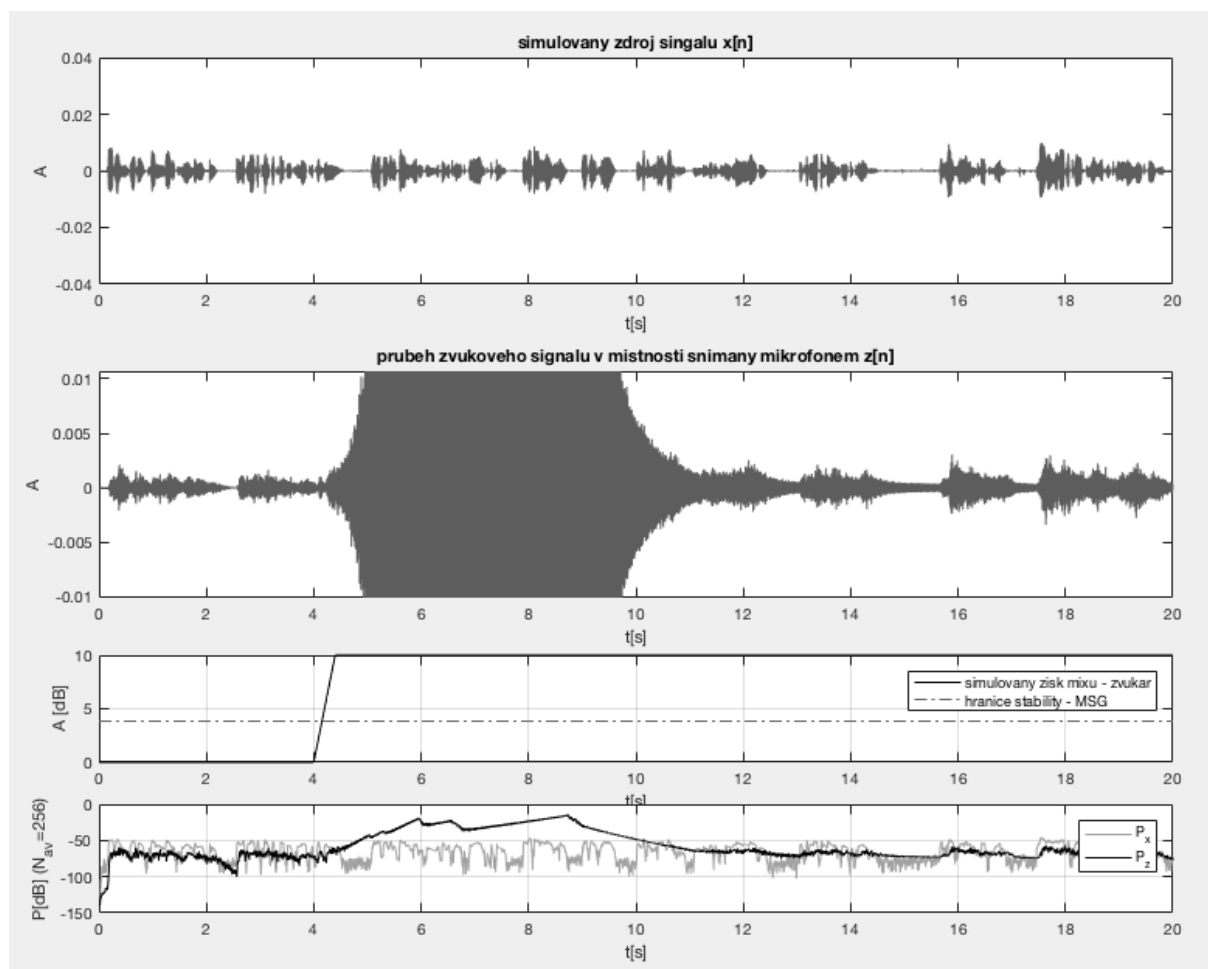
Po skočení každé simulace jsou zobrazeny výsledky, kde lze nalézt průběhy různých signálů po celou dobu simulace  $t_{sim}$ . Lze vidět, jak se měnily filtry FS bloku v průběhu simulace a průběžné mezivýsledky při detekování (před i po verifikaci) pískání nestabilní zpětné vazby FS bloku.

<sup>2</sup> volně dostupné z: <http://www.openairlib.net/auralizationdb>

<sup>3</sup> volně dostupné z: <http://www.openairlib.net/anechoicdb>

<sup>4</sup> volně dostupné z: <http://research.cs.aalto.fi/acoustics/virtual-acoustics/research/acoustic-measurement-and-analysis/85-anechoic-recordings.html#>

<sup>5</sup> volně dostupné z: [http://freemusicarchive.org/music/Grant\\_Duncan/Alliance/](http://freemusicarchive.org/music/Grant_Duncan/Alliance/)

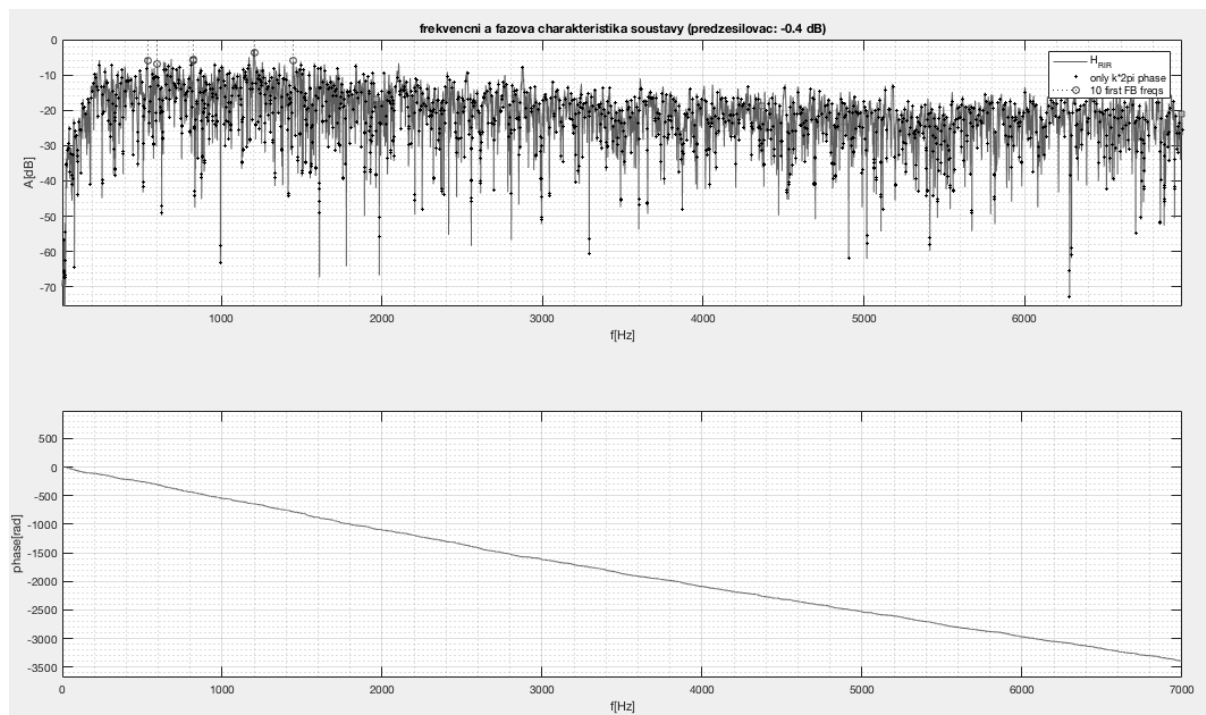


Obr. 14: Ukázka signálů  $x[n]$  výstupních dat MATLAB simulace. Parametry algoritmu:  $N_{win}=1000$ ,  $N_{sh}=500$ ,  $N_{fft}=1024$ ,  $N_{iir}=32$ ,  $N_b=16$ ,  $q_m=4$ ,  $db_{min}=-10$ dB,  $fd_{a_{min}}=1,01$ ,  $fd_{p_{max}}=5\%$ . Pro  $h[n]$  je použita RIR2 a pro vstupní signál  $x[n]$  zvuková nahrávka č. 1. Parametry simulace:  $t_{sim}=20$ s,  $x_m=1$ ,  $t_f=4$ s,  $t_g=0,4$ s,  $a_{ydb}=-0,4$ dB,  $a1_{db}=0$ dB,  $a2_{db}=10$ dB.

Na obr. 14 jsou zobrazeny signály z ukázkové simulace omezovače. Algoritmus zjevně po několika sekundách dokázal utlumit pískání a úspěšně vrátit systém do stability. Tato simulace prokázala teoretické  $\Delta MSG \geq 6$ dB, ale pískání způsobené dočasně nestabilní smyčkou očividně dosáhlo „rušivé“ intenzity pro posluchače. Ve čtvrté sekundě dochází kvůli změně zisku  $a[n]$  ve smyčce ke ztrátě stability, protože se zisk zvětšil nad hranici MSG o 6 dB. Z původních 4 dB MSG je potom systém na -6dB MSG. Bez omezovače by vznikala nežádoucí jevy akustické zpětné vazby a situace by dopadla přesně podle spektrogramu na obr. 18. Omezovač po nějaké době rozpozná exponenciální vzrůstání amplitudy několika nestabilních frekvencí, ty potlačí pomocí FS bloku a smyčka je díky tomu po deváté sekundě opět stabilní. V této ukázce doba reakce algoritmu byla dost dlouhá na to, aby pískání vzrostlo do „rušivé“ hlasitosti, ale přesto ještě nedošlo k jevu saturace. V grafu na obr. 14 není vidět celý rozsah signálu  $z[n]$ , protože je přiblížen tak, aby užitečná složka  $x[n]$  i  $z[n]$  byla vizuálně srovnatelná. Z průběhu výkonu na posledním grafu na obr. 14 je dobře vidět, že k saturaci nedošlo.

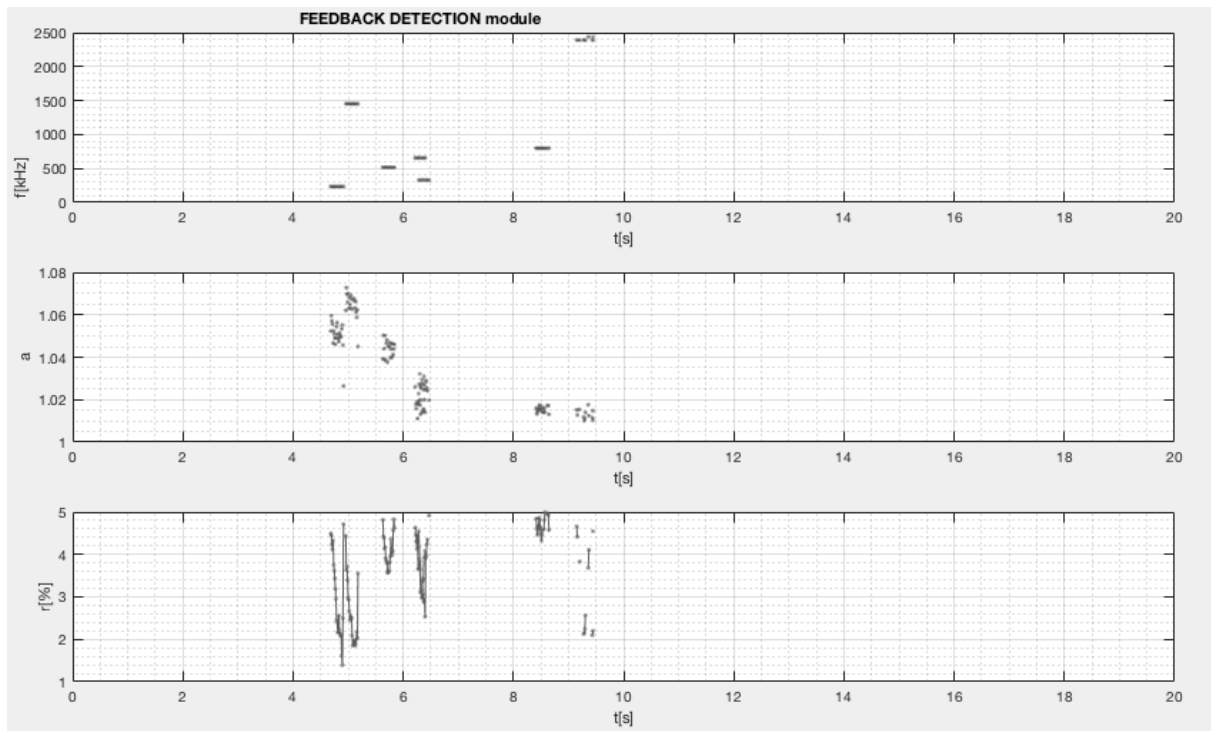
Aby bylo možné počítat teoretické  $\Delta MSG$  pro ověření, zda algoritmus danou simulovanou situaci zvládl (ověřit, že jeho opravdové  $\Delta MSG$  v dané situaci je větší nebo rovno teoretickému  $\Delta MSG$ ), je třeba znát MSG a provedenou změnu zisku ve smyčce  $a2_{db} - a1_{db}$ . Ukázka RIR č. 2 (ze seznamu v předchozí kapitole) převedená do frekvenční oblasti pro výpočet MSG, ke kterému je vztažen výpočet  $\Delta MSG$  ve všech simulacích s RIR2, je na obr. 15:





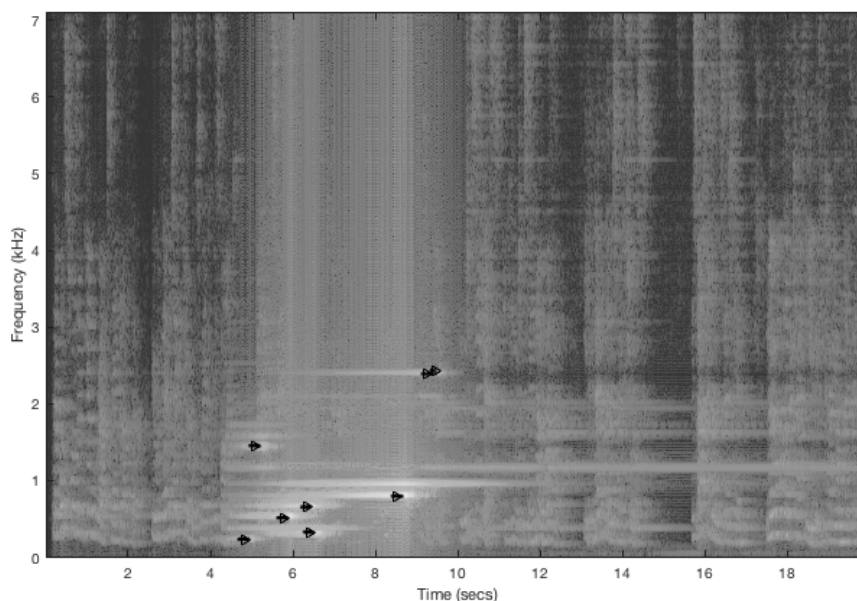
**Obr. 15:** Ukázka ze analýzy  $h_{rir}[n]$  ve spektrální oblasti. Kolečka značí několik největších hodnot amplitudy, které zároveň splňují kritérium fáze (5). Pokud se zisk ve smyčce zvětší natolik, že kolečkem označené maximum překročí hranici 0dB, budou splněny obě kritéria (5) i (6). Všechny kolečkem nebo tečkou označené frekvence budou pískat, pokud se zvednou nad 0dB. Z hodnoty největšího maxima označeného kolečkem lze vypočítat MSG systému.

Pomocí grafů na obr. 16 lze přehledněji ve fázi ladění algoritmu najít vhodné hodnoty parametrů  $fd\_p\_max$  a  $fd\_a\_min$ . Lze pozorovat všechny rozpoznané frekvence pískání zpětné vazby a jim odpovídající hodnoty průměrného  $q$  a  $p_{rd}$ . Jsou to pouze mezivýsledky. Algoritmus se na základě těchto hodnot, udávajících, na jakých frekvencích se aktuálně objevuje nežádoucí jev, s jakou pravděpodobností je to zpětná vazba a jak je rychlá, v budoucnu rozhodne zda zařadí nový filtr nebo změní již zařazený filtr FS bloku.

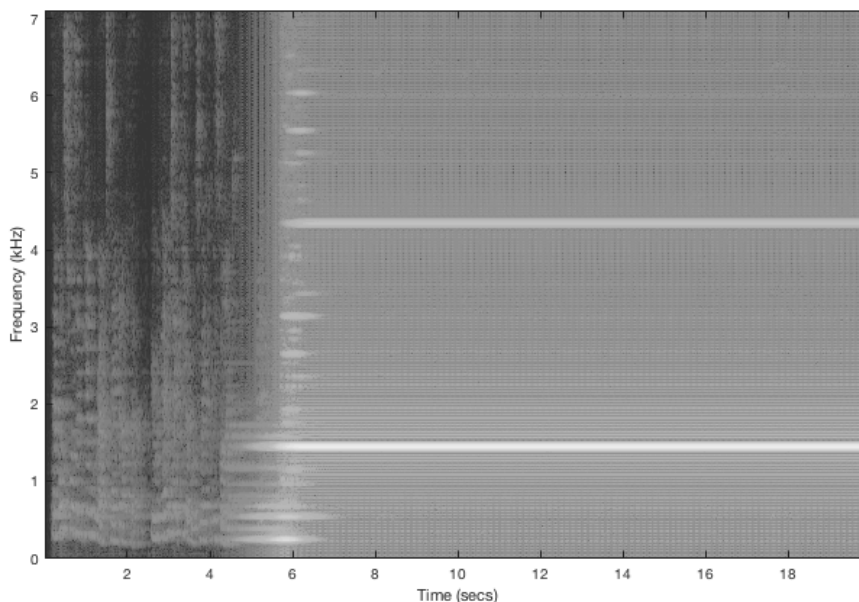


Obr. 16: Ukázka mezivýsledků FD bloku algoritmu. První graf zobrazuje všechny algoritmem rozpoznané frekvence v průběhu simulace, které splnily podmínky definované v (12), a těmto frekvencím přiřazené hodnoty – průměrný kvocient  $q$  v druhém grafu a relativní průměrná absolutní odchylka  $p_{rd}$  v třetím grafu. (Parametry simulace definovány pod obr. 14.)

Nejvíce průběh celé simulace demonstruje spektrogram signálu  $z[n]$  na obr. 17. Ve spektrogramu jsou vyznačena místa, kde v určitém čase na určité frekvenci došlo k vložení filtru omezovače. Signál  $z[n]$  demonstruje zvuk, který by slyšel divák v místnosti poslouchající zvuk vydávaný reproduktorem. V simulaci nelze oddělit  $H_{mic}$  z FIR filtru simulujícího prostředí i se všemi prvky ozvučení. Pokud je potřeba najít signál poslechu diváka v místnosti, musí se použít signál  $y_2[n]$  nebo  $z[n]$ .  $z[n]$  se velmi blíží hledanému signálu, protože  $H_{mic}$  v reálné situaci způsobuje pouze změnu hlasitosti bez výrazného vlivu na signál. Vliv bloku simulovaného prostředí na signál je způsoben hlavně  $H_{fb}$ .



Obr. 17: Spektrogram signálu  $z[n]$  s označením místa vložení filtrů pomocí šipek. (Parametry simulace definovány pod obr. 14.)



Obr. 18: Spektrogram signálu  $z[n]$  při vypnutém FD bloku, vzniklého ve stejné simulaci se stejnými parametry jako signál z předchozího obrázku. Je vidět, že vzniká nestabilní frekvence 1,45 kHz a během jedné sekundy dosáhne amplituda saturační intenzity a nelinearita vytváří liché vyšší harmonické frekvence. (Parametry simulace definovány pod obr. 14, bez omezovače.)

### 3.4 Výsledky testování a dosažené $\Delta$ MSG

Aby bylo možné posoudit velké množství proběhlých simulací s různými simulovanými situacemi, je třeba definovat objektivní rozlišování úspěšnosti omezovače. Experimentálně byla stanovena hranice nepřijemnosti zvuku na  $z[n] = 0,4$ . Pokud pro jakýkoliv vzorek signálu bude platit  $|z[n]| \geq 0,4$ , tak je daná simulace označena jako pouze „rušivá“. Pokud pro jakýkoliv vzorek v posledních 400ms signálu bude platit  $|z[n]| \geq 0,4$ , tak je daná simulace označena jako „nestabilní“. Lze tím pokrýt i úspěšné simulace, kdy omezovači trvá déle, než se s nestabilní situací vypořádá a mezi dobou vzniku nestability a dobou do znovuzískání stability dojde k tak velké intenzitě pískání, že by zdánlivě mohla být považována za neúspěch.

Algoritmus byl testován pomocí 96 simulovaných situací. Z celkového počtu situací bylo vypočítáno procento těch situací, které byly posouzeny jako úspěšné, tj. „stabilní“, a procento těch simulací, které byly posouzeny jako „nerušivé“. Byla vytvořena tab. 4 se statisticky zpracovanými výsledky, kde jsou vypsána procenta úspěšnosti pro situace, které by teoreticky požadovaly různé  $\Delta$ MSG. Proběhlé simulace měly ověřit, zda omezovač bude úspěšný v situacích vyžadujících určité  $\Delta$ MSG teoretického omezovače. Ukázka výsledných signálů vybraných simulací je na obr. 19.

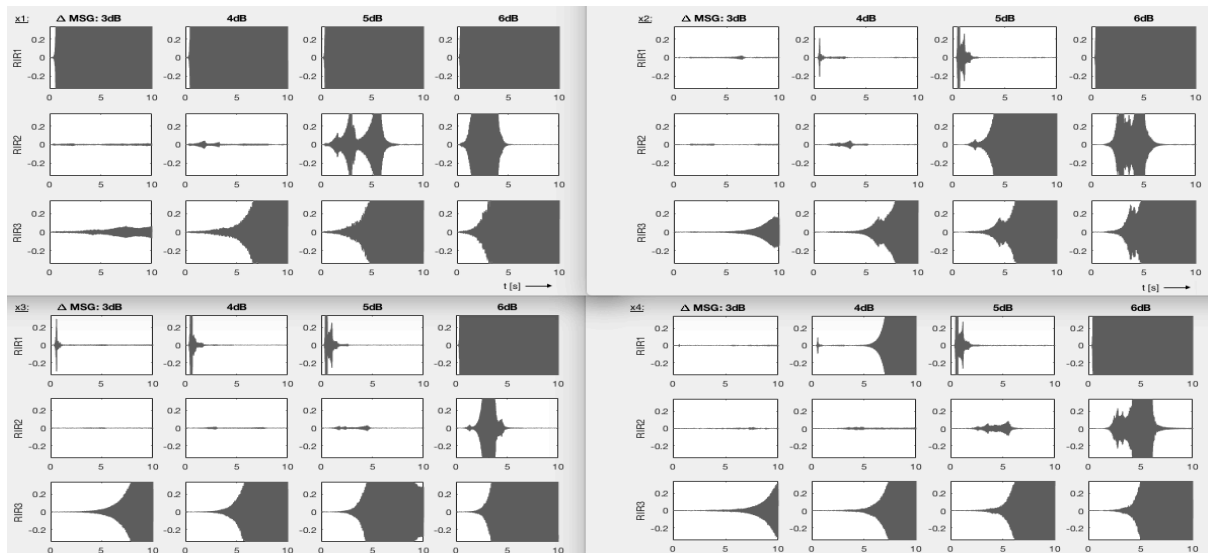
požadované $\Delta$ MSG		2 dB	3 dB	4 dB	5 dB	6 dB	7 dB	8 dB	9 dB
nerušivost	RIR1	100 %	75%	25 %	0 %	0 %	0 %	0 %	0 %
	RIR2	100 %	100 %	100 %	50 %	0 %	0 %	0 %	0 %
	RIR3	100 %	75 %	0 %	0 %	0 %	0 %	0 %	0 %
úspěšnost (stabilní)	RIR1	100 %	75 %	50 %	75 %	0 %	25 %	0 %	0 %
	RIR2	100 %	100 %	100 %	75 %	100 %	0 %	25 %	25 %
	RIR3	100 %	75%	0 %	25 %	0 %	0 %	0 %	0 %
	průměrná	100 %	83 %	50 %	58 %	33 %	8 %	8%	8 %

Tab. 4: Úspěšnost algoritmu pro různé stupně teoretického  $\Delta$ MSG. U nerušivosti je dané procento „nerušivých“ (a zároveň „stabilních“) z celkového počtu proběhlých simulací. U úspěšnosti je dané procento „stabilních“ (nehlédě na „rušivé“) z celkového počtu proběhlých simulací. Průměrování v posledním řádku je z 12 různých simulací.

Pro každou RIR lze v simulacích vysledovat přibližnou hranici  $\Delta$ MSG, kterou algoritmus omezovače ještě zvládne. Hranici nelze určit přesně, protože je závislá na konkrétním použitém vstupním signálu  $x[n]$ . V tab. 5 je vyhodnocení výsledků přibližných  $\Delta$ MSG ze všech simulací.

RIR č.	1	2	3	průměrná
dosažené $\Delta$ MSG	3 - 5 dB	5 – 6 dB	3 dB	4,2 dB

Tab. 5: Vyhodnocení dosažených  $\Delta$ MSG pro jednotlivá simulovaná prostředí



Obr. 19: Časové průběhy výstupních signálů  $z[n]$  ze simulací - vybrané kombinace vstupních signálů  $h[n]$ ,  $x[n]$  a zesílení  $a2\_db$  [dB] v simulacích (viz schéma na obr. 7, seznam vstupních signálů v kapitole 3.2 nebo popis parametrů v tab. 2).  $\Delta$ MSG je v tomto testu teoretická hodnota, kterou je třeba podle výsledného signálu ze simulace potvrdit nebo vyvrátit. Na obrázku není vidět celý rozsah signálů, ale jen do hodnoty  $\pm 0,4$  na ose y. Popisky x1 až x4 označují použitý vstupní signál.

### 3.5 Porovnání dvou verzí ekvalizace v algoritmu omezovače

Pro základní porovnání alternativní verze FS bloku (s OLS spektrální filtrací místo kaskády IIR notch filtrů) lze provést úplně stejný test, jako je popsán v předchozí kapitole 3.4 a porovnat tab. 5 s nově vypočítanou tab. 7, jediný rozdíl bude v nastavení simulace na `iir_fft_enable = true`.

Z porovnání tabulek lze usoudit, že alternativní verze FS bloku přináší lepší výsledky než původní verze.

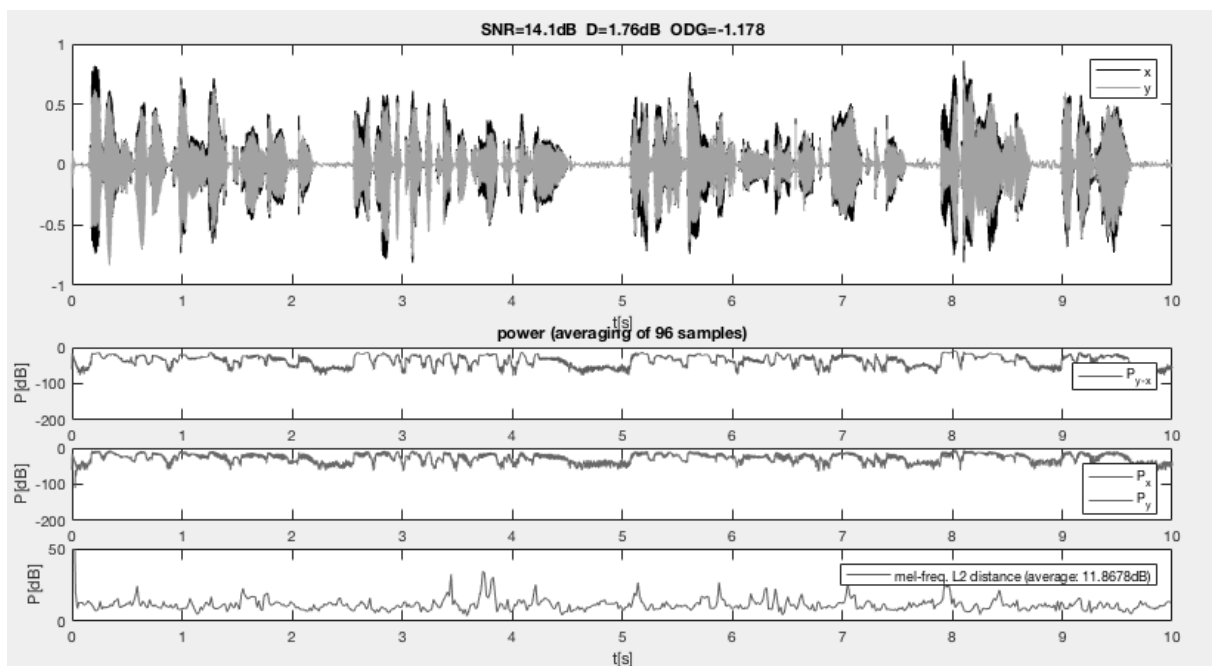
požadované $\Delta$ MSG	2 dB	3 dB	4 dB	5 dB	6 dB	7 dB	8 dB	9 dB
úspěšnost (průměrná)	100 %	92 %	92 %	67 %	58 %	17 %	17 %	0 %

Tab. 6: Úspěšnost algoritmu pro různé stupně teoretického  $\Delta$ MSG – FFT OLS verze FS bloku. Průměrováno je z 12 různých simulací.

RIR č.	1	2	3	průměrná
dosažené $\Delta$ MSG	5 – 8 dB	6 – 8 dB	3 – 4 dB	5,7 dB

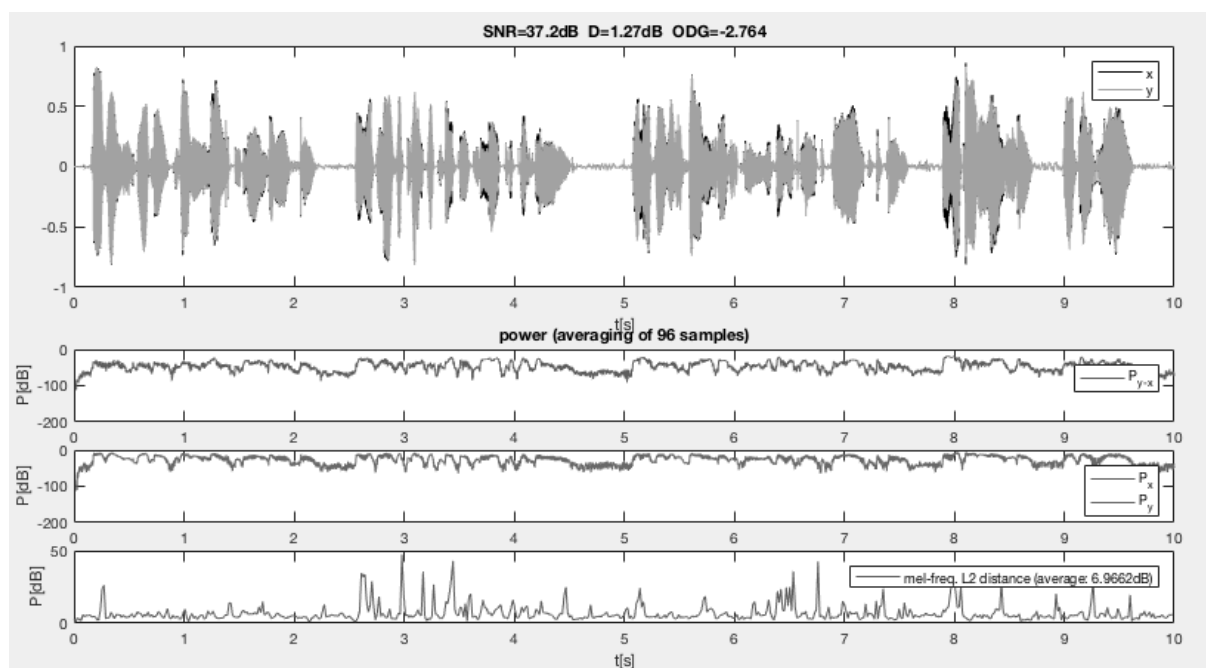
Tab. 7: Vyhodnocení dosažených  $\Delta$ MSG pro jednotlivá simulovaná prostředí - FFT OLS verze FS bloku.

Další porovnání obou verzí FS bloku lze provést pomocí speciálních skriptů `iir_benchmark.m` a `ols_benchmark.m`. Oba skripty slouží k testování zkreslení samostatného FS bloku z algoritmu. Není zde simulovaná žádná signálová smyčka, ani žádné prostředí, pouze je z \*.wav souboru načten vstupní signál  $x[n]$  a prohnán jednou z verzí FS bloku. Výstupní signál  $y_2[n]$  je porovnán s původním signálem  $x[n]$  a jsou vypočítány tři hodnoty objektivního hodnocení zkreslení – SNR, keprstrální vzdálenost a PEAQ. Hodnota SNR je základní měřítko v porovnávání signálů obecně, ale jeho výsledky zdaleka nemusí odpovídat subjektivnímu hodnocení při poslechu zvuků člověkem. Keprstrální vzdálenost je vhodnější měřítko, které se často používá při měření rozdílů mezi dvěma řečovými signály, ale pro hudební signály už není o moc lepší než SNR. PEAQ je složitý výpočet objektivního hodnocení zvukového signálu, který se snaží co nejvíce přiblížit subjektivnímu hodnocení zvukových signálů člověkem.



Obr. 20: Zobrazení výsledků měření zkreslení FS bloku s kaskádou IIR filtrů. Parametry algoritmu použity stejné jako na obr. 14. Horní graf zobrazuje průběh originálního signálu  $x[n]$  (použit je řečový signál x1) i zkresleného výstupního signálu  $y[n]$ . Nad grafem jsou uvedené výsledné hodnoty SNR, klasické

kepstrální vzdálenosti  $D$  i PEAQ hodnocení s ODG stupněm. Druhý graf zobrazuje výkon šumu, třetí graf výkon  $x[n]$  a  $y[n]$  a poslední graf zobrazuje průběh krátkodobé melfrekvenční kepstrální vzdálenosti.



Obr. 21: Zobrazení výsledků měření zkreslení alternativní verze FS bloku se spektrální filtrací OLS. Viz popis grafů předchozím obrázku.

Porovnané výsledky se z hlediska varianty kritéria velmi liší. V horní části obr. 20 a 21 lze najít konkrétní výsledné hodnoty pro jednotlivá kritéria. SNR kritérium pro OLS verzi vyhodnotilo výrazně menší zkreslení než u IIR verze FS bloku. Kepstrální vzdálenost pro OLS verzi vyšla horší než pro IIR verzi FS bloku. Podle nepercepčních objektivních kritérií se zdá, že OLS verze je lepší, percepční hodnocení PEAQ ale vyšlo úplně opačně. ODG stupeň pro OLS verzi odpovídá „vnímatelnému, ale nerušivému“ zkreslení signálu, pro IIR verzi odpovídá „rušivému“ zkreslení. Při subjektivním hodnocení poslechem signálů se potvrdilo objektivní hodnocení PEAQ. IIR verze FS bloku v signálu vytváří pouze velmi malé zvýraznění vyšších frekvencí, ale OLS verze do signálu přidává poměrně rušivé cvrkání o intenzitě úměrné aktuální intenzitě užitečného signálu.

Během testování s dvěma verzemi filtrace se objevily různé výhody a nevýhody. Pokud bude zapnut IIR notch filtr, bude vytvářet posun fáze (pro každou frekvenci jiný) a tím ovlivní celou zpětnovazební smyčku. Znamená to, že aktivace IIR filtrů pro potlačení některých frekvencí může sekundárně vyvolat menší změny frekvencí, které měly být původně potlačovány. OLS spektrální filtrace žádnou změnu fáze procházejícího nevytváří. Pouze nutnost blokového zpracování do signálu přidává konstantní zpoždění o  $N_{win}$  vzorků, které je neměnné během celého provozu algoritmu a pro všechny frekvence stejné. OLS spektrální filtrace oproti IIR notch filtrům však přidává rušivý šum, jehož výkon je závislý na velikosti útlumu nejhlubších notch filtrů (při filtraci ve spektru). Původ tohoto šumu se nepodařilo odhalit.

Lepší výsledky alternativní verze FS bloku při hodnocení dosažitelného  $\Delta$ MSG tak převáží velké sluchem vnímané zkreslení. Přestože by algoritmus omezovače s OLS verzí FS bloku zvládl i situace vyžadující větší  $\Delta$ MSG, než by zvládl algoritmus omezovače s IIR verzí FS bloku, musel by použít notch filtry s větším útlumem, což by velmi rušivě zkreslovalo zvukový signál. Alternativní verze FS

bloku tak algoritmu omezovače nepřinese obecně lepší výsledky, ale pouze potenciální výpočetní úsporu pro procesor.

Alternativní verzi FS bloku nebylo možné otestovat na reálném prostředí, protože nezbyl čas na dokončení potřebných optimalizací v C kódu, aby bylo možné OLS verzi FS bloku provozovat v reálném čase. Pokud by kód byl dokončen, bylo by možné zvětšit počet filtrů  $N_{iir}$  na mnohem větší počet než u kaskády IIR filtrů, protože výpočetní náročnost spektrální filtrace pomocí FFT není vůbec závislá na  $N_{iir}$ .

### 3.6 Zkušenosti z testování v reálných podmínkách

Nalezení optimálních hodnot pro parametry algoritmu je mnohem efektivnější na DSP implementaci algoritmu, protože lze operativně zkoušet reakce algoritmu při častém přenastavení parametrů. Simulace by velmi složitě napodobovaly podobné testování a navíc výpočet několika set simulací (popisovaném v kapitole 3.4) trval přibližně 5 dní na poměrně výkonném počítači.

Při testování funkce algoritmu s různými hodnotami parametrů se ukázalo, že ideální hodnota  $db\_min$  je závislá na sestavě ozvučení a běžných úrovních signálů používaných v konkrétní sestavě ozvučení. Při používání omezovače v praxi by tak bylo potřeba nastavit tuto hodnotu uživatelem nebo vytvořit na zařízení svítící ukazatel, který by poradil zvukaři jakou úroveň signálu má na mixu nastavit pro výstup do omezovače. Pokud by ke správné synchronizaci parametru  $db\_min$  s běžně používanou úrovní nedošlo, mohl by algoritmus přestat fungovat při velmi nízkých úrovních nebo vykazovat větší chybovost při úrovních příliš velkých a hrozilo by přebuzení AD převodníku.

Během testování se ukázalo, že vyvinutý algoritmus ve většině případů zvládne zpětnovazební smyčku udržet stabilní, pokud zisk ve smyčce nezvýšíme příliš moc. Pro dané ozvučení a místnost (obr. 11), ve kterém testování probíhalo, algoritmus dosahuje hodnot  $\Delta MSG$  6-10 dB, záleží na konkrétní poloze a natočení mikrofonu. Hodnota je určena experimentálně, pomocí několika testů v různých vzdálenostech mikrofonu a reproduktoru. Jako zdroj signálu slouží nahrávky  $x_1$  až  $x_3$  (uvedené v kapitole 3.2) pouštěné z malého reproduktoru 10 cm od mikrofonu. Testování bylo prováděno se stejnými hodnotami parametrů jako v simulaci. Pouze zisk ovládaný na mixážním pultu nebyl zvýšen podle simulovaného průběhu  $a[n]$  (v jednom schodu), ale postupně byl zisk zvyšován v intervalu 5 sekund o 1 dB, aby bylo možné rozpoznat hranici zisku, při které omezovač přestane správně fungovat. Hodnota  $\Delta MSG$  byla odečtena pomocí měřítka decibelů uvedeného u posuvníku, kterým se nastavuje zisk, a porovnána s hodnotou hranice stability při vypnutém omezovači. Experimentálně změřené  $\Delta MSG$  algoritmu v reálných podmínkách vychází překvapivě lepší, než výsledky měření ze simulací.

Byly provedeny i pokusy s výměnou některých částí aparatury a byl zkoumán vliv těchto změn na úspěšnost algoritmu. Tyto změny však neměly rozpoznatelný vliv na úspěšnost algoritmu. Místo dynamického mikrofonu originální sestavy (kapitola 2.6.2) s kardioidní směrovou charakteristikou byl vyzkoušen kondenzátorový mikrofón Samson C02 se superkardioidní směrovou charakteristikou. Dokonce byl vyzkoušen i scénář se třemi mikrofony zapnutými zároveň. V dalším scénáři byl nahrazen zesilovač a pasivní reproduktor originální sestavy (kapitola 2.6.2) menším aktivním reproduktorem, opět bez rozeznatelného ovlivnění funkčnosti omezovače.

Na trhu existuje několik komerčních automatických omezovačů zpětné vazby. Porovnat definovatelné výsledky omezovače vytvořeného v této práci s komerčními zařízeními bohužel nemohu, protože jsou poměrně drahé a nikdy jsem tato zařízení neměl ani nikde nezkoušel. Výrobci těchto komerčních zařízení neuvádějí žádnou hodnotu podobnou parametru  $\Delta MSG$ , recenze na

internetu neuvádí žádné konkrétní výsledky měření a z videí na YouTube lze o zhodnocení  $\Delta$ MSG komerčních zařízení pouze spekulovat. Je pravděpodobné, že algoritmus Auto-Notchingu v komerčních omezovačích je více vyladěn a propracován, než algoritmus Auto-Notchingu z této práce. Pravděpodobně nevytvářejí tolik šumu, jako vytváří DA převodník na použité desce s DSP, která je určena obecně pro vývoj zařízení k zpracování signálů, nikoliv na kvalitní zpracování zvuku. Ve chvílích ticha je tento šum slyšet a působí rušivě. Tento šum ale nijak nesouvisí s kvalitou algoritmu omezovače a stačilo by k DSP připojit kvalitnější externí zvukový DA převodník. Jediné srovnání, které lze obecně z dostupných videí posoudit, je, že omezovač vytvořený v rámci této práce je schopný zhruba napodobit chování komerčních omezovačů.



## Závěr

Cíle práce a úkoly z jejího zadání byly úspěšně splněny. V první kapitole jsou shrnuty teoretické poznatky o problematice potlačování akustické zpětné vazby. Důraz byl kladen na automatické metody. Na základě těchto informací byla vybrána metoda Auto-Notching (kapitola 1.4.4), která byla oproti dostupným informacím rozšířena o detaily pro implementaci (kapitola 2). Konkrétně byla vytvořena speciální kombinace tří upravených kritérií pro správnou identifikaci nežádoucího jevu akustické zpětné vazby. Pro algoritmus byl vymyšlen způsob verifikace identifikovaného jevu před zařazením příslušného filtru a nový modul algoritmu pro časování a recyklace nepotřebných filtrů. Nutnost modulu časování a recyklace se projevila až při realizaci algoritmu, proto nejspíš většina pramenů o metodách potlačování akustické zpětné vazby aspekt omezeného času filtrů v algoritmu ani nezmiňuje.

Vytvořený algoritmus omezovače akustické zpětné vazby není úplně bezchybný, ale rozhodně prokázal schopnost práce v reálném čase. Vytvořený omezovač umí potlačit vznikající pískání nestabilních frekvencí elektroakustické zpětnovazební smyčky pouze v méně náročných scénářích a při náročnějších situacích algoritmus nezvládá pískání ani po určité době odstranit (zjištěná hranice úspěšnosti omezovače v kapitole 3.4). Při středně náročných situacích dochází k slyšitelným až rušivým intenzitám pískání, které mohou trvat až několik sekund. Další krátké zapískání může následovat při krátkodobé deaktivaci filtru v rámci procesu verifikace, zda některý z aktivních filtrů je potřeba i nadále. Přesto lze vytvořený omezovač považovat za úspěšný, protože přechodné rušivé zvuky nejsou tak závažné, jako neřízené nepřetržité pískání zpětné vazby o maximální hlasitosti reproduktoru. Pokud by omezovač pracoval v podmínkách, kde je mikrofon i reproduktor upevněn a kde se akustika místnosti příliš nemění, lze omezovač použít při postupném pomalém zvyšování zisku k dosažení velmi dobrých výsledků (popisuje kapitola 3.6).

Při získávání zkušeností s číslicovým signálovým procesorem během dřívějších let studia na ČVUT se ukázalo, že vyvinout algoritmus a realizovat funkční ladičku hudebního nástroje zabere dva semestrální projekty i velkou část bakalářské práce. Proto je vývoj algoritmu automatického omezování akustické zpětné vazby a jeho realizace na signálovém procesoru uskutečněno až v této diplomové práci – po třech letech od první myšlenky. Předchozí zkušenosti s číslicovým signálovým procesorem byly nezbytné pro realizaci omezovače. Přesto vývoj algoritmu zabral první polovinu roku a realizace na signálovém procesoru druhou polovinu roku.

Kvůli náročnosti implementace nezbyl prostor na další testování algoritmu, který má ještě několik potenciálních možností zdokonalení. Kvůli velkému počtu parametrů ovlivňujících algoritmus by bylo třeba provést hlubší zkoumání, jak docílit správné kombinace hodnot parametrů, nebo dokonce nalézt vhodné presety hodnot pro různé situace nasazení omezovače nebo pro různé typy zvukových signálů. Do budoucna bych rád na vylepšování omezovače pokračoval. Tato práce mi poskytla poměrně dobrý základ k budoucímu vývoji zařízení omezovače a vlastní elektronické desky se signálovým procesorem. Je možné dále optimalizovat C kód realizovaného omezovače na signálovém procesoru a získat tak prostor pro přesnější spektrální analýzu, aniž by došlo k nepřijatelné latenci výstupního signálu. Je možné vyzkoušet jiné způsoby, jak implementovat dílčí bloky algoritmu, a provést větší počty simulací s širším pokrytím možných situací pomocí různých simulovaných testovacích dat. Rád bych v budoucnu vytvořil kompaktní zařízení, které by bylo možné použít na koncertech či jiných akcích. Experimenty s vytvořeným omezovačem na reálné zvukové aparatuře prokázaly potenciál omezovače k dotažení do fáze, kdy by byl schopný konkurovat komerčně prodávaným zařízením.

## Zkratky a termíny

ozvučení – takto se v praxi označuje ozvučovací aparatura, tedy všechna elektronika snímající, zpracovávající a reprodukuje zvukový signál

jev, zpětná vazba, vazbení – termíny vysvětluje kapitola 1.2

omezovač – algoritmus nebo implementace algoritmu potlačování akustické zpětné vazby

OLS – Overlap and Save, je metoda zpracování signálu po segmentech [7], použitá pro spektrální filtraci v této práci

mix – mixážní pult, používaný jako ústřední součást ozvučení, kde zvukař může nastavovat zesílení všech mikrofonů, ekvalizaci signálů a poměr při sloučení těchto signálů do reproduktoru

MSG – Maximum Stable Gain, jeho hodnota udává, jak moc lze v elektroakustické zpětnovazební smyčce zvýšit zesílení do hranice stability – kolik dB dalšího zesílení zvukaři zbývá do meze stability smyčky [14]

$\Delta$ MSG – hodnota vyjadřuje, o kolik dB se zvýší MSG použitím omezovače akustické zpětné vazby oproti MSG v této zpětnovazební smyčce bez omezovače

DSP – číslicový signálový procesor, v některých případech je tato zkratka použita pro celou desku vývojového přípravku s číslicovým signálovým procesorem, pomocnou elektronikou i konektory

LCDK – Low-Cost Development Kit, v překladu vývojový přípravek

SNR – Signal-to-Noise Ratio, odstup výkonu signálu od výkonu šumu v decibelové stupnici [13]

PEAQ – Perceptual Evaluation of Audio Quality [4]

ODG – Objective Difference Grade, je stupnice používaná pro souhrnný výsledek PEAQ s hodnotami 0 až -4, kde 0 odpovídá nevnímatelnému zkreslení zvukového signálu

FS – Feedback Suppression

FD – Feedback Detection

RIR – Room Impulse Response

## Použité zdroje

- [1] ÅSTRÖM, K. J., a WITTENMARK, B. *Computer-controlled systems: theory and design*. Upper Saddle River, Prentice Hall, 1997. ISBN 0-13-314899-8.
- [2] FRANKLIN, Gene F., POWELL, J. David aj. *Feedback control of dynamic systems*. Prentice Hall, PEARSON. 2009, ISBN-13: 978-0-13-601969-5, ISBN-10:0-13-601969-2.
- [3] GREEN, M. C., J. SZYMANSKI a M. SPEED. 'Assessing the suitability of the magnitude slope deviation detection criterion for use in automatic acoustic feedback control', Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16). pp. 85-92, Sept. 2016.
- [4] ITU-R Rec. BS-1387-1: "Method for Objective Measurements of Perceived Audio Quality", International Telecommunications Union, Geneva, Switzerland (2001)
- [5] POLICARDI, F. *MLS and Sine-Sweep Technique Comparison in Room-Acoustic Measurements*, Elektrotehniski Vestnik, vol. 78, pp. 91-95, 2011.
- [6] REKTORYS, Karel a spol. *Přehled užití matematiky I*. Vyd. 7. Prometheus, spol. s.r.o. Praha, 2003. ISBN 80-7196-180-9
- [7] SOVKA Pavel, Petr POLLÁK. *Vybrané metody číslicového zpracování signálů*, Vyd. 2. přeprac. Praha: Vydavatelství ČVUT, 2003. ISBN 80-01-02821-6
- [8] TEXAS INSTRUMENTS. L138/C6748 Development Kit (LCDK). In: *Texas Instruments Wiki* [online]. poslední aktualizace 17. 2. 2016 [cit. 2016-05-02]. Dostupné z: [http://processors.wiki.ti.com/index.php/L138/C6748\\_Development\\_Kit\\_\(LCDK\)](http://processors.wiki.ti.com/index.php/L138/C6748_Development_Kit_(LCDK))
- [9] TEXAS INSTRUMENTS. *TMS320C67x DSP Library Programmer's Reference Guide* [online]. Rev. C. 31 Jan 2010. Dostupné z: <http://www.ti.com/lit/ug/spru657c/spru657c.pdf>
- [10] TEXAS INSTRUMENTS. StarterWare 01.10.01.01 User Guide. In: *Texas Instruments Wiki* [online]. poslední aktualizace 2. 5. 2013 [cit. 2016-05-02]. Dostupné z: [http://processors.wiki.ti.com/index.php/StarterWare\\_01.10.01.01\\_User\\_Guide](http://processors.wiki.ti.com/index.php/StarterWare_01.10.01.01_User_Guide)
- [11] TEXAS INSTRUMENTS. *OMAP-L138 DSP+ARM Processor Technical Reference Manual* [online]. Rev. A. 12 Dec 2011. Dostupné z: <http://www.ti.com/lit/ug/spruh77a/spruh77a.pdf>
- [12] TROXEL, Dana. *Understanding Acoustic Feedback & Suppressors* [online]. Dostupné z: <http://www.rane.com/note158.html>
- [13] UHLÍŘ Jan a Pavel SOVKA. *Číslicové zpracování signálů*. Vyd. 1. Praha: Vydavatelství ČVUT. 1995.
- [14] VAN WATERSCHOOT, T. a M. MOONEN. 'Fifty years of acoustic feedback control: State of the art and future challenges', Proc. IEEE, vol. 99, no. 2, pp. 288-327, Feb. 2011
- [15] VASEGHI, S. V. *Advanced Signal Processing and Digital Noise Reduction*. Wiley Teubner, 1997