

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2018

Denis Kramář



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

RESTAURACE SIGNÁLU PO PRŮCHODU LIMITEREM S POUŽITÍM PSYCHOAKUSTICKÉHO MODELU

RESTORATION OF SIGNALS AFTER PASSING THROUGH THE LIMITER WITH THE USE OF
PSYCHOACOUSTIC MODEL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Denis Kramář

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Závíška

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Audio inženýrství**
Ústav telekomunikací

Student: Denis Kramář

ID: 186622

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Restaurace signálu po průchodu limiterem s použitím psychoakustického modelu

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte znehodnocení zvukového signálu po průchodu omezovačem signálu, tedy jev nazývaný též jako clipping. Tento problém analyzujte a seznamte se s obvyklými metodami rekonstrukce takovýchto signálů. Dále se seznamte s metodami založenými na tzv. řídkých reprezentacích signálů.

Prostudujte také analyzující a syntezující model signálu a pro oba modely v prostředí MATLAB implementujte rekonstrukční algoritmy a rozšiřte je o váhování koeficientů podle absolutního prahu slyšení a následně do váhování zakomponujte efekt kmitočtového maskování. Výsledky rekonstrukce vyhodnoťte pomocí vhodných objektivních i subjektivních ukazatelů.

DOPORUČENÁ LITERATURA:

[1] Hrbáček, R., Rajmic, P., Veselý, V., Špiřík, J. Řídké reprezentace signálů: úvod do problematiky, Elektrevue, 2011. ISSN 1213-1539

[2] Defraene, B.; Mansour, N.; De Hertogh, S.; Van Waterschoot, T.; Diehl, M.; Moonen, M. Declipping of Audio Signals Using Perceptual Compressed Sensing. IEEE Transactions on Audio, Speech, and Language Processing [online]. 2016. s 2627–2637. Dostupné z URL: <<http://ieeexplore.ieee.org/document/6600777>>

Termín zadání: 5.2.2018

Termín odevzdání: 29.5.2018

Vedoucí práce: Ing. Pavel Závíška

Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce se zabývá využitím řídkých reprezentací signálu za účelem restaurace audiosignálu poškozeného clippingem. Nejprve je zde probrána teorie týkající se limiteru a samotné limitace signálu. Poté jsou zde uvedeny některé současné metody založené na teorii řídkých reprezentací. Ta je popsána v následující kapitole. Následně je zde popsán psychoakustický model a jeho využití pro declipping. Na závěr teoretické části jsou zde představeny dvě metody řešení této úlohy. První je založená na syntezujícím modelu signálu a využívá algoritmus Douglas-Rachford. Druhá je založená na analyzujícím modelu signálu a byl pro ni zvolen algoritmus Chambolle-Pock. V další části je popsána jejich implementace v prostředí Matlab. Na závěr jsou vyhodnoceny výsledky dosažené oběma algoritmy.

KLÍČOVÁ SLOVA

audiosignál, clipping, declipping, psychoakustický model, řídké reprezentace, řídkost, zpracování signálu, Douglas-Rachford, Chambolle-Pock

ABSTRACT

This Bachelor thesis deals with the use of sparse representations for the purpose of restoration clipping-damaged audiosignal. First, a theory of limiter and signal limiting itself is discussed. Subsequently, some of present methods based on sparse representations theory are given. The theory of sparse representations is discussed in following chapter. After that is here described a psychoacoustic model and its use for declipping. At the end of theoretical part, two methods dealing with this problem are introduced. First is based on synthesis model of signal using Douglas-Rachford algorithm. Second is based on analysis signal model using Chambolle-Pock algorithm. In the next part is their implementation in the Matlab environment. Finally, the result achieved by both algorithms are evaluated.

KEYWORDS

audio, clipping, declipping, psychoacoustic model, signal processing, sparse representations, sparsity, Douglas-Rachford, Chambolle-Pock

KRAMÁŘ, Denis. *Restaurace signálu po průchodu limiterem s použitím psychoakustického modelu*. Brno, 2018, 70 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Pavel Závíška

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Restaurace signálu po průchodu limitem s použitím psychoakustického modelu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto semestrálního projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Pavlu Závíškovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

podpis autora



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	10
1 Teoretický úvod	11
1.1 Limiter	11
1.1.1 Stavba limiteru	11
1.1.2 Využití limiteru	11
1.2 Clipping	12
1.2.1 Hard clipping	13
1.2.2 Soft clipping	14
2 Metody restaurace signálu	17
2.1 Declipping	17
2.2 Metody založené na řídkých reprezentacích	17
2.2.1 A Constrained Matching Pursuit to Audio Declipping	18
2.2.2 Recovering a Clipped Signal in Sparseland	18
2.2.3 Least Squares with Examples in Signal Processing	19
2.2.4 Consistent Iterative Hard Thresholding for Signal Declipping	19
2.2.5 Declipping of Audio Signals Using Perceptual Compressed Sensing	19
2.2.6 Audio Declipping with Social Sparsity	20
2.2.7 Audio Declipping by Cosparsity Hard Thresholding	21
2.2.8 Sparsity and Cosparsity for Audio Declipping	22
3 Úvod do řídkých reprezentací	23
3.1 Základní značení a znalosti	23
3.2 Normy vektorů	23
3.3 Řídká řešení	24
3.4 Báze a framy	27
3.5 Syntezující a analyzující model	28
3.6 Algoritmy pro hledání řídkých řešení	28
4 Psychoakustika	30
4.1 Absolutní práh slyšitelnosti	30
4.2 Kmitočtové maskování	30
4.3 Psychoakustický model	32
4.3.1 Váhování	32
4.3.2 Rozdělení tónových a šumových složek a výpočet jejich SPL	33
4.3.3 Výpočet maskovacího prahu	33
4.3.4 Decimace maskovacích složek	33

4.3.5	Výpočet poměru signálu k maskující složce	33
5	Návrh řešení	34
5.1	Declipping pomocí proximálních algoritmů	34
5.2	Návrh pro syntezující model	36
5.3	Návrh pro analyzující model	37
5.4	Projekce na množinu přípustných řešení	39
6	Programové řešení	41
6.1	Hlavní soubor	41
6.1.1	Načtené zvukového souboru	41
6.1.2	Nastavení parametrů	41
6.1.3	Hard clipping	42
6.1.4	Konstrukce framu	42
6.2	Funkce pro hard clipping	43
6.3	Váhovací funkce	44
6.4	Algoritmus Douglas-Rachford	44
6.5	Algoritmus Chambolle-Pock	45
6.6	Projekce na množinu přípustných řešení	45
7	Výsledky testování	46
7.1	Testování	46
7.1.1	Ukazatel Δ SNR	46
7.2	Výsledky rekonstrukce bez váhování	47
7.3	Výsledky restaurace s využitím váhovacích funkcí	49
7.3.1	Váhování pomocí křivky ATH	50
7.3.2	Váhování pomocí modelu MPEG	54
7.4	Shrnutí výsledků	56
7.5	Výsledky poslechových testů	59
7.5.1	Způsob testování	59
7.5.2	Výsledky poslechových testů	60
8	Závěr	63
	Literatura	65
	Seznam symbolů, veličin a zkratk	68
	Seznam příloh	69
A	Obsah přiloženého CD	70

SEZNAM OBRÁZKŮ

1.1	Blokové schéma běžného limiteru	12
1.2	Převodní charakteristika hard clippingu.	13
1.3	Vliv hard clippingu na reálný signál.	14
1.4	Převodní charakteristika soft clippingu.	15
1.5	Vliv soft clippingu na reálný signál.	16
3.1	Zvětšující se koule norem ℓ_0 , ℓ_1 a ℓ_2 dotýkající se nadroviny $\mathbf{Ax} = \mathbf{y}$	27
4.1	Kmitočtová charakteristika absolutního prahu slyšitelnosti	31
5.1	Převodní charakteristika – soft thresholding	35
5.2	Blokové schéma declippingu pomocí syntezujícího modelu	37
5.3	Blokové schéma declippingu pomocí analyzujícího modelu	39
7.1	Typický průběh hodnot ΔSNR v závislosti na počtu iterací.	47
7.2	Graf závislosti velikosti ℓ_1 -normy vektoru koeficientů na počtu iterací pro vzorek <code>acoustic_guitar.wav</code> a $\pm\theta_c = 0,6$	48
7.3	Hodnoty ukazatele ΔSNR pro algoritmus Douglas-Rachford.	49
7.4	Hodnoty ukazatele ΔSNR pro algoritmus Chambolle-Pock.	49
7.5	Průměr hodnot ΔSNR jednotlivých testovacích zvuků rekonstruovaných pomocí obou algoritmů.	50
7.6	Závislost výsledných hodnot ΔSNR na dynamickém rozsahu křivky ATH.	51
7.7	Porovnání změn ℓ_1 -normy vektoru koeficientů s využitím lineární normalizace a bez ní pro algoritmus Douglas-Rachford a Chambolle-Pock.	52
7.8	Frekvenční závislost normalizované křivky ATH převedené na hodnoty akustického tlaku [Pa].	52
7.9	Srovnání algoritmu Douglas-Rachford a Chambolle-Pock při váhování pomocí hodnot křivky ATH v decibelech a pascálech.	53
7.10	Frekvenční závislost normalizované maskovací křivky modelu MPEG-1.	54
7.11	Porovnání hodnot ΔSNR pro váhování pomocí maskovací křivky vytvořené z clipovaného signálu – MPEG a křivky vytvořené z dříve restaurovaného signálu – MPEG*.	55
7.12	Srovnání algoritmu Douglas-Rachford a Chambolle-Pock při váhování pomocí hodnot maskovací křivky vytvořené pomocí psychoakustického modelu MPEG-1.	57
7.13	Srovnání obou testovaných algoritmů a metod váhování podle ukazatele ΔSNR	58
7.14	Výsledky poslechových testů pro velké zkreslení při $\theta_c = 0,1$	60
7.15	Výsledky poslechových testů pro menší zkreslení při $\theta_c = 0,4$	61

ÚVOD

Při zpracovávání audiosignálu, ať už v analogové nebo digitální podobě, se mnohdy nevyhneme jistým poškozením nebo zkreslením. K tomu může docházet z mnoha důvodů a jedním z nich je poškození signálu při průchodu limiterem, tedy jev známý jako clipping. Tímto problémem a jeho řešením (declipping), se tato práce bude zabývat.

Existuje několik způsobů, jak takto poškozené signály restaurovat, aby se alespoň přiblížily původním signálům, protože úplná rekonstrukce signálu zatím není možná. V poslední době nejpoužívanějším přístupem je využití modelu řídkých reprezentací signálu.

V první kapitole je obecně popsán limiter a jeho funkce. Dále také dva druhy clippingu, jejich vliv na signál a případy, kdy k těmto limitacím dochází.

Kapitola druhá obsahuje výčet významných publikací, které se zabývají využitím řídkých reprezentací při declippingu. Je zde také stručně vysvětleno, v čem spočívají metody v těchto článcích uvedené.

Třetí kapitola obsahuje úvod do teorie řídkých reprezentací a základní značení. Jsou zde definovány vztahy, které jsou dále využity pro samotný declipping. Kapitola obsahuje i části popisující báze, syntezující a analyzující model signálu a také algoritmy pro hledání řídkých řešení, které jsou dále popsány v kapitole 5.

Čtvrtá kapitola je lehký úvod do problémů psychoakustiky. Je zde vysvětlen jev *kmitočtového maskování* a je zde popsán psychoakustický model, který bude využit při restauraci signálu.

Pátá kapitola obsahuje bližší seznámení s algoritmy používanými při declippingu s využitím řídkých reprezentací a jsou zde představeny řešení pro syntezující a analyzující model signálu. Pro každé z těchto řešení je v závěru navrženo blokové schéma.

V šesté kapitole je popsána samotná implementace těchto algoritmů v prostředí Matlab a jsou zde vysvětleny možnosti nastavení programu pro declipping.

V poslední kapitole jsou demonstrovány výsledky restaurace signálu pomocí představených algoritmů. Jsou zde porovnány výsledky různých nastavení programu pro declipping. Výsledné restaurované signály jsou porovnány pomocí hodnot ΔSNR a pomocí subjektivních poslechových testů.

1 TEORETICKÝ ÚVOD

V teoretickém úvodu popíšu stavbu limiteru a jeho funkci při zpracování audio signálu. Dále se budu zabývat jevem zvaným clipping, který vzniká při průchodu signálu limiterem. Popíšu dva druhy clippingu – tzv. soft clipping a hard clipping. Vysvětlení obou těchto jevů pro názornost doplním i grafickým zobrazením vlivu clippingu na amplitudu a frekvenční spektrum reálného signálu a porovnáám ho s jeho původními charakteristikami. Uvedu také případy, kdy k těmto limitacím dochází a k čemu se případně používají.

1.1 Limiter

Pojmem limitace se rozumí jakýkoliv proces, při kterém zabráníme překročení předem stanovené hodnoty dané charakteristiky (v našem případě amplitudy) na výstupu zařízení. Limiter je jedním z druhů kompresorů dynamiky s vysokou hodnotou kompresního poměru. V ideálním případě by zpracovávaný signál neměl přesáhnout nastavenou úroveň. Takto tvrdou limitací dochází neodvratně i k zkreslení vstupního signálu [1].

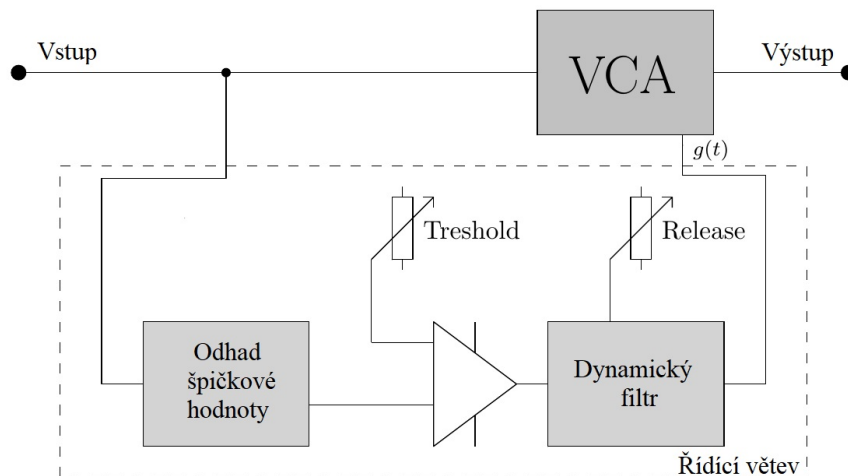
1.1.1 Stavba limiteru

Limiter má obecně vzato tři parametry nastavení. Práh limitace – *threshold*, kterým nastavujeme, na jaké úrovni bude signál omezen, rychlost sepnutí limiteru – *attack* a dobu vypuštění limiteru – *release*. Rychlost, jakou limiter sepne (*attack*) většinou volíme co nejkratší, aby limiter byl schopný zaznamenat i nejkratší změny signálu a potlačit všechna překročení prahu (*threshold*) [2]. Na obrázku 1.1 vidíme blokové schéma základního analogového limiteru. Jak je z obrázku zřejmé, využívá se zde odhad špičkové hodnoty vstupního signálu, díky kterému je limiter schopný měnit prahovou hodnotu – *threshold* a zamezit tak tvrdé saturaci signálu.

1.1.2 Využití limiteru

Limiter se ve většině případů používá jako ochranný prvek, aby nedošlo k poškození dalších článků řetězce. Bývá součástí koncových zesilovačů audio systémů, kde zabráňuje zkreslení výstupního signálu nebo poškození reproduktorové soustavy.

V tomto případě je vstupní signál limitován, pokud je jeho úroveň vyšší než úroveň, kterou je zesilovač schopný zpracovat. Dále se limiter objevuje například jako součást baskytarových aparátů, mixážních pultů nebo jako samostatný pedálový efekt, kterým můžeme docílit záměrného zkreslení signálu za uměleckým účelem.



Obr. 1.1: Blokové schéma běžného limiteru

V FM rádiovém vysílání se limiter využívá v přijímačích při příjmu dvou a více signálů zároveň k snížení rozdílů hlasitostí na výstupu a potlačení šumu [1].

1.2 Clipping

Clipping je jev, který vzniká při limitaci signálu. Jedná se o nelineární zkreslení, ke kterému dochází, pokud je hodnota vstupního signálu vyšší, než je dynamický rozsah zpracovatelný systémem. Části signálu, které tento rozsah překročí, jsou limitovány na prahových úrovních systému, čímž ztrácíme část původního signálu. Ke clippingu zejména dochází v případě, že je zpracováván signál upravován po dynamické stránce pomocí změny zesílení (*gainu*) nebo když je signál převáděn z analogové formy do digitální nebo naopak [3].

Při práci s audiosignálem se clipping projevuje jako praskání nebo zkreslení signálu. K tomuto nejčastěji dochází pokud je zesilovač nucen zesílit signál na úroveň vyšší, než je jeho vlastní napájecí napětí. Pokud k tomuto dojde, přechází zesilovač do tzv. stavu saturace. Maximální výstupní napětí tedy odpovídá napájecímu napětí zesilovače a všechny úrovně vstupního signálu, které by ho po zesílení přesáhly, budou na této hodnotě napětí limitovány [4].

Záměrně se clippingu využívá při zkreslování elektrické kytary efektem *distortion*.

Mimo oblast audio techniky se s clippingem setkáváme například při záznamu digitálních fotografií. Pokud je focený objekt moc světlý nebo naopak moc tmavý (tj. nenachází se v dynamickém rozsahu snímače fotoaparátu), dojde opět k limitaci a na výsledné fotografii budou objekty nad horní úrovní tohoto rozsahu zobrazeny

jako bílá barva, objekty pod spodní úrovní jako černá. Mluvíme poté o přexponování, respektive podexponování. Ve filmové fotografii představuje dynamický rozsah citlivost filmu.

V praxi rozlišujeme mezi dvěma druhy clippingu podle ostroty limitace vstupního signálu na prahové úrovni. Pokud je signál tvrdě oříznut po překročení úrovně limitace, mluvíme o *hard clippingu* neboli tvrdé limitaci, někdy také označované jako digitální clipping.

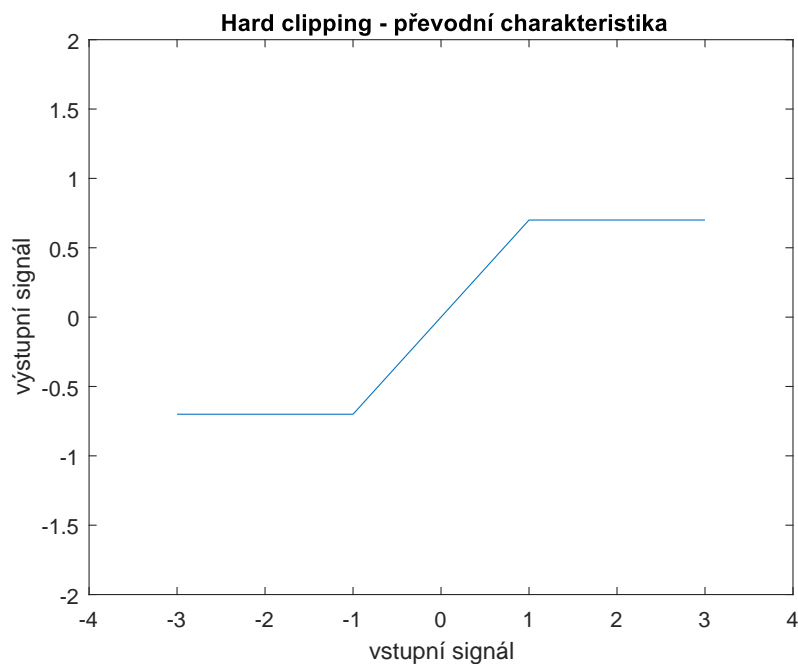
Druhým případem je jev zvaný *soft clipping*. Vstupní signál není tvrdě limitován, ale čím více se blíží prahové úrovni, tím více jsou jeho špičky komprimovány.

1.2.1 Hard clipping

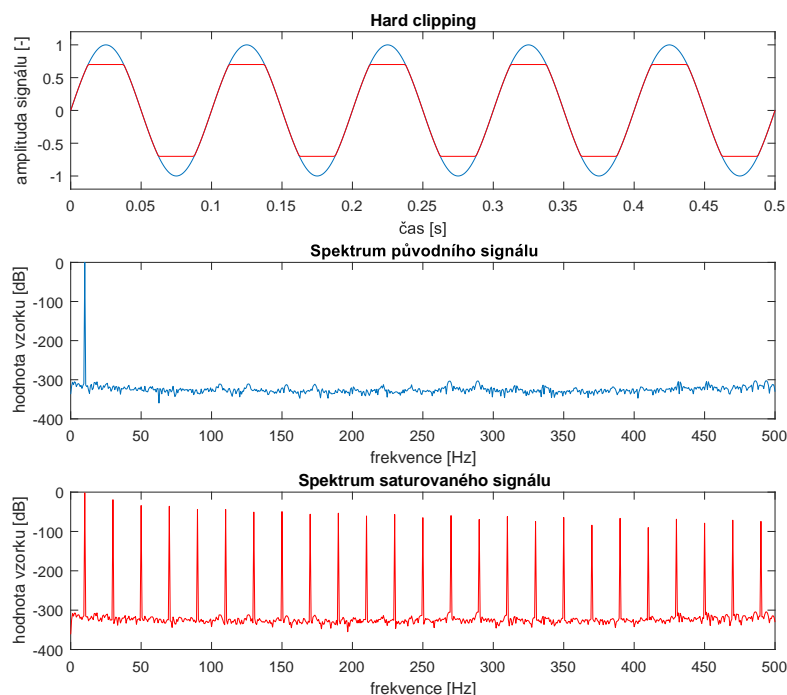
Jak již bylo řečeno, hard clipping je případ dynamické limitace signálu, při které dochází k tvrdému oříznutí špiček vstupního signálu na prahové úrovni θ . Dochází tak ke ztrátám veškeré informace komprimovaných špiček signálu a úplná zpětná rekonstrukce takto limitovaného signálu je takřka nemožná.

Převodní charakteriku hard clippingu můžeme vidět na obrázku 1.2 a matematicky lze tuto funkci zapsat jako [5]:

$$x_h[n] = \begin{cases} x & \text{pro } |x[n]| < \theta, \\ \theta \cdot \text{sgn}(x[n]) & \text{pro } |x[n]| \geq \theta. \end{cases} \quad (1.1)$$



Obr. 1.2: Převodní charakteristika hard clippingu.



Obr. 1.3: Vliv hard clippingu na reálný signál.

Lze vidět, že mezi hodnotami $+\theta_c = 0,7$ a $-\theta_c = -0,7$ je přenosová funkce lineární a mimo tuto oblast je rovna prahovým hodnotám $\pm\theta_c$. Pokud tedy vstupní signál překročí meze dané těmito prahy, výstupní signál bude limitován na hodnotách těchto prahů $\pm\theta_c$ [3].

Na obrázku 1.3 je zobrazen vliv hard clippingu na reálný harmonický signál s kmitočtem 10 Hz a amplitudou $\pm 1[-]$. Práh limitace θ_c je opět $\pm 0,7$. Na prvním grafu je vidět vliv limitace na časovou oblast signálu, zajímavější jsou však zbylé dva grafy, na kterých je znázorněno frekvenční spektrum původního a saturovaného signálu zobrazené pomocí fourierovy transformace. Zatímco spektrum nezkráceného signálu logicky obsahuje pouze jednu složku na kmitočtu 10 Hz, spektrum signálu po zpracování přenosovou funkcí (1.1) obsahuje nekonečně mnoho harmonických složek. Oříznutím na hodnotách $\pm\theta_c$ funkce přestává být hladká a z toho důvodu vznikají vyšší harmonické složky tvořené lichými násobky fundamentu, tedy 30 Hz, 50 Hz, 70 Hz, atd.

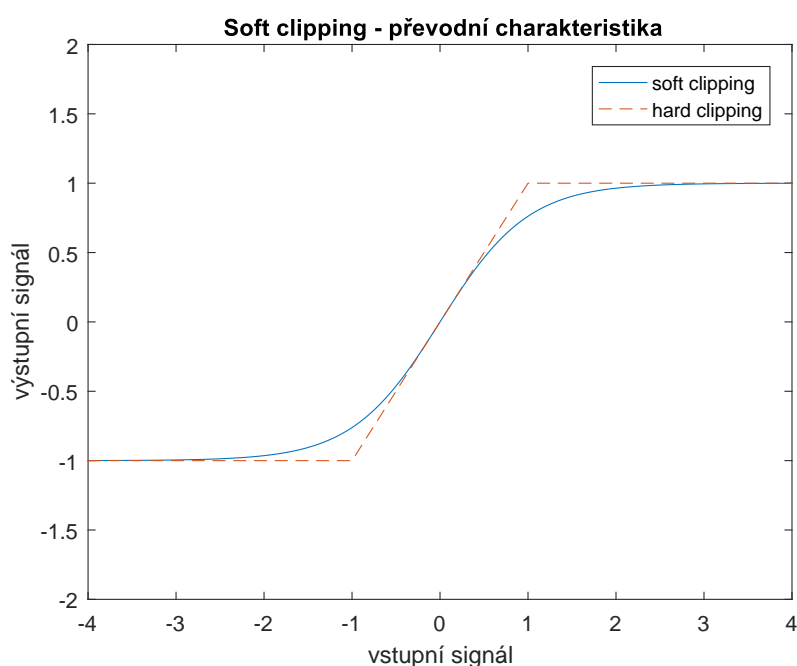
1.2.2 Soft clipping

Při soft clippingu dochází stejně jako u hard clippingu k dynamické limitaci signálu na hodnoty $\pm\theta_c$, na rozdíl od hard clippingu však nedochází k tvrdému oříznutí špi-

ček, ale postupné kompresi. Čím více se dynamická úroveň vstupního signálu bude blížit hodnotám $\pm\theta_c$, tím více bude signál komprimován, k tvrdé limitaci však nikdy nedojde. Při malém zkreslení tedy nemusí dojít k žádné ztrátě informace vstupního signálu. Při znalosti převodní charakteristiky jsme dokonce schopni dokonalé rekonstrukce původního signálu [3].

Na obrázku 1.4 vidíme převodní charakteristiku soft clippingu, pro kterou jsem zvolil jednoduchou funkci

$$x_s[n] = \begin{cases} \tanh x_s[n] & \text{pro } |x[n]| \leq 1, \\ 1 \cdot \text{sgn}(x[n]) & \text{pro } |x[n]| > 1. \end{cases} \quad (1.2)$$



Obr. 1.4: Převodní charakteristika soft clippingu.

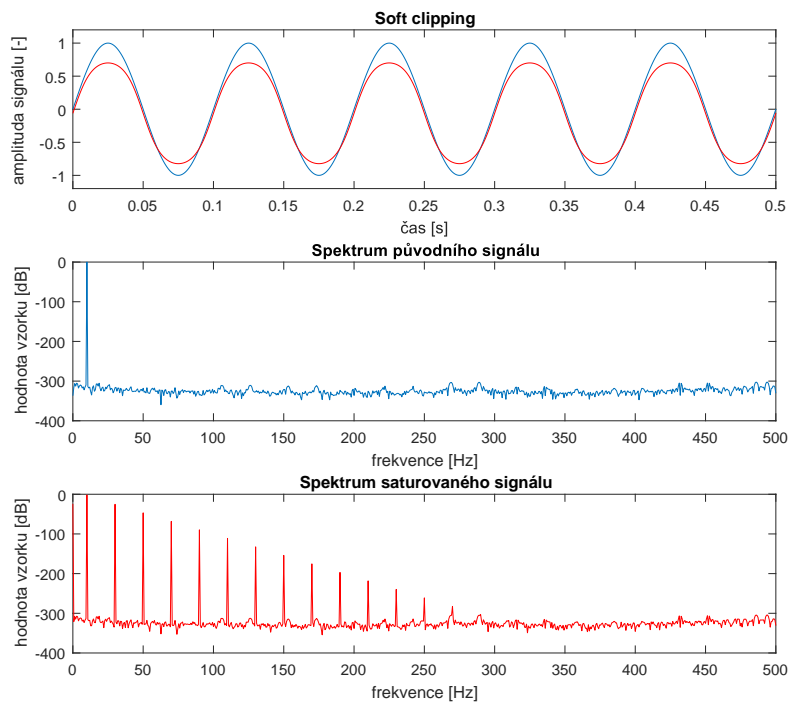
Tato funkce vykazuje poměrně lineární charakteristiku pro nízké hodnoty vstupního signálu a velmi hladkou křivku v okolí mezních hodnot. Nicméně převodní funkci soft clippingu lze vyjádřit jakoukoliv sigmoidou, např. $\arctan x$ nebo $\frac{x}{1+|x|}$ [3].

Na obrázku 1.5 je opět vidět vliv limitace na harmonický signál, tentokrát pro soft clipping. Frekvence vstupního signálu je opět 10 Hz a amplituda $\pm 1[-]$. Převodní charakteristika (1.2) byla lehce upravena pro mezní hodnoty $\pm\theta_c = 0,7$.

Jak je vidět na prvním grafu, signál na této hodnotě není ostře oříznut, ale zachovává si hladký průběh.

Tato skutečnost zapřičiňuje ve frekvenčním spektru, že signál po zpracování systémem 1.4 sice obsahuje vyšší harmonické složky, nicméně jich už není nekonečně

mnoho a navíc jejich amplituda postupně klesá. Opět lze ale vidět, že limitací se objevují liché násobky kmitočtu původního signálu.



Obr. 1.5: Vliv soft clippingu na reálný signál.

2 METODY RESTAURACE SIGNÁLU

V této kapitole nejprve vysvětlím pojem declipping a uvedu některé metody řešení této problematiky. Dále se již budu věnovat metodám založeným na řídkých reprezentacích. Uvedu zde chronologicky několik nejdůležitějších publikací řešících tuto problematiku a popíšu, v čem spočívají představené metody.

2.1 Declipping

Pojmem declipping se rozumí proces, při němž se ze signálu poškozeného limitací snažíme rekonstruovat původní signál nebo alespoň signál co nejpodobnější. Zároveň je účelem declippingu odstranění co nejvíce, pro posluchače, rušivých složek vzniklých zkreslením saturovaného signálu [3].

Pokud se bavíme o restauraci signálu poškozeného clippingem, ve většině případů máme na mysli hard clipping. Jak již bylo výše zmíněno, při soft clippingu nedochází k tak radikálnímu poškození signálu a pokud je známa převodní charakteristika, jsme schopni původní signál jednoduše rekonstruovat za použití inverzní funkce. Navíc pro posluchače není vliv soft clippingu natolik nepříjemný jako tvrdá limitace.

Poprvé se s pojmem declippingu a jeho řešením setkáváme v článku *Restoring a clipped signal* [6], vydaném v roce 1991 Abelem a Smithem. Od té doby se objevilo mnoho různých přístupů k této problematice a jejího řešení. Nejčastěji se setkáváme s využíváním statistiky, kdy se na základě parametrů vstupního signálu snažíme pomocí algoritmu nahradit chybějící vzorky těmi nejvíce pravděpodobnými [3].

Často se také setkáváme s využitím lineární predikce, jakožto metody analýzy audiosignálu. Tato metoda vychází z předpokladu, že po navzorkování signálu lze každý další vzorek diskrétního signálu vyjádřit jako lineární kombinaci jeho předchozích vzorků. Tudíž každý další vzorek je předvídatelný (*predictable*), odtud pochází název lineární predikce [7].

V poslední době však nejlepší výsledky vykazují algoritmy využívající princip řídkých reprezentací, kterými se tato práce zabývá.

2.2 Metody založené na řídkých reprezentacích

V této části budou popsány nejdůležitější metody restaurace signálu využívající řídké reprezentace, které se dodnes objevily. Jednotlivé články budou pokud možno seřazeny chronologicky dle data vzniku.

2.2.1 A Constrained Matching Pursuit to Audio Declipping

Poprvé se v oblasti rekonstrukce audio signálů objevilo využití řídkých reprezentací v článku *A Constrained Matching Pursuit to Audio Declipping* [8] vydaném roku 2011. Rekonstrukce byla aplikována na obecný audio signál, u kterého bylo známo, které vzorky byly limitovány (z tohoto předpokladu vycházejí všechny současné metody). Metoda, kterou autoři publikace využívají je založená na zpracovávání signálu po blocích, které se ze 75 % překrývají a jejichž délku udávají 64 ms. Tento princip se nazývá OLA (*Overlap-add approach*). K nalezení řídkých řešení byl na každý zpracovaný blok použit omezený OMP algoritmus (*Constrained Orthogonal Matching Pursuit*) a diskrétní kosinová transformace (DCT), jakožto slovník.

Pro porovnání autoři restaurovali signál i pomocí metody založené na lineární predikci. Výsledky aplikace metody na řečový a hudební signál byly porovnány podle hodnot SNR (*signal-to-noise ratio*) a rekonstruované signály vykazovaly oproti původnímu saturovanému signálu zlepšení okolo 4 dB. Zatímco rekonstrukce signálu pomocí lineární predikce pouze 1,9 dB.

2.2.2 Recovering a Clipped Signal in Sparseland

O několik měsíců později autoři Weinstein a Wakin zveřejnili článek [9], ve kterém představili své dvě metody založené opět na řídkých reprezentacích a výsledky porovnali s výše zmíněnou metodou. Testovanými signály byly uměle generované řídké signály o délce $N = 128$ vzorků s náhodně generovanou amplitudou (mezi hodnotami 0,5 a 1,5), fází a frekvencí jednotlivých složek pro každou k úroveň řídkosti. První z uvedených metod je založená na ℓ_1 -minimalizaci a dostala název $R\ell_1CC$ (*Reweighted ℓ_1 minimization with clipping constraints*). Jako slovník byla zvolena diskrétní fourierova transformace (DFT).

Druhou, zde představenou metodou je, *Trivial Pursuit with Clipping Constraints* (TPCC). Princip této metody je rozdělen do dvou částí. Nejprve se pomocí DFT určí umístění jednotlivých harmonických složek v kmitočtové oblasti a poté se pomocí metody nejmenších čtverců (*least-squares approach*) vypočítají hodnoty těchto koeficientů. Podobně jako u jiných hladových algoritmů, jako např. Matching Pursuit nebo OMP (viz 3.6). Pokud je nám známa hladina řídkosti původního signálu k , spočívá řešení v nalezení k největších harmonických složek poškozeného signálu. Pokud nám tato hladina není známa, je třeba použít hladový (greedy) algoritmus, dokud nebude chyba rekonstrukce na dostatečně nízké úrovni.

Oba algoritmy byly aplikovány na 100 různých simulací a výsledky ukazují, že podle změřených parametrů $R\ell_1CC$ fungoval lépe než druhý TPCC algoritmus, který však byl mnohem méně výpočetně náročný. Oproti algoritmu uvedenému v předchozím článku vykazovaly obě zde popsané metody lepší výsledky dle daných kritérií.

2.2.3 Least Squares with Examples in Signal Processing

Článek *Least Squares with Examples in Signal Processing* [10], publikovaný roku 2013 se zaměřuje na využití metody nejmenších čtverců při zpracovávání signálu. Autor uvádí několik možných využití této metody, a to při lineárních predikcích, vyhlazování, dekonvoluci nebo pro nás nejzajímavějším odhadu chybějících dat. V rámci této problematiky je uveden i případ využití pro declipping audiosignálů, a to za účelem minimalizace třetí derivace saturovaného signálu.

Metoda nejmenších čtverců vykazuje výborné výsledky z hlediska rychlosti výpočtu, kvalitu rekonstrukce však nemůžeme posoudit, jelikož autor neuvedl praktické výsledky.

2.2.4 Consistent Iterative Hard Thresholding for Signal Declipping

Další článek, který se opět přímo věnuje využití řídkých reprezentací na úlohu declippingu je publikace [11] autorů z belgické univerzity. Představují zde nový algoritmus *Iterative Hard Thresholding – Declipping Constraints* (IHT-DC), pro který jako slovník opět využívají DCT s délkou bloků 64 ms a překryvem 75 %, stejně jako výše zmíněný algoritmus C-OMP, který je v této práci využit jako srovnání. Tato délka bloku při vzorkovací frekvenci 16 kHz odpovídá délce $N = 1024$ vzorků.

Výhodou této metody oproti uvedenému C-OMP je fakt, že IHT-DC využívá pro výpočet i informace ze saturovaného signálu, jako například práh limitace každého vzorku.

Algoritmus byl testován na jednom uměle generovaném zvukovém signálu s náhodně volenou řídkostí a na dvou reálných hudebních nahrávkách. Výsledky těchto aplikací byly porovnány s C-OMP algoritmem, metodou založenou na autoregresním modelu (AR) a programem Adobe DeClipper. Pro porovnání byly použity hodnoty SNR a subjektivní poslechový test. Ze všech testovaných metod vyšel nejlépe právě představený IHT-DC algoritmus.

2.2.5 Declipping of Audio Signals Using Perceptual Compressed Sensing

Pro tuto práci velice zajímavá metoda byla představena v článku *Declipping of Audio Signals Using Perceptual Compressed Sensing* [12] roku 2013, skupinou autorů kolem Bruna Defraene. V této publikaci je představen algoritmus pro restauraci audiosignálů doplněný o poznatky z oboru psychoakustiky, který dostal název *perceptual compressed sensing* (PCS1). Jak z názvu vyplývá, měl by vycházet z metody komprimovaného snímání, k tomu zde však nedochází.

Hledání chybějících vzorků je zde řešeno pomocí ℓ_1 -minimalizace, jako slovník je použita DFT a délka bloků zpracovávaného signálu je $N = 512$. Poprvé je pro tuto úlohu použité jiné okno než obdélníkové, a to Hannovo. Překryv jednotlivých bloků byl zvolen 50 %.

Jak již bylo zmíněno, nejzajímavější na této metodě je využití psychoakustického modelu. V potaz se zde berou dva aspekty lidského vnímání zvuku, absolutní práh slyšení a jev maskování. Tyto dva jevy autoři spojili v jeden parametr kmitočtové (simultální) maskování (*instantaneous global masking threshold*). Tato hodnota je zjišťována pomocí standardu MPEG-1 Layer 1 z každého zpracovávaného bloku. Nejprve jsou zjištěny tónové a netónové složky signálu, jejich lokální maskovací prahy a poté se jejich kombinací s absolutním prahem slyšení vypočítá okamžitý globální maskovací práh daného bloku. Ten je pak, jako parametr, vložen do výpočtu dalšího bloku při minimalizaci.

Při hledání řídkého řešení pomocí ℓ_1 -minimalizace zde nevystupuje pouze hledaný vektor \mathbf{z} , ale součet tohoto vektorů s váhovací maticí \mathbf{P} , tedy \mathbf{Pz} . Tato matice je tvořena reciprokými hodnotami okamžitého globálního maskovacího prahu zasaženými v hlavní diagonále. Použití této matice je z důvodu upřednostnění složek s vyšším maskovacím prahem. Díky tomu jsou při rekonstrukci zvýrazněny složky signálu, které jsou pro lidský sluch a vnímání zvuku důležité.

Pro porovnání výsledků dosažených představeným algoritmem PCSL1 s psychoakustickým modelem byly vytvořeny další dva algoritmy založené na řídkých reprezentacích. První, CSL0, využívající ℓ_0 -minimalizaci a druhý CSL1 založený na ℓ_1 -minimalizaci.

Jako objektivní srovnávací parametry zde byly opět použity hodnoty SNR, ale také hodnoty PEAQ ODG (*Perceptual Evaluation of Audio Quality – Objective Difference*), které lépe vyjadřují perceptuální hodnocení signálu. Hodnocení bylo opět doplněno subjektivními výsledky poslechových testů.

U všech použitých algoritmů došlo k nárůstu hodnot SNR i PEAQ ODQ u rekonstruovaného signálu oproti saturovanému. Nejvyšších hodnot SNR dosahoval algoritmus CSL0, z hlediska hodnot PEAQ ODQ zdaleka nejlépe uspěl model PCSL1. Výsledky poslechových testů také hovořily jasně pro algoritmus používající psychoakustický model. Docházíme tedy k závěru, že parametr SNR zcela nevyhovává o kvalitě restaurace signálu z hlediska sluchového vnímání.

2.2.6 Audio Declipping with Social Sparsity

Další zajímavý článek [13] publikovaný roku 2014 přináší opět nové poznatky v oblasti rekonstrukce signálu pomocí řídkých reprezentací. Tentokrát se jedná o využití tzv. sociální řídkosti (*social sparsity*) v minimalizační úloze. Jedná se o metodu, při

níž se v algoritmu hledajícím řídká řešení berou v potaz i sousední vzorky a sdružují se do skupin, k tomuto úkolu slouží tzv. *shrinkage* operátory [14]. Porovnání použití různých *shrinkage* operátorů je také obsahem této publikace.

Cílem autorů zároveň bylo omezit množinu přípustných řešení. Pro tento účel je zde využita *square hinge* funkce. Jako optimalizační algoritmus založený na sociální řídkost je zde použit relaxovaný (F)ISTA *Fast Iterative Shrinkage–Thresholding* algoritmus, jako slovník je zde poprvé namísto báze využít frame, a to Gaborův frame. Délka bloků byla 64 ms a vzorkovací frekvence 16 kHz, okno zde opět bylo zvoleno Hannovo.

Výsledky byly porovnány s dalšími metodami představenými v článku [11] a metodou představenou v článku [15]. Jak již bylo zmíněno, druhou částí hodnocení bylo i porovnání různých *shrinkage* operátorů, jmenovitě *Lasso* (L), *Windowed Group-Lasso* (WGL), *Empirical Weiner* (EW) a jeho variace *Persistent Empirical Weiner* (PEW).

Testování bylo provedeno na hudebních i řečových ukázkách pomocí rozdílů hodnot SNR restaurovaného signálu oproti saturovanému. Toto bylo provedeno pro různé úrovně clippingu.

Nejlépeších výsledků bylo dosaženo pomocí operátorů PEW a EW. L a WGL už tak uspokojivé výsledky nepodávaly. U PEW došlo při úrovni clippingu $\theta_c = 0,7$ k průměrnému zvýšení SNR až o 16 dB oproti saturovanému signálu u vzorku mluveného slova. Co se týče vlivu na hudební vzorek, dosahoval PEW průměrných hodnot SNR až 10 dB při úrovních clippingu $\theta_c = \{0,3, 0,4, 0,5, 0,6\}$.

V porovnání s metodami představenými v článcích [8] a [15] dosahovaly výsledky s použitými parametry PEW a EW značného zlepšení, L a WGL však zaostávaly. Pouze v případě hudebního signálu byly výsledky L a WGL srovnatelné s metodou z článku [15].

2.2.7 Audio Declipping by Cospase Hard Thresholding

Téhož roku vyšel článek [16] od autorů výše zmíněné publikace *Consistent Iterative Hard Thresholding for Signal Declipping*. Metoda, která je zde představená, vychází z již zmíněného algoritmu *Iterative Hard Thresholding – Declipping Constraints*, tentokrát je však založená na analyzujícím modelu místo syntezujícího (viz 3.5). Tato metoda je založená na *Alternating Direction Method of Multipliers* z důvodu výpočetní složitosti řídké projekce signálu, která je NP-hard. Kombinací původní metody představené v článku [11], analytického modelu a využití *Alternating Direction Method of Multipliers* vznikl nový algoritmus pojmenovaný *Cospase Declipping by Hard Thresholding* (CoDec - HT).

Výsledky aplikace tohoto algoritmu na dva hudební signály o délce deseti sekund (vzorkovací frekvence byla 16 kHz a počet vzorků $N = 1024$) byly porovnány s původním algoritmem dle [11] a metodou popsanou článkem [13] pomocí rozdílů hodnot SNR.

Z výsledných grafů je jasné, že nová metoda jasně předčila původní *Iterative Hard Thresholding*. Výsledky navíc byly srovnatelné s metodou *social sparsity* navzdory nižší výpočetní náročnosti.

2.2.8 Sparsity and Cosparsity for Audio Declipping

V roce 2015 byl publikován další článek s novou metodou rekonstrukce audiosignálu založenou na řídkých reprezentacích [17]. Mezi autory byl opět Srđan Kitić, autor předchozí metody *Audio Declipping by Cosparsity Hard Thresholding* a její původní verze [11]. Nyní spolu s kolegy v tomto článku představují přizpůsobivý algoritmus založený na nekonvexní heuristice, který dostal název SPADE (*Sparse Audio DEclipper*). Tento algoritmus, po přizpůsobení parametrů, lze aplikovat na oba modely signálu, tedy analyzující i syntezující. V publikaci jsou tedy uvedeny a dále porovnávány dvě verze algoritmu, a to A-SPADE pro analyzující a S-SPADE pro syntezující datový model.

Oba modely jsou identické, pokud jsou jejich slovníky čtvercové, navzájem inverzní matice a platí $\mathbf{A} = \mathbf{D}^{-1}$, tedy \mathbf{A} je báze. Dále pro analyzující model je žádoucí, pokud slovník tvoří těsný frame.

Testování probíhalo na deseti různých hudebních vzorcích s délkou 10 s a vzorkovací frekvencí 16 kHz. Jako slovník byla pro oba modely zvolena STFT (*Short Time Fourier Transformation*), Hammingovo okno a délka bloků byla zvolena 1024 vzorků s překryvem 75 %.

Výsledky byly porovnány s už dříve představenou metodou *Audio Declipping with Social Sparsity* [13] a algoritmem C-IHT (*Consistent Iterative Hard Thresholding*) pomocí rozdílů hodnot SNR, v článku uváděném jako SDR (*signal-to-distortion ratio*), restaurovaného signálu vůči saturovanému. Dalším porovnávacím parametrem byla nadbytečnost použitého slovníku. Oba představené algoritmy ve výsledcích předčily obě srovnávací metody C-IHT a Social Sparsity. Nejlepších výsledků pak dosahoval S-SPADE, nicméně jeho výpočetní doba se pohybovala v řádu hodin, zatím co A-SPADE dospěl k výsledkům jen o méně horším průměrně za tři minuty. Jak autoři sami poznamenávají, mohl by tento algoritmus mít, po úpravách, využití v oblasti rekonstrukce signálu v reálném čase.

3 ÚVOD DO ŘÍDKÝCH REPREZENTACÍ

V této kapitole se čtenáři pokusím přiblížit problematiku řídkých reprezentací, jelikož metody na nich založené budu dále používat k samotnému declippingu audi-osignálu. Definuji zde základní značení, vysvětlím zde dále používané pojmy, jako např. norma vektoru nebo řídkost. Nejobsáhlejší část této kapitoly se bude zabývat samotným hledáním řídkých řešení signálu. V krátkosti zde představím báze a framy vektorů, vysvětlím a porovnam zde dva modely signálu, syntezující a analyzující. Na závěr zde uvedu principy algoritmů pro hledání řídkých řešení. Pro přehlednost vše rozdělím do samostatných podkapitol.

3.1 Základní značení a znalosti

V této části definuji základní pojmy, se kterými budu v této práci pracovat. Pro správné porozumění dále definuji značení, které budu dále používat.

Skalární veličiny budeme značit kurzívou, např. m , N atd. Vektory budeme zapisovat tučným řezem tzn. \mathbf{x} , \mathbf{y} .

Pokud nebude v textu jinak řečeno, budeme vektory uvažovat jako sloupcové, s indexováním prvků počínaje jedničkou, tj. $\mathbf{y} = [y_1, y_2, y_3, \dots, y_n]$. Počet prvků vektoru nebo množiny (tzv. kardinalitu) budeme značit jako absolutní hodnotu, např. $|\mathbf{x}| = \{-6, 0, 2, 7\} = 4$ [18].

Matice budeme naproti tomu označovat velkými tučnými písmeny, např. \mathbf{A} , \mathbf{B} . Pokud se v textu budou využívat komplexní sdružení čísla c , jejich značení bude \bar{c} , tedy kurzívou s pruhem.

Dále je nutné definovat nosič vektoru. Tímto pojmem je myšlena množina indexů, v nichž má daný vektor nenulové hodnoty. Tato množina bude označena $\text{supp}(\mathbf{x})$. Pro nosič vektoru tedy platí $\text{supp}(\mathbf{x}) = \{i | x_i \neq 0\}$ [18]. Pokud tedy máme vektor $\mathbf{x} = \{0, 0, 1, 2, 0, 0, 7, 0, 6, 0\}$, bude $\text{supp}(\mathbf{x}) = \{3, 4, 7, 9\}$ a velikost nosiče vektoru $|\text{supp}(\mathbf{x})| = 4$.

Vektorové prostory budeme vždy považovat za neprázdné a zapisovat je budeme jako \mathbb{V} , \mathbb{R} , atd.

3.2 Normy vektorů

Nyní je potřeba definovat ℓ_p normu vektoru, což je klíčový pojem v teorii řídkých reprezentací. Tato norma je pro libovolný vektor $\mathbf{x} \in \mathbb{C}^N$ definována jako: [18]

$$\begin{aligned}
\|\mathbf{x}\|_p &:= \left(\sum_{i=1}^N |x_i|^p \right)^{1/p} && \text{pro } 1 \leq p < \infty, \\
\|\mathbf{x}\|_p &:= \sum_{i=1}^N |x_i|^p && \text{pro } 0 < p < 1, \\
\|\mathbf{x}\|_\infty &:= \max_i |x_i|, \\
\|\mathbf{x}\|_\infty &:= |\text{supp}(x)|.
\end{aligned} \tag{3.1}$$

Přísně vzato se jedná o ℓ_p -normu vektoru pouze v případě $1 \leq p < \infty$. Pro zjednodušení však budeme pro všechna p používat jednotné označení ℓ_p -norma.

Zcela nejčastěji využívanou normou je euklidovská ℓ_2 -norma, tj. vzdálenost vektoru. Tuto normu však v této práci používat nebudeme. Mnohem více nás bude zajímat ℓ_1 -norma, která vyjadřuje součet absolutních hodnot prvků vektoru, tedy $\|\mathbf{x}\|_1 = \sum_i |x_i|$ a dále také ℓ_0 -norma, představující počet nenulových složek vektoru. Z výše definovaných vztahů tedy pro vektor $|\mathbf{x}| = \{0, 0, 1, 2, 0, 0, 7, 0, 6, 0\}$ je $\|\mathbf{x}\|_1 = 16$ a $\|\mathbf{x}\|_0 = 4$. Za zmínku také stojí $\|\cdot\|_\infty$ -norma, jejímž výsledkem je největší prvek vektoru.

Když už máme definovanou normu vektoru, můžeme definovat i pojem řídkosti (*sparsity*) vektoru. Vektor nazveme k -řídkým, pokud má nejvýše k nenulových složek, tedy každý vektor $\mathbf{x} \in \mathbb{C}^N$ musí splňovat podmínku:

$$\|\mathbf{x}\|_0 \leq k. \tag{3.2}$$

Relativní řídkostí vektoru rozumíme poměr $\frac{k}{N}$, tedy poměr nenulových složek vektoru k ku jeho celkové délce N [18].

3.3 Řídká řešení

Hledáním řídkého řešení systému lineárních rovnic je myšleno řešení obvyklé soustavy $\mathbf{Ax} = \mathbf{y}$, přičemž chceme, aby hledaný vektor \mathbf{x} byl co nejřidší, tedy obsahoval co nejméně nenulových složek. Obecně lze tuto úlohu zapsat jako:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{vzhledem k } \mathbf{Ax} = \mathbf{y}, \tag{3.3}$$

přičemž známe výsledný vektor $\mathbf{y} \in \mathbb{C}^m$ a matici $\mathbf{A} \in \mathbb{C}^{m \times N}$ definující lineární zobrazení. Budeme předpokládat pouze případy, kdy $m < N$ případně $m \ll N$

a matice \mathbf{A} je plné řádkové hodnosti. Matice \mathbf{A} se běžně nazývá slovník (*dictionary*) a sloupce této matice označujeme jako atomy (*atoms*) [18].

Pokud je zpracováváný signál zašuměný, není výsledek řídký v doslovném smyslu, ale obsahuje odchylku vzniklou právě tímto zašuměním. Z tohoto důvodu do výpočtu problému $\mathbf{Ax} = \mathbf{y}$ zahrnujeme aproximační odchylku δ .

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \|\mathbf{Ax} - \mathbf{y}\|_p \leq \delta, \quad (3.4)$$

kde obvykle uvažujeme $p = 2$ [18].

Nyní je potřeba definovat pojem *spark*, což je důležitá vlastnost při hledání řídkého řešení. Jedná se o číslo udávající nejmenší počet lineárně závislých sloupců matice \mathbf{A} . Formálně tuto vlastnost zapíšeme [18]:

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{z} \in \mathbf{A}, \mathbf{z} \neq \mathbf{0}} \|\mathbf{z}\|_0. \quad (3.5)$$

Z této definice vyplývá, že pro nenulovou matici $\mathbf{A} \in \mathbb{C}^{m \times N}$, kde $m < N$, pro číslo *spark* platí $\text{spark}(\mathbf{A}) \in \{2, \dots, m + 1\}$. Hodnoty $\text{spark}(\mathbf{A}) = 2$ je dosaženo, pokud jeden sloupec je násobkem jiného [18].

Řešení \mathbf{x} je nejřidší možné řešení a žádné jiné stejně řídké řešení neexistuje, pokud \mathbf{x} splňuje

$$\|\mathbf{x}\|_0 < \frac{\text{spark}(\mathbf{A})}{2}. \quad (3.6)$$

Pokud se tedy podaří najít řešení splňující tuto podmínku, víme, že bylo zároveň nalezeno řešení původního problému $\mathbf{Ax} = \mathbf{y}$. Vzhledem k tomu, že výpočetní náročnost hledání čísla $\text{spark}(\mathbf{A})$ je srovnatelná s řešením rovnice (3.3), v praxi se příliš nepoužívá.

Je tedy potřeba využít jiný způsob ověření jedinečnosti signálu, který je výpočetně jednodušší. K tomuto účelu nám slouží vlastnost matice \mathbf{A} zvaná vzájemná koherence, která je definována jako největší absolutní normalizovaný skalární součin dvou různých sloupců matice \mathbf{A} , tedy

$$\mu(\mathbf{A}) = \max_{1 \leq j, k \leq N, j \neq k} \frac{|\mathbf{a}_j^\top \mathbf{a}_k|}{\|\mathbf{a}_j\|_2 \cdot \|\mathbf{a}_k\|_2}, \quad (3.7)$$

kde \mathbf{a}_j označuje j -tý sloupec matice \mathbf{A} [19].

Mezi číslem *spark* a vzájemnou koherencí libovolné matice \mathbf{A} platí:

$$\text{spark}(\mathbf{A}) \leq 1 + \frac{1}{\mu(\mathbf{A})}. \quad (3.8)$$

Jak vyplývá z výše uvedených vztahů, pokud nalezneme takové řešení \mathbf{x} soustavy $\mathbf{Ax} = \mathbf{y}$ splňující podmínku

$$\|\mathbf{x}\|_0 < \frac{1}{2} \left(1 + \frac{1}{\mu(\mathbf{A})} \right), \quad (3.9)$$

pak se jedná o nejřidší možné a zároveň jedinečné řešení. Z této nerovnosti plyne, že pro nalezení nejřidšího řešení je potřeba používat co nejvíce nekoherentní slovníky. Jak ukážeme později, tohoto řešení lze dosáhnout i ℓ_1 -minimalizací [19].

Předpokládejme, že existuje k_0 -řidké řešení problému (3.3), o kterém díky splnění podmínky (3.6) víme, že se jedná o jedinečné a nejřidší možné. Kdybychom trvali na nalezení přesného řešení (3.3), bylo by třeba projít všech $\binom{N}{k_0}$ možných kombinací. Toto odpovídá výpočetní složitosti NP-hard. V praxi, kdy N nabývá vysokých hodnot to není přijatelné, jelikož s dnešní technikou by výpočet trval při nejmenším v řádech týdnů [18]. Za účelem zrychlení výpočtu je tedy třeba vyměnit přesnost za rychlost použitím aproximačních metod. Dalším problémem je nekonvexnost ℓ_0 -normy, díky čemuž není možné použití některé z dnes dostupných metod konvexní optimalizace [19].

Je tedy vhodné nahradit nekonvexní ℓ_0 -normu nejbližší konvexní ℓ_p -normou, abychom se jen s malou odchylkou přiblížili k původnímu NP-složitému řešení. Nejbližší konvexní norma je ℓ_1 -norma. Původní (3.3) úlohu tedy můžeme aproximovat na

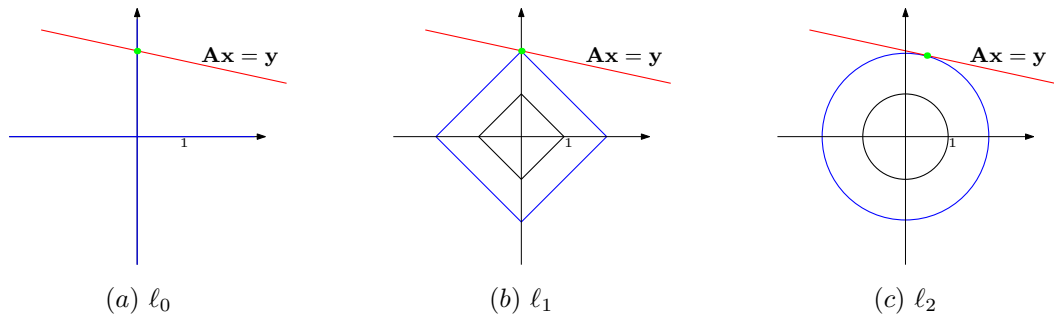
$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{vzhledem k} \quad \mathbf{Ax} = \mathbf{y}. \quad (3.10)$$

Ve většině případů dochází ke shodě řešení pomocí ℓ_0 a ℓ_1 minimalizací (relaxací). Vyřešení úlohy (3.10) však nezaručuje, že nalezené řešení bude nejřidší možné a může dojít i k situaci, kdy výsledkem ℓ_1 -minimalizace bude nekonečně mnoho řešení a nejřidší možná řešení budou pouze dvě krajní [3].

Na obrázku 3.1 je znázorněno řešení úlohy $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ vzhledem k $\mathbf{Ax} = \mathbf{y}$ postupně pro normy ℓ_0 , ℓ_1 a ℓ_2 v prostoru \mathbb{R}^2 .

Jak je vidět, řešení pomocí ℓ_0 a ℓ_1 jsou shodná a nejřidší možná. Pomocí euklidovské ℓ_2 -normy jsme však našli odlišné řešení, které není řidké, ale má nejmenší energii [18].

Podmínky, za jakých se řešení pomocí norem ℓ_0 a ℓ_1 shodují, jsou stále předmětem zkoumání [3].



Obr. 3.1: Zvětšující se koule norem l_0 , l_1 a l_2 dotýkající se nadroviny $\mathbf{Ax} = \mathbf{y}$.

3.4 Báze a framy

Při restauraci signálu budeme využívat vlastností vektorových prostorů, které je nejprve potřeba definovat. Jedná se o pojmy báze a frame vektorového prostoru (VP). V této sekci budeme dále uvažovat pouze konečně rozměrné vektory a dimenze VP.

Nejprve si definujeme samotný vektorový prostor. Jedná se algebraický objekt, jehož základním prvkem je vektor. Důležitou vlastností vektorového prostoru je, že pro vektory platí lineární matematické operace, tj. sčítání, odčítání, atd. Výsledek těchto operací je opět součástí vektorového prostoru.

Každý vektor $\mathbf{x} \in \mathbb{V}$ je lineární kombinací generátorů vektorového prostoru. Podmnožinu těchto generátorů $\mathbf{E} \in \mathbb{V}$ pak zapisujeme jako $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_1, \dots, \mathbf{e}_m\}$ a pro souřadnice c_i vektoru \mathbf{x} vzhledem k \mathbf{E} platí

$$\mathbf{x} = c_1\mathbf{e}_1 + c_2\mathbf{e}_2 + \dots + c_m\mathbf{e}_m = \mathbf{E}\mathbf{c}. \quad (3.11)$$

Bází vektorového prostoru je nejmenší množina vektorů, jejichž lineární kombinací lze vyjádřit jakýkoli vektor tohoto prostoru. Jedná se tedy o minimální systém generátorů \mathbf{E} . Vektory tvořící bázi musí být lineárně nezávislé a jejich počet je roven dimenzi daného prostoru [18].

V praxi se využívají dva speciální druhy bází, a to *ortogonální* a *ortonormální*. První zmíněná se vyznačuje tím, že všechny její vektory jsou na sebe kolmé. Ortonormální báze je speciálním případem báze ortogonální, tedy všechny její vektory jsou na sebe taktéž kolmé, ale zároveň jsou i jednotkové, tedy $\|\mathbf{b}_i\| = 1$.

Nyní je možné si definovat pojem frame, který budeme později potřebovat při rekonstrukci audiosignálu.

Pokud k bázi VP přidáme další vektory, vznikne nám podmnožina generátorů, jejichž počet bude větší než dimenze prostoru. Tímto systémem budeme stále schopni vyjádřit jakýkoli vektor VP, generátory však již nebudou lineárně nezávislé.

Framy nejsou tolik omezené jako báze a právě pro svoji flexibilitu se často využívají. S jejich nadbytečností však přichází problémy s numerickou nestabilitou a výpočetní složitost [18].

Abychom mohli mluvit o množině vektorů jako o framu VP, musí pro konstanty $0 < A \leq B < \infty$ platit:

$$A\|\mathbf{x}\|^2 \leq \sum_{k \in \mathbb{I}} |\langle \mathbf{x}, \mathbf{F}_k \rangle|^2 \leq B\|\mathbf{x}\|^2, \forall \mathbf{x} \in \mathbb{V}. \quad (3.12)$$

Tyto konstanty A, B označujeme jako meze framu a pokud platí $A = B$, tak tento frame nazýváme *těsným*. Tento druh framu odpovídá případu ortogonální báze. I zde opět existuje speciální druh těsného framu, kdy, stejně jako u ortonormální báze, platí $A = B = 1$. Tyto framy se jmenují *Parsevalovy těsné framy*. Prvky framu jsou většinou označovány jako *atomy* [2].

3.5 Syntezující a analyzující model

V předchozí části bylo vysvětleno, jak se pomocí řídkých reprezentací řeší soustava rovnic $\mathbf{A}\mathbf{x} = \mathbf{y}$. Takto lze zapsat každý diskretní signál, tedy jako součin vektoru souřadnic a matice. Tomuto procesu se říká transformace a použité matici se pak nazývá transformační.

Existují dva odlišné modely tohoto vyjádření. Pokud máme libovolný signál \mathbf{y} vyjádřený pomocí součinu vektoru souřadnic \mathbf{x} a transformační matice \mathbf{A} , nazýváme tento model syntezující (neboli aditivní), neboť signál \mathbf{y} skládáme z několika složek. Příkladem takové transformace je Fourierova transformace, při které je signál vyjádřen jako součet funkcí sinus a kosinus o různých frekvencích.

Druhým, méně častým, modelem signálu je model analyzující. V tomto případě transformace signálu \mathbf{y} nevzniká skládáním vektorů, namísto toho jsou tyto vektory hledány. Tento model zapisujeme jako $\mathbf{\Omega}\mathbf{y} = \mathbf{x}$, kde $\mathbf{\Omega}$ je transformační matice (popř. analytický slovník) a \mathbf{x} je řídký vektor reprezentující signál \mathbf{y} vzhledem k matici $\mathbf{\Omega}$ [20].

Při použití bází jsou oba modely signálu \mathbf{y} shodné, tedy $\mathbf{A} = \mathbf{\Omega}^{-1}$ [3].

Při hledání řídkých řešení pak pro oba modely využíváme jiné algoritmy, které budou popsány v další části.

3.6 Algoritmy pro hledání řídkých řešení

V této části uvedu některé algoritmy, které se v současné době využívají pro hledání řídkých řešení. Jelikož tato práce není zaměřena na zkoumání a srovnávání různých

algoritmů, budou pouze rozděleny do skupin podle způsobu řešení problému. Specifika jednotlivých algoritmů tu však popisována nebudou.

První skupinou jsou relaxační algoritmy, které, jak název napovídá, využívají ℓ_1 -relaxace. Tyto algoritmy vycházejí z poznatků popsaných částí 3.3. Mezi tyto algoritmy patří BP (*Basic Pursuit*), modifikovaný LARS (*Least Angle Regression, homotopy method*), IRLS (*Iterative Reweight Least Squares*) a další [19].

Druhá skupina je zastoupena tzv. hladovými (greedy) algoritmy. Princip těchto algoritmů je hledání nejméně jednoho „nejvýznamnějšího“ prvku vektoru v každé iteraci. Takto nalezený atom se automaticky podílí na výsledném řešení bez ohledu na další průběh algoritmu. Tyto metody jsou využívány pro svou nízkou složitost, nicméně nalezení globálního optima nezaručují [19]. Nejčastěji používaný zástupce této kategorie je OMP (*Orthogonal Matching Pursuit*) a jako dalšího zástupce můžeme uvést například MP (*Matching Pursuit*)

Za zmínku ještě stojí algoritmy kombinující obě skupiny a algoritmy využívající metody prahování (*thresholding*).

4 PSYCHOAKUSTIKA

V této kapitole přiblížím základní principy vědního oboru psychoakustiky, které následně budou využity při rekonstrukci signálu. Vysvětlím pojem *absolutní práh slyšení*, jev kmitočtového maskování a na závěr představím psychoakustický model.

4.1 Absolutní práh slyšitelnosti

Absolutní práh slyšitelnosti je definován jako minimální průměrná hladina akustického tlaku, kterou je schopný člověk s normálním sluchem slyšet. Tato hladina se mění v závislosti na kmitočtu. Je závislá na charakteru zvuku (čistý tón, šum, řeč, atd.), prostředí (volné nebo difúzní pole), věku posluchače a rozdíl je i mezi monaurálním a binaurálním poslechem [21].

Při výpočtu akustického tlaku se jako vztažná hodnota využívá průměrná hodnota prahu slyšitelnosti v okolí kmitočtu 1 kHz, tj $2 \cdot 10^{-5}$ Pa. Nejvyšší citlivost lidského sluchu je v okolí kmitočtu 3 kHz.

Na obrázku 4.1 je znázorněna frekvenční charakteristika prahu slyšitelnosti v oblasti do 5 kHz. Na vodorovné ose jsou uvedeny hodnoty frekvence v kHz a na svislé ose je hladina akustického tlaku L v dB(SPL) s vztažnou hodnotou $p_0 = 2 \cdot 10^{-5}$ Pa. Zobrazená nelineární funkce je průměrným výsledkem subjektivních testů v odhlučněném prostředí a zapsat ji lze jako [22]:

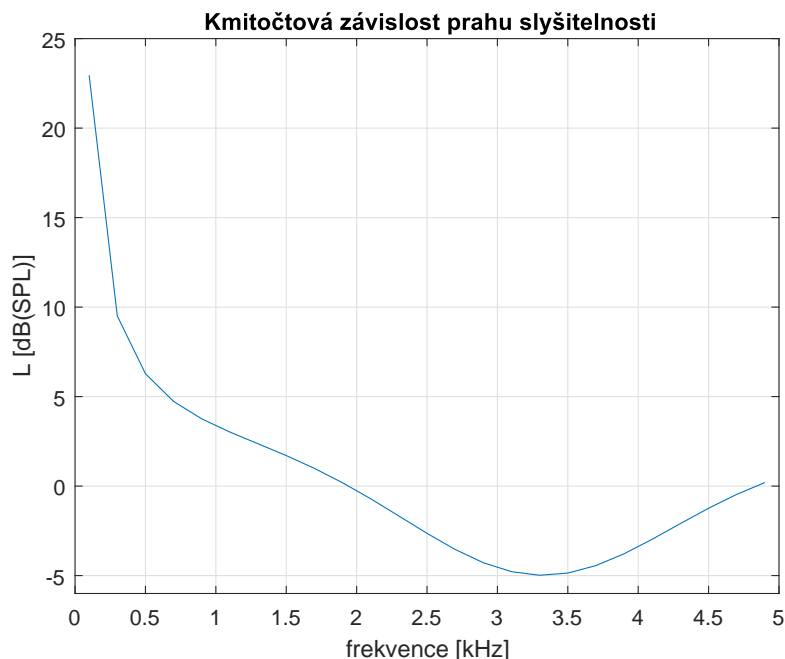
$$S_a(f) = 3,64 \cdot f^{-0,8} - 6,5 \cdot \exp(-0,6 \cdot (f - 3,3)^2) + 10^{-3} \cdot f^4, \quad (4.1)$$

kde $S_a(f)$ je hladina akustického tlaku v dB(SPL) a f je kmitočet v kHz, přičemž $f \in [100 \text{ Hz}, 5 \text{ kHz}]$.

4.2 Kmitočtové maskování

Další z vlastností lidského sluchového ústrojí je jev zvaný *maskování*. Pokud do lidského ucha přicházejí dva zvuky současně, může se stát, že se díky vnímání jednoho zvuku (*maskující*) zeslabí nebo úplně ztratí vjem ze zvuku druhého (*maskovaný*).

Toto je způsobeno fyziologickou stavbou vnitřního ucha. Zvuk se do mozku přenáší pomocí rozkmitání bazilární membrány a jeho výška určuje oblast, která je rozkmitána. Hluboké tóny rozkmitávají střed hlemýždě, vysoké zase jeho vstup. Při zpracovávání zvuku není však podrážděno pouze jedno konkrétní místo membrány, ale i jeho okolí. Pokud jsou buňky této oblasti podrážděny, po nějakou dobu nejsou schopné přijmout další impulz. To zapříčiňuje, že druhý, slabší zvuk dráždící oblast rozkmitanou prvním zvukem bude zamaskován (nebude slyšet). K maskování tedy dochází v případě, že dva zvuky rozkmitávají tutéž oblast bazilární membrány



Obr. 4.1: Kmitočtová charakteristika absolutního prahu slyšitelnosti

a zároveň amplituda tohoto kmitání je u maskujícího zvuku větší než u maskovaného [21].

Rozlišujeme dva druhy maskování, *kmitočtové (simultánní)* a *dočasné*. K maskování nedochází pouze při současném vnímání více zvuků, ale i před zazněním maskovacího zvuku a po jeho odeznění. Tyto jevy nazýváme *pre-masking* a *post-masking* a k jejich měření se využívají krátké šumové impulzy.

Jev maskování se využívá v tzv. *psychoakustickém modelu* při ztrátové kompresi zvuku k určení úrovně *kvantovacího šumu* na takovou hodnotu, aby tento šum byl zamaskován zvukovým signálem a nebyl slyšet. Kvantovací šum vzniká při převodu diskretních zvukových vzorků se spojitou úrovní na vzorky s diskretní úrovní (*kvantování*). Velikost kvantovacího šumu je tedy dána odchylkou takto nakvantovaného signálu od původního [21].

Účinky maskování ve frekvenční oblasti se zakreslují pomocí *maskovací křivky*, která ohraničuje kmitočtové pásmo, ve kterém je slyšet pouze maskovací tón. Šířka kritického pásma není pro všechny kmitočty signálu stejná, pro signály s nižší frekvencí je řádově menší než pro signály s frekvencí vyšší. Tato křivka se liší pro tón a pro šum, proto je součástí psychoakustického modelu rozpoznávání charakteru signálu, zda se jedná o šumový nebo harmonický signál. Toto se provádí v samostatných kmitočtových pásmech a pro každé se volí daná maskovací křivka (harmonického signálu nebo úzkopásmového šumu).

4.3 Psychoakustický model

Psychoakustický model se využívá ve standardu MPEG pro kódování zvukového signálu k určení složek signálu, které jsou vjemově důležité a těch, které je možné vynechat, aniž by to lidský sluch zaznamenal. Vyhledává hlasité zvuky, které mohou zamaskovat zvuky slabší nebo naopak příliš slabé zvuky, které mohou být zcela vynechány, protože by je lidské ucho nebylo schopné zaznamenat. To se odráží i na určení velikosti bitového prostoru pro danou část signálu. Čím méně důležitá pro celkový sluchový vjem bude, tím méně bitů jí bude z celkového prostoru přiděleno.

Při kódování zvuku pomocí MPEG je vstupní signál pomocí banky filtrů rozdělen do subpásem (*subpásmové kódování*) a pro každé z nich se pomocí psychoakustického modelu určí počet přidělených bitů z celkového počtu. Určující je rozdíl mezi celkovou energií pásma a jeho maskovacím prahem, čím menší tento odstup bude, tím vyšší kvalita kódování je potřeba, tedy i vyšší počet bitů. Maskovací práh se určuje pomocí FFT vstupních vzorků signálu a probíhá paralelně s rozdělováním do subpásem. Použití obou těchto metod zároveň zaručuje dostatečné spektrální rozdělení pro výpočet minimálního maskovacího prahu i rozlišení časové oblasti pro kódování [23].

Existuje velké množství psychoakustických modelů, které se používají pro různé velikosti vzorkovacích frekvencí. Nejčastěji používanými jsou PAM 1 a PAM 2, druhý jmenovaný se využívá ve standardu MPEG Layer 3 (formát MP3).

Nyní bude uveden výčet operací, které obsahuje psychoakustický model pro MPEG. Jelikož není obsahem této práce implementace samotného modelu, ale pouze jeho využití při restauraci audio signálu, nebude zde matematicky vyjádřeno, jak se jednotlivé operace provádějí.

Nejprve je signál nakvantován (jak bylo popsáno výše), poté nastává *váhování*.

4.3.1 Váhování

Jedná se o proces, při němž je signál zpracováván po blocích, které mají rozdílnou velikost. Tato velikost závisí na velikosti periody vstupního signálu. Tento proces je založen na předpokladu, že délka jednotlivých bloků je větší než perioda signálu. Toto je zajištěno použitím váhovacího okénka, v případě MP3 se jedná o Hannovo okno. Díky tomu, že toto okno dává téměř nulovou váhu signálu na svých okrajích, nedochází při překryvu jednotlivých oken ke zkreslení.

4.3.2 Rozdělení tónových a šumových složek a výpočet jejich SPL

Poté co je signál rozdělen pomocí subpásmového kódování (viz výše) je potřeba rozdělit tónové složky od šumových za účelem správného určení maskovací křivky. Toto probíhá pomocí určení lokálních maxim spektra FFT daného bloku a zjištění složek podobných funkci sinus, které reprezentují tón. Poté se vypočítá SPL jednotlivých tónových složek.

Ze zbylých složek se určí šumové složky s využitím kritických pásem. Pro každé kritické pásmo je pak učeno SPL dané šumové složky.

4.3.3 Výpočet maskovacího prahu

Nejprve je potřeba zjistit individuální maskovací prahy jednotlivých tónových a šumových složek signálu. Z těchto hodnot a absolutního prahu slyšení je pak určena celková maskovací křivka. Maskovací práh celého subpásma je pak vypočítán z jednotlivých individuálních maskovacích prahů a absolutního prahu slyšení. Z celkového maskovacího prahu je pak pro každé subpásmo vypočítána minimální slyšitelná kmitočtová hodnota, tedy *minimální maskovací práh*.

4.3.4 Decimace maskovacích složek

Snížení počtu maskovacích složek se říká *decimace*. Tento proces je aplikován pouze na maskující složky, které jsou nad úrovní maskovacího prahu. Pokud jsou maskovací složky příliš blízko u sebe (blíže než 0,5 Bark), je zachována pouze ta nejhlasitější. Ostatní slabší složky v kritickém pásmu jsou odstraněny, protože jsou maskovány nejsilnější složkou, tudíž nejsou důležité a zbytečně by zabíraly bitový prostor [23].

4.3.5 Výpočet poměru signálu k maskující složce

Nakonec je pro každé subpásmo od hodnoty jeho akustického tlaku odečten daný minimální maskovací práh, čímž dojdeme k hodnotě SMR (*signal to mask ratio*). Tato hodnota udávaná v dB představuje odstup slyšitelné kmitočtové složky (tónu) od vypočítané maskovací křivky. Jinými slovy se jedná o dynamický rozsah dané složky signálu.

Tento údaj se posléze používá při kódování signálu k určení složek signálu, které jsou nepotřebné.

5 NÁVRH ŘEŠENÍ

V této kapitole představím řešení problému declippingu pro syntežující a analyzující model. Nejprve však uvedu matematické vyjádření declippingu, stručně vysvětlím princip proximálních algoritmů, které budou použity pro oba modely. Dále popíšu tzv. *proximální operátory* pro úlohu declippingu, uvedu výchozí parametry, se kterými budou oba algoritmy pracovat a na závěr pro oba modely signálu navrhnou bloková schémata, která doplním o psychoakustický model.

5.1 Declipping pomocí proximálních algoritmů

Nejprve je tedy potřeba si úlohu declippingu definovat matematicky. Uvažujme tedy signál \mathbf{y} , jakožto původní signál před saturací a \mathbf{y}_c , představující tvrdě limitovaný signál podle (1.1). Horní práh limitace označíme jako $\theta_H > 0$ a dolní $\theta_L < 0$ a budeme předpokládat, že vzorky, které nebyly poškozeny, se shodují s původním signálem \mathbf{y} .

Pro další práci se vzorky saturovaného signálu budeme potřebovat tzv. masky signálu. Ty si lze představit jako jednotkové matice s odstraněnými nepotřebnými řádky [3].

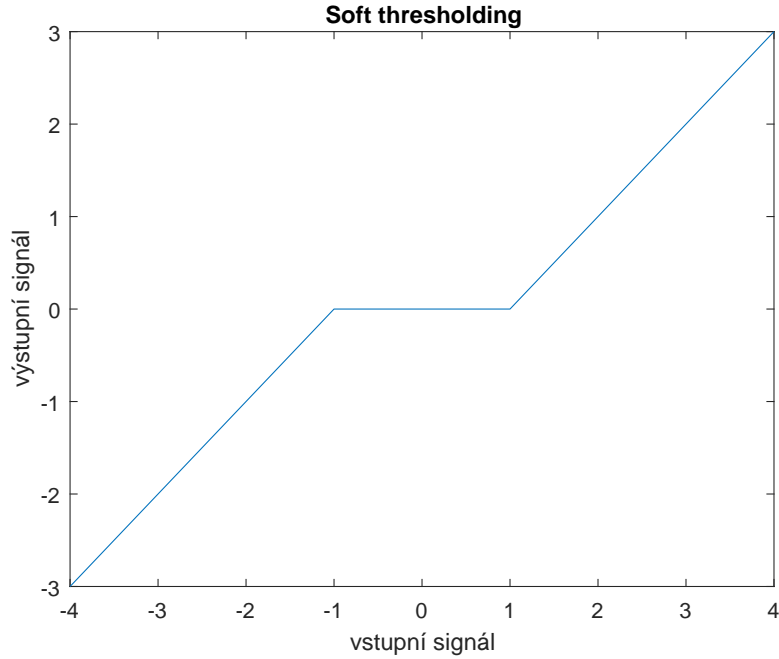
Vytvoříme si tedy tři masky, a to M_R (*reliable*), představující nepoškozené vzorky signálu, které nebyly ovlivněny clippingem, M_H (*higher*) pro vzorky přichycené na úrovni θ_H a M_L (*lower*) pro vzorky s hodnotou θ_L . Budeme také předpokládat, že zpracovávaný signál je řídký například v Gaborově transformaci a budeme tedy pro rekonstrukci signálu \mathbf{y}_c hledat nejřidší vektor koeficientů \mathbf{c} této transformace.

Jak již bylo dříve uvedeno, pro hledání řídkého řešení je vhodné použít ℓ_1 -normu. Pokud tedy úlohu (3.10) obohatíme o nově vytvořené masky, můžeme problém pro syntežující model zapsat jako:

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{vzhledem k} \quad \begin{cases} M_R G \mathbf{c} = M_R \mathbf{y}_c, \\ M_H G \mathbf{c} \geq \theta_H, \\ M_L G \mathbf{c} \leq \theta_L, \end{cases} \quad (5.1)$$

kde G představuje syntézu signálu. Pokud do této úlohy vstoupí váhování, neprobíhá minimalizace pouze vektoru \mathbf{c} , ale váhovaného vektoru \mathbf{c} . Rekonstruovaný signál získáme z výsledku (5.1) pomocí zpětné transformace $G\hat{\mathbf{c}}$. Vyjádření, upravené pro analyzující model, bude popsáno v části 5.3.

Aby bylo možné najít řešení této úlohy, je nutné použít algoritmy založené na proximálním dělení. Ty vyházejí z předpokladu, že lze většinu optimalizačních problémů převést na tzv. neomezený tvar, který představuje minimalizaci součtu m konvexních



Obr. 5.1: Převodní charakteristika – soft thresholding

funkcí. Pro úlohu declippingu budeme využívat algoritmy ze skupiny tzv. dopředno-zpětných (*forward-backward*) algoritmů, které hledají minimum součtu dvou konvexních funkcí

$$\min_{\mathbf{x} \in \mathbb{R}^N} f_1(\mathbf{x}) + f_2(\mathbf{x}). \quad (5.2)$$

Jako řešení těchto funkcí se používají proximální operátory, které jsou řešením minimalizační úlohy [3]:

$$\text{prox}_f(\mathbf{x}) = \arg \min_{\mathbf{y} \in \mathbb{R}^N} f(\mathbf{y}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2. \quad (5.3)$$

Pro declipping budou funkcemi v minimalizační úloze (5.2) ℓ_1 -norma vektoru a indikátorová funkce ι_C . Proximálním operátorem pro $\|\mathbf{x}\|_1$ bude měkké prahování (*soft thresholding*) a pro ι_C projekce na množinu $C \subset \mathbb{R}^N$.

Měkké prahování je definováno jako [3]:

$$\text{soft}(\mathbf{x}) = \begin{cases} \mathbf{x} - \underline{\omega} & \text{pokud } \mathbf{x} < \underline{\omega}, \\ 0 & \text{pokud } \mathbf{x} \in \langle \underline{\omega}, \bar{\omega} \rangle, \\ \mathbf{x} - \bar{\omega} & \text{pokud } \mathbf{x} > \bar{\omega}. \end{cases} \quad (5.4)$$

Převodní charakteristika pro $\omega = 1$ je zobrazena na obrázku 5.1. Indikátorová funkce ι_C je definována jako

$$\iota_C(\mathbf{x}) = \begin{cases} 0 & \text{pro } \mathbf{x} \in C, \\ +\infty & \text{pro ostatní,} \end{cases} \quad (5.5)$$

a jejím proximálním operátorem je projekce na množinu přípustných řešení C , tedy $\text{proj}_C\{\mathbf{x}\}$.

Neomezený tvar pro úlohu declippingu bude následující:

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{w}\mathbf{c}\|_1 + \iota_C(\mathbf{c}). \quad (5.6)$$

Množina přípustných řešení C je množina všech koeficientů \mathbf{c} splňujících podmínky (5.1). Tuto množinu tedy lze zapsat jako:

$$C = \{\mathbf{c} | \mathbf{M}_R \mathbf{G} \mathbf{c} = \mathbf{M}_R \mathbf{y}_c, \mathbf{M}_H \mathbf{G} \mathbf{c} \geq \theta_H, \mathbf{M}_L \mathbf{G} \mathbf{c} \leq \theta_L\}, \quad (5.7)$$

kde \mathbf{c} je vektor koeficientů Gaborovy transformace.

Uvažujme tedy množinu C splňující tuto podmínku.

Indikátorová funkce ι_C má tu vlastnost, že pokud nalezené řešení leží v množině C , bude rovna 0, a pokud nalezené řešení bude ležet mimo množinu řešení C , hodnota ι_C bude $+\infty$. Pro úlohu declippingu to znamená, že pokud bude řešení z množiny C , v rovnici 5.6 bude figurovat pouze minimalizace $\|\cdot\|_1$. Naopak pokud nebude ležet v množině C , nabude minimalizace hodnot $+\infty$, čímž je zaručeno, že řešení nebude minimální [3].

5.2 Návrh pro synteuzující model

Pro synteuzující model bude k hledání řídkých řešení použit algoritmus Douglas-Rachford (DR), který pro úlohu (5.6), kde vystupuje minimalizace ℓ_1 -normy a indikátorové funkce ι_C , předpokládá obě funkce konvexní a diferencovatelné, což je splněno.

S využitím konkrétních proximálních operátorů pro declipping můžeme tento algoritmus zapsat následovně:

Algoritmus 1: Algoritmus Douglas-Rachford pro Declipping

Input: Funkce f_1 , f_2 a váhy koeficientů \mathbf{w} funkce f_2

Output: $\hat{\mathbf{c}}^{(n)}$

Nastav parametry $\epsilon \in (0, 1)$, $\lambda > 0$, $\mathbf{c}_0 \in \mathbb{C}^N$;

Nastav iterační čítač $n = 0$;

repeat

$\hat{\mathbf{c}}^{(n)} = \text{proj}_C \mathbf{c}^{(n)}$;

$\sigma^{(n)} \in \langle \epsilon, 2 - \epsilon \rangle$;

$\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} + \sigma^{(n)} (\text{soft}_{\lambda \mathbf{w}}(2\hat{\mathbf{c}}^{(n)} - \mathbf{c}^{(n)}) - \mathbf{c}^{(n)})$;

$n \leftarrow n + 1$;

until *convergence*;

return $\hat{\mathbf{c}}^{(n)}$

kde \mathbf{w} značí vektor použitého psychoakustického modelu na jejíž hodnotu je aktuální hodnota vstupního signálu prahována.

Na obrázku 5.2 je navržené blokové schéma algoritmu pro úlohu declippingu pomocí DR. Psychoakustický model je označen standardem MPEG-1.

5.3 Návrh pro analyzující model

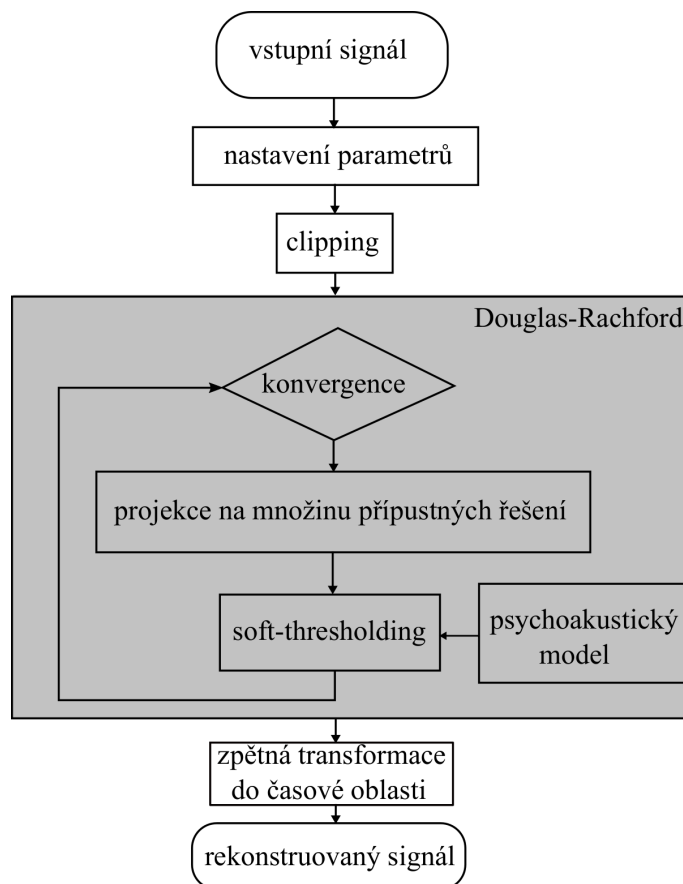
Pro analyzující model je třeba řešit úlohu

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{w}G^*\mathbf{x}\|_1 + \iota_{\Gamma}(\mathbf{x}) \quad (5.8)$$

kde G^* představuje analyzující operátor, ι_{Γ} je indikátorová funkce, pro kterou platí:

$$\iota_C(\mathbf{x}) = \begin{cases} 0 & \text{pro } \mathbf{x} \in \Gamma, \\ +\infty & \text{pro ostatní} \end{cases} \quad (5.9)$$

a \mathbf{w} je váhovací vektor psychoakustického modelu.



Obr. 5.2: Blokové schéma declippingu pomocí syntezujícího modelu

Pro tento model bude použit algoritmus Chambolle-Pock (CP), který řeší minimalizační úlohy typu

$$\min_{\mathbf{x}} f_1(\mathbf{x}) + f_2(L\mathbf{x}), \quad (5.10)$$

kde f_1 a f_2 jsou konvexní funkce, které nemusí být hladké a L je lineární operátor.

Při přizpůsobení tohoto problému úloze declippingu, bude f_1 indikátorová funkce a její proximální operátor bude projekce na množinu přípustných řešení Γ , platí tedy $\text{prox}_{f_1} = \text{proj}_{\Gamma}$. Funkce f_2 pak představuje váhovanou ℓ_1 -normu $\|\mathbf{w} \cdot\|_1$, jejíž proximální operátor je měkké prahování, což lze zapsat jako $\text{prox}_{f_2} = \text{soft}_{\mathbf{w}}$, kde \mathbf{w} je vektor vah. Jelikož však v CP nevystupuje přímo funkce f_2 ale Fenchell-Rockafellarovo sdružení f_2^* , je nutné jeho proximální operátor odvodit následovně:

$$\begin{aligned} \text{prox}_{\alpha f_2^*}(\mathbf{u}) &= \mathbf{u} - \alpha \cdot \text{prox}_{f_2/\alpha}(\mathbf{u}/\alpha) = \mathbf{u} - \alpha \cdot \text{prox}_{\|\mathbf{w} \cdot\|_1/\alpha}(\mathbf{u}/\alpha) = \\ &= \mathbf{u} - \alpha \cdot \text{soft}_{\mathbf{w}/\alpha}(\mathbf{u}/\alpha) = \mathbf{u} - \alpha \cdot \frac{1}{\alpha} \cdot \text{soft}_{\mathbf{w}}(\mathbf{u}) = \mathbf{u} - \text{soft}_{\mathbf{w}}(\mathbf{u}) = \text{clip}_{\mathbf{w}}(\mathbf{u}). \end{aligned} \quad (5.11)$$

Funkce clip pak má následující podobu:

$$\text{clip}_{\mathbf{w}}(\mathbf{x}) = \text{sgn}(\mathbf{x}) \odot \min(|\mathbf{x}|, \mathbf{w}), \quad (5.12)$$

kde operátor \odot je násobení vektorů po složkách. Tato funkce je vlastně funkce hard-clip definovaná podle (1.1), jejíž převodní charakteristika je zobrazená v grafu 1.2.

Algoritmus Chambolle-Pock pro účely declippingu lze zapsat například takto:

Algoritmus 2: Algoritmus Chambolle-Pock pro Declipping

Input: Funkce f_1, f_2 , váhy koeficientů \mathbf{w} funkce f_2 , lineární operátor

$$G : \mathbb{C}^q \rightarrow \mathbb{C}^p$$

Output: $\hat{\mathbf{p}}^{(n)}$

Nastav parametry $\zeta, \sigma > 0, \gamma \in [0, 1]$;

Nastav počáteční hodnotu primární proměnné $\mathbf{p}^{(0)} \in \mathbb{C}^p$ a duální proměnné $\mathbf{q}^{(0)} \in \mathbb{C}^q$;

Nastav hodnotu výstupní proměnné $\hat{\mathbf{p}}^{(0)} = \mathbf{p}^{(0)}$;

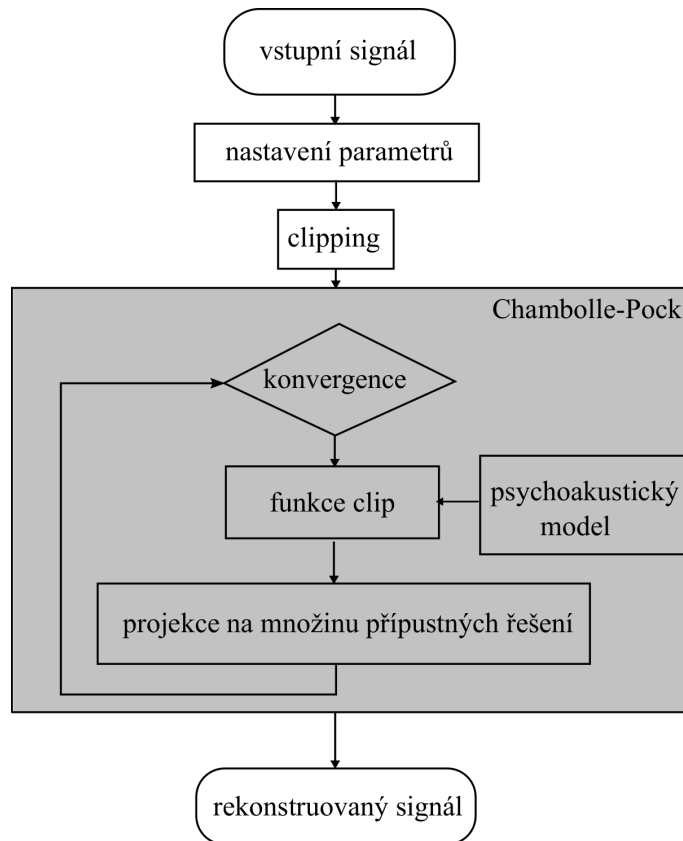
Nastav iterační čítač $n = 0$;

repeat

$$\left| \begin{array}{l} \mathbf{q}^{(n+1)} = \text{clip}_{\mathbf{w}} \left(\mathbf{q}^{(n)} + \sigma G^* \hat{\mathbf{p}}^{(n)} \right); \\ \mathbf{p}^{(n+1)} = \text{proj}_{\Gamma} \left(\mathbf{p}^{(n)} - \zeta G^* \mathbf{q}^{(n+1)} \right); \\ \hat{\mathbf{p}}^{(n+1)} = \mathbf{p}^{(n+1)} + \gamma \left(\mathbf{p}^{(n+1)} - \mathbf{p}^{(n)} \right); \\ n \leftarrow n + 1; \end{array} \right.$$

until convergence;

return $\hat{\mathbf{p}}^{(n)}$



Obr. 5.3: Blokové schéma declippingu pomocí analyzujícího modelu

Navržené blokové schéma na obrázku 5.3 představuje řešení declippingu pomocí CP. Schéma je stejné jako v případě DR (5.2) jen s tím rozdílem, že projekce na množinu přípustných řešení Γ a funkce clip jsou v opačném pořadí. Jelikož CP pracuje s analyzujícím modelem signálu, je výstupem algoritmu signál v časové oblasti a není třeba provádět další syntézu, jako v případě DR.

5.4 Projekce na množinu přípustných řešení

V této části budou definovány projekce pro syntezující a analyzující model. Nejprve je ale potřeba několik definic. Nejprve mějme prostor komplexních matic \mathbb{G} definovanou jako [3]:

$$\mathbb{G}^{m,n,N} = \{[\mathbf{B}_0 \dots \mathbf{B}_{N-1}] \mid \mathbf{B}_1 \in \mathbb{K}^{m \times n}\}, \quad (5.13)$$

kde n je sudé číslo. Matice \mathbf{B} jsou tedy vyskládány za sebe do řádku. Pro jeden řádek $\mathbb{G}^{n,N}$ tedy platí [3]:

$$\mathbb{G}^{n,N} = \left\{ \begin{bmatrix} \mathbf{b}_0 \\ \vdots \\ \mathbf{b}_{N-1} \end{bmatrix} \mid \mathbf{b}_i \in \mathbb{K}^n \right\}. \quad (5.14)$$

Pokud budeme předpokládat takovouto matici plné hodnosti $A \in \mathbb{G}^{m,n,N}$, pokud $m \leq n$ a $\mathbf{A}\mathbf{A}^*$ je diagonální a \mathbf{A} tedy tvoří těsný frame. Projekci vektoru $\mathbf{z} \in \mathbb{G}^{n,N}$ danou jako:

$$\text{proj}_{\{\mathbf{x} \mid \mathbf{A}\mathbf{x} \in [\mathbf{b}_1, \mathbf{b}_2]\}}(\mathbf{z}) := \arg_{\mathbf{u}} \min \|\mathbf{z} - \mathbf{u}\|_2 \quad \text{vzhledem k } \mathbf{A}\mathbf{u} \in [\mathbf{b}_1, \mathbf{b}_2] \text{ a } \mathbf{u} \in \mathbb{G}^{n,N}, \quad (5.15)$$

můžeme vyjádřit pro syntezující model jako [3]:

$$\text{proj}_{\{\mathbf{x} \mid \mathbf{A}\mathbf{x} \in [\mathbf{b}_1, \mathbf{b}_2]\}}(\mathbf{z}) = \mathbf{z} - \mathbf{A}^+(\mathbf{A}\mathbf{z} - \text{proj}_{[\mathbf{b}_1, \mathbf{b}_2]}(\mathbf{A}\mathbf{z})), \quad (5.16)$$

kde $\mathbf{b}_1 \leq \mathbf{b}_2$ a $\mathbf{b}_1, \mathbf{b}_2 \in \tilde{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. Tato projekce může být také vypočtena jako [3]:

$$\text{proj}_{[\mathbf{b}_1, \mathbf{b}_2]}(\mathbf{y}) = \min(\max(\mathbf{b}_1, \mathbf{y}), \mathbf{b}_2). \quad (5.17)$$

Pro tuto projekci je nutné vektory \mathbf{b}_1 a \mathbf{b}_2 definovat jako:

$$(\mathbf{b}_1)_i = \begin{cases} \mathbf{y}_i & \text{pokud } i \in M_R, \\ \theta_H & \text{pokud } i \in M_H, \\ -\infty & \text{pokud } i \in M_L. \end{cases} \quad (\mathbf{b}_2)_i = \begin{cases} \mathbf{y}_i & \text{pokud } i \in M_R, \\ \infty & \text{pokud } i \in M_H, \\ \theta_L & \text{pokud } i \in M_L, \end{cases} \quad (5.18)$$

kde i označuje i -tý vzorek signálu. Vektor \mathbf{b}_1 tedy omezuje signál „zdola“, \mathbf{b}_2 „shora“.

V případě analyzujícího modelu pro množinu Γ definovanou jako:

$$\Gamma = \{\mathbf{x} \mid M_R \mathbf{x} = M_R \mathbf{y}_c, M_H \mathbf{x} \geq \theta_H M_L \mathbf{x} \leq \theta_L\}, \quad (5.19)$$

může projekce vektoru vzorků v časové oblasti \mathbf{x} vypadat následovně:

$$\text{proj}_{\Gamma}(\mathbf{z}) = \begin{cases} \mathbf{y}_{cn} & \text{pro } n \in M_R, \\ \max(\mathbf{z}_n, \theta_c) & \text{pro } n \in M_H \\ \min(\mathbf{z}_n, -\theta_c) & \text{pro } n \in M_L. \end{cases} \quad (5.20)$$

Kde \mathbf{z}_n označuje n -tý vzorek vektoru \mathbf{z} .

Vzorky na spolehlivých pozicích jsou tedy nahrazeny vzorky vstupního signálu na pozicích M_R . Pokud je hodnota vzorků na pozicích M_H menší než θ_c , jsou nahrazeny hodnotou θ_c , vzorky na pozicích M_L větší než $-\theta_c$ jsou opět touto hodnotou nahrazeny.

6 PROGRAMOVÉ ŘEŠENÍ

V této kapitole je popsána struktura a funkce jednotlivých částí implementace obou algoritmů pro declipping v prostředí Matlab. Program byl vytvořen ve verzi R2016a s využitím toolboxu LTFAT (*The Large Time-Frequency Analysis Toolbox*). Jedná se o nástroj pro práci s časově-frekvenční syntézou a analýzou. Tento toolbox je ke stažení na webové stránce <<https://http://ltfat.github.io/>> a pro spuštění programu je nezbytný.

Tato kapitola je rozdělena do několika částí, z nichž každá je věnována jednomu z m-souborů, ze kterých je samotný program pro declipping složen. U každého m-souboru je vysvětlena jeho funkce, účel jednotlivých parametrů a možnosti jejich nastavení.

6.1 Hlavní soubor

Hlavním souborem programu pro declipping je `main.m`, který je rozdělen do sedmi částí. Nejprve je načten zvukový vzorek, ten následně normalizován. Poté jsou nastaveny parametry pro jednotlivé funkce. Poté je na vstupním signále provedena tvrdá limitace na předem nastavenou hodnotu. V další části je zkonstruován frame podle přednastavených parametrů. Poslední věc, která je provedena ještě před samotných proximálním algoritmem, je vytvoření vektoru `vah`. Ten potom společně se vstupním signálem a framem vstupuje do zvoleného algoritmu. Po samotné restauraci je v poslední části signál přehrán, vykreslen do grafu a jsou vypsány hodnoty SDR a délka času potřebná k restauraci.

6.1.1 Načtené zvukového souboru

Vstupním souborem musí být jednokanálový zvukový soubor ve formátu WAVE (*.wav) nebo jiném zvukovém formátu, např. MP3, FLAC, AAC či MPEG-4.

Načtení vzorku vstupního zvukového souboru probíhá pomocí funkce `audioread`, jejímž výstupem je vektor vzorků a hodnota vzorkovací frekvence. Dále je do struktury `param` uložena délka vektoru vstupního signálu `dataL` pomocí funkce `length`.

Takto načtený zvukový soubor je následně normalizován, aby jeho maximální absolutní hodnota byla 1.

6.1.2 Nastavení parametrů

V této části jsou nastaveny parametry jednotlivých funkcí programu. Parametry jsou uloženy ve struktuře `param`, která je následně použita na vstupu většiny funkcí.

Prvním parametrem je `thr`, což je hodnota prahu tvrdé limitace $\pm\theta_c$. Dalším parametrem je `lambda`, jakožto hodnota prahu měkkého prahování u Douglas-Rachfordova algoritmu.

Dále jsou nastavovány parametry potřebné pro konstrukci framu. Nejprve je to proměnná `frametype`, která udává typ framu. V našem případě je nastavená na hodnotu `'dgtreal'` pro Gaborův frame. Dále je zde proměnná `wtype`, což je druh použitého okna. Šířka okna `w`, posuv okna `a` a počet kanálů `M` jsou pak parametry potřebné pro samotné DGT.

V proměnné typu string `algorithm` se volí, který proximální algoritmus bude pro declipping použit. Pro syntezující model (viz 5.2) je to možnost `'Douglas-Rachford'` a pro model analyzující je to `'Chambolle-Pock'`, který je popsán v části 5.3. Parametr `maxit` udává maximální počet iterací vybraného algoritmu.

Poslední dva parametry se týkají nastavení váhování. Proměnná `vah` nabízí tři možnosti. Při nastavení hodnoty `'1'` je vektor koeficientů DGT váhován pouze hodnotou `lambda`. Další možností je `'ATH'`, což je váhování křivkou absolutního prahu slyšitelnosti. Poslední možností je `'MASK'`, čímž je nastaveno váhování pomocí vektoru maskovacího prahu.

6.1.3 Hard clipping

V této části je volána funkce `hard_clip`, která provádí limitaci vstupního signálu na předem nastavené hodnotě $\pm\theta_c$ v parametru `thr`. Vstupními parametry této funkce jsou tedy vstupní zvukový soubor `data` a struktura `param`. Funkce vrací tvrdě clipovaný signál v proměnné `data_clip` a strukturu obsahující masky clipovaného signálu.

6.1.4 Konstrukce framu

V této části je pomocí funkcí toolboxu LTFAT konstruován frame podle již dříve nastavených parametrů. Samotnou konstrukci framu obstarává funkce `frame`, jejíž výstup je následně uložen ve struktuře `param.F`. Jelikož při hledání řídkých řešení je ideální pracovat s parsevalovsky těsnými framy, je následně nutné použít funkci `frametight`, která vytvoří Parsevalův těsný frame.

Funkce `frameaccel` pak předpočítá struktury framu pro rychlejší analýzu a syntézu signálu.

Posledním krokem v této části je kontrola těsnosti vytvořeného framu, kterou provádí funkce `framebounds`, která vrací hodnoty mezi framu. Ty se pro parsevalovsky těsné framy rovnají jedné.

Váhování

Zde dochází k vytvoření váhovacího vektoru `vah_vec`, který je později použit pro proximální operátory. Jsou zde tři možnosti váhování podle nastavené hodnoty proměnné `vah`, uložené ve struktuře `param`. Při nastavení hodnoty na 'ATH' je konstruována křivka absolutního prahu slyšitelnosti pomocí funkce `ath` a při zadané hodnotě 'MASK' je to vektor maskovacích křivek vytvořený funkcí `masking2`, jejímž autorem je doc. Jiří Schimmel. Pokud je hodnota proměnné `vah` nastavená na hodnotu '1', probíhá prahování algoritmu Douglas-Rachford pouze pomocí proměnné `lambda` a pro Chambolle-Pock je prahem funkce `clip` hodnota 1.

Proximální algoritmus

V této části, jak již název napovídá, je volán samotný proximální algoritmus. Dle hodnoty nastavené v proměnné `algorithm` je to Douglas-Rachfordův algoritmus nebo algoritmus Chambolle-Pock. Při první variantě je volána funkce `douglas.m` (viz 6.4), v případě druhém se jedná o funkci uloženou m-souboru `chambolle.m` (6.5).

Jsou zde také nastaveny funkce jednotlivých proximálních operátorů, které jsou uloženy do struktur `f1` a `f2`. Tyto struktury následně vstupují do funkce vybraného proximálního algoritmu.

Přehrání a vykreslení grafů

Poslední částí hlavního souboru je úsek, ve kterém dochází k vykreslení grafů, výpočtům hodnot ΔSNR pomocí funkce `snr.m` a přehrání restaurovaného zvukového signálu pomocí funkce `soundsc`.

6.2 Funkce pro hard clipping

Omezení signálu na předem nastavení hodnotě $\pm\theta_c$ obstarává funkce `hard_clip`.

V této funkci jsou nejprve do struktury `mask` uloženy masky clippovaného signálu, které jsou později použity při projekci na množinu přípustných řešení. Jedná se o pozice jednotlivých vzorků vstupního signálu, uložených ve vektoru logických jedniček a nul. Pozice, na kterých jsou hodnoty větší než θ_c jsou uloženo do proměnné `Mh`, pozice vzorků menších než $-\theta_c$ zase do proměnné `Ml`. Pozice vzorků, které zůstanou po limitaci nezměněné, jsou uloženy jako `Mr`.

Následně dochází k samotné limitaci, kdy je limitovaný signál uložen do proměnné `data_clip`, která je spolu se strukturou `mask` výstupem této funkce.

6.3 Váhovací funkce

Jak již bylo zmíněno, konstrukci váhovacího vektoru obstarávají dvě funkce, v závislosti na hodnotě proměnné `vah`.

Jednou variantou je použití funkce `ath.m`, na jejímž výstupu je vektor ATH (*The absolute threshold of hearing*). V této funkci je možné nejprve nastavit tři parametry. Při nastavené parametru `normalize` na hodnotu 1, výsledná křivka bude normalizována. Při potřebě převést výslednou křivku z hodnot SPL na hodnoty akustického tlaku, je potřeba nastavit nenulovou hodnotu proměnné `pascal`. Posledním parametrem je `dyn`, který udává maximální hodnotu v decibelech, na které je následně tato křivka tvrdě limitována za účelem snížení dynamického rozsahu.

Následně je zde dle vzorce (4.1) sestavena křivka ATH s počtem vzorků odpovídajícím počtu nastavených kmitočtových kanálů `fram` a vzorkovací frekvenci. Takto vznikne váhovací křivka pro jeden posuv okna. Aby bylo možné váhovat celý vektor koeficientů DGT, je nutné křivku ATH poskládat za sebe do jednoho vektoru s délkou odpovídající danému vektoru koeficientů. O to se stará funkce `repmat` z Matlab knihovny.

Druhou variantou je použití psychoakustického modelu, který generuje funkce `masking2.m`. Tato funkce vytváří maskovací práh vstupního signálu podle psychoakustického modelu MPEG. Princip tvoření maskovacích křivek je popsán v části 4.3, není tedy potřeba ho znovu vysvětlovat. Vstupními parametry jsou zvukový signál, hodnota vzorkovací frekvence a druh psychoakustického modelu, který má být použit. Opět lze nastavit dynamický rozsah výsledné křivky nebo zapnout lineární normalizaci.

Funkce ze vstupních parametrů vyčte počáteční a finální frekvenci vstupního signálu a také počet jeho vzorků. Jako maximální hodnota frekvence je brána hodnota $f_{\max} = f_s/2$, tedy poloviční hodnota vzorkovací frekvence zvukového souboru.

6.4 Algoritmus Douglas-Rachford

Funkce `douglas.m` je implementace Douglas-Rachfordova algoritmu popsaného v části 5.2, proto zde nebude znovu jeho funkce popisována. Vstupními argumenty této funkce jsou saturovaný signál, funkce `f1` a `f2` s proximálními operátory, váhovací vektor a struktura `param`. Z této struktury si funkce načte všechny potřebné parametry, jako například `frame` `F`, úroveň soft-thresholdingu `lambda` nebo maximální počet iterací `maxit`.

Na začátku funkce je opět sekce nastavení, obsahující několik parametrů. Prvním je parametr `verbose`, který po nastavení na hodnotu 1 zapne vypisování do konzoly. Následuje parametr `sigma` určující velikost kroku v jednotlivých iteracích

a posledním parametrem je `coefstop`, který určuje minimální rozdíl mezi koeficienty jednotlivých iterací, po jehož dosažení se algoritmus ukončí. Všechny tyto algoritmy jsou pro přehlednost ukládány do struktury `param.DR`.

Počátečním bodem algoritmu jsou koeficienty saturovaného signálu, ty jsou spočítány analýzou pomocí framu `F`. Lineární analýza a syntéza je v jednotlivých iteracích počítána pomocí funkcí `frana` a `frsyn` toolboxu `LTFAT`. Po překročení minimálního rozdílu koeficientů nebo maximálního počtu iterací se algoritmus ukončí a výsledný restaurovaný signál je nakonec oříznut na velikost původního signálu.

6.5 Algoritmus Chambolle-Pock

Algoritmus Chambolle-Pock, popsáný v části 5.3, je implementován v m-souboru `chambolle.m`. Vstupní parametry jsou stejné jako v případě algoritmu Douglas-Rachford. Tedy saturovaný signál, vektor `vah`, struktura `param` a funkce proximálních operátorů `f1`, `f2` upravené pro potřeby analyzujícího algoritmu.

Samotnému algoritmu pro declipping opět předchází část s nastavením parametrů funkce, kterou jsou následně uloženy do struktury `param.CP`. Stejně jako v případě Douglas-Rachfordova algoritmu jsou zde parametry `verbose`, `maxit` a ukončovací parametr `coefstop`. Dále jsou zde interní parametry samotného proximálního algoritmu `zeta`, `sigma`, `gamma`, které jsou definované v části 5.3.

Jako ukončovací parametr je zde opět nastavena relativní změna koeficientů mezi iteracemi nebo počet iterací. Po překročení jednoho z těchto parametrů se algoritmus ukončí a vypíše do konzoly aktuální rozdíl mezi koeficienty a dobu trvání algoritmu. Na závěr je restaurovaný signál oříznut na délku původního saturovaného signálu. Výstupem celé funkce je restaurovaný signál a konečný rozdíl koeficientů.

6.6 Projekce na množinu přípustných řešení

Jedním z proximálních operátorů obou algoritmů je projekce na množinu přípustných řešení (viz 5.1). K tomuto účelu byla vytvořena funkce `proj.m`, která je volaná v každé iteraci proximálního algoritmu. Vstupními argumenty této funkce jsou saturovaný signál `data_clip`, aktuální rekonstrukce signálu a struktura `param`, ze které si funkce čte předem uložené masky signálu.

V této funkci probíhá syntéza pro analyzující algoritmus popsána v části 5.4.

Pro účely syntezujícího algoritmu byla vytvořena funkce `proj_dr.m` provádějící projekci podle vzorce (5.16). Tato funkce si v jedné fázi také volá funkci `proj_dr.m`.

7 VÝSLEDKY TESTOVÁNÍ

V této kapitole budou popsány výsledky testování obou výše popsaných algoritmů pro declipping.

Nejprve zde budou krátce popsány zvukové signály, na kterých byl declipping testován. V další části budou porovnány výsledky různých nastavení váhování pro oba algoritmy. Budou srovnány výsledky bez použití váhování, s využitím křivky ATH a s využitím psychoakustického modelu.

V následující části bude provedeno porovnání výsledků implementací Douglas-Rachfordova a Chambolle-Pockova algoritmu a jejich výpočetní náročnosti.

Výsledky budou porovnávány nejprve pomocí hodnot ΔSNR a v závěru této kapitoly budou shrnuty výsledky poslechových testů.

7.1 Testování

Testování probíhalo na pěti jednokanálových zvukových signálech. Jednotlivé vzorky byly: mluvené slovo, znělka, zpěv s doprovodem klavíru, kontrabas a akustická kytara. Tyto nahrávky byly ve formátu *.wav a jejich vzorkovací frekvence byla 16 kHz.

Testování probíhalo postupně pro úrovně prahu $\pm\theta_c = \{0,1; 0,2; \dots; 0,9\}$. Ukončovací parametr algoritmů byl nastaven na 500 iterací, protože jak bylo vyzorováno, při větším počtu iterací se již hodnoty ΔSNR nijak znatelně nemění (viz graf 7.1). Další možností, jak došlo k ukončení algoritmu, bylo dosažení hodnoty rozdílu koeficientů menší než 0,00001. V této situaci již bylo jasné, že algoritmus dokonvergoval.

7.1.1 Ukazatel ΔSNR

Jak již bylo výše zmíněno, výsledné rekonstrukce budou porovnávány pomocí hodnot SNR, tedy velikosti odstupů užitečného signálu od šumu.

Vzorec pro výpočet této hodnoty je:

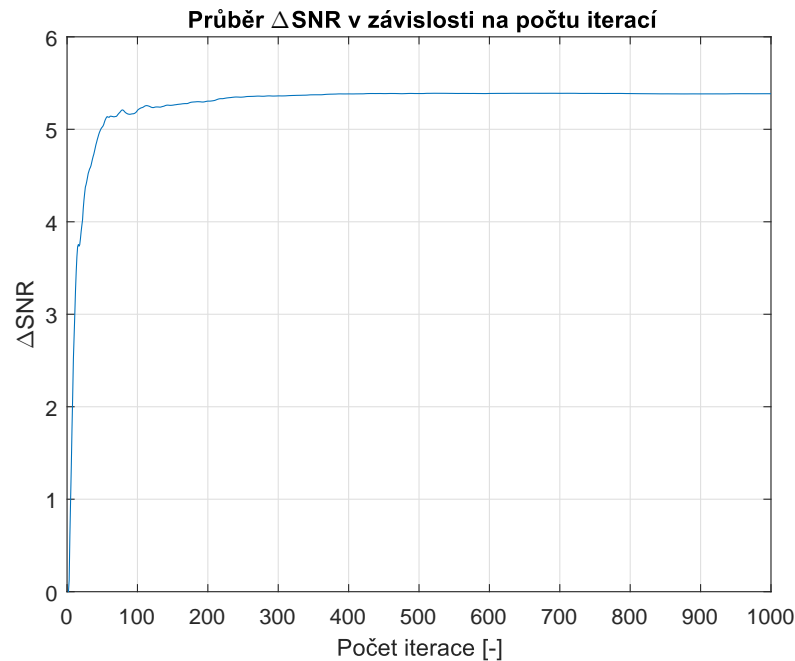
$$\text{SNR}(\mathbf{u}, \mathbf{v}) = 20 \log_{10} \left(\frac{\|\mathbf{u}\|_2}{\|\mathbf{u} - \mathbf{v}\|_2} \right), \quad (7.1)$$

kde \mathbf{u} značí vektor vzorků původního signálu a \mathbf{v} toho restaurovaného. Pro hodnocení výsledků však nebude použit ukazatel SNR, ale ΔSNR . Ten je definován jako rozdíl SNR rekonstruovaného a clipovaného signálu. Vzhledem k tomu, že předpokládáme, že jsou vzorky těchto signálů na nepoškozených pozicích M_R shodné, jedná se tedy o vylepšení hodnot SNR pouze poškozených vzorků.

Tento ukazatel je tedy vypočten jako

$$\Delta\text{SNR} = \text{SNR}(\mathbf{y}, \hat{\mathbf{y}}) - \text{SNR}(\mathbf{y}, \mathbf{y}_c), \quad (7.2)$$

kde \mathbf{y} je původní nepoškozený signál, $\hat{\mathbf{y}}$ značí rekonstruovaný signál a \mathbf{y}_c signál clipovaný. Někdy tento ukazatel označován jako SDR, tedy odstup signálu od zkreslení.



Obr. 7.1: Typický průběh hodnot ΔSNR v závislosti na počtu iterací.

7.2 Výsledky rekonstrukce bez váhování

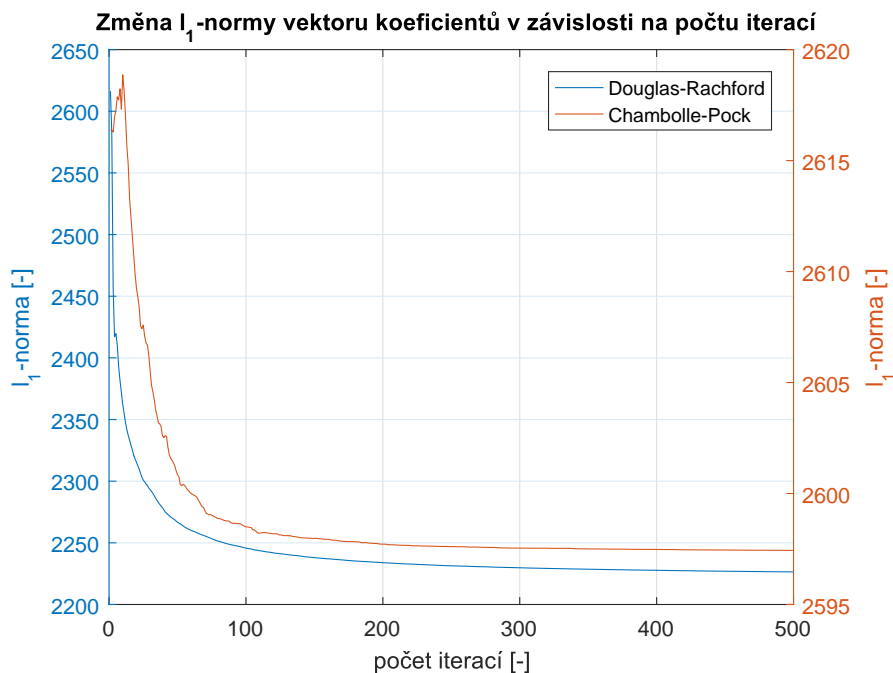
Nejprve byla testována restaurace pouze za přítomnosti parametru λ , jakožto parametru proximálního operátoru soft-thresholdingu pro syntezující model. V případě analyzujícího modelu kvůli Fenchell-Rockafellarovu sdružení byl signál funkcí clip prahován na hodnotu prahu 1. Jelikož se jedná o skaláry, všechny koeficienty DGT byly prahovány stejnou hodnotou. Ukázalo se, že Douglas-Rachford vykazoval nejlepší výsledky při nastavení parametru `lambda` na hodnotu 1.

Nastavování vnitřních parametrů proximálních algoritimů probíhalo empiricky. Nejlepších výsledků bylo pro syntezující algoritmus dosaženo při nastavení parametru `sigma=1`.

U algoritmu analyzujícího modelu byly vnitřní parametry nastaveny na hodnoty `sigma = 1`; `zeta = 0,5`; `gamma = 1,2`.

Na grafu 7.2 jsou znázorněny typické průběhy konvergence algoritimů Douglas-Rachford a Chambolle-Pock. Je zde vidět rychlost poklesu relativní změny koeficientů v závislosti na počtu iterací.

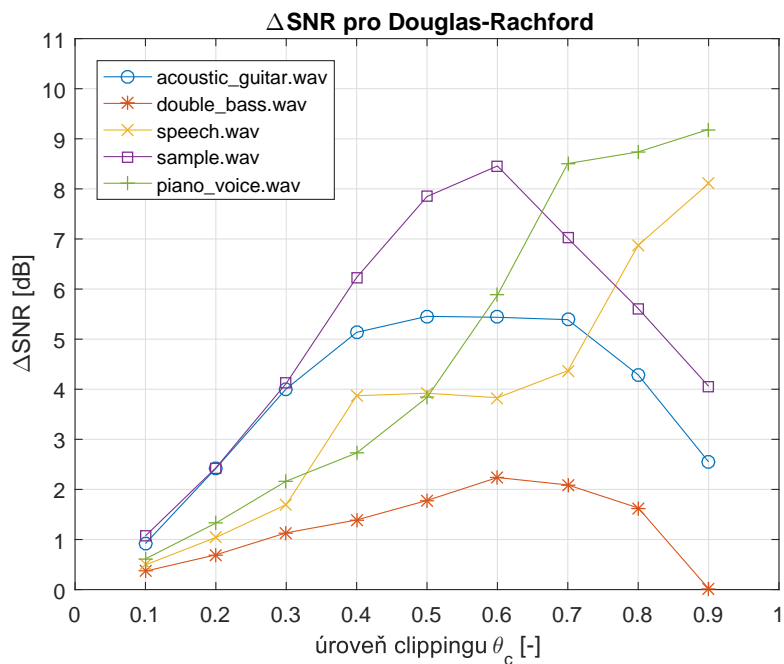
Z tohoto grafu je patrné, že algoritmus Chambolle-Pock konverguje mnohem než Douglas-Rachford, nicméně nemá tak pravidelný průběh.



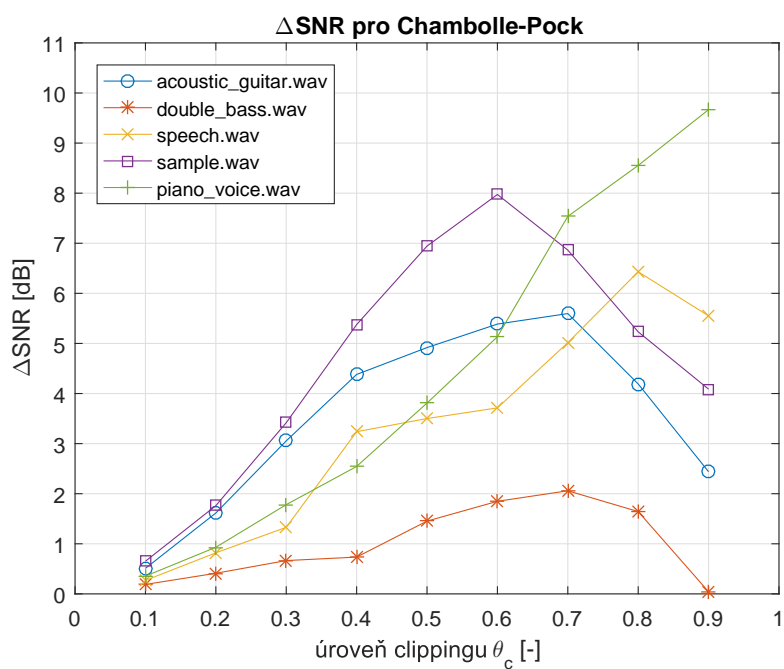
Obr. 7.2: Graf závislosti velikosti ℓ_1 -normy vektoru koeficientů na počtu iterací pro vzorek `acoustic_guitar.wav` a $\pm\theta_c = 0,6$.

V následujících dvou grafech jsou zobrazeny výsledky rekonstrukcí jednotlivých testovacích signálů v závislosti na úrovni θ_c . Výsledky jsou porovnány pomocí ukazatele ΔSNR . Graf 7.3 prezentuje výsledky restaurace pomocí DR algoritmu. Jak je z grafu patrné, s rostoucí hodnotou thresholdu rostou i hodnoty ΔSNR . Pokles při vyšších hodnotách θ_c je způsobený tím, že již samotná limitace nemá i přes normalizaci signálu na hodnotu SNR vliv. Zkreslení je u těchto zvukových vzorků při vyšších hodnotách clippingové prahu natolik malé, že výsledný rozdíl SNR, který je počítán jako rozdíl SNR clipovaného a restaurovaného signálu, je zanedbatelný. Výsledky algoritmu Chambolle-Pock jsou zobrazeny v grafu 7.4. Opět jsou zde zobrazeny výsledky restaurací všech testovacích signálů pro různé hodnoty thresholdu. Z grafu je patrné, že průběhy jednotlivých křivek ΔSNR jsou až na výjimky totožné s výsledky algoritmu DR. Je zde také patrné, že restaurované signály pomocí algoritmu CP dosahují nepatrně menších hodnot než výsledky dosažené algoritmem DR. Srovnání průměrných hodnot ΔSNR je znázorněno v grafu 7.5.

Doba potřebná pro nalezení řešení úlohy declippingu byla pro oba v podstatě shodná. Z poslechových testů je patrné, že pro některé zvukové vzorky, například `speech.wav` a `piano_voice.wav`, je patrné potlačení nižších frekvenčních složek a naopak mírné zvýraznění těch vysokých. Rekonstrukce pro nižší úroveň clippingové prahu obsahují velké množství šumové složky v celém rozsahu pásma.



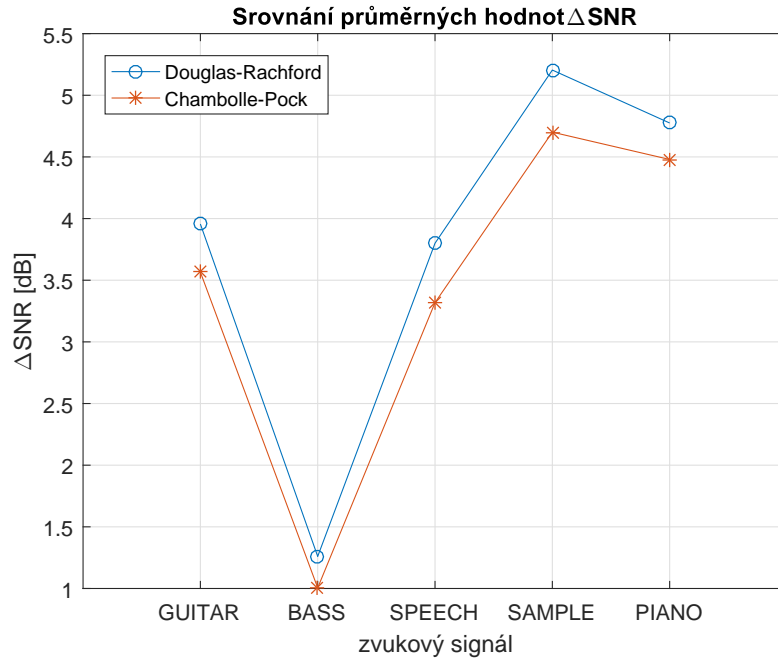
Obr. 7.3: Hodnoty ukazatele Δ SNR pro algoritmus Douglas-Rachford.



Obr. 7.4: Hodnoty ukazatele Δ SNR pro algoritmus Chambolle-Pock.

7.3 Výsledky restaurace s využitím váhovacích funkcí

V této části budou porovnány výsledky testování dvou druhů váhování koeficientů DGT. V rámci algoritmu Douglas-Rachford se jedná o váhy funkce soft-thresholding.



Obr. 7.5: Průměr hodnot ΔSNR jednotlivých testovacích zvuků rekonstruovaných pomocí obou algoritmů.

V případě algoritmu Chambolle-Pock se zase jedná o váhování funkce clip.

V prvním případě bylo testováno využití křivky ATH. Pro každý vzorek vektoru koeficientů byla tedy přiřazena jiná hodnota prahu v závislosti na frekvenci.

Druhou metodou prahování bylo využití psychoakustického modelu. V hlavním souboru byly nejprve vypočítány hodnoty maskovacích křivek pro celý zvukový signál, ze kterých byl vytvořen váhovací vektor.

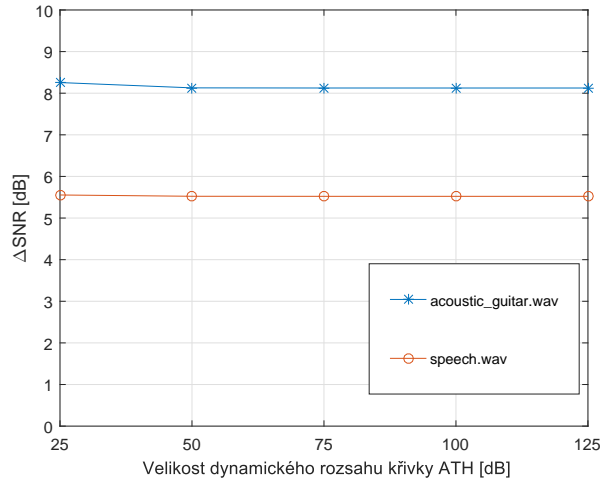
V obou těchto případech byl tedy vektor koeficientů váhován pro DR součinem parametru λ a váhovacího vektoru a pro CP součinem prahu funkce clip, tedy hodnotou 1 a váhovacího vektoru.

7.3.1 Váhování pomocí křivky ATH

Jak již bylo výše zmíněno, při váhování pomocí ATH je konstruována křivka podle vztahu (4.1). Tato křivka je dále převrácena podél osy x a upravena tak, aby vykazovala pouze kladné hodnoty.

Dalším krokem je omezení dynamického rozsahu funkce. Vliv různých dynamických rozsahů na výslednou hodnotu ΔSNR je znázorněn na grafu (7.6). Jak je z tohoto grafu patrné nemá tento parametr na kvalitu rekonstrukce žádný vliv, proto budou další měření pouze pro hodnotu dynamického rozsahu 25 dB.

Následně se ještě nabízela možnost aplikovat na křivku lineární normalizaci.



Obr. 7.6: Závislost výsledných hodnot ΔSNR na dynamickém rozsahu křivky ATH.

Při zavedení normalizace se základní úloha (5.8) upraví do tvaru

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|c\mathbf{w}G^*\mathbf{x}\|_1 + \iota_{\Gamma}(\mathbf{x}), \quad (7.3)$$

kde c je koeficient normalizace. Vzhledem k tomu, že se jedná o skalár, je možné jej vytknout před minimalizací a tato úloha se dá přepsat jako:

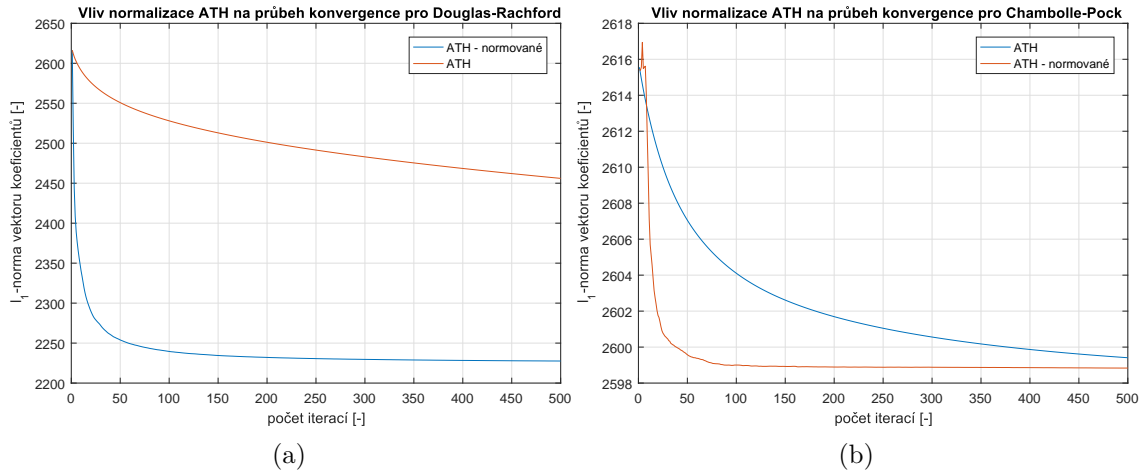
$$\hat{\mathbf{x}} = c \cdot \arg \min_{\mathbf{x}} \|\mathbf{w}G^*\mathbf{x}\|_1 + \iota_{\Gamma}(\mathbf{x}). \quad (7.4)$$

Z toho vyplývá, že normalizace nemá na výsledek minimalizace žádný vliv, jelikož hledáme argument tohoto minima. Z testování se však prokázalo, že při použití normalizované křivky ATH algoritmy konvergují rychleji. Porovnání rychlosti konvergence normované a nenormované křivky ATH, použité při rekonstrukci pomocí obou algoritmů, lze pozorovat v grafu 7.7. Jak již bylo ale výše zmíněno, na výslednou kvalitu rekonstrukce nemá lineární normalizace žádný vliv. Z důvodu rychlejší konvergence budeme dále pracovat s normovanou ATH.

Posledním aspektem, kterým je možné ovlivnit výsledek restaurace, je převedení jejích hodnot z decibelů na pascaly, tedy na hodnoty akustického tlaku. To se ukázalo jako velmi užitečné, protože rekonstrukce s použitím křivky akustického tlaku dosahovaly lepších výsledků ΔSNR .

Frekvenční závislost výsledné váhovací funkce pro normovanou křivku ATH je zobrazena v grafu 7.8.

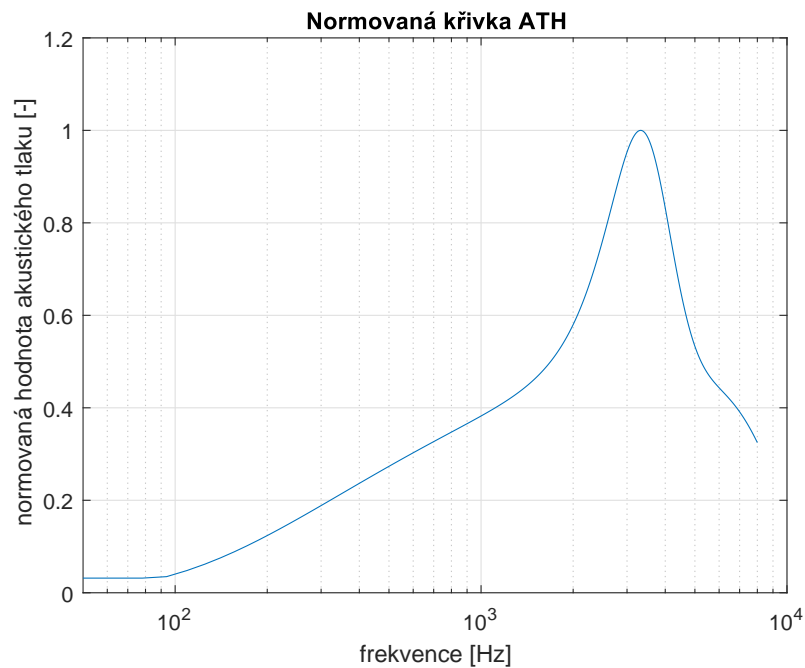
Srovnání výsledků při váhování oběma křivkami je znázorněno v grafech na obrázku 7.9. Jsou zde porovnány výsledky obou algoritmů v kombinaci s váhovacími křivkami. Jak je z těchto grafů patrné, ve všech případech dokazovaly jednotlivé algoritmy lepších výsledků při použití křivky ATH převedené na normované hodnoty akustického tlaku.



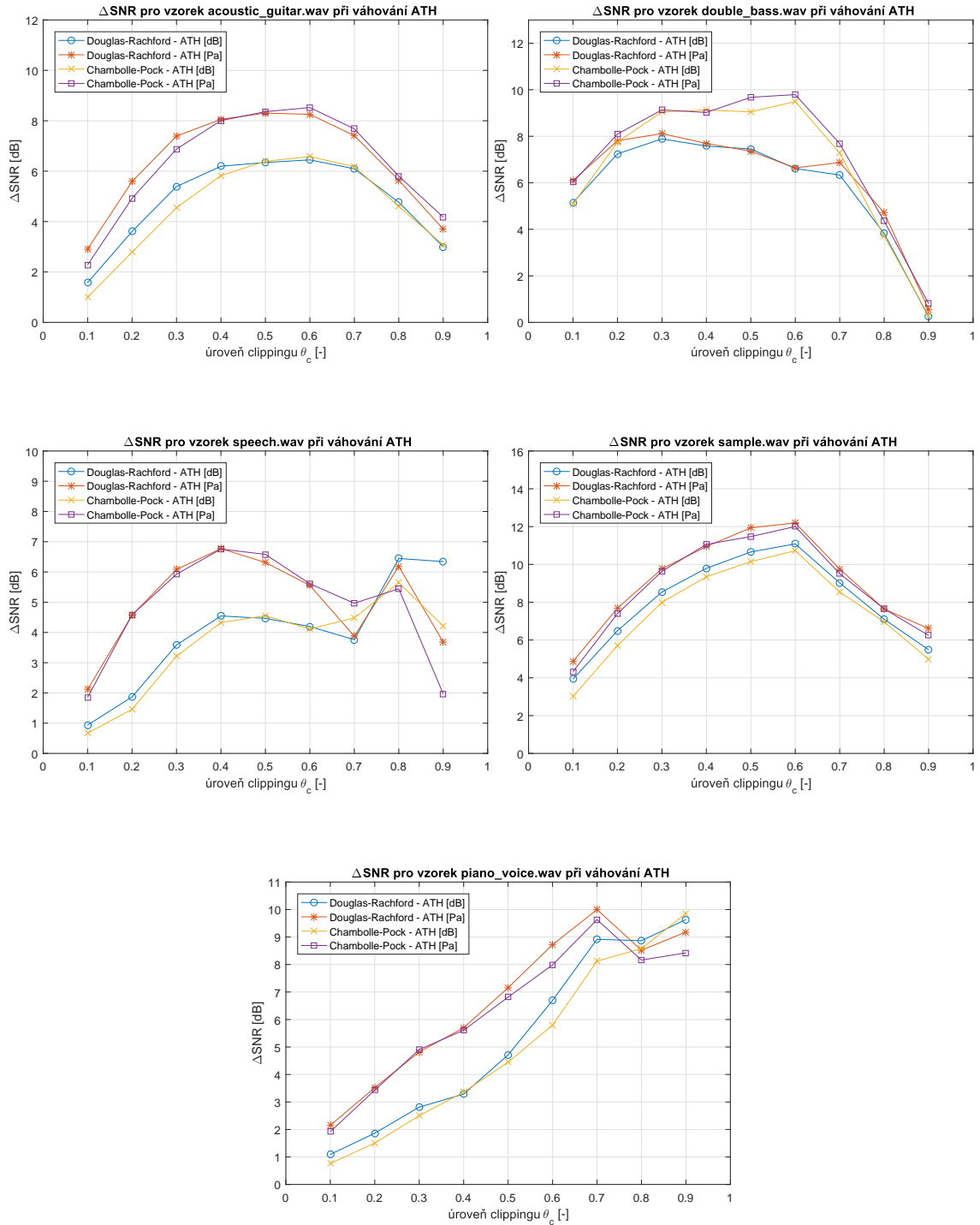
Obr. 7.7: Porovnání změn ℓ_1 -normy vektoru koeficientů s využitím lineární normalizace a bez ní pro algoritmus Douglas-Rachford a Chambolle-Pock.

Z grafů je také patrný nárůst hodnot ΔSNR oproti restauracím bez použití váhovací funkce.

V poslechu se u některých vzorků projevuje vyrovnaní dynamiky jednotlivých frekvenčních složek a naproti restauraci bez váhování je výsledný signál mnohem čitelnější.



Obr. 7.8: Frekvenční závislost normalizované křivky ATH převedené na hodnoty akustického tlaku [Pa].



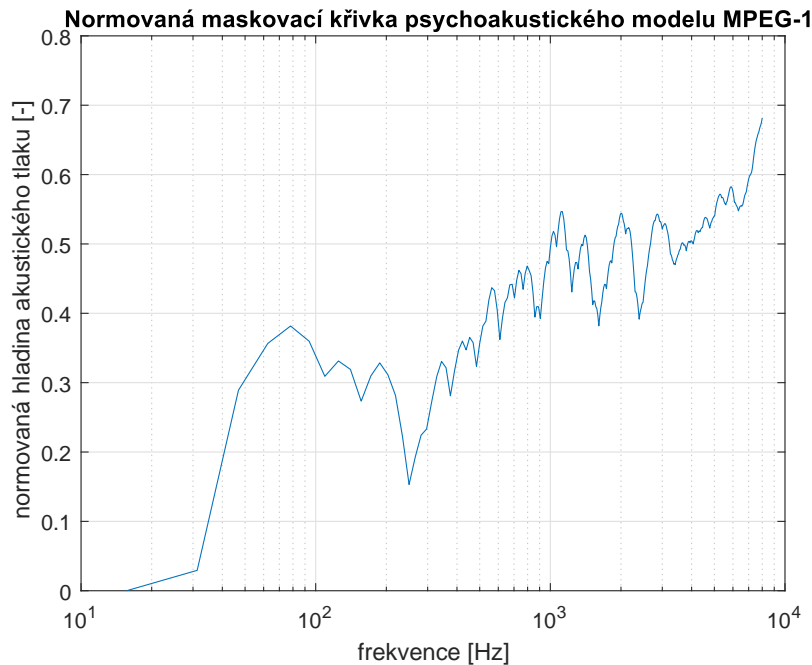
Obr. 7.9: Srovnání algoritmu Douglas-Rachford a Chambolle-Pock při váhování pomocí hodnot křivky ATH v decibelech a pascalech.

7.3.2 Váhování pomocí modelu MPEG

Posledním předmětem testování bylo využití samotného psychoakustického modelu MPEG jako vektoru váhovací funkce. Jak již bylo výše zmíněno, k tomuto účelu byla využita funkce `masking2.m`.

Nejprve bylo nutné nastavit její parametry. Empiricky bylo zjištěno, že nejlepších výsledků funkce vykazovala při omezení dynamického rozsahu hodnotou `dyn=60 dB`, lineární normalizaci a převedení hodnot výsledné maskovací křivky na decibely. Následně bylo zapotřebí tuto křivku stejně jako v případě ATH převrátit podél osy x a posuvem ve směru osy y zajistit pouze kladné hodnoty.

Vzniklá maskovací křivka je zobrazena v grafu 7.10. Jedná se o maskovací křivku prvních 1024 vzorků zvukového signálu `acoustic_guitar`. Již po několika testo-



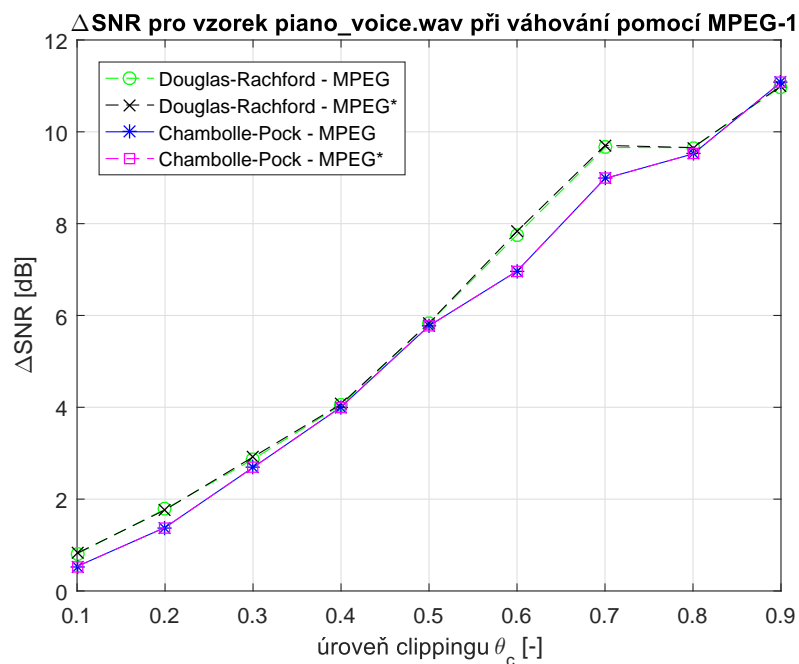
Obr. 7.10: Frekvenční závislost normalizované maskovací křivky modelu MPEG-1.

váních se ukázalo, že výsledné hodnoty ΔSNR signálů restaurovaných s využitím váhování pomocí této maskovací křivky dosahují nižších hodnot než v případě využití křivky ATH.

Problémem mohlo být, že maskovací křivka byla počítána z clipovaného signálu, kde mohlo docházet k rozdílnému kmitočtovému maskování než v případě původního signálu. Lepších výsledků tedy mohlo být dosaženo s využitím maskovací křivky vytvořené z již restaurovaného signálu bez využití váhování. Nejprve byl tedy audiosignál restaurován bez možnosti váhování a z výsledného restaurovaného signálu byla vytvořena maskovací křivka. Ta byla následně využita jako váhovací vektor při dalším testu.

Ve výsledku se však ukázalo, že tato metoda nepřinesla skoro žádné vylepšení hodnot ΔSNR a rozdíl nebyl znatelný ani při poslechu restaurovaných audiosignálů.

Ukázka výsledků declippingu pro maskovací křivku vytvořenou z clipovaného a restaurovaného signálu pro oba algoritmy je znázorněno v grafu 7.11. Od této



Obr. 7.11: Porovnání hodnot ΔSNR pro váhování pomocí maskovací křivky vytvořené z clipovaného signálu – MPEG a křivky vytvořené z dříve restaurovaného signálu – MPEG*.

metody bylo tedy opuštěno a všechny následné testy probíhaly pro maskovací křivku vytvořenou z clipovaného signálu.

Jak již bylo zmíněno, výsledky declippingu s využitím MPEG nedosahovaly průměrně tak vysokých hodnot ΔSNR , jako v případě ATH, oproti rekonstrukci bez použití váhování však došlo ke zlepšení. Poslechově byly výsledky pro vyšší clipovací prahy srovnatelné s rekonstrukcemi s využitím ATH, avšak pro nízké hodnoty θ_c byly restaurace výrazně zkrácené a objevovala se i nepříjemná zvýraznění vyšších frekvenčních složek.

Mírně lepších výsledků opět dosahoval algoritmus Douglas-Rachford. Porovnání obou algoritmů pro všechny testovací zvuky je znázorněno v grafem 7.12.

7.4 Shrnutí výsledků

V této části budou shrnuty výsledky testování obou algoritmů a bude zde srovnána varianta s váhování pomocí ATH a MPEG. Jako porovnávací parametr budou opět použity hodnoty ukazatele ΔSNR .

Rozdíly ΔSNR mezi výsledky obou algoritmů byly pozorovatelné pro variantu bez váhování. Při použití váhovacích funkcí byly dosažené hodnoty oběma algoritmy mnohem vyrovnanější. Při declippingu bez použití váhování dosahoval algoritmus Douglas-Rachford o něco lepších výsledků než Chambolle-Pock. Při obohacení o váhování pomocí křivky ATH se výsledky vyrovnaly. Výjimkou byl vzorek `double_bass.wav`, pro který vykazoval algoritmus Chambolle-Pock výrazně lepších výsledků než Douglas-Rachford. V případě váhování maskovací křivkou modelu MPEG-1 opět lépe vycházel DG algoritmus.

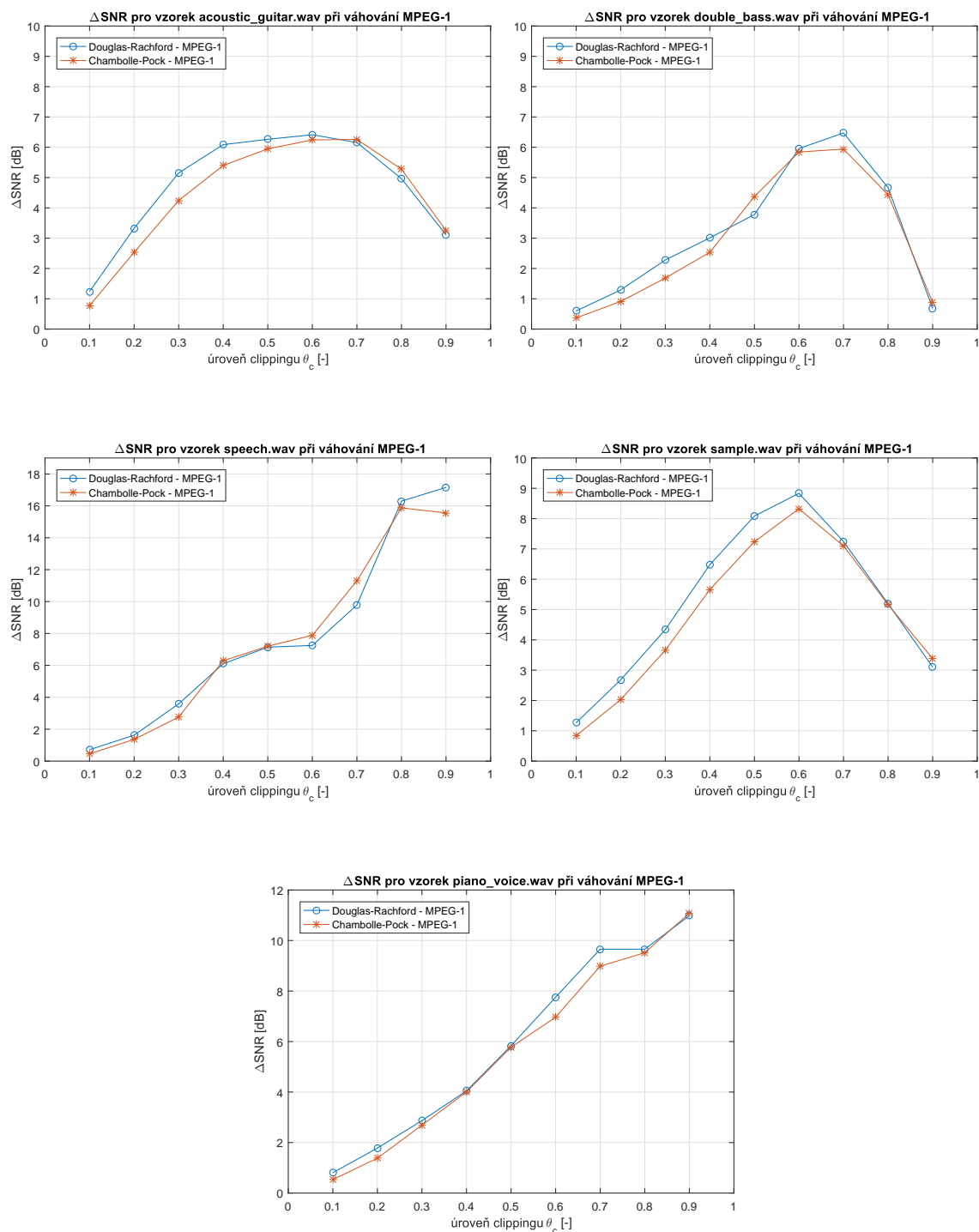
Porovnání všech variant je vyneseno v grafech 7.13. Jak je již na první pohled patrné, pro některé zvukové signály zaznamenávaly jednotlivé varianty velké rozdíly, zatímco u některých zvukových vzorků byly rozdíly minimální. Zajímavé je, že pro některé signály bez ohledu na použítou metody declippingu je výsledný tvar křivky závislosti hodnot ΔSNR na úrovni thresholdu stejný. Toto můžeme pozorovat například u vzorku `acoustic_guitar.wav`, `sample.wav` nebo `piano_voice.wav`.

U vzorků `speech.wav` a `double_bass.wav` však využití váhování způsobilo, že algoritmy dosahovaly pro některé úrovně clipovacího prahu mnohem lepších hodnot než dříve.

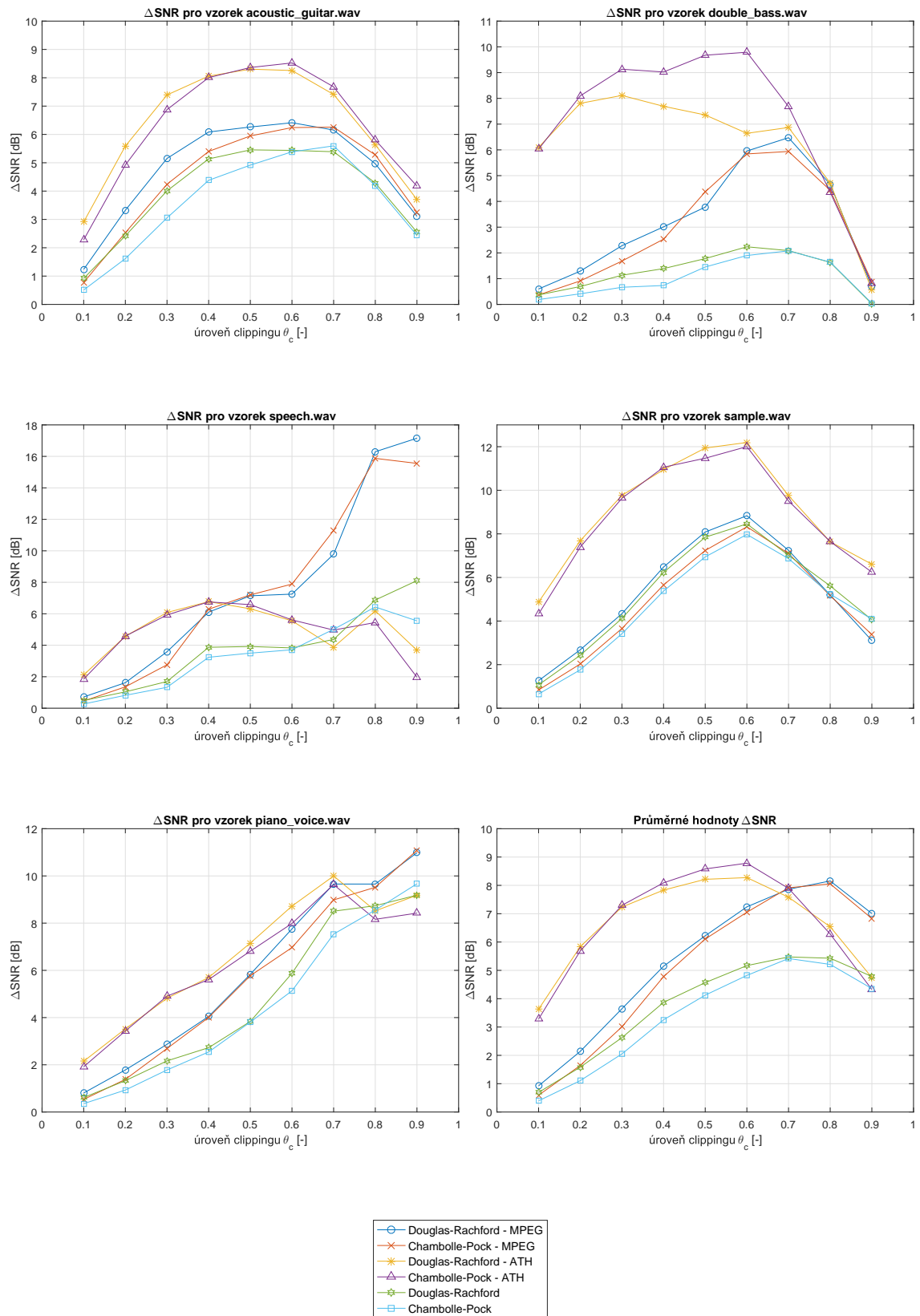
Jak je vidět z grafu pro vzorek `double_bass.wav`, při využití křivky ATH bylo oběma algoritmy dosaženo mnohem lepších výsledků než pomocí zbylých variant. Největší rozdíl je patrný hlavně pro nízké hodnoty θ_c .

U vzorku `speech.wav` bylo pro nízké úrovně θ_c opět nejlépe vycházelo váhování ATH, avšak pro $\theta_c \geq 0,5$ vyházely nejlepší hodnoty ΔSNR pro MPEG. Největší rozdíl můžeme pozorovat pro hodnoty $\theta_c = 0,8$ a $0,9$, kde byl rozdíl mezi MPEG a zbylými výsledky skoro 10 dB.

Poslední z grafů 7.13 je vyjádření průměrných hodnot ΔSNR pro jednotlivé metody v závislosti na úrovni clippingu. Z tohoto grafu je jasně patrné, že s výjimkou nejnižších úrovní zkrácení při $\theta_c = 0,8$ a $0,9$, bylo nejlepších výsledků dosaženo využitím křivky ATH coby váhovacího vektoru proximálních operátorů. V případě vysokých hodnot úrovně clippingu se zase nejvíce osvědčilo váhování maskovací křivkou vytvořenou pomocí psychoakustického modelu MPEG-1.



Obr. 7.12: Srovnání algoritmu Douglas-Rachford a Chambolle-Pock při váhování pomocí hodnot maskovací křivky vytvořené pomocí psychoakustického modelu MPEG-1.



Obr. 7.13: Srovnání obou testovaných algoritmů a metod váhování podle ukazatele ΔSNR .

7.5 Výsledky poslechových testů

Poté co jsme výsledky obou algoritmů zhodnotili pomocí objektivních parametrů, bylo nutné ještě provést subjektivní hodnocení pomocí poslechových testů.

V této části bude nejprve popsán způsob testování a poté zde budou prezentovány výsledky poslechových testů.

7.5.1 Způsob testování

Náplní poslechových testů byly výsledky obou testovaných algoritmů a jejich modifikací. Hodnoceno bylo tedy šest restaurovaných vzorků. Restaurace pomocí analyzujícího a syntezujícího algoritmu bez použití váhování, s váhováním pomocí křivky ATH a s váhováním maskovací křivkou modelu MPEG-1.

Testování probíhalo pro restaurace dvou úrovní clippingu pro hodnoty $\theta_c = 0,1; 0,4$. Tedy pro velmi zkreslené signály a také pro mírné zkreslení. Pro vyšší hodnoty clipovacího prahu by nemělo smysl provádět poslechové testy, protože zkreslení by nebylo pro některé zvukové vzorky takřka patrné ani v případě clipovaného signálu.

Hodnocení kvality restaurace probíhalo metodou MOS (*Mean Opinion Score*), která obsahuje stupnici od 1 do 5. Hodnota 5 znamená nejlepší možnou kvalitu, 1 nejhorší a jedná se o hodnoty ODG (*Objective Difference Grade*). Jednotlivé stupně této škály pro účely hodnocení výsledků declippingu jsou uvedeny v tabulce 7.1.

Aby bylo zajištěno uživatelsky pohodlné testování, byla využita speciální Java aplikace pro subjektivní hodnocení kvality zvukového signálu. Tuto aplikaci vytvořil Viktor Černý v rámci své bakalářské práce [24].

Jako referenční zvuk byl zvolen původní nezkreslený signál a jako kotva clipovaný signál na dané úrovni θ_c , který byl zařazen mezi restaurované zvuky pro lepší srovnání. Respondent tedy porovnával 6 restaurovaných vzorků a jeden clipovaný. Test se skládal z deseti stran. Pro každý z pěti testovacích zvuků se hodnotily zvlášť výsledky restaurací pro dvě hodnoty clipovacího prahu.

Tab. 7.1: Pětistupňová škála hodnocení MOS pro hodnocení kvality declippingu.

Význam snížení kvality(Impairment)	ODG
Degradace je nepostřehnutelná (imperceptible)	5,0
Degradace je slyšitelná, ale neruší (perceptible, but not annoying)	4,0
Degradace příliš neruší (slightly annoying)	3,0
Degradace rušivá (annoying)	2,0
Degradace je velmi rušivá (very annoying)	1,0

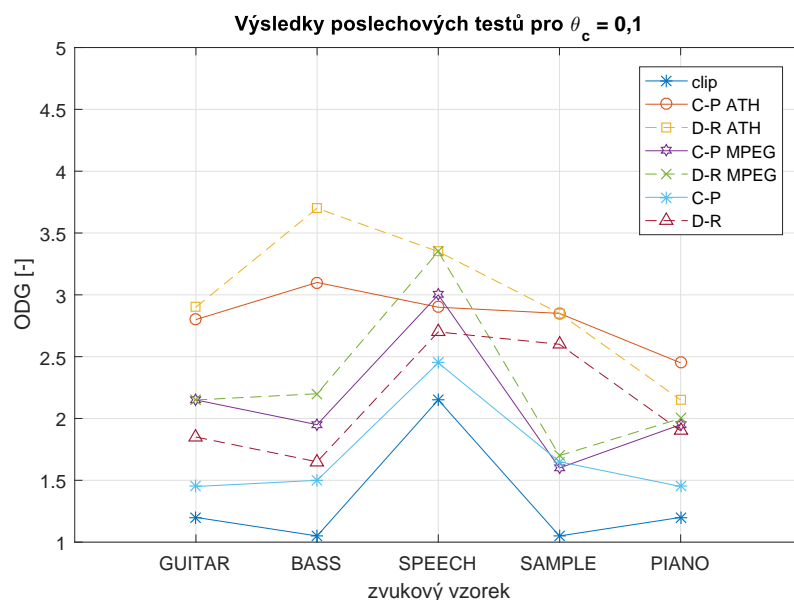
Poslechové testy byly provedeny na skupině 20 respondentů. Pro všechny byly zajištěny stejné podmínky z hlediska prostředí, hlasitosti a hardware. Testování probíhalo na notebooku Acer Aspire V15, externí zvukové kartě Focusrite Scarlett 2i4 a sluchátkách Audio-Technica ATH-M50X.

Celková délka čistého poslechového času činila 406 sekund (6,8 minut). Průměrná doba testování jednoho respondenta se pohybovala okolo 15 minut.

7.5.2 Výsledky poslechových testů

Výsledné grafy obsahují kromě výsledků všech testovaných metod i poslechové výsledky pro clipovaný signál, ten je v grafech označen černou plnou čarou. Pro lepší porovnání proximálních algoritmů je Douglas-Rachford označen čárkovaně a Chambolle-Pock naopak plnou čarou.

Graf 7.14 obsahuje výsledky poslechových testů pro clipovací úroveň $\theta_c = 0,1$. Na první pohled je z grafu patrné, že pro všechny testované zvuky dosáhl podle



Obr. 7.14: Výsledky poslechových testů pro velké zkreslení při $\theta_c = 0,1$.

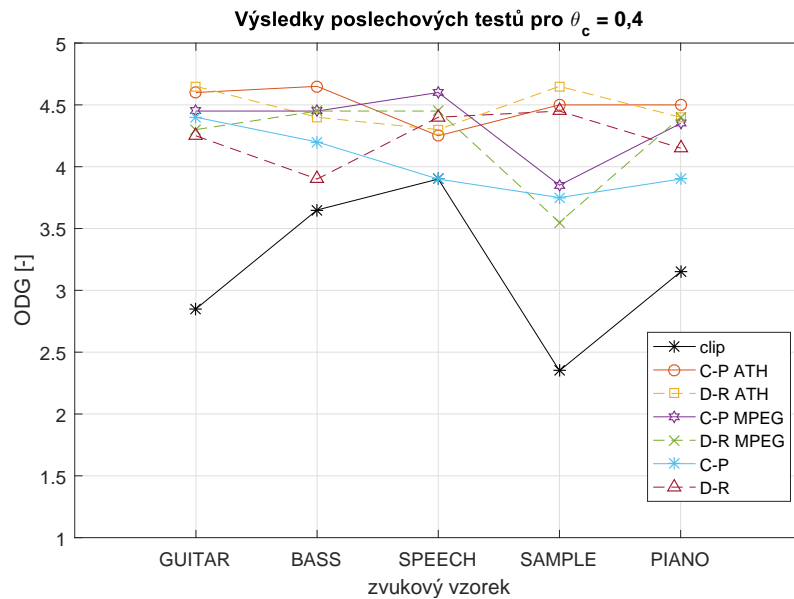
očekávání na nejnižší hodnoty ODG clipovaného signálu.

Nejvyšších hodnot ODG dosahoval algoritmu Douglas-Rachford s váhováním pomocí ATH. Jedinou výjimku tvořil vzorek `piano_voice.wav`, u kterého nejlepších výsledků dosáhl Chambolle-Pock váhovaný taktéž pomocí ATH.

Největší rozdíl mezi hodnotami ODG clipovaného a rekonstruovaného zvuku bylo dosaženo u vzorku `double_bass.wav`. Nejvíce viditelný je tento rozdíl pro oba algoritmy váhované pomocí ATH. Nejhorších výsledky ze všech testovaných metod podává Chambolle-Pock bez váhování.

Při porovnání jednotlivých metod váhování, nejlepší výsledky vykazovaly algoritmy váhované pomocí ATH, což také nasvědčovaly výsledky ukazatele ΔSNR . Nejhorších výsledků naopak dosahovaly neváhované algoritmy. Výjimku tvořil vzorek `sample.wav`, kde neváhovaný Douglas-Rachford předčil oba algoritmy váhované pomocí MPEG.

Nicméně výsledky všech algoritmů dokazují, že při velkém zkreslení dosahují rekonstrukce hodnot ODG kolem 2, maximálně 3. Tedy degradace je stále rušivá. Potvrdilo se ovšem, že výsledky declippingu přinesly snížení úrovně poslechové degradace. V grafu 7.15 jsou prezentovány výsledky poslechových testů pro menší



Obr. 7.15: Výsledky poslechových testů pro menší zkreslení při $\theta_c = 0,4$.

úroveň zkreslení při $\theta_c = 0,4$.

Již při prvním pohledu je jasné, že všechny testované zvuky, včetně clipovaného, dosahovaly mnohem lepších výsledků ODG než v prvním případě.

Opět nejhorších výsledků dosahoval saturovaný signál, který dosahoval průměrně hodnoty ODG 3, degradace byla tedy stále mírně rušivá. Naproti tomu testované metody declippingu dosahovaly, až na výjimky, hodnot okolo ODG = 4.

Nejvyrovnanějších výsledků dosahovaly algoritmy u vzorku `acoustic_guitar.wav`, kde byl také zaznamenán poměrně velký rozdíl mezi clipovaným signálem a těmi restaurovanými. Naopak nejmenší zaznamenaný rozdíl byl u vzorku `speech.wav`. V tomto případě totiž i původní clipovaný zvukový vzorek dosahoval hodnot ODG skoro 4. Výrazné zlepšení už tedy nebylo příliš možné. Zcela největšího zlepšení bylo dosaženo u vzorku `sample.wav`. Ovšem pouze pro algoritmy váhované pomocí ATH a neváhovaný Douglas-Rachford, ty v tomto případě poměrně znatelně předčily zbylé tři metody.

Průměrně nejhorší byl ze všech testovaných metod declippingu opět neváhovaný Chambolle-Pock. Pro nižší úroveň zkreslení však rozdíly mezi výsledky jednotlivých testovaných rekonstrukcí nejsou až tak velké. Nejlépe opět dopadl Douglas-Rachford váhovaný křivkou ATH. Chambolle-Pock váhovaný stejnou metodou však dosahoval takřka srovnatelných výsledků, pouze v případě vzorku mluveného slova zaznamenal menší propad.

8 ZÁVĚR

Tato bakalářská práce se zabývá problémem restaurace audiosignálu po průchodu limiterem. K řešení této úlohy je zde použita teorie řídkých reprezentací a psychoakustický model. V první kapitole je obecně probrána teorie týkající se limiteru a limitace signálu. Jsou zde vysvětleny dva druhy clippingu – hard clipping a soft clipping a případy, kdy k těmto poškozením signálu dochází.

V druhé kapitole je nejprve rozebrána samotná úloha declippingu a obvyklé metody jejího řešení. Dále je zde představeno osm publikací představujících současné metody declippingu založených na řídkých reprezentacích signálu. U každého z těchto článků je ve stručnosti představena použitá metoda a její výsledky, případně porovnání s jinými metodami.

Jedním z cílů této práce bylo seznámení se s řídkými reprezentacemi signálu a jejich využitím pro úlohu declippingu. Toto téma je rozebráno v kapitole třetí, která obsahuje úvod do problematiky, včetně základního značení a výchozích definic. Je zde probrána veškerá teorie potřebná k porozumění algoritmů pracujících na základě řídkých reprezentací. Tato kapitola také obsahuje část zabývající se syntetizujícím a analyzujícím modelem signálu a jejich využitím.

Čtvrtá kapitola obsahuje popis důležitých psychoakustických jevů, jejichž využití pro restauraci audiosignálu je velkým přínosem. Důležitou částí této kapitoly je také seznámení s psychoakustickým modelem, které je rozděleno na popis jednotlivých procesů zpracovávání signálu, které jsou v tomto modelu obsaženy.

V páté kapitole je nejprve popsáno využití proximálních algoritmů pro úlohu declippingu a následně jsou zde představena dvě řešení. První je založené na syntetizujícím modelu signálu a využívá algoritmus Douglas-Rachford. Druhé je založené na analyzujícím modelu signálu a byl pro ně zvolen algoritmus Chambolle-Pock. Je zde přiblížen princip obou těchto algoritmů a pro každý z nich je zde navrženo blokové schéma včetně použitého psychoakustického modelu.

Šestá kapitola obsahuje popis samotného programového řešení, implementovaného v prostředí Matlab. Jsou zde popsány jednotlivé m-soubory programu pro declipping a jejich nastavení.

Sedmou kapitolou je prezentace výsledků obou proximálních algoritmů pro úlohu declippingu. Výsledky jsou zde nejprve srovnány pomocí objektivního ukazatele Δ SNR. Dále tato kapitola obsahuje subjektivní porovnání výsledků. Nejprve je popsána metodika testování poslechových testů a následně jsou zde vyhodnoceny jejich výsledky.

Algoritmus Douglas-Rachford vykazoval skoro ve všech případech lepší výsledky než Chambolle-Pock, a to jak v případě ukazatele *Delta*SNR, tak i při poslechových testech. Dále bylo prokázáno, že vhodně zvolená váhová funkce může vý-

razně přispět kvalitě rekonstrukce signálu. Ze dvou testovaných váhovacích křivek se prokázala jako nejpřínosnější křivka absolutního prahu slyšitelnosti. Výsledky při váhování maskovací křivkou psychoakustického modelu již tak přesvědčivé nebyly. Nejúspěšnější testovanou variantou programu pro declipping byl tedy algoritmus Douglas-Rachford, váhovaný normalizovanou křivkou ATH.

Řešení, prezentovaná touto bakalářskou prací, podávají vcelku uspokojivé výsledky pro menší úrovně zkreslení. Co se týče vysoce zkreslených signálů, je zde rozhodně prostor pro zlepšení. Velkou výhodou zde prezentovaných řešení je jejich poměrně nízká časová náročnost.

Zajímavým využitím algoritmu pro declipping by například bylo jeho obohacení o funkci, která by byla schopná sama určit poškozené vzorky signálu. Vstupním signálem tohoto programu by pak mohl být přímo clipovaný signál. Pokud by se také podařilo výrazně snížit výpočetní náročnost tohoto programu, bylo by například možné vytvořit mobilní aplikaci pro restaurování zvukových nahrávek nebo zvukových stop videa.

LITERATURA

- [1] *Limiter*, Wikipedia [online]. 2017, poslední aktualizace 7.8.2017 [cit. 6.11.2017]. Dostupné z URL: [<https://en.wikipedia.org/wiki/Limiter>](https://en.wikipedia.org/wiki/Limiter).
- [2] KRÁL, V. *Kompresory dynamiky zvukových signálů: bakalářská práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 69 stran, 2 přílohy. Vedoucí práce doc. Ing. Jaroslav Koton, Ph.D.
- [3] ZÁVIŠKA, P. *Restaurace audiosignálů založená na řídkých reprezentacích: diplomová práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 94 stran, 2 přílohy. Vedoucí práce Mgr. Pavel Rajmic, Ph.D.
- [4] *Clipping (signal processing)* Wikipedia [online]. 2017, poslední aktualizace 24.3.2009 [cit. 19.5.2017]. Dostupné z URL: [<https://en.wikipedia.org/wiki/Clipping_\(signal_processing\)>](https://en.wikipedia.org/wiki/Clipping_(signal_processing)).
- [5] HARVILLA, M. J.; STERN, R. M. M. Efficient audio declipping using regularized least squares *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015. s 221–225. Dostupné z URL: [<http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?arnumber=7177964>](http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/stamp/stamp.jsp?arnumber=7177964).
- [6] ABEL, J. S.; SMITH, J.O. Restoring a clipped signal. *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. vydání 3, 1991. s. 1745–1748. Dostupné z URL: [<http://ieeexplore.ieee.org/document/150655/>](http://ieeexplore.ieee.org/document/150655/).
- [7] MAKHOUL J. Linear prediction: A tutorial review, *IEEE*, [online]. vol. 63, no. 4, s. 561–580, 1975. Dostupné z URL: [<http://ieeexplore.ieee.org/document/150655/>](http://ieeexplore.ieee.org/document/150655/).
- [8] ADLER, A.; EMIYA, V.; JAFARI, M. G.; ELAD, M.; GRIBONVAL, R.; PLUMBLEY, M. D. A constrained matching pursuit approach to audio declipping. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, s. 329–332. Dostupné z URL: [<http://ieeexplore.ieee.org/document/5946407/>](http://ieeexplore.ieee.org/document/5946407/).
- [9] WEINSTEIN, A. J.; WAKIN, M. B. Recovering a Clipped Signal in Sparse and Sampling Theory in Signal and Image Processing. ročník 12, s. 55–69, 2013. Dostupné z URL: [<https://arxiv.org/pdf/1110.5063>](https://arxiv.org/pdf/1110.5063).

- [10] SALESNICK, I. Least Squares with Examples in Signal Processing. [online]. OpenStax CNX. 28. 4. 2013, [cit. 20. 11. 2017]. Dostupné z URL: <<https://cnx.org/exports/5d13ca71-5821-41d6-a40a-d5278652fc11@1.pdf/least-squares-with-examples-in-signal-processing-1.pdf>> Least Squares with Examples in Signal Processing>.
- [11] KITIĆ, S.; JACQUES, L.; MADHU, N.; HOPWOOD, M. P.; SPRIET, A.; DE VLEESCHOUWER, C. Consistent Iterative Hard Thresholding for Signal Declipping. 2013. Dostupné z URL: <<http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/document/6638804/>>.
- [12] DEFRAENE, B.; MANSOUR, N.; DE HERTOOGH, S.; VANWATERSCHOOT, T.; DIEHL, M.; MOONEN, M. Declipping of Audio Signals Using Perceptual Compressed Sensing. *IEEE Transactions on Audio, Speech, and Language Processing* [online]. 2016. s 2627–2637. Dostupné z URL: <<http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/document/6600777/>>.
- [13] SIEDENBURG, K.; KOWALSKI, M.; DÖFLER, M. Audio Declipping With Social Sparsity. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, s. 1577–1581 Dostupné z URL: <<http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/document/6853863/>>.
- [14] KOWALSKI, M.; SIEDENBURG, K.; DÖFLER, M. Neighborhood Systems Enrich Structured Shrinkage Operators. 2012. Dostupné z URL: <<https://hal.archives-ouvertes.fr/hal-00691774v1/document>>.
- [15] ADLER, A.; EMIYA, V.; JAFARI, M. G.; ELAD, M.; GRIBONVAL, R.; PLUMBLEY, M. D. Audio Inpainting *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, s. 922-932, 2012. Dostupné z URL: <<http://ieeexplore.ieee.org/document/6020748/>>.
- [16] KITIĆ, S.; BERTIN, N.; GRIBONVAL, R. Audio Declipping by Cosparsity Hard Thresholding. *iTwist – Traveling Workshop on Interactions between Sparse models and Technology*. 2014. Dostupné z URL: <<https://hal.inria.fr/hal-00922497>>.

- [17] KITIĆ, S.; BERTIN, N.; GRINBONVAL, R. Sparsity and Cosparsity for Audio Clipping: A Flexible Non-convex Approach. *LVA/ICA 2015, LNCS 9237* s. 243–250, 2015. Dostupné z URL: <https://hal.archives-ouvertes.fr/hal-01159700/document>.
- [18] RAJMIC, P. *Řídké a nízkohodnotní reprezentace signálů s aplikacemi: habilitační práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 139 stran.
- [19] HRBÁČEK, R.; RAJMIC, P.; VESELÝ, V.; ŠPIŘÍK, J. *Řídká reprezentace signálů: úvod do problematiky* [online]. *Elektrorevue*, 2011. ISSN 1213-1539. Dostupné z URL: <http://www.elektrorevue.cz/cz/download/ridke-reprezentace-signalu-uvod-do-problematiky/.e.cz/>.
- [20] DONG, J.; WANG, W.; DAI, W.; PLUMBLEY, M. D.; HAN, Z.F.; CHAMBERS, J. Analysis SimCO Algorithms for Sparse Analysis Model Based Dictionary Learning. *IEEE Transactions on Signal Processing*, vol. 64, no. 2, s. 417-431, 2016. Dostupné z URL: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/document/7169499/>.
- [21] SCHIMMEL, J. *Elektroakustika: skripta k předmětu* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 200 stran.
- [22] ALOUI, N.; RAOOF, K.; BOUALLEGUE, A.; LATOURNEUR, S.; ZAIBI, S. A Time Reversal-Based Acoustic Indoor Localization System at the Absolute Threshold of Hearing. *IEEE Sensors Journal* vol. 15, no. 11, S. 6472-6484, 2015. Dostupné z URL: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/document/7279182/>.
- [23] LEPA, O. *Psychoakustický model lidského sluchu* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 52 stran, 2 přílohy. Vedoucí práce Ing. Petr Sysel, Ph.D.
- [24] ČERNÝ, Viktor. *Nástroj pro subjektivní hodnocení kvality restaurace audio signálů: bakalářská práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 50 s, 1 příloha. Vedoucí práce Ing. Pavel Záviška.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AR	Autoregressive model
A-SPADE	Analysis Sparse Audio Declipper
ATH	The absolute threshold of hearing
BP	Basis Pursuit
C-IHT	Consistent Iterative Hard Thresholding
C-OMP	Constrained Orthogonal Matching Pursuit
CBAR	Constrained Blind Amplitude Reconstruction
CoDec-HT	Cospase Declipping by Hard Thresholding
CP	Chambolle-Pock
CSL0	Compressed Sensing based method using ℓ_0 norm
CSL1	Compressed Sensing based method using ℓ_1 norm
DCT	Discrete Cosine Transform – Diskrétní kosinová transformace
DFT	Discrete Fourier Transform – Diskrétní Fourierova transformace
DR	Douglas-Rachford
EW	Empirical Weiner
IHT-DC	Iterative Hard Thresholding for Declipping
IRLS	Iterative Reweight Least Squares
(F)ISTA	(Fast) Iterative Shrinkage/Thresholding Algorithm
LARS	Least Angle Regression
MOS	Mean Opinion Score
MPEG-1	Motion Pictures Experts Group – Layer 1
NP-hard	Non-Deterministic Polynomial-Time Hardness
ODG	Objective Difference Grade
PCSL1	Perceptual Compressed Sensing based method using ℓ_1 norm
PEW	Persistent Empirical Weiner
$R\ell_1$ CC	Reweighted ℓ_1 minimization with Clipping Constraints
S-SPADE	Synthesis Sparse Audio Declipper
SDR	Signal to Distortion Ratio – Odstup signálu od zkreslení
SNR	Signal to Noise Ratio – Odstup signálu od šumu
TPCC	Trivial Pursuit with Clipping Constraints
SPL	Sound pressure level
VP	Vektorový prostor
WGL	Windowed Group-Lasso

SEZNAM PŘÍLOH

A Obsah přiloženého CD

70

A OBSAH PŘILOŽENÉHO CD

Na přiloženém CD jsou v adresáři `declipping_program` uloženy všechny m-soubory programu pro declipping popsané v kapitole 6.

V adresáři `test_zvuky` jsou přiloženy všechny testovací zvuky ve formátu `*.wav`.

Posledním adresářem je `declipped_samples`, který obsahuje ukázky restaurovaných testovacích signálů pomocí výše popsáných metod. Tento adresář obsahuje pět podadresářů rozdělených podle testovacích zvuků. Adresář s restauracemi daného testovacího zvuku je dále ještě rozdělen na dvě složky pro clipovací threshold θ_c 0,1 a 0,4. Součástí složky s restaurovanými zvuky je pro srovnání i původní clipovaný vzorek. Všechny zvukové soubory jsou ve formátu `*.wav`.

```
/ ..... kořenový adresář přiloženého CD
├── declipped_samples ..... restaurované zvuky
│   ├── BASS ..... restaurace vzorku double_bass.wav
│   ├── GUITAR ..... restaurace vzorku acoustic_guitar.wav
│   ├── PIANO ..... restaurace vzorku piano_voice.wav
│   ├── SAMPLE ..... restaurace vzorku sample.wav
│   └── SPEECH ..... restaurace vzorku speech.wav
├── declipping_program ..... balík pro declipping
│   ├── ath.m
│   ├── douglas.m
│   ├── hard_clip.m
│   ├── chambolle.m
│   ├── main.m
│   ├── masking2.m
│   ├── proj.m
│   ├── proj_dr.m
│   └── snr.m
└── test_zvuky ..... testovací zvukové signály
    ├── acoustic_guitar.wav
    ├── double_bass.wav
    ├── piano_voice.wav
    ├── sample.wav
    └── speech.wav
```