

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020

Bc. Jakub Šiška



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 ZVUKOVÝCH SIGNÁLŮ POŠKOZENÝCH KVANTIZACÍ

RESTORATION OF AUDIO SIGNALS DAMAGED BY QUANTIZATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Šiška

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Závíška

BRNO 2020



Diplomová práce

magisterský navazující studijní obor **Audio inženýrství**

Ústav telekomunikací

Student: Bc. Jakub Šiška

ID: 186636

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Restaurace zvukových signálů poškozených kvantizací

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte jev znehodnocení zvukového signálu pomocí kvantizace v časové oblasti a seznamte se s různými přístupy zvyšování bitové hloubky, neboli tzv. dekvantizace. Důležité zástupce takových metod popište a implementujte.

Dále se seznamte s teorií řídkých reprezentací signálů a aplikujte ji v kontextu dekvantizace zvukových signálů. Implementujte algoritmy založené na konvexní i nekonvexní optimalizaci pro syntezující i analyzující model signálu. Analyzujte dosažené výsledky pomocí vhodných objektivních i subjektivních poslechových testů a srovnajte je se stávajícími metodami, popř. standardními plug-in moduly pro zvyšování bitové hloubky.

DOPORUČENÁ LITERATURA:

[1] Rajmic, P., Daňková, M. Úvod do řídkých reprezentací signálů a komprimovaného snímání. Skriptum, Vysoké učení technické v Brně, 2014. ISBN 978-80-214-5169-8.

[2] Závíška, P., Mokrý, O., Rajmic, P. "A proper version of synthesis-based sparse audio declipper," in Acoustics, Speech and Signal Processing (ICASSP), 2019 IEEE International Conference on, May 2019, pp. 591–595.

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

Vedoucí práce: Ing. Pavel Závíška

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

UPOZORNĚNÍ:

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

ABSTRAKT

Táto práca sa zaoberá reštauráciou zvukových signálov poškodených kvantizáciou. V teoretickej časti je popísaná kvantizácia a dekvantizácia všeobecne, rôzne existujúce metódy dekvantizácie audio signálov a opis teórie riedkych reprezentácií signálov. Ďalšia časť sa zaoberá algoritmi pre rekonštrukciu signálu Douglas–Rachford, Chambolle–Pock, SPADEQ a ich implementáciou v programe MATLAB v nasledujúcej kapitole. V poslednej časti práce sú tieto algoritmy otestované pomocou objektívnych ukazovateľov Δ SDR, PEMO-Q, PEAQ a subjektívneho posluchového testu MUSHRA.

KLÚČOVÉ SLOVÁ

Kvantizácia, dekvantizácia, riedka reprezentácia, Douglas–Rachford, Chambolle–Pock, SPADEQ, zvukový signál, spracovanie signálu

ABSTRACT

This master's thesis deals with the restoration of audio signals damaged by quantization. The theoretical part starts with a description of quantization and dequantization in general, few existing methods of dequantization of audio signals and theory of sparse representations of signals are also presented. The next part introduces algorithms suitable for dequantization, specifically Douglas–Rachford, Chambolle–Pock, SPADEQ and implementation of these algorithms in MATLAB application in the next chapter. In the last part of this thesis, testing of reconstructed signals using the algorithms takes place and results are evaluated by objective measures Δ SDR, PEMO-Q, PEAQ and subjective listening test MUSHRA.

KEYWORDS

Quantization, dequantization, sparse representation, Douglas–Rachford, Chambolle–Pock, SPADEQ, audio signal, signal processing

ŠIŠKA, Jakub. *Restaurace audiosignálů poškozených kvantizací*. Brno, 2020, 66 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Pavel Závíška

VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Restaurovanie audiosignálov poškodených kvantizácií“ som vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi Ing. Pavlovi Záviškovi za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Obsah

Úvod	9
1 Kvantizácia a dekvantizácia audiosignálov	10
1.1 Metódy dekvantizácie audiosignálov	12
1.1.1 Bayesian restoration of quantised audio signals using a sinusoidal model with autoregressive residuals	13
1.1.2 Sparse reconstruction of quantized speech signals	13
1.1.3 Dequantization of speech signals by primal–dual networks	13
1.1.4 Fast iterative shrinkage for signal declipping and dequantization	14
2 Teória riedkych reprezentácií signálov v kontexte k dekvantizácii audiosignálov	15
2.1 Značenie a základné poznatky	15
2.2 Normy vektorov	15
2.3 Bázy a framy	16
2.4 Riedkosť signálu a riedke riešenia	18
2.5 ℓ_1 relaxácia	20
2.6 Proximálne algoritmy pre konvexnú optimalizáciu	21
2.6.1 Proximálne algoritmy, proximálny operátor	22
2.6.2 Proximálny operátor ℓ_1 normy	23
2.7 Syntezujúci a analyzujúci model signálu	24
3 Algoritmy pre rekonštrukciu signálu	25
3.1 Algoritmy založené na konvexnej optimalizácii	25
3.1.1 Algoritmus pre syntezujúci model signálu	25
3.1.2 Algoritmus pre analyzujúci model signálu	27
3.2 Algoritmy založené na nekonvexnej optimalizácii – SPADE	29
3.2.1 ADMM	29
3.2.2 A–SPADEQ algoritmus pre analyzujúci model signálu	31
3.2.3 S–SPADEQ a S–SPADEQ _{DR} algoritmy pre syntezujúci model signálu	32
4 Implementácia v programe MATLAB	34
4.1 Hlavný súbor	34
4.2 Kvantizácia	35
4.3 Douglas–Rachfordov algoritmus	35
4.4 Chambolle–Pockov algoritmus	36
4.5 Projekcia na množinu prípustných riešení v časovej oblasti	36

4.6	Projekcia na množinu prípustných riešení v oblasti koeficientov	37
4.7	Segmentácia signálu pre SPADEQ algoritmy	37
4.8	A-SPADEQ algoritmus	38
4.9	S-SPADEQ a S-SPADEQ _{DR} algoritmy	38
4.10	Hard thresholding	38
5	Testovanie algoritmov, výsledky testovania	39
5.1	Testovanie podľa objektívnych ukazovateľov	39
5.1.1	SDR	39
5.1.2	PEMO-Q	39
5.1.3	PEAQ	42
5.2	Subjektívne posluchové testy	43
5.2.1	MUSHRA	43
5.3	Nastavovanie parametrov algoritmov	45
5.3.1	Nastavenie počtu iterácií	46
5.3.2	Nastavenie parametrov γ a ζ	47
5.3.3	Nastavenie veľkosti a posunu okna	48
5.3.4	Nastavenie počtu kvantizačných hladín	50
5.3.5	Nastavenie parametrov pre SPADEQ algoritmy	51
5.4	Výsledky testovania	52
5.4.1	Výsledky podľa hodnôt Δ SDR a SDR	52
5.4.2	Výsledky podľa metódy PEMO-Q	54
5.4.3	Výsledky podľa metódy PEAQ	56
5.4.4	Výsledky podľa testu MUSHRA	56
5.5	Porovnanie dosiahnutých výsledkov ukazovateľa ODG a MUSHRA testu	57
6	Záver	60
	Literatúra	62
	Zoznam symbolov, veličín a skratiek	65
A	Elektronická príloha	66

Zoznam obrázkov

1.1	Kvantizácia signálu v čase, vzniknuté kvantizačné skreslenie	11
2.2	Vrstevnice noriem ℓ_p a ich dotyk s nadrovinou určenou $\mathbf{Ax} = \mathbf{y}$	21
2.3	Prevodná charakteristika mäkkého prahovania	24
5.1	Priebeh hodnôt normy $\ \mathbf{c}\ _1$ a ΔSDR pri správne nastavenom prahu mäkkého prahovania	46
5.2	Priebeh hodnôt normy $\ \mathbf{c}\ _1$ a ΔSDR pri príliš vysokom prahu mäkkého prahovania	46
5.3	Závislosť ΔSDR na počte opakovaní algoritmu pre signál huslí	47
5.4	Zistené hodnoty parametra γ pre rôzny počet iterácií algoritmu	48
5.5	Hodnoty $\ \mathbf{c}\ _1$ pre daný počet iterácií	48
5.6	Závislosť ΔSDR na veľkosti okna pre signál huslí pre 8 hladín	49
5.7	Výsledky testovania pre rôzne hodnoty veľkosti a posunu okna	49
5.8	Výsledky rekonštrukcie signálu huslí DR algoritmom pre rôzny počet kvantizačných hladín	50
5.9	Hodnoty SDR po rekonštrukcii signálu huslí DR algoritmom pre rôzny počet kvantizačných hladín	51
5.10	Hodnoty ΔSDR zrekonštruovaného signálu huslí algoritmom S-SPADEQ _{DR} v závislosti na parametri ϵ	52
5.11	Priemerná hodnota ΔSDR rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín	53
5.12	Priemerná hodnota SDR rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín	53
5.13	Priemerná hodnota ODG(PEMO-Q) rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín	55
5.14	Priemerná hodnota PSM _t rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín	55
5.15	Priemerná hodnota ODG(PEAQ) rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín	56
5.16	Priemerné hodnotenie stimulov z testu MUSHRA pre rôzne signály a počty kvantizačných hladín	57
5.17	Porovnanie priemerných hodnotení stimulov z testu MUSHRA transformovaných na SDG stupnicu a hodnôt ODG z metód PEMO-Q a PEAQ	58

Úvod

S kvantizáciou sa dá dnes stretnúť v mnohých oblastiach, či už v spracovaní signálov alebo inde. Je to nezvratný proces a pri nesprávne zvolenej kvantizácii nie je rekonštrukcia signálu do pôvodného stavu možná. Táto práca sa bude zaoberať riešením dekvantizácie s využitím riedkych reprezentácií.

V prvej kapitole tejto práce je opísaná kvantizácia všeobecne, jej použitie v spracovaní číslicových signálov, a aký má kvantizácia vplyv na signál. Taktiež sa pojednáva o dekvantizácii a potom je spísaný výber existujúcich metód pre dekvantizáciu.

Druhá kapitola obsahuje teoretickú časť o riedkych reprezentáciách, najprv základné poznatky, ďalej časti popisujúce normy vektorov, bázy a framy a popis riedkych riešení, ktorými sa dnes zaoberá čoraz viac prác. V predposlednom bode sú zhrnuté algoritmy, ktoré sú využívané pre konvexnú optimalizáciu.

V tretej kapitole sa pojednáva o použití rôznych algoritmov s popisom ich fungovania. Opísané algoritmy zahŕňajú Douglas–Rachford, Chambolle–Pock a SPADEQ pre analyzujúci aj syntezujúci model signálu.

Vo štvrtej kapitole je popísaná implementácia týchto algoritmov v programe MATLAB a opísanie vstupných parametrov, s ktorými sa mení správanie algoritmov.

Posledná kapitola pojednáva o rôznych možnostiach testovania rekonštrukčných algoritmov podľa objektívnych aj subjektívnych ukazovateľov. Najskôr je popísaný proces hľadania najlepších nastavení vstupných parametrov algoritmov. V rámci tejto kapitoly sú obsahom aj výsledky testovania signálov rekonštruovaných pomocou algoritmov s rôznymi hodnotami vstupných parametrov. Na hodnotenie kvality rekonštrukcie sú využité objektívne ukazovatele (Δ)SDR, PEMO-Q, PEAQ a subjektívny posluchový test podľa metódy MUSHRA. Výsledky sú spracované pre niekoľko signálov hudobných nástrojov.

1 Kvantizácia a dekvantizácia audiosignálov

Kvantizácia má v súčasnosti veľké využitie v rôznych sférach, najmä v matematike a spracovaní signálov. Kvantizácia sa dá chápať ako proces priradenia určitého množstva hodnôt na menší počet diskrétnych hodnôt, pričom ich počet býva často konečný. Typickým príkladom procesu kvantizácie v matematike je zaokrúhľovanie.

Pri spracovaní číslicových signálov sa kvantizácia využíva takmer vždy, keďže reprezentácia signálu v číslicovej forme vyžaduje zaokrúhľovanie hodnôt. Zariadenie alebo algoritmus, ktorý vykonáva kvantizáciu sa nazýva kvantizátor. Príkladom kvantizátora je napríklad A/D prevodník.

Najstarším príkladom použitia kvantizácie je zaokrúhľovanie, ktoré analyzoval Sheppard pre získavanie odhadu hustôt podľa histogramu. Každé reálne číslo x je možné zaokrúhliť na najbližšie celé číslo $q(x)$ s tým, že vznikne kvantizačná chyba $c = q(x) - x$, a potom platí $q(x) = x + c$.

Všeobecne môžeme kvantizátor definovať ako vektor hodnôt, ktorý obsahuje množinu intervalov $\mathbf{S} = \{S_i; i \in \mathbf{I}\}$, kde vektor indexov \mathbf{I} je zvyčajne rad po sebe idúcich celých čísel začínajúcich 0 alebo 1, spolu s radom reprodukčných hodnôt $\mathbf{C} = \{y_i; i \in \mathbf{I}\}$ tak, že celkový kvantizátor q je definovaný $q(x) = y_i$ pre $x \in S_i$, čo sa dá vyjadriť ako [1]

$$q(x) = \sum_i y_i 1_{S_i}(x), \quad (1.1)$$

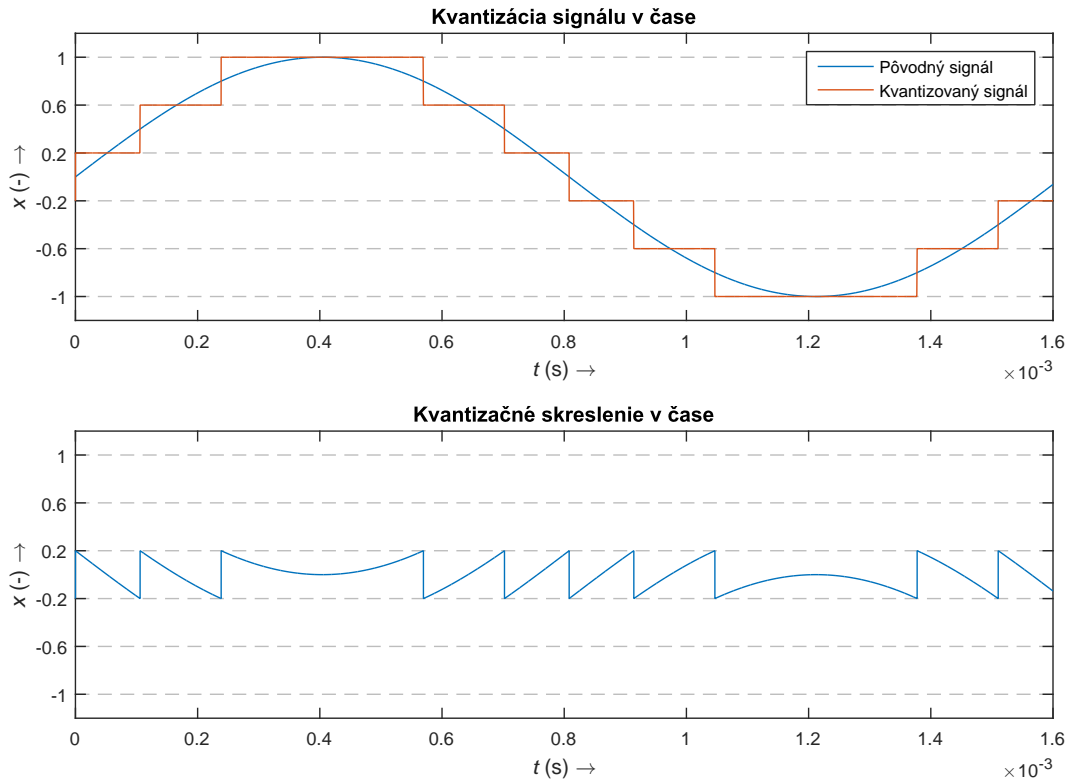
kde $1_S(x)$ je indikátorová funkcia, ktorá má hodnotu 1 pre $x \in S$ a v opačnom prípade 0. U tejto definície predpokladáme, že množina kvantizačných krokov (intervalov) \mathbf{S} je časťou reálneho radu, pričom jednotlivé intervaly sú nesúvislé a kompletne.

Všeobecne platí, že intervaly môžu mať podobu $S_i = (a_{i-1}, a_i]$, kde a_i sa nazývajú prahy a utvárajú rastúcu postupnosť. Šírka intervalu S_i je jeho dĺžka $a_i - a_{i-1}$. Funkcia $q(x)$ sa často nazýva kvantizačné pravidlo (*quantization rule*).

Kvantizátor je uniformný (na obr. 1.1 hore), keď kvantizačné hladiny y_i sú od seba rovnomerne rozložené o Δ a prahy a_i sú v strede medzi prilahlými hladinami. Keby sa počítalo s nekonečným množstvom hladín, potom všetky intervaly S_i budú mať šírku rovnú rozdielu medzi dvoma susediacimi hladinami Δ . Ak počítame s konečným množstvom hladín, všetky okrem dvoch intervalov budú mať šírku Δ a krajné intervaly budú polootvorené. [1]

U neuniformného kvantizátora budú mať intervaly S_i rôznu šírku Δ , a to kvôli nerovnomerne rozloženým hladinám y_i .

Kvalita kvantizátora sa dá zmerať ako presnosť výsledného signálu v porovnaní s originálom. Jedným zo spôsobov je definovanie miery skreslenia $d(x, \hat{x})$, ktoré stanovuje skreslenie vzniknuté pri reprodukcii pôvodnej vzorky signálu x na kvantizovanú vzorku \hat{x} , a uváženie priemerného skreslenia ako miery kvality systému, pričom



Obr. 1.1: Kvantizácia harmonického signálu uniformným kvantizátorom so 6 kvantizačnými hladinami a vzniknuté kvantizačné skreslenie.

čím menšie je priemerné skreslenie, tým vyššia je kvalita kvantizátora. Najčastejšou mierou skreslenia je kvadrát chyby $d(x, \hat{x}) = |x - \hat{x}|^2$ [1].

Kvantizačné skreslenie je tvorené radom pridaných harmonických, spolu s ich obrazmi, ktoré vzniknú pri aliasingu. U komplikovanejších signálov je takéto skreslenie pomerne neškodné, podobá sa bielu šumu. Ak je ale signál jednoduchý, skreslenie je zreteľné.

Naviac sa môžu relatívne úrovne pridaných harmonických významne líšiť s malými zmenami v úrovni vstupného signálu. To môže byť rušivé v hudobných signáloch, ktoré vytvárajú hudobné nástroje ako napríklad klavír, ktorého tóny sa s doznievaním stávajú viac sínusoidné a nastáva u nich amplitúdová modulácia skreslených harmonických zložiek. Tento efekt sa nazýva granulačný šum (*granulation noise*) [2].

Je žiadúce získať čo najmenšie skreslenie, pričom to sa dá dosiahnuť zmenšením dĺžky intervalov a zvýšením ich počtu. Avšak, to má za následok zvýšené požiadavky na počet bitov pre popisovanie signálu z výstupu kvantizátora do dekódera, kvôli čomu treba brať do úvahy obmedzenú kapacitu úložiska a médií pre uloženie reprodukcie.

Jednoduchou metódou pre zistenie požiadavky na kapacitu úložiska je predpo-

kladanie, že kvantizátor kóduje vstup x na binárnu reprezentáciu indexu kvantizácie i , ktorý stanovuje hladinu reprodukcie, ktorá sa použije pri rekonštrukcii. Ak existuje N možných hladín a všetky binárne reprezentácie majú rovnakú dĺžku, binárne vektory budú potrebovať $\log n$ bitov. Z toho sa dá definovať *tok* kódu v bitoch na vstupnú vzorku

$$R(q) = \log N. \quad (1.2)$$

Kvantizátor s pevne stanovenou dĺžkou binárneho slova má pevne stanovený tok, pretože všetky kvantizačné hladiny majú predpokladanú dĺžku binárneho slova rovnakú [1].

Cieľom kvantizácie je zakódovať informáciu zo zdroja na čo najmenej bitov, tj. s čo najnižším dátovým tokom tak, aby sa reprodukcia zo zakódovaných bitov mohla vykonať s čo najvyššou kvalitou, tj. s malým priemerným skreslením. Je potrebné zabezpečiť kompromis medzi priemerným skreslením a tokom dát, čo sa dá stanoviť ako *operational distortion-rate function* $\delta(R)$, ktorá je definovaná ako najmenšie skreslenie $D(q)$ skalárneho kvantizátora q s tokom R alebo menej:

$$\delta(R) = \inf_{q: R(q) \leq R} D(q). \quad (1.3)$$

Počet kvantizačných hladín N odpovedá počtu použitých bitov kvantizátoru m , pričom $N = 2^m - 1$. S týmito veličinami súvisí odstup signálu od šumu, dá sa vyjadriť ako

$$\frac{S}{N} = 20 \log N = 20 \log (2^m - 1) \cong 6m + k, \quad (1.4)$$

kde konštanta k závisí na definícii kvantizačného skreslenia [3].

Kvantizácia je nezvratný proces, signál ktorý bol poškodený je prakticky nemožné spracovať do pôvodnej podoby, dá sa k nej iba priblížiť. Na to sa využíva dekvantizácia, proces opačný ku kvantizácii. Pri dekvantizácii sa zvyšuje počet kvantizačných hladín, ktoré môžu jednotlivé vzorky poškodeného signálu nadobudnúť, spolu s tým sa zvýši počet bitov, zlepší sa odstup signálu od šumu a zmenší sa kvantizačné skreslenie. Úlohou pri dekvantizácii bude sa čo najviac priblížiť k hodnotám vzoriek pôvodného signálu.

1.1 Metódy dekvantizácie audiosignálov

Existuje viacero výskumov, u ktorých sa autori pokúšali rekonštruovať kvantizovaný signál. Niektoré z výskumov budú v tejto sekcii predstavené a uvedené budú ich dosiahnuté výsledky. Prvým z nich je *Bayesian restoration of quantised audio signals using a sinusoidal model with autoregressive residuals* [4], ďalším je *Dequantization of speech signals by primal-dual networks* [6], a tretia metóda je nazvaná *Fast iterative shrinkage for signal declipping and dequantization* [7].

1.1.1 Bayesian restoration of quantised audio signals using a sinusoidal model with autoregressive residuals

Táto metóda rekonštrukcie signálu je založená na Bayesovskej metóde zmenšenia kvantizačného skreslenia pri presúvaní zvukového signálu na médium s vyšším rozlíšením. Výstupný signál je modelovaný ako súčet autoregresívneho procesu neznámeho rádu vyvolaného vstupným poškodeným signálom, ktorý je zvolený ako biely gausovský šum a sínusoidových komponentov, ktoré korešpondujú so signálmi obsahujúcimi najmä sínusoidy, u ktorých je skreslenie zreteľné. Odhad výsledného signálu sa vykonáva použitím metódy *Markov chain Monte Carlo* (MCMC).

Na testovanie bol použitý komerčne nahratý zvuk klavírnej skladby s doznievajúcimi tónmi. Kvantizáciou tohto signálu sa zvýšilo skreslenie na 23 dB pod hodnotu nepoškodeného signálu a bolo jasne počuteľné. Signál bol spracovaný v blokoch zložených z 1024 vzoriek. V každom bloku bolo vybratých 30 kmitočtov pre sínusoidové komponenty a algoritmus sa zopakoval 400-krát.

U výsledného rekonštruovaného signálu sa podarilo skreslenie znížiť priemerne o 11 dB (rms), v niektorých tichších častiach signálu až o 17 dB (rms) [4].

1.1.2 Sparse reconstruction of quantized speech signals

V tejto metóde sa autori zamerali na hľadanie rečového kódeku pre bezdrôtové akustické senzory. V sieťach akustických sensorov musia najmä vysielajúce pracovať s čo najnižšou výpočtovou náročnosťou kvôli obmedzeniam v životnosti batérií, preto bol signál kvantizovaný na nízke bitové rozlíšenie. Na rekonštrukciu takéhoto signálu sa využila riedkosť signálu a prístup podobný komprimovanému snímaniu.

Na dekvantizáciu signálu bol využitý Chambolle–Pockov algoritmus pre konvexnú optimalizáciu bez váhovacích filtrov. Rekonštrukcia bola testovaná na sade 720 viet mužského hlasu pri vzorkovaní na 16 kHz. Algoritmus bol nastavený na 25 iterácií a výsledky testovania boli vyjadrené pomocou ukazovateľa vnemové ohodnotenie kvality reči (*Perceptual Evaluation of Speech Quality*, PESQL). Testy ukázali významné zlepšenie hodnoty PESQL a zároveň je možné túto metódu použiť v reálnom čase. [5]

1.1.3 Dequantization of speech signals by primal–dual networks

Autori tejto metódy vylepšili predošlú metódu rekonštrukcie rečových signálov 1.1.2. Na vylepšenie kvality skalárnej kvantizácie pre siete akustických sensorov boli využité riedka rekonštrukcia, umelé neurónové siete a váhovacie filtre. Najskôr sa zistilo, že optimalizácia založená na riedkych rekonštrukciách je podobná štruktúre neurónových sietí. Na základe vytvorenia takéhoto algoritmu sa to následne využilo

pre vybudovanie neurónovej siete, ktorá bola trénovaná pre získanie vylepšeného dekodovania. Taktiež bol do stratovej funkcie neurónovej siete pridaný váhový filter z kódom budeného lineárneho prediktívneho kódovania reči (*code-excited linear predictive speech coding*, CELP), čím sa získal vylepšený zrekonštruovaný rečový signál. Metóda pracuje s krátkymi oknami signálu, čo umožňuje jej fungovanie v reálnom čase.

Pre dekvantizáciu signálu bol znova použitý Chambolle–Pockov algoritmus. Tiež sa využíva objektívna funkcia pre trénovanie založené na váhovacom filtri postavenom na CELP kódovaní, napríklad typu *adaptive multi-rate* (AMR) a *wideband AMR* (AMR–WB).

Celkový algoritmus bol vyskúšaný na sade 720 viet mužského hlasu, vzorkovaných na 16 kHz. Časť z nich bola použitá na trénovanie neurónovej siete, ďalšia na vývoj a posledná na testovanie. Výsledky ukázali zlepšenie v odstupe signálu od šumu (SNR) pri akomkoľvek dátovom toku oproti predošlej metóde s použitím Chambolle–Pockovho algoritmu bez váhovacieho filtra a priemeru kvadrátu chyby (*mean squared error*, MSE). Pri porovnaní druhého ukazovateľa PESQL bola táto metóda lepšia pri bitovom toku 4 a 5 bitov na vzorku [6].

1.1.4 Fast iterative shrinkage for signal declipping and dequantization

Táto práca sa zaoberá riešením declippingu a dekvantizácie signálu použitím iteratívneho zmenšovacieho/prahovacieho algoritmu (*Iterative Shrinkage/Thresholding Algorithm*, ISTA) a jeho zrýchlenej verzie (FISTA), a ich porovnaním s algoritmom ADMM (*Alternating Direction Method of Multipliers*). Všetky algoritmy boli nastavené maximálne na 400 iterácií, prípadne dovtedy, kým sa algoritmus nekonvergoval.

Ukázalo sa, že kvôli nízkej rýchlosti konvergenzie bol algoritmus ISTA ďaleko od optimálnych hodnôt aj po maximálnom počte iterácií, zatiaľ čo algoritmu FISTA stačilo 150 iterácií na dosiahnutie optima, čo zároveň znamenalo výrazné zlepšenie v rýchlosti prevedenia rekonštrukcie signálu.

Autori ukazujú, že použitie ADMM algoritmu je výpočetne oveľa náročnejšie a zároveň veľmi podobné vo výsledkoch odstupe signálu od šumu oproti ostatným algoritmom. Čas potrebný pre výpočet ADMM algoritmu bol okolo 16 minút pre declipping aj dekvantizáciu, pre algoritmy ISTA aj FISTA sa doba výpočtu skrátila na jednotky sekúnd, konkrétne u ISTA to bolo 3,8 s a pri FISTA 1,56 – 1,81 s.

Výhodou týchto metód je teda hlavne oveľa nižšia výpočetná náročnosť s 250 – 600-krát menším časom potrebným na výpočet [7].

2 Teória riedkych reprezentácií signálov v kontexte k dekvantizácii audiosignálov

V tejto kapitole sa bude pojednávať o riedkych reprezentáciách, nakoľko metódy riešení na nich založené budú využité k dekvantizácii signálu. Najskôr budú zhrnuté základné značenie a pojmy, vektorové normy, bázy a framy. Následne bude zhrnutá riedkosť signálu a hľadanie riedkych riešení. V závere kapitoly budú stručne uvedené princípy algoritmov pre hľadanie riedkych riešení, proximálny operátor a porovnanie syntezujúceho a analyzujúceho modelu signálu.

2.1 Značenie a základné poznatky

Počet prvkov množiny (kardinalita) bude značený ako absolútna hodnota, napr. $|\{-2, 3, -9, 8\}| = 4$. Skalárne veličiny budú značené kurzívou, napríklad m, N . Vektory budú značené tučným písmom, napr. \mathbf{x}, \mathbf{y} . Vektory s konečnými rozmermi sa budú uvažovať ako stĺpcové, ak nebude napísané inak.

Indexovanie prvkov vektorov bude začínať jednotkou, ako napr. $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$. Nosič vektora \mathbf{x} je množina jeho indexov nenulových hodnôt, označuje sa $\text{supp}(\mathbf{x})$; $\text{supp}(\mathbf{x}) = \{i \mid x_i \neq 0\}$. Napríklad pre $\mathbf{x} = [x_1, \dots, x_5]^T = [1, 0, 0, 5, 2]^T$ platí $\text{supp}(\mathbf{x}) = \{1, 4, 5\}$ a $|\text{supp}(\mathbf{x})| = 3$.

Maticy budú značené veľkým tučným písmenom, napr. \mathbf{A}, \mathbf{Y} . Vektorové priestory sa budú značiť ako \mathbb{V}, \mathbb{R}^N a všade budú uvažované ako neprázdne.

2.2 Normy vektorov

Norma vektora je číslo vyjadrujúce jeho veľkosť. ℓ_p -norma vektora $\mathbf{x} \in \mathbb{C}^N$ je definovaná ako

$$\begin{aligned}\|\mathbf{x}\|_p &:= \left(\sum_{i=1}^N |x_i|^p\right)^{(1/p)} \text{ pre } 1 \leq p < \infty, \\ \|\mathbf{x}\|_p &:= \sum_{i=1}^N |x_i|^p \text{ pre } 0 < p < 1, \\ \|\mathbf{x}\|_\infty &:= \max_i |x_i|, \\ \|\mathbf{x}\|_0 &:= |\text{supp}(\mathbf{x})|.\end{aligned}\tag{2.1}$$

O normu sa prísne vzaté jedná len v prípade $1 \leq p < \infty$, pre zjednodušenie sa ale pre všetky p bude používať jednotné označenie ℓ_p -norma [8].

Norma $\|\cdot\|_1$ predstavuje súčet absolútnych hodnôt prvkov vektora ($\|\mathbf{x}\|_1 = \sum_i |x_i|$) a norma $\|\cdot\|_0$ reprezentuje počet nenulových zložiek vektora. Eukleidovská norma $\|\cdot\|_2 = \sqrt{\sum_i |x_i|^2}$ predstavuje vzdialenosť vektora od počiatku súradníc.

2.3 Bázy a framy

Vektorový priestor (VP) je algebraická štruktúra, ktorej prvkami sú vektory. Každý vektor $\mathbf{x} \in \mathbb{V}$ je nejakou lineárnou kombináciou generátorov vektorového priestoru \mathbb{V} . Ak je generátorov \mathbb{V} viac ako dimenzií n , potom sa dá vektor $\mathbf{x} \in \mathbb{V}$ vyjadriť ako vzájomne rôzne lineárne kombinácie, preto jeden vektor môže mať viac reprezentácií, čo sa nazýva nedourčenosť. Ak sa v systéme generátorov $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m\}$ dá \mathbf{x} zapísať ako

$$\mathbf{x} = c_1 \mathbf{e}_1 + c_2 \mathbf{e}_2 + \dots + c_m \mathbf{e}_m = \mathbf{E} \mathbf{c}, \quad (2.2)$$

potom skaláry c_i sú súradnice \mathbf{x} v \mathbf{E} [8]. Zápis $\mathbf{E} \mathbf{c}$ znamená násobenie vektoru $\mathbf{c} = [c_1, \dots, c_m]^\top$ maticou \mathbf{E} , ktorá vznikne zoradením stĺpcov \mathbf{e}_i vedľa seba. To súhlasí s najčastejšie používaným spôsobom modelovania signálu, tzv. aditívnym, kde signál \mathbf{x} je lineárne kombinovaný z dielčích signálov:

$$\mathbf{x} = \sum_{i=1}^m c_i \mathbf{e}_i. \quad (2.3)$$

Vektorové priestory môžu mať určenú normu, ktorá informuje o veľkosti vektora, a skalárny súčin, ktorý udáva uhol medzi dvoma vektormi. Vtedy sa jedná o normované priestory, resp. priestory s vnútorným súčinom [12]. Norma sa zavádza indukovanou pomocou skalárneho súčinu $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. Vektory $\mathbf{x} = [x_1, \dots, x_N]^\top$ nachádzajúci sa v priestore \mathbb{C}^N sa väčšinou doplní o skalárny súčin [8]

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^N x_i \bar{y}_i = \mathbf{y}^* \mathbf{x} \quad (2.4)$$

a indukovanú normu

$$\|\mathbf{x}\|_2 := \|\mathbf{x}\| = \sqrt{\sum_{i=1}^N |x_i|^2} = \sqrt{\mathbf{x}^* \mathbf{x}}. \quad (2.5)$$

Báza vektorového priestoru je minimálny systém jeho generátorov; množina lineárne nezávislých vektorov, ktorých lineárnou kombináciou sa dá určiť každý vektor v danom vektorovom priestore. V priestore s konečnými rozmermi a dimenziou N obsahuje báza N vektorov, ktoré sú lineárne nezávislé. Ak je $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_N\}$ báza, potom sa každý prvok $\mathbf{x} \in \mathbb{V}$ v tej bázi dá vyjadriť pomocou jednoznačných súradníc c_i [8]:

$$\mathbf{x} = \sum_{i=1}^N c_i \mathbf{b}_i = \mathbf{B} \mathbf{c}. \quad (2.6)$$

Najviac sa používajú bázy ortogonálne a ortonormálne. Ak pre ľubovoľné 2 vektory z bázy $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ platí

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \text{ pre } i \neq j, \quad \langle \mathbf{b}_i, \mathbf{b}_i \rangle \neq 0, \quad (2.7)$$

potom všetky dvojice báзовých vektorov sú na seba kolmé a táto báza je ortogonálna. V ortonormálnej báze navyše platí pre každý jej prvok $\|\mathbf{b}_i\| = 1$, alebo $\mathbf{B}^* = \mathbf{B}^{-1}$ [8].

Dve rôzne bázy v rovnakom vektorovom priestore $\mathbf{E} = \{\mathbf{e}_1, \dots, \mathbf{e}_N\}$ a $\mathbf{F} = \{\mathbf{f}_1, \dots, \mathbf{f}_N\}$ sú navzájom biortogonálne, ak spĺňajú nasledujúcu podmienku:

$$\langle \mathbf{e}_i, \mathbf{f}_j \rangle = \begin{cases} 1 & \text{pre } i = j, \\ 0 & \text{pre } i \neq j, \end{cases} \quad (2.8)$$

alebo $\mathbf{F}^* \mathbf{E} = \mathbf{I}$. Nie je potrebné aby dvojice vektorov z \mathbf{E} a \mathbf{F} na seba boli kolmé, no každý vektor z \mathbf{E} musí byť kolmý na všetky vektory z \mathbf{F} okrem jedného. Existujú aj bázy biortonormálne.

Pri väčšom počte generátorov ako dimenzie je stále možné vyjadriť hocikáky vektor vo vektorovom priestore, no generátory budú lineárne závislé. Takáto množina vektorov sa volá frame. Framy sú menej obmedzené ako bázy a vďaka ich flexibilitě sa využívajú, s tým ale prichádzajú nevýhody v podobe väčšej numerickej nestability a ťažších výpočtových metód.

Frame je vytvorený v priestore \mathbb{V} množinou vektorov $\{\phi_k\}_{k \in \mathbb{I}}$, ak existujú také konštanty $0 < A \leq B < \infty$, že platí [8]

$$A \|\mathbf{x}\|^2 \leq \sum_{k \in \mathbb{I}} |\langle \mathbf{x}, \phi_k \rangle|^2 \leq B \|\mathbf{x}\|^2, \quad \forall \mathbf{x} \in \mathbb{V}. \quad (2.9)$$

Prvky framu sa nazývajú atómy a konštanty A, B sú medze framu [8].

Pre lineárny obal framu je frame každá konečná množina vektorov. Z každého framu v konečnej dimenzii sa dá zvoliť báza [13].

Frame je tesný (*tight*), ak platí $A = B$. Táto rovnosť platí aj pre ortogonálne bázy s rovnako veľkými prvkami. Zvláštny prípad nastáva ak $A = B = 1$, frame je potom 1-tesný, taktiež nazývaný ako Parsevalov tesný frame.

Ďalej sa bude pracovať so separabilným Hilbertovým priestorom \mathcal{H} , ktorého vlastnosti sú unitárnosť, úplnosť a separabilita [8]. Definované bázy v \mathcal{H} sú napr. Schauderova alebo Riezsova.

Príkladom úplného separabilného Hilbertovho priestoru je priestor funkcií s konečnou energiou $L^2(\mathbb{R})$. Je definovaný ako

$$L^2(\mathbb{R}) = \left\{ f : \sqrt{\int_{-\infty}^{\infty} |f(t)|^2 dt} < \infty \right\}, \quad (2.10)$$

pričom f je komplexná funkcia reálnej premennej.

Systémy generátorov pre sú často stavané na princípoch dilatácie a translácie. Takéto systémy pre $L^2(\mathbb{R})$ existujú v časovo–kmitočtovom pojatí signálov, kde sú framy vytvárané pomocou translácie a modulácie fixnej funkcie $g \in L^2(\mathbb{R})$. Súbor takýchto funkcií [8]

$$\left\{ e^{j2\pi mbx} g(x - na) \right\}_{m,n \in \mathbb{Z}} \quad (2.11)$$

pre zvolené hodnoty a a b sa nazýva Gaborov systém. Iné názvy sú krátkodobá Fourierova analýza alebo transformácia, STFT (*Short-Time Fourier Transform*). Tento typ systému sa používa kvôli tomu, že klasická Fourierova transformácia pracuje s harmonickými funkciami s globálnym dosahom, tzn. jeden koeficient dokáže ovplyvniť celý signál. Človek vníma zvuk jeho kmitočtovou charakteristikou, umiestnenou v čase, preto je potrebné analyzovať časovo lokálne spektrá signálu. Odtiaľ je pôvod pojmu časovo–kmitočtová reprezentácia.

Ak je gaborovský systém frame, dá sa každý signál zrekonštruovať z jeho časovo–kmitočtových koeficientov. Gaborova transformácia sa však väčšinou používa na analýzu signálu. Koeficienty majú mriežkovú štruktúru, vďaka čomu sa dá signál vyobraziť v podobe spektrogramu – dvojrozmerné pole Gaborových koeficientov.

2.4 Riedkosť signálu a riedke riešenia

K moderným metódam v spracovaní signálov patria reprezentácie signálov pomocou nedourčených systémov lineárnych rovníc. V takýchto systémoch nie je reprezentácia signálu jednoznačne určená, pretože v nich existuje viac neznámych ako lineárnych vzťahov medzi nimi. Reprezentácia signálu preto nie je jednoznačne určená a kvôli tomu sa hľadá reprezentácia, ktorá je z nejakého pohľadu optimálna. Dnes sa často využívajú algoritmy, ktoré hľadajú tzv. riedku reprezentáciu signálu, tzn. keď je signál v nejakom systéme určený len niekoľkými prvkami [8].

Riedke riešenia systémov lineárnych rovníc sú také, ktoré majú čo najviac neznámych súčasne nulových. Uplatňuje sa to pri uľahčení interpretácie dát, pri jednoduchšej a silnej kompresii a pri veľa iných aplikáciách.

Vektor $\mathbf{x} \in \mathbb{C}^N$ môžeme nazvať k -riedkym (k -sparse), ak platí

$$\|\mathbf{x}\|_0 \leq k. \quad (2.12)$$

k -riedky vektor je teda taký, ktorý má najviac k nenulových zložiek. Reálna riedkosť vektora \mathbf{x} s dĺžkou N je potom definovaná ako pomer $\frac{k}{N}$.

Podstatnú časť informácie o signáli obsahuje len zlomok koeficientov. Nie je to však presné pojetie riedkosti, teda ako počet nenulových koeficientov podľa vzťahu (2.2). Väčšina koeficientov signálu je síce zanedbateľných, no nie sú nulové. Preto sa zavádza pojem aproximatívna riedkosť alebo kompresibilita signálu. Charakterizácia

takého signálu $\mathbf{y} \in \mathbb{R}^N$ je, že $\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon$, pričom $\|\epsilon\|_2 \ll \|\mathbf{y}\|_2$ a $\|\mathbf{x}\|_0 \ll N$, teda signál je možné až na malú chybu reprezentovať riedkou lineárnou kombináciou stĺpcov \mathbf{A} [8].

Pri riedkych riešeniach má byť obvyklá sústava rovníc $\mathbf{A}\mathbf{x} = \mathbf{y}$ riešená tak, aby neznámy hľadaný vektor \mathbf{x} bol čo najviac riedky, čiže aby obsahoval, ak je možné, čo najviac nulových zložiek. Ide teda o úlohu

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhľadom k } \mathbf{A}\mathbf{x} = \mathbf{y}, \quad (2.13)$$

pričom poznáme vektor $\mathbf{y} \in \mathbb{C}^m$ (pozorovanie, meranie, signál) a maticu $\mathbf{A} \in \mathbb{C}^{m \times N}$ definujúcu lineárne zobrazenie. Bude sa predpokladať s prípadmi, keď $m < N$ alebo $m \ll N$, a \mathbf{A} má plnú hodnotu. Matica \mathbf{A} je najčastejšie nazvaná slovník (*dictionary*) a stĺpce matice sú nazvané atómy (*atoms*). Všetky \mathbf{x} , ktoré spĺňajú $\mathbf{A}\mathbf{x} = \mathbf{y}$ nazývame prípustné riešenia. Pri vyššie uvedených podmienkach pre maticu \mathbf{A} je prípustných riešení nekonečno a tvoria afinný priestor (priestor, na ktorom je definované sčítanie bodov a vektorov; jedná sa o zobecnenie Euklidovského priestoru).

Pri riešení sa dá pracovať aj s aproximačnou úlohou, kde splnenie $\mathbf{A}\mathbf{x} = \mathbf{y}$ nie je striktné a povolená je odchýlka:

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhľadom k } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_p \leq \delta, \quad (2.14)$$

kde najčastejšie sa uvažuje $p = 2$. Takáto úloha väčšinou platí pre prípad, keď je signál zašumený alebo chybné zmeranie, a to môže spôsobiť odchýlku od presného riešenia [8].

Ďalším pojmom je *spark* („iskra“), je to vlastnosť pre rozhodovanie o existencii a vlastnostiach riešení. Číslo $\text{spark}(\mathbf{A})$ je definované ako najmenší počet lineárne závislých stĺpcov matice \mathbf{A} :

$$\text{spark}(\mathbf{A}) = \min_{\substack{\mathbf{z} \in \ker \mathbf{A}, \\ \mathbf{z} \neq \mathbf{0}}} \|\mathbf{z}\|_0. \quad (2.15)$$

Pre maticu $\mathbf{A} \in \mathbb{C}^{m \times N}$, ktorá je nenulová a $m < N$, platí $\text{spark}(\mathbf{A}) \in \{2, \dots, m+1\}$, pričom nadobudne hodnotu 2, ak je jeden stĺpec násobkom iného. Čím menší je spark , tým redší musí byť vektor \mathbf{x} pre zaistenie jedinečnosti riešenia [8].

Ak riešenie \mathbf{x} v sústave $\mathbf{A}\mathbf{x} = \mathbf{y}$ spĺňa

$$\|\mathbf{x}\|_0 < \frac{\text{spark}(\mathbf{A})}{2}, \quad (2.16)$$

potom platí, že \mathbf{x} je najredšie možné riešenie a žiadne iné rovnako riedke riešenie neexistuje. Ak sa podarí nájsť riešenie, ktorého kardinalita vyhovuje predošlému vzťahu, potom toto riešenie je zároveň riešenie pôvodného problému (2.13). Výpočetná náročnosť hľadania hodnoty $\text{spark}(\mathbf{A})$ je ale podobne veľká ako riešenie samotného problému, preto je potrebné použiť jednoduchší spôsob overenia unikátneho riešenia.

Toto je možné vďaka vzájomnej koherencii (*mutual coherence*), vlastnosti matice \mathbf{A} , definovanej ako najväčší absolútny normalizovaný skalárny súčin dvoch rôznych stĺpcov matice \mathbf{A} [14]:

$$\mu(\mathbf{A}) = \max_{\substack{1 \leq j, k \leq N \\ j \neq k}} \frac{|\mathbf{a}_j^\top \mathbf{a}_k|}{\|\mathbf{a}_j\|_2 \cdot \|\mathbf{a}_k\|_2}, \quad (2.17)$$

kde \mathbf{a}_j značí j -tý stĺpec matice \mathbf{A} . Vzájomnou koherenciou sa teda môže získať „miera lineárnej nezávislosti“ medzi stĺpcami matice [8].

Pre číslo $\text{spark}(\mathbf{A})$ a maticu \mathbf{A} platí vzťah

$$\text{spark}(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{A})}. \quad (2.18)$$

Ak riešenie \mathbf{x} sústavy $\mathbf{A}\mathbf{x} = \mathbf{b}$ spĺňa podmienku

$$\|\mathbf{x}\|_0 < \frac{1}{2} \left(1 + \frac{1}{\mu(\mathbf{A})}\right), \quad (2.19)$$

potom \mathbf{x} je jediné najredšie riešenie, a je dosiahnuteľné ℓ_1 -minimalizáciou [8].

Nižšia výpočetná náročnosť u tejto metódy je ale vyvážená väčšou nerovnosťou. Ak sa blíži koherencia nule, pravá strana vzťahu (2.19) sa zvyšuje nad všetky medze, a opačne ak sa koherencia blíži 1, pravá strana sa tiež približuje k 1. Existuje teda snaha hľadať slovníky, ktoré majú koherenciu čo najnižšiu.

Predpokladajme, že platí $\text{spark}(\mathbf{A}) > 2k_0$ a existuje nejaké riešenie sústavy, ktoré je k_0 -riedke, teda podľa podmienky (2.16) je toto riešenie jednoznačné a najredšie možné. Ak by bolo potrebné nájsť presné riešenie, muselo by sa prejsť všetkých $\binom{N}{k_0}$ kombinácií podmnožín stĺpcov matice [8].

Toto odpovedá NP-zložitosti, kde ak N je vysoké číslo, je to v praxi neprijateľné vzhľadom k výpočetnému času, ktorý by so súčasnou technikou mohol trvať minimálne niekoľko týždňov. Nezostáva teda nič iné, než zvýšiť rýchlosť procesu pri zväčšení rizika úbytku presnosti, ak chceme výsledok získať v normálnom čase.

2.5 ℓ_1 relaxácia

Keďže norma ℓ_0 nie je konvexnou funkciou, nie je možné pre problém (2.13) využiť žiadnu z metód a algoritmov konvexnej optimalizácie (konvexného programovania), ktoré sú v dnešnej dobe k dispozícii [15].

Pretože normy ℓ_p sú konvexné pre $p \geq 1$, je možné využiť najbližšiu konvexnú normu, v tomto prípade ℓ_1 , aspoň k približnému, no z časového hľadiska priaznivejšiemu riešeniu problému (2.13). Bude sa teda riešiť úloha

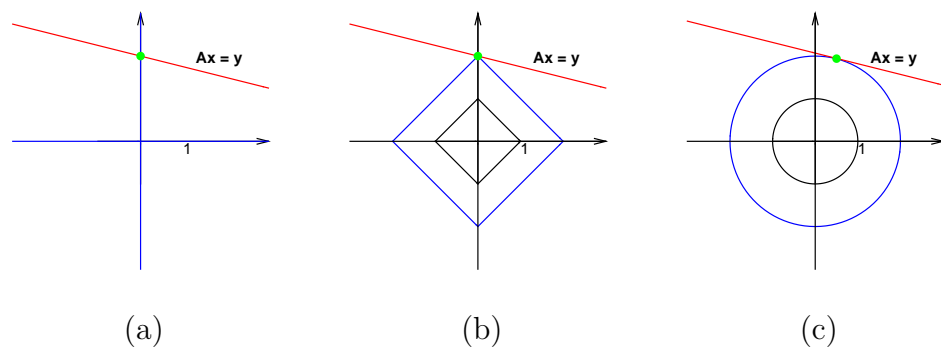
$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhľadom k } \mathbf{A}\mathbf{x} = \mathbf{y}. \quad (2.20)$$

Vo väčšine prípadov budú riešenia úloh (2.13) a (2.20) zhodné [8]. Ak vstupné dáta budú zašumené, povolí sa odchýlka vplyvom šumu alebo nepresností:

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhľadom k } \|\mathbf{Ax} - \mathbf{y}\|_2 \leq \delta. \quad (2.21)$$

Tento problém sa nazýva LASSO (*Least Absolute Shrinkage and Selection Operator*) [8].

Na obrázku 2.2 je vyobrazené riešenie úlohy $\min_{\mathbf{x}} \|\mathbf{x}\|_p$ vzhľadom k $\mathbf{Ax} = \mathbf{y}$ v priestore \mathbb{R}^2 pre $p = 0, 1$ a 2 . Prípustné riešenia sú dané červenou priamkou a z obrázka je zrejmé, že riešenie problémov (2.13) a (2.20) je zhodné. U euklidovskej normy $p = 2$ je riešenie odlišné, ide o riešenie s minimálnou energiou.



Obr. 2.2: Zobrazenie vrstevníc (a) ℓ_0 , (b) ℓ_1 a (c) ℓ_2 noriem a ich dotyk s nadrovinou $\mathbf{Ax} = \mathbf{y}$.

Na maticu \mathbf{A} a mieru riedkosti vektora \mathbf{x} sa vzťahujú ďalšie podmienky, ako napr. vlastnosť nulového priestoru (NSP) a vlastnosť zoslabenej izometrie (RIP) [8].

2.6 Proximálne algoritmy pre konvexnú optimalizáciu

Žiadna metóda riešenia optimalizačných úloh nemôže nájsť vždy presné riešenie, ktoré vedie k získaniu riedkej reprezentácie signálu. Preto sa používajú metódy aproximatívne, špeciálny prípad sú proximálne algoritmy, ktoré sa používajú v aplikáciách pri praktických výpočtoch. Algoritmy sa dajú rozdeliť do troch kategórií [8].

Prvou kategóriou sú tzv. „hladné“ (*greedy*) algoritmy, ktoré sú založené na princípe, že v každej iterácii nájdú jeden alebo viac najdôležitejších atómov. V každom ďalšom priebehu algoritmu je konečné riešenie závislé na vybraných atómoch a nie je možné sa ich zbaviť. Tieto metódy nie sú veľmi zložité, no nie je u nich zaručené

dosiahnutie globálneho optima, a v niektorých prípadoch môžu úplne zlyhať. Algoritmy v tejto kategórii sú napríklad *Matching Pursuit* (MP), *Orthogonal Matching Pursuit* (OMP) a rôzne ďalšie.

Ďalšie metódy využívajú ℓ_1 -relaxáciu a podľa toho sú aj nazvané ako relaxačné algoritmy. U týchto algoritmov sa pracuje s tým, že za určitých podmienok sa dostanú k presnému riešeniu, prípadne aspoň relatívne blízke. V tejto kategórii existujú metódy *Basis Pursuit* (BP), *Least Angle Regression* (LARS), *Iterative Reweighted Least Squares* (IRLS), niekedy tiež nazvaná FOCUSS (*FOCal Underdetermined System Solver*) alebo *Dantzig Selector*. Na tieto úlohy je možné použiť aj proximálne algoritmy, keďže ℓ_1 -norma je konvexná funkcia [8].

Mimo uvedených kategórií sú algoritmy využívajúce prahovanie (*thresholding*) a tie, ktoré sú postavené na výhodách oboch skupín algoritmov. Táto kategória sa uvádza ako hybridná. Jedným z algoritmov je A*OMP, ktorý využíva A*-algoritmus prehľadávania informačných stromov. Ďalšie algoritmy sú postavené na myšlienke aproximatívneho posielaní správ v grafovom modeli (*approximate message passing*).

2.6.1 Proximálne algoritmy, proximálny operátor

Proximálne algoritmy sú postupy z teórie optimalizácie a zahŕňajú celý rad konvexných úloh, vrátane relaxovaných úloh s ℓ_1 normou. Optimalizácia si zakladá na iteratívnej minimalizácii funkcionálov pri známych podmienkach konvergenie algoritmu k optimu úlohy. Tieto metódy sú flexibilné, no ich rýchlosť spracovania je o niečo menšia ako u špecializovaných algoritmov.

Pracovať sa bude s algoritmom, ktorý dokáže nájsť celkom obecné riešenie tzv. konvexnej úlohy bez obmedzujúcich (*constrained*) podmienok [8]

$$\arg \min_{\mathbf{x}} g(\mathbf{x}) + h(\mathbf{x}). \quad (2.22)$$

Úloha (2.20) je optimalizačná úloha s obmedzujúcimi podmienkami, no neodpovedá (2.22). Na formu bez obmedzujúcich podmienok sa ju však dá transformovať s využitím indikátorovej (charakteristickej) funkcie. Indikátorová funkcia množiny Γ , ktorá je neprázdna podmnožina \mathbb{R}^N , je

$$\iota_{\Gamma}: \mathbf{x} \mapsto \begin{cases} 0 & \text{pre } \mathbf{x} \in \Gamma, \\ \infty & \text{inak.} \end{cases} \quad (2.23)$$

Obecné riešenie úlohy (2.23) je potom

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 + \iota_{\Gamma}. \quad (2.24)$$

Namiesto (2.20) sa potom rieši úloha, ktorá už je v potrebnom tvare, a druhý sčítanec zabezpečuje riešenie v prípustnej množine:

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 + \iota_{\{\mathbf{x}: \mathbf{Ax}=\mathbf{y}\}}. \quad (2.25)$$

Medzi obmedzenou úlohou (2.21) a neobmedzenou existuje ekvivalencia, sčasti nesprávne tiež nazývaná LASSO [8]. Ekvivalencia znamená, že obidve úlohy dávajú rovnaký vektor optimálnych \mathbf{x} . Táto úloha sa nazýva tiež regularizovaná lineárna inverzná úloha, keďže dodatočným členom sa v nej penalizujú niektoré riešenia z nekonečne mnohých, čím sa úloha tzv. regularizuje.

Čím väčší bude dodatočný člen, tým viac budú penalizované neriedke riešenia, na druhú stranu pri moc vysokej hodnote tohto člena bude príliš veľká odchýlka od dát. Voľba správnej hodnoty regularizačného parametra tohto člena je teda veľmi dôležitá.

Proximálny operátor konvexnej funkcie je základ algoritmov. Prirodzene rozširuje pojem projekčný operátor na konvexnú množinu, ktorý nachádza bod v tejto množine ležiaci najbližšie k predvolenému bodu.

Nech $f: \mathbb{R}^N \rightarrow (-\infty, \infty)$ je konvexná funkcia, zdola polospojité s neprázdny definíčným oborom. Pre každé $\mathbf{x} \in \mathbb{R}^N$ má minimalizačná úloha

$$\arg \min_{\mathbf{y} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + f(\mathbf{y}) \quad (2.26)$$

riešenie, ktoré je jednoznačné a bude sa značiť ako $\text{prox}_f \mathbf{x}$. Operátor prox_f sa nazýva proximálny operátor funkcie f [8].

Vzťah (2.26) je možné považovať za regularizované potlačovanie šumu (*regularized denoising*). Ak daný signál \mathbf{x} vznikol pridaním šumu do čistého signálu, potom člen $\|\mathbf{x} - \mathbf{y}\|_2^2$ zabezpečuje, aby výsledok odšumovania bol „blízko“ od známeho signálu \mathbf{x} , no pridaný šum musí byť rozumne veľký. Regularizér $f(\mathbf{y})$ penalizuje \mathbf{y} , ktoré sú hľadané za pomoci konvexnej funkcie, a tým vynucuje niektoré vlastnosti \mathbf{y} , ako napr. energiu, relaxovanú riedkosť alebo totálnu variáciu.

2.6.2 Proximálny operátor ℓ_1 normy

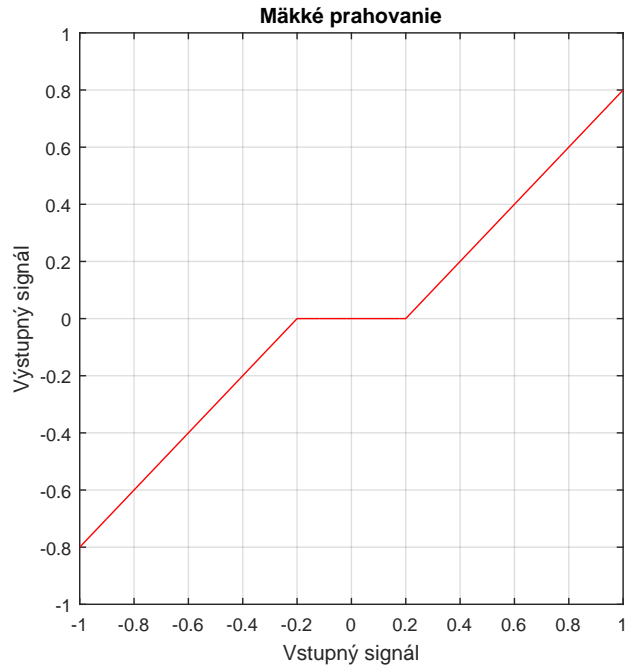
Pri hľadaní riedkych reprezentácii signálu je veľmi dôležitá norma ℓ_1 . Proximálny operátor pre prípad $f(\mathbf{y}) = \gamma \|\mathbf{y}\|_1$, kde γ je regularizačný skalár, ktorý sa využíva v aplikáciách, sa dá podľa definície odvodiť ako [8]

$$\text{prox}_{\gamma \|\cdot\|_1}(\mathbf{x}) = \arg \min_{\mathbf{y}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \gamma \|\mathbf{y}\|_1. \quad (2.27)$$

Analýzou po intervaloch, kde je funkcia $|\cdot|$ diferencovateľná, dostaneme funkciu

$$y_i = \frac{x_i}{|x_i|} \max(|x_i| - \gamma, 0), \quad (2.28)$$

ktorá sa nazýva mäkké prahovanie (*soft thresholding*). Tá sa bude značiť ako $y_i = \text{soft}_\gamma(x_i)$, alebo vektorovo $\mathbf{y} = \text{soft}_\gamma(\mathbf{x})$, pričom operátor bude prahovať každú zložku \mathbf{x} zvlášť [8]. Príklad prevodnej charakteristiky mäkkého prahovania je uvedený na obrázku 2.3.



Obr. 2.3: Prevodná charakteristika mäkkého prahovania pre $\gamma = 0,2$

2.7 Syntezujúci a analyzujúci model signálu

Každý diskretný signál \mathbf{y} sa dá zapísať ako súčin vektora súradníc a tzv. transformačnej matice, pričom tento proces sa nazýva transformácia. Sústava rovníc $\mathbf{A}\mathbf{x} = \mathbf{y}$ sa dá riešiť pomocou riedkych reprezentácií signálu a využívajú sa dva modely pre toto riešenie.

Prvý model je syntezujúci alebo aditívny, keďže signál \mathbf{y} je vyjadrený ako súčin transformačnej matice \mathbf{A} a vektora súradníc \mathbf{x} , a je poskladaný z niekoľkých zložiek.

Druhý model sa nazýva analyzujúci. V ňom sa hľadá vektor súradníc \mathbf{x} , ktorý je daný ako súčin vektora reprezentácie signálu \mathbf{y} a transformačnej matice $\mathbf{\Omega}$. Pri použití báz sú oba modely signálu zhodné, platí $\mathbf{A} = \mathbf{\Omega}^{-1}$. Príkladom tohto modelu je napríklad Fourierova transformácia, kde signál je určený ako súčet harmonických signálov rôznych kmitočtov [9]. Pre hľadanie riedkych riešení sa potom používajú algoritmy, ktorých typy sú uvedené v sekcii 2.6.1.

3 Algoritmy pre rekonštrukciu signálu

V tejto časti budú popísané priebehy rekonštrukčných algoritmov.

K dispozícii je vstupný číslicový signál $\mathbf{x} \in \mathbb{R}^N$, kvantizovaný na bežne používané 16-bitové rozlíšenie, teda počet kvantizačných hladín, ktoré jednotlivé vzorky signálu môžu nadobnúť je $2^{16} = 65536$. Tento signál sa bude najprv kvantizovať na n kvantizačných hladín \mathbf{q} s hodnotami $\langle -1, 1 \rangle$, ktoré môžu byť rôzne rozložené a vzdialené od seba s rovnakými alebo rôznymi intervalmi, a tým sa získa kvantizovaný signál $\mathbf{y} \in \mathbb{R}^N$.

Vytvorí sa vektor kvantizačných hladín \mathbf{q} a zároveň sa vypočítajú prahy kvantizačných hladín \mathbf{b} – rozhodovacie úrovne medzi kvantizačnými hladinami, na ktoré sa budú vzorky signálu zaokrúhľovať.

Pre kvantizovaný signál budú vytvorené dva vektory najbližších kvantizačných prahov každej vzorky, pre spodné prahy a pre vrchné prahy. Pre rekonštrukciu je to dôležité v tom, že každá vzorka poškodeného signálu má hodnotu medzi korešpondujúcim spodným a vrchným prahom, a toto platí aj u pôvodného vstupného signálu.

3.1 Algoritmy založené na konvexnej optimalizácii

Algoritmy budú pracovať so syntezujúcim alebo analyzujúcim modelom signálu. Pre syntezujúci model sa bude používať Douglas–Rachfordov (DR) algoritmus, pre analyzujúci sa využije Chambolle–Pockov (CP) algoritmus.

3.1.1 Algoritmus pre syntezujúci model signálu

Bude sa predpokladať, že spracovaný signál je riedky napr. v Gaborovej transformácii a pre rekonštrukciu \mathbf{y} sa bude hľadať čo najredší vektor koeficientov \mathbf{c} tejto reprezentácie. Pre hľadanie riedkeho riešenia sa využije ℓ_1 norma. Problém pre syntezujúci model riešenia sa dá zapísať ako

$$\tilde{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 \text{ vzhľadom k } \|\mathbf{y} - D\mathbf{c}\|_\infty \leq \frac{\Delta}{2}, \quad (3.1)$$

kde Δ je rozdiel medzi dvoma susediacimi kvantizačnými hladinami a $D: \mathbb{C}^P \rightarrow \mathbb{R}^N$ je operátor syntézy Parsevalovho tesného framu. Reštaurovaný signál sa potom získa spätnou transformáciou $D\tilde{\mathbf{c}}$.

Množina prípustných riešení pre koeficienty bude

$$\Gamma^* = \left\{ \mathbf{c} \mid \|\mathbf{y} - D\mathbf{c}\|_\infty \leq \frac{\Delta}{2} \right\}. \quad (3.2)$$

Pre použitie vyhovujúceho algoritmu je vhodné prepísať problém (3.1) do neobmedzeného tvaru [10]

$$\arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 + \iota_{\Gamma^*}(\mathbf{c}), \quad (3.3)$$

a indikátorová funkcia je potom

$$\iota_{\Gamma^*}(\mathbf{c}) = \begin{cases} 0 & \text{pre } \mathbf{c} \in \Gamma^*, \\ +\infty & \text{pre } \mathbf{c} \notin \Gamma^*. \end{cases} \quad (3.4)$$

K ℓ_1 norme v probléme (3.3) je možné pridať vektor váh \mathbf{w} , ktorá môže značne ovplyvniť výsledok behu algoritmu. Táto funkcia vo vlastnom riešení nebude použitá.

Vlastnosť indikátorovej funkcie je taká, že ak riešenie leží v množine Γ^* , bude sa rovnať 0, a potom v probléme (3.1) sa bude brať do úvahy len ℓ_1 minimaliácia. Ak riešenie neleží v množine Γ^* , potom hodnota ι_{Γ^*} bude $+\infty$. To znamená, že minimalizácia nadobudne hodnotu $+\infty$ a riešenie nebude zaručene minimálne.

Pre hľadanie riešenia úlohy tvaru (3.3) je pre syntezujúci model vhodné použiť Douglas–Rachfordov algoritmus, pričom je nutné aby obe funkcie boli konvexné, čo je pri úlohe (3.3) splnené.

Obecný zápis DR algoritmu [16] je nasledovný:

Algoritmus 1: Obecný Douglas–Rachfordov algoritmus

Input: Funkcie f_1, f_2

Output: $\tilde{\mathbf{c}}^{(i+1)}$

Nastav parametre $\lambda = 1$ a $\gamma > 0$;

Nastav počiatočnú hodnotu premennej $\mathbf{c}^{(0)} \in \mathbb{C}^P$;

Nastav hodnotu iteračného čítača $i = 0$;

repeat

$$\left| \begin{array}{l} \tilde{\mathbf{c}}^{(i)} = \text{prox}_{\gamma f_2} \mathbf{c}^{(i)}; \\ \mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} + \lambda(\text{prox}_{\gamma f_1}(2\tilde{\mathbf{c}}^{(i)} - \mathbf{c}^{(i)}) - \tilde{\mathbf{c}}^{(i)}); \end{array} \right.$$

until convergence;

return $\tilde{\mathbf{c}}^{(i+1)}$

Algoritmus iteruje v dvoch krokoch. V prvom kroku proximálny operátor prox_{f_2} , kde f_2 značí indikátorovú funkciu ι_{Γ^*} , odpovedá projekcii na množinu prípustných riešení proj_{Γ^*} , ktorá korešponduje s ι_{Γ^*} . Tento krok sa dá vypočítať ako [11]

$$\text{proj}_{\Gamma^*}(\mathbf{c}) = \mathbf{c} - D^*(D\mathbf{c} - \text{proj}(D\mathbf{c})), \quad (3.5)$$

kde D^* je operátor analýzy a vnútorná projekcia $\text{proj}(D\mathbf{c})$ je jednoduchá projekcia v časovej doméne, ktorá sa vypočíta postupne pre každú vzorku ako

$$\left(\text{proj}(D\mathbf{c}) \right)_n = \begin{cases} (D\mathbf{c})_n & \text{pre } n \in M_R(D\mathbf{c})_n, \\ (\mathbf{b}_H)_n & \text{pre } n \in M_H(D\mathbf{c})_n, \\ (\mathbf{b}_L)_n & \text{pre } n \in M_L(D\mathbf{c})_n, \end{cases} \quad (3.6)$$

kde \mathbf{b}_H a \mathbf{b}_L sú vektory najbližších spodných a vrchných kvantizačných prahov pre celú dĺžku signálu.

V druhom kroku proximálny operátor ℓ_1 normy prox_{f_1} predstavuje mäkké prahovanie soft_γ opísané v sekcii 2.6.2, ktoré sa vypočíta podľa

$$\text{soft}_\gamma(\mathbf{c}) = \text{sgn}(\mathbf{c}) \odot \max(|\mathbf{c} - \gamma|, 0). \quad (3.7)$$

DR algoritmus pre dekvantizáciu má nasledujúci zápis:

Algoritmus 2: Douglas–Rachfordov algoritmus pre dekvantizáciu

Input: Funkcie f_1, f_2

Output: $\tilde{\mathbf{c}}^{(i+1)}$

Nastav parametre $\lambda = 1$ a $\gamma > 0$;

Nastav počiatočnú hodnotu premennej $\mathbf{c}^{(0)} \in \mathbb{C}^P$;

Nastav hodnotu iteračného čítača $i = 0$;

repeat

$$\left| \begin{array}{l} \tilde{\mathbf{c}}^{(i)} = \text{proj}_{\Gamma^*} \mathbf{c}^{(i)}; \\ \mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} + \lambda(\text{soft}_\gamma(2\tilde{\mathbf{c}}^{(i)} - \mathbf{c}^{(i)}) - \tilde{\mathbf{c}}^{(i)}); \end{array} \right.$$

until convergence;

return $\tilde{\mathbf{c}}^{(i+1)}$

3.1.2 Algoritmus pre analyzujúci model signálu

Pri použití analyzujúceho modelu je riešenie úlohy dekvantizácie nasledovné [17]:

$$\arg \min_{\mathbf{x}} \|A\mathbf{x}\|_1 \text{ vzhľadom k } \mathbf{x} \in \Gamma, \quad (3.8)$$

alebo v neobmedzenom tvare

$$\arg \min_{\mathbf{x}} \|A\mathbf{x}\|_1 + \iota_\Gamma(\mathbf{x}), \quad (3.9)$$

kde $A: \mathbb{R}^N \rightarrow \mathbb{C}^P$ je operátor analýzy.

Množina prípustných riešení pre vzorky v časovej doméne bude daná

$$\Gamma = \left\{ \mathbf{x} \mid \|\mathbf{y} - \mathbf{x}\|_\infty \leq \frac{\Delta}{2} \right\}. \quad (3.10)$$

Pre nájdenie riešenia úlohy tvaru (3.9) je vhodné použiť Chambolle–Pockov algoritmus, pričom obe funkcie musia byť konvexné, čo je pri úlohe (3.9) splnené. Žiadna z týchto funkcií nemusí byť hladká.

Obecný zápis CP algoritmu [17] je nasledovný:

Algoritmus 3: Obecný Chambolle–Pockov algoritmus

Input: Funkcie f_1, f_2 , lineárny operátor $A: \mathbb{R}^N \rightarrow \mathbb{C}^P$

Output: $\tilde{\mathbf{x}}^{(i)}$

Nastav parametre $\zeta, \sigma > 0$ a $\theta \in [0, 1]$;

Nastav počiatočnú hodnotu primárnej premennej $\mathbf{x}^{(0)} \in \mathbb{R}^N$ a duálnej premennej $\mathbf{c}^{(0)} \in \mathbb{C}^P$;

Nastav hodnotu výstupnej premennej $\tilde{\mathbf{x}}^{(0)} = \mathbf{x}^{(0)}$;

Nastav hodnotu iteračného čítača $i = 0$;

repeat

$$\begin{cases} \mathbf{c}^{(i+1)} = \text{prox}_{\sigma f_2^*}(\mathbf{c}^{(i)} + \sigma A \tilde{\mathbf{x}}^{(i)}); \\ \mathbf{x}^{(i+1)} = \text{prox}_{\zeta f_1}(\mathbf{x}^{(i)} - \zeta A^* \mathbf{c}^{(i+1)}); \\ \tilde{\mathbf{x}}^{(i+1)} = \mathbf{x}^{(i+1)} + \theta(\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}); \end{cases}$$

until *convergence*;

return $\tilde{\mathbf{x}}^{(i)}$

Splnenie konvergenzie pri $\theta = 1$ nastane, ak platí podmienka $\zeta \sigma \|A\|^2 < 1$. A nadobúda hodnoty $\|A\| < 1$ a parametre ζ, σ sa nastavujú podľa vzťahu $\zeta = 1/\sigma$. Označenie f^* reprezentuje Fenchell–Rockafellarovo združenie ku konvexnej funkcii f . Proximálny operátor f^* sa potom získa pomocou Moreauovej identity [17].

Funkcia f_1 je indikátorová funkcia a takisto ako u DR algoritmu, jej proximálny operátor je projekcia na množinu prípustných riešení Γ , proj_Γ . Projekcia je jednoduchá, vypočíta sa podľa vzťahu (3.6).

Funkcia f_2 predstavuje ℓ_1 normu a jej proximálnym operátorom je mäkké prahovanie soft_σ , pre ktoré platí vzťah (3.7). Keďže CP algoritmus neobsahuje priamo funkciu f_2 ale jej Fenchell–Rockafellerovské združenie, potom proximálny operátor tejto funkcie bude funkcia clip [17], ktorá je definovaná ako

$$\text{clip}_\gamma(\mathbf{c}) = \text{sgn}(\mathbf{c}) \odot \min(|\mathbf{c}|, \gamma). \quad (3.11)$$

Pri využití týchto funkcií je CP algoritmus pre dekvantizáciu popísaný nasledovne:

Algoritmus 4: Chambolle–Pockov algoritmus pre dekvantizáciu

Input: Funkcie f_1, f_2 , lineárny operátor $A: \mathbb{R}^N \rightarrow \mathbb{C}^P$

Output: $\tilde{\mathbf{x}}^{(i)}$

Nastav parametre $\zeta, \sigma > 0$ a $\theta \in [0, 1]$;

Nastav počiatočnú hodnotu primárnej premennej $\mathbf{x}^{(0)} \in \mathbb{R}^N$ a duálnej premennej $\mathbf{c}^{(0)} \in \mathbb{C}^P$;

Nastav hodnotu výstupnej premennej $\tilde{\mathbf{x}}^{(0)} = \mathbf{x}^{(0)}$;

Nastav hodnotu iteračného čítača $i = 0$;

repeat

$$\begin{cases} \mathbf{c}^{(i+1)} = \text{clip}(\mathbf{c}^{(i)} + \sigma A \tilde{\mathbf{x}}^{(i)}); \\ \mathbf{x}^{(i+1)} = \text{proj}_{\Gamma}(\mathbf{x}^{(i)} - \zeta A^* \mathbf{c}^{(i+1)}); \\ \tilde{\mathbf{x}}^{(i+1)} = \mathbf{x}^{(i+1)} + \theta(\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}); \end{cases}$$

until *convergence*;

return $\tilde{\mathbf{x}}^{(i)}$

3.2 Algoritmy založené na nekonvexnej optimalizácii – SPADE

SPADE (*S*Parse *A*udio *D*Eclipper) [18] je heuristický algoritmus, pôvodne určený pre declipping, využívajúci riedkosť signálu. V tejto časti bude opísaná metóda ADMM – základ algoritmov SPADE, a použitie týchto algoritmov pre analyzujúci aj syntezujúci model signálu.

3.2.1 ADMM

SPADE je odvodený od metódy ADMM (*A*lternating *D*irection *M*ethod of *M*ultipliers). ADMM dokáže riešiť problémy formy

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(L\mathbf{x}), \quad (3.12)$$

kde $\mathbf{x} \in \mathbb{R}^N$ a $L: \mathbb{R}^N \rightarrow \mathbb{C}^P$ je lineárny operátor. Bude sa predpokladať, že f a g sú reálne konvexné funkcie reálnych, prípadne komplexných premenných [19]. Problém (3.12) sa dá prepracovať na

$$\min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \text{ vzhľadom k } L\mathbf{x} - \mathbf{z} = 0. \quad (3.13)$$

Pre riešenie optimalizačnej úlohy (3.13) sa vytvorí tzv. *Augmented Lagrangian* rovnica v tvare

$$L_{\rho}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^{\top}(L\mathbf{x} - \mathbf{z}) + \frac{\rho}{2} \|L\mathbf{x} - \mathbf{z}\|_2^2, \quad (3.14)$$

kde $\rho > 0$ je penalizačný parameter. ADMM pozostáva z troch krokov:

$$\mathbf{x}^{(i+1)} = \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}, \mathbf{z}^{(i)}, \mathbf{y}^{(i)}), \quad (3.15a)$$

$$\mathbf{z}^{(i+1)} = \arg \min_{\mathbf{z}} L_{\rho}(\mathbf{x}^{(i+1)}, \mathbf{z}, \mathbf{y}^{(i)}), \quad (3.15b)$$

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \rho(L\mathbf{x}^{(i+1)} - \mathbf{z}^{(i+1)}). \quad (3.15c)$$

Pre prepísanie rovnice do zmenšenej (*scaled*) formy, ktorá je praktickejšia, je potrebné zdefinovať reziduál $\mathbf{r} = L\mathbf{x} - \mathbf{z}$ a duálnu premennú $\mathbf{u} = \mathbf{y}/\rho$. Rovnica *Augmented Lagrangian* v zmenšenej forme potom bude

$$L_{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{r} + \mathbf{u}\|_2^2 - \frac{\rho}{2} \|\mathbf{u}\|_2^2, \quad (3.16)$$

Zmenšená (*scaled*) verzia ADMM má nasledujúci tvar [19]:

$$\mathbf{x}^{(i+1)} = \arg \min_{\mathbf{x}} \left(f(\mathbf{x}) + \frac{\rho}{2} \|L\mathbf{x} - \mathbf{z}^{(i)} + \mathbf{u}^{(i)}\|_2^2 \right), \quad (3.17a)$$

$$\mathbf{z}^{(i+1)} = \arg \min_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\rho}{2} \|L\mathbf{x}^{(i+1)} - \mathbf{z} + \mathbf{u}^{(i)}\|_2^2 \right), \quad (3.17b)$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + L\mathbf{x}^{(i+1)} - \mathbf{z}^{(i+1)}. \quad (3.17c)$$

Pri minimalizácii (3.14) podľa \mathbf{x} v prvom kroku (3.17a) je možné opomenúť $g(\mathbf{z})$, pretože nie je závislé na \mathbf{x} . Podobne v druhom kroku (3.17b) je možné vypustiť $f(\mathbf{x})$, pretože nie je závislé na \mathbf{z} , podľa ktorého sa minimalizuje. To isté platí aj pre člen $\frac{\rho}{2} \|\mathbf{u}\|_2^2$.

Pred rekonštrukciou signálu je potrebné signál rozdeliť na predom definované časti (bloky), pričom pre rozdelenie a zloženie naspäť – segmentáciu signálu sa používa metóda OLA (*overlap-add*). Signál sa rozdelí na bloky s nejakou dĺžkou a časti blokov sa prekrývajú so susednými blokmi. Bloky sú potom váhované, používa sa najmä Hannovo okno. Po reštaurácii sa bloky sčítajú do výsledného signálu [18].

SPADE algoritmus aproximuje riešenie NP-hard optimalizačných problémov pre analyzujúci, resp. syntezujúci model signálu:

$$\min_{\mathbf{x}, \mathbf{z}} \|\mathbf{z}\|_0 \quad \text{vzhľadom k } \mathbf{x} \in \Gamma(\mathbf{y}) \text{ a } \|A\mathbf{x} - \mathbf{z}\|_2 \leq \epsilon, \quad (3.18)$$

$$\min_{\mathbf{x}, \mathbf{z}} \|\mathbf{z}\|_0 \quad \text{vzhľadom k } \mathbf{x} \in \Gamma(\mathbf{y}) \text{ a } \|\mathbf{x} - D\mathbf{z}\|_2 \leq \epsilon. \quad (3.19)$$

Γ označuje množinu prípustných riešení (3.2), $\mathbf{x} \in \mathbb{R}^N$ značí signál v časovej doméne a $\mathbf{z} \in \mathbb{C}^P$ obsahuje koeficienty. Lineárne operátory analýzy $A: \mathbb{R}^N \rightarrow \mathbb{C}^P$ a syntézy $D: \mathbb{C}^P \rightarrow \mathbb{R}^N$ majú vzťah $D = A^*$ a pracuje sa s Parsevalovými tesnými framami. Na transformáciu signálu sa používa DFT a IDFT [18].

Po úprave kroku s projekciou je SPADE algoritmus možné použiť aj pre dekvantizáciu signálu, v tejto práci bude verzia pre dekvantizáciu nazvaná SPADEQ (*SParse Audio DEQuantizer*). Výpočetná náročnosť všetkých verzií SPADEQ algoritmov je rovnaká a výpočetne najnáročnejšie procesy sú transformácie.

3.2.2 A–SPADEQ algoritmus pre analyzujúci model signálu

Problém (3.18) sa rieši cez tri ADMM kroky:

$$\mathbf{z}^{(i+1)} = \arg \min_{\mathbf{z}} \left\| A\mathbf{x}^{(i)} - \mathbf{z} + \mathbf{u}^{(i)} \right\|_2^2 \text{ vzhľadom k } \|\mathbf{z}\|_0 \leq k, \quad (3.20a)$$

$$\mathbf{x}^{(i+1)} = \arg \min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{z}^{(i+1)} + \mathbf{u}^{(i)} \right\|_2^2 \text{ vzhľadom k } \mathbf{x} \in \Gamma, \quad (3.20b)$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + A\mathbf{x}^{(i+1)} - \mathbf{z}^{(i+1)}. \quad (3.20c)$$

Riešenie prvého kroku (3.20a) sa dá dosiahnuť implementáciou operátora tvrdého prahovania (*hard thresholding*) \mathcal{H}_k na $(A\mathbf{x}^{(i+1)} + \mathbf{u}^{(i)})$. Vyberie sa k najväčších hodnôt koeficientov, ktoré budú nezmenené a všetkým ostatným sa hodnota prepíše na nulu. Na začiatku sa stanoví nízka hodnota k a algoritmus pracuje iteratívne s rastúcim k tak dlho, kým algoritmus nedosiahne kritérium pre termináciu. Hodnota k narastá každú r -tú iteráciu algoritmu.

Ďalší krok (3.20b) je v skutočnosti projekcia $A^*(\mathbf{z}^{(i)} + \mathbf{u}^{(i)})$ na Γ , implementovaná postupne pre každú vzorku v časovej doméne [19]. Je to jednoduchá projekcia, prevedená podľa (3.6).

Po tomto kroku nasleduje podmienka, ktorá skončí priebeh algoritmu, ak rozdiel signálu po projekcii a pred ňou je menší ako parameter ϵ .

Posledný krok algoritmu predstavuje aktualizáciu duálnej premennej \mathbf{u} , čítača čísla iterácie a počtu koeficientov, ktoré vstupujú do operátora tvrdého prahovania.

Celý algoritmus je popísaný nasledovne:

Algoritmus 5: A–SPADEQ

Input: $A, \mathbf{y}, M_R, M_H, M_L, s, r, \epsilon$

- 1 $\tilde{\mathbf{x}}^{(0)} = \mathbf{y}, \mathbf{u}^{(0)} = 0, i = 1, k = s$
- 2 $\mathbf{z}^{(i+1)} = \mathcal{H}_k \left(A\tilde{\mathbf{x}}^{(i)} + \mathbf{u}^{(i)} \right)$
- 3 $\tilde{\mathbf{x}}^{(i+1)} = \arg \min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{z}^{(i+1)} + \mathbf{u}^{(i)} \right\|_2^2$ vzhľadom k $\mathbf{x} \in \Gamma$
- 4 **if** $\left\| A\tilde{\mathbf{x}}^{(i+1)} - \mathbf{z}^{(i+1)} \right\|_2 \leq \epsilon$ **then**
- 5 | terminate
- 6 **else**
- 7 | $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + A\tilde{\mathbf{x}}^{(i+1)} - \mathbf{z}^{(i+1)}$
- 8 | $i \leftarrow i + 1$
- 9 | **if** $i \bmod r = 0$ **then**
- 10 | | $k \leftarrow k + s$
- 11 | **end**
- 12 | go to 2
- 13 **end**
- 14 **return** $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(i+1)}$

3.2.3 S–SPADEQ a S–SPADEQ_{DR} algoritmy pre syntezujúci model signálu

Podobne pre syntezujúci model sa problém (3.19) rieši cez tri ADMM kroky:

S–SPADEQ	S–SPADEQ _{DR}
$\mathbf{z}^{(i+1)} = \arg \min_{\mathbf{z}} \left\ \mathbf{z} - \mathbf{z}^{(i)} + \mathbf{u}^{(i)} \right\ _2^2$ <p style="text-align: center;">vzhľadom k $\ \mathbf{z}\ _0 \leq k$, (3.21a)</p>	$\mathbf{z}^{(i+1)} = \arg \min_{\mathbf{z}} \left\ D\mathbf{z} - \mathbf{x}^{(i)} + \mathbf{u}^{(i)} \right\ _2^2$ <p style="text-align: center;">vzhľadom k $\ \mathbf{z}\ _0 \leq k$, (3.22a)</p>
$\tilde{\mathbf{z}}^{(i+1)} = \arg \min_{\mathbf{z}} \left\ \mathbf{z} - \mathbf{z}^{(i+1)} + \mathbf{u}^{(i)} \right\ _2^2$ <p style="text-align: center;">vzhľadom k $D\mathbf{z} \in \Gamma$, (3.21b)</p>	$\mathbf{x}^{(i+1)} = \arg \min_{\mathbf{x}} \left\ D\mathbf{z}^{(i+1)} - \mathbf{x} + \mathbf{u}^{(i)} \right\ _2^2$ <p style="text-align: center;">vzhľadom k $\mathbf{x} \in \Gamma$, (3.22b)</p>
$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \tilde{\mathbf{z}}^{(i+1)} - \mathbf{z}^{(i+1)}. \quad (3.21c)$	$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + D\mathbf{z}^{(i+1)} - \mathbf{x}^{(i+1)}. \quad (3.22c)$

V pôvodnej S–SPADE verzii sa oba minimalizačné kroky (3.21a), (3.21b) vykonávajú pomocou premennej \mathbf{z} . Aj keď je tento postup založený na ADMM, algoritmus rieši odlišný problém od (3.19) [20]. Preto sa o S–SPADE nedá hovoriť ako o syntezujúcom náprotivku verzie využívajúcej analyzujúci model. Kvôli tomu bola vytvorená nová verzia S–SPADE, nazvaná S–SPADE_{DR} (S–SPADE *done right*) [19], kde sa autori snažili odstrániť chyby pôvodnej verzie. Rozdiel je najmä v prvom kroku (3.22a), kde kvôli neortogonalite lineárneho operátora D je výsledný vektor iba aproximovanými riešením, na rozdiel od A–SPADE, kde tvrdé prahovanie \mathcal{H}_k rieši rovnicu (3.20a) presne. Druhý krok (3.22b) sa potom získa jednoduchou projekciou v čase, podobne ako u A–SPADEQ. Zápis S–SPADEQ algoritmov je nasledovný:

Algorithmus 6: S-SPADEQ	Algorithmus 7: S-SPADEQ _{DR}
Input: $D, \mathbf{y}, M_R, M_H, M_L, s, r, \epsilon$	Input: $D, \mathbf{y}, M_R, M_H, M_L, s, r, \epsilon$
1 $\tilde{\mathbf{z}}^{(0)} = D^* \mathbf{y}, \mathbf{u}^{(0)} = 0, i = 1, k = s$	1 $\tilde{\mathbf{x}}^{(0)} = D * \mathbf{y}, \mathbf{u}^{(0)} = 0, i = 1, k = s$
2 $\mathbf{z}^{(i+1)} = \mathcal{H}_k \left(\tilde{\mathbf{z}}^{(i)} + \mathbf{u}^{(i)} \right)$	2 $\mathbf{z}^{(i+1)} = \mathcal{H}_k \left(D^* (\tilde{\mathbf{x}}^{(i)} - \mathbf{u}^{(i)}) \right)$
3 $\tilde{\mathbf{z}}^{(i+1)} = \arg \min_{\mathbf{z}} \left\ \mathbf{z} - \mathbf{z}^{(i+1)} + \mathbf{u}^{(i)} \right\ _2^2$ vzhľadom k $D\mathbf{z} \in \Gamma$	3 $\tilde{\mathbf{x}}^{(i+1)} = \arg \min_{\mathbf{x}} \left\ D\mathbf{z}^{(i+1)} - \mathbf{x} + \mathbf{u}^{(i)} \right\ _2^2$ vzhľadom k $\mathbf{x} \in \Gamma$
4 if $\left\ \tilde{\mathbf{z}}^{(i+1)} - \mathbf{z}^{(i+1)} \right\ _2 \leq \epsilon$ then	4 if $\left\ D\mathbf{z}^{(i+1)} - \tilde{\mathbf{x}}^{(i+1)} \right\ _2 \leq \epsilon$ then
5 terminate	5 terminate
6 else	6 else
7 $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \tilde{\mathbf{z}}^{(i+1)} - \mathbf{z}^{(i+1)}$	7 $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + D\mathbf{z}^{(i+1)} - \tilde{\mathbf{x}}^{(i+1)}$
8 $i \leftarrow i + 1$	8 $i \leftarrow i + 1$
9 if $i \bmod r = 0$ then	9 if $i \bmod r = 0$ then
10 $k \leftarrow k + s$	10 $k \leftarrow k + s$
11 end	11 end
12 go to 2	12 go to 2
13 end	13 end
14 return $\tilde{\mathbf{x}} = D\tilde{\mathbf{z}}^{(i+1)}$	14 return $\tilde{\mathbf{x}} = D\tilde{\mathbf{z}}^{(i+1)}$

4 Implementácia v programe MATLAB

Kód bol vytvorený vo verzii MATLAB R2015a. Pre priebeh niektorých funkcií sa využíva toolbox LTFAT (*Large Time-Frequency Analysis Toolbox*), vytvorený pre prácu časovo-frekvenčnou analýzou a syntézou. Tento toolbox je možné získať na stránke <<https://ltfat.github.io/>>.

4.1 Hlavný súbor

Hlavný súbor programu je `projekt.m`. Na začiatku prebehne načítanie zvukového súboru. Súbor musí byť jednokanálový vo formáte `*.wav`, alebo v inom formáte podporovanom MATLABom. Pre načítanie zvuku sa používa funkcia `audioread`, výstup funkcie je vektor vzoriek signálu a vzorkovací kmitočet. Následne je signál normalizovaný.

V ďalšej časti je možné nastaviť vstupné parametre. Parameter `alg` slúži pre výber dekvantizačného algoritmu, `distrib` vyberie rozloženie kvantizačných hladín, `quantize` určuje, či sa podľa hodnoty `n_level` vyberie priamo počet kvantizačných hladín, alebo toto číslo bude určovať bitové rozlíšenie kvantizovaného signálu. Následne sa vyberajú parametre `frame`, `window` udáva veľkosť okna, `timeShift` posun okna, `red` redundanciu kmitočtových kanálov reprezentácie signálu, `channels` počet kanálov `frame` a `windowType` určuje typ okna. Následne je prevedená kvantizácia signálu (viď obr. 1.1) a konštrukcia `frame` s využitím LTFAT toolboxu podľa nastavených parametrov na začiatku súboru a prednastavených parametrov. `Frame` je uložený v premennej `f`.

Pre DR a CP algoritmy je nastavený typ `frame` `'dgtreal'`, ktorý označuje, že sa jedná o Gaborov `frame`, a typ okna Hannovo (`'hann'`). Bude sa pracovať s Parsevalovým tesným `framom`, čo je zabezpečené funkciou `frametight`. Pre rýchlejšie operácie využívajúce `frame` je použitá funkcia `frameaccel`.

SPADEQ algoritmy na rozdiel od predchádzajúcich algoritmov pre ℓ_1 minimalizáciu spracovávajú signál po blokoch, ktoré sú oknované. Pre každé okno sa použije DFT transformácia, kde operátor analýzy bude DFT a operátor syntézy IDFT. To sa dosiahne cez typ `frame` `'dft'`, alebo `'reddft'` pri redundancii koeficientov.

Po vytvorení `frame` je poškodený signál rekonštruovaný pomocou algoritmu vybratého na začiatku výberu vstupných parametrov. Vstupné premenné do algoritmov sú vektory pôvodného signálu `xd`, kvantizovaného signálu `xq`, `frame` `f`, prahov najbližších kvantizačných hladín `stepL` a `stepH` a dĺžka pôvodného signálu `xLength`. SPADEQ algoritmy využívajú navyše vstupné parametre `frame` pre vytvorenie okna pre segmentáciu signálu do blokov, ktoré sú detailnejšie popísané v časti 4.7.

Po priebehu algoritmu sa do príkazového okna vypíše výsledná hodnota ΔSDR , získaná zo vzťahu (5.1.1), a objektívnej funkcie a čas trvania priebehu algoritmu. Rekonštruovaný signál je možné prehrať a tiež je možné zapísať zreštaurovaný signál na pevný disk. Posledným krokom hlavného skriptu je možnosť zobrazíť si postupy objektívnej funkcie a ΔSDR po iteráciách algoritmu, a spektrogramy koeficientov pôvodného, kvantizovaného a reštaurovaného signálu.

4.2 Kvantizácia

Funkcia `quant` sa nachádza v súbore `quant.m`. Vstupmi tejto funkcie sú pôvodný signál `xd`, číslo `n_level` pre parameter `quantize` a rozloženie hladín `distrib`. Na začiatku funkcia zistí počet hladín na kvantizáciu a skontroluje, či počet vstupných parametrov je dostatočný. Ak nie, priradí predvolené hodnoty `n_level = 16` a lineárne rozloženie hladín. Potom je vytvorený vektor s kvantizačnými hladinami podľa nastaveného rozloženia. Vytvorí sa vektor prahov kvantizačných hladín `b`, označený `step` a následne prebehne kvantizácia vstupného signálu `xd`, uloženie poškodeného signálu do premennej `xq` a vytvorenie vektorov `bH` a `bL` ako premenné `stepH` a `stepL`. Výstupom tejto funkcie sú vektory `xq`, `stepH` a `stepL`.

4.3 Douglas–Rachfordov algoritmus

Nasledujú funkcie algoritmov pre dekvantizáciu. Prvým z nich je DR algoritmus, nachádzajúci sa v súbore `douglas.m`. Vstupmi tejto funkcie sú vektory pôvodného `xd` a kvantizovaného signálu `xq`, frame `f`, vektory najbližších kvantizačných prahov signálu `stepH` a `stepL` a dĺžka pôvodného signálu `xLength`.

Vo funkcii je najskôr určený počet opakovaní algoritmu `iter`, prahová hodnota mäkkého prahovania `gamma` a váhový parameter `lambda`. Nasleduje funkcia pre mäkké prahovanie `soft_thr` (3.7), vytvorenie reprezentácie signálu `cq` a inicializácia vektora `c` pre druhý krok algoritmu. Následne je vytvorená funkcia pre výpočet SDR, vyjadrený v rovnici (5.1), a logický operátor pre vykresľovanie grafov počas priebehu algoritmu `render`.

Ďalej sa vo funkcii nachádza samotný DR algoritmus, opísaný v časti 3.1.1. Projekcia na množinu prípustných riešení sa ukladá do vektora `x` a kroky algoritmu sa ukladajú do vektorov `c_tilde` a `c`. Výsledný reštaurovaný signál sa zapíše do vektora `xr`.

V poslednom kroku algoritmu sa vykresľujú grafy objektívnej funkcie a ΔSDR , objektívneho ukazovateľa popísaného v sekcii 5.1.1, a po každej iterácii algoritmu sa prekreslia s novozískanými hodnotami. Algoritmus sa opakuje pre zadaný počet

iterácií a po jeho dokončení sa výstupy uložia do hlavnej funkcie. Výstupy sú vektor zrekonštruovaného signálu \mathbf{x}_r , vektor priebehu objektívnej funkcie \mathbf{obj} a vektor priebehu ΔSDR , dSDR .

4.4 Chambolle–Pockov algoritmus

Podobne ako pre funkciu DR algoritmu opísanú v časti 4.3, vstupmi funkcie sú \mathbf{x}_d , \mathbf{x}_q , \mathbf{f} , stepH , stepL a axLength .

Na začiatku je nastavenie počtu iterácií algoritmu a váhovacích parametrov ζ , σ , θ a λ . Nasleduje zápis funkcie pre mäkké prahovanie `soft_thr`, ktorého zápis je uvedený vo vzťahu (3.7), vytvorenie reprezentácie signálu \mathbf{c}_q a inicializácia vektorov pre jednotlivé kroky algoritmu. Potom sa vytvorí funkcia pre výpočet SDR, opísaný v časti (5.1), a nastavenie operátora pre vykreslenie hodnôt objektívnej funkcie a ΔSDR počas priebehu algoritmu.

CP algoritmus, ktorý je detailnejšie popísaný v časti 3.1.2, je ďalším krokom tejto funkcie, pričom projekcia na množinu prípustných riešení sa ukladá do vektora \mathbf{x} a kroky algoritmu sa ukladajú do vektorov $\mathbf{x_tilde}$ a \mathbf{c} . Zrekonštruovaný signál sa zapíše do vektora \mathbf{x}_r .

Posledná časť funkcie obsahuje vykreslenie objektívnej funkcie a ΔSDR , ktoré sa po každej iterácii algoritmu prekreslia s nadobudnutými hodnotami. Po dokončení algoritmu sú výstupmi funkcie vektor zrekonštruovaného signálu \mathbf{x}_r , vektor priebehu objektívnej funkcie \mathbf{obj} a vektor priebehu ΔSDR , dSDR .

4.5 Projekcia na množinu prípustných riešení v časovej oblasti

Vstupmi funkcie `proj.m` sú vektory signálu \mathbf{x} a najbližších kvantizačných prahov stepH , stepL . Vektory `stepH` a `stepL` je najprv potrebné upraviť tak, aby ich dĺžka korešpondovala s vektorom \mathbf{x} .

Projekcia spracováva každú vzorku signálu. Logické operátory M_R , M_H a M_L pre danú vzorku sú určené tým, či má vzorka signálu vyššiu, resp. nižšiu hodnotu než najbližší vyšší (alebo nižší) kvantizačný prah, alebo sa nachádza medzi nimi. Vzorky signálu s hodnotou medzi kvantizačnými prahmi budú zohľadnené v M_R a ich hodnota ostane nezmenená. M_H je určené vzorkami, ktorých hodnota je nad najbližším vrchným kvantizačným prahom b_H a tieto vzorky nadobudnú hodnotu b_H . Pre M_L to budú vzorky s hodnotou pod najbližším spodným kvantizačným prahom b_L a na túto hodnotu sa aktualizujú.

Nové hodnoty signálu po projekcii sa uložia do vektora \mathbf{x} ako jediný výstup tejto funkcie.

4.6 Projekcia na množinu prípustných riešení v oblasti koeficientov

Vstupom funkcie `proj_c.m` sú vektory koeficientov \mathbf{c} a najbližších kvantizačných prahov `stepH`, `stepL`. Pred samotnou projekciou je potrebné vektor koeficientov \mathbf{c} transformovať na vektor signálu \mathbf{x}_p , ktorý spolu s `stepH` a `stepL` vstupujú do funkcie projekcie v časovej oblasti, opísanej v sekcii 4.5. Tá sa následne vykoná a jej výstupný vektor \mathbf{x} je súčasťou kroku (3.5) ako vektor `proj(Dc)`.

Výstupom funkcie bude vektor koeficientov `c_tilde`.

4.7 Segmentácia signálu pre SPADEQ algoritmy

Pred volaním SPADEQ algoritmov je potrebné signál rozdeliť do blokov, čo sa deje vo funkcii `spadeq_segment.m`. Vstupmi tejto funkcie sú vektor pôvodného \mathbf{x}_d a kvantizovaného signálu \mathbf{x}_q , frame \mathbf{f} , typ algoritmu `alg`, parametre framu `window`, `timeShift`, `windowType`, vektory najbližších kvantizačných prahov signálu `stepH`, `stepL` a dĺžka pôvodného signálu `xLength`.

Najskôr je vo funkcii niekoľko parametrov, ktorých hodnoty sa dajú meniť. Premenná `stepSize`, odpovedajúca premennej s v popisoch priebehov algoritmov v sekciiach 3.2.2 a 3.2.3, udáva hodnotu, o koľko sa zvýši počet koeficientov, ktoré sa budú brať do úvahy pri tvrdom prahovaní. Hodnota `stepRate`, odpovedajúca premennej r v popisoch algoritmov v častiach 3.2.2 a 3.2.3 určuje, po koľkých iteráciách sa zvýši počet koeficientov o hodnotu `stepSize`. `epsilon` označuje prah pre funkciu na prerušenie algoritmu a `iter` je maximálny počet opakovaní algoritmu.

Potom sa vykonáva úprava dĺžiek signálu a najbližších kvantizačných prahov na ich konci, aby posledné bloky boli rovnako veľké ako ostatné. Vytvorí sa tiež okno pre analýzu `gana` a syntézu `gsyn` a premenná nahrádzajúca posun okna `idxrange`. Následne sa inicializujú bloky signálu, najbližších kvantizačných prahov, zároveň aj matice ΔSDR a objektívnej funkcie. Tiež je možné zapnúť vykreslenie grafu ΔSDR .

Cyklus s algoritmom obsahuje najskôr počítadlo zrekonštruovaných blokov a vynásobenie signálu a najbližších kvantizačných hladín oknom pre analýzu. Následne je prevedený jeden zo SPADEQ algoritmov podľa nastavenia `alg`, jednotlivé algoritmy sú označené '`aspadeq`', '`sspadeq`' a '`sspadeq_new`'.

Po priebehu algoritmu sa bloky signálu skladajú naspäť do pôvodnej podoby s využitím metódy OLA. Výstupmi funkcie pre segmentáciu signálu sú vektor zrekonštruovaného signálu \mathbf{x}_r , priebeh ΔSDR po každej iterácii $d\text{SDR}$ a vektor objektívnej funkcie obj .

4.8 A–SPADEQ algoritmus

Do tejto funkcie vstupujú bloky pôvodného \mathbf{x}_d_block a poškodeného signálu \mathbf{x}_q_block , frame \mathbf{f} , hodnota zvýšenia počtu koeficientov pre tvrdé prahovanie stepSize a počet iterácií, po ktorých nastane zvýšenie počtu koeficientov stepRate , prah pre funkciu na prerušenie algoritmu epsilon , maximálny počet iterácií algoritmu iter a bloky vektorov najbližších kvantizačných prahov stepL_block , stepH_block .

Inicializujú sa vstupné vektory podľa prvého riadku algoritmu 5 a vytvorí sa funkcia pre výpočet SDR, opísaná v sekcii 5.1.1. Nasleduje algoritmus popísaný v časti 3.2.2 a ukladanie postupov objektívnej funkcie a ΔSDR do vektorov. Výstupom tejto funkcie je vektor blokov zrekonštruovaného signálu \mathbf{x}_r_block a vektory objektívnej funkcie obj a ΔSDR , $d\text{SDR}$.

4.9 S–SPADEQ a S–SPADEQ_{DR} algoritmy

Vstupy a priebehy týchto funkcií sú rovnaké ako ako u funkcie pre A–SPADEQ algoritmus 4.8. Inicializujú sa vstupné vektory podľa prvého riadku algoritmu 14 alebo 14 a vytvorí sa funkcia pre výpočet SDR vyjadrený v rovnici (5.1). Nasleduje jeden z algoritmov popísaných v časti 3.2.3 a ukladanie postupov objektívnej funkcie a ΔSDR do vektorov. Výstupy S–SPADEQ funkcií sú vektor blokov zrekonštruovaného signálu \mathbf{x}_r_block a vektory objektívnej funkcie obj a ΔSDR , $d\text{SDR}$.

4.10 Hard thresholding

Pomocná funkcia pre SPADEQ algoritmy využívajúce tvrdé prahovanie má ako vstupy vektor DFT koeficientov signálu \mathbf{a} a počet koeficientov, ktoré budú prahované k . Najprv sa vyberie polovica spektra koeficientov a DC koeficientu, následne sa koeficienty vytriedia podľa ich veľkosti a všetkým okrem k najväčším koeficientom sa prideli nula. Prebehne výpočet konjugátu vybraných koeficientov a takto upravené koeficienty sa uložia do vektora \mathbf{s} , ktorý je aj výstupom tejto funkcie.

5 Testovanie algoritmov, výsledky testovania

V tejto kapitole budú podrobnejšie opísané ukazovatele a metódy, podľa ktorých boli algoritmy pre rekonštrukciu signálu otestované, a predstavené výsledky testovania. V prvej časti kapitoly bude opis objektívnych ukazovateľov a subjektívneho posluchového testu. V ďalšej časti bude popísané hľadanie najlepšieho nastavenia parametrov algoritmov a v poslednej časti budú uvedené a popísané dosiahnuté výsledky.

Testovanie algoritmov prebiehalo na desiatich signáloch rôznych hudobných nástrojov, medzi nimi boli husle, klarinet, hoboje, harfa, zvonkohra, celesta, akordeón, gitara, klavír a orchester. Nahrávky boli jednokanálové, vo formáte .wav so vzorkovacím kmitočtom 44,1 kHz.

5.1 Testovanie podľa objektívnych ukazovateľov

5.1.1 SDR

Výsledná rekonštrukcia signálu sa bude porovnávať s pomocou ukazovateľa SDR – hodnota odstupe signálu od skreslenia. SDR pre dva signály \mathbf{a} a \mathbf{b} je možné vypočítať ako [10]

$$\text{SDR}(\mathbf{a}, \mathbf{b}) = 10 \log_{10} \frac{\|\mathbf{a}\|_2^2}{\|\mathbf{a} - \mathbf{b}\|_2^2}. \quad (5.1)$$

ΔSDR vyhodnocuje kvalitu rekonštrukcie, ako veľmi sa reštaurovaný signál \mathbf{z} podobá na pôvodný signál \mathbf{x} , a vypočíta sa ako rozdiel SDR hodnôt reštaurovaného a poškodeného signálu \mathbf{y} voči pôvodnému, $\Delta\text{SDR} = \text{SDR}(\mathbf{x}, \mathbf{z}) - \text{SDR}(\mathbf{x}, \mathbf{y})$.

5.1.2 PEMO-Q

Jednou z metód pre objektívne posúdenie a predikciu kvality signálu podľa subjektívneho vnemu je PEMO-Q (*Perceptual Model–Quality*) [21]. Na vyhodnotenie kvality skresleného signálu voči referenčnému signálu sa využíva sluchový model na výpočet *vnútorných reprezentácií* signálu, ktoré označujú informácie, o ktorých sa predpokladá, že sú na výstupe periférneho sluchového systému v zmysle neurálnej aktivity a slúžia ako vstup pre vyššie kognitívne úrovne. Vnútorné reprezentácie sú ďalej spracovávané a väčšinou porovnané mierou podobnosti. Výsledná miera v ideálnom prípade korešponduje s vnímanými odlišnosťami signálu, ktoré sú interpretované ako zhoršenie kvality zvuku testovaného signálu voči referenčnému, ktorý predstavuje optimálnu kvalitu zvuku.

MATLAB kódy s funkciami pre vyhodnotenie signálu pomocou PEMO-Q je možné stiahnuť z <https://www.hoertech.de/de/f-e-produkte/pemo-q.html>.

Základný postup tejto metódy je aplikovanie modelu sluchového spracovávanía na pár signálov tvorený testovacím a referenčným. Vnímaná podobnosť signálov bude zmeraná pomocou koeficientu korelácie výstupov modelu. Každá vnímaná zmena signálu voči referencii sa bude pokladať za zhoršenie kvality zvuku. Koeficient korelácie preto slúži ako objektívna miera zhoršenia kvality zvuku [21].

Pred spracovaním signálov sluchovým modelom je potrebné odstrániť ich možné vzájomné časové oneskorenie a rozdiel medzi úrovňami. Tieto odchýlky sú pri vnímaní irelevantné, no pri objektívnom hodnotení kvality môžu celkom výrazne ovplyvniť výsledok. Časové oneskorenie sa dá opraviť posunutím referenčného signálu o známu alebo odhadovanú hodnotu. Toto platí v prípade, že sa oneskorenie v čase nemení, v opačnom prípade je oneskorenie potrebné opravovať blok po bloku. Vyrovnanie úrovni signálu sa vykonáva dosiahnutím rovnakých RMS hodnôt. Tento postup bude fungovať, ak úroveň signálu v čase príliš nekolíše. Posledným krokom predspracovania spočíva v odstránení tichých častí signálu. Z referenčného aj testovacieho signálu sa vymažú časti, ktorých úrovne sú zjavne pod prahom počutia. Predpokladá sa, že tiché časti nijak neprispievajú k hodnoteniu zvukovej kvality poslucháčom. Kvôli možným dočasným maskovacím efektom nie sú pauzy odstránené úplne, ale iba skrátené na dobu 200 ms [21].

Pre simuláciu transformácie zvukového stimulu na nervovú aktivitu ľudského ucha je použitý kvantitatívny model spracovania zvukového signálu z výskumu popísaného v [22]. Tento model transformuje oba signály na korešpondujúce vnútorné reprezentácie zložené z trojdimenzionálnych vzorcov meniacich sa v čase, frekvencii a modulačnej frekvencii. Vstupný signál je rozložený na 35 kritických pásiem, zodpovedajúcich charakteristike pásmových priepustí bazilárnej membrány. Každé pásmo je potom spracované jednotlivo, najskôr korigovaním polvlny a filtrovaním dolnou priepustou. V tejto časti sa zachová obálka signálu pre kmitočty nad dolnou priepustou a amplitúda a fáza pre kmitočty pod dolnou priepustou. Vo fáze adaptácie sa berú do úvahy zmeny amplitúdy, rýchle zmeny sú zdôraznené a stacionárne časti signálu sú potlačené. V poslednom kroku spracovania je signál s obálkou analyzovaný bankou ôsmich filtrov, použitou pre detekciu amplitúdovej modulácie. Pre zníženie výpočtovej náročnosti v ďalšom spracovaní sú vnútorné reprezentácie podvzorkované pre každý modulačný kanál [21].

Simulované subjektívne hodnotenie vnímanej podobnosti signálov je nutné predpovedať. Na to sa využije koeficient lineárnej vzájomnej korelácie vnútorných reprezentácií. Predtým sa však vnútorná reprezentácia testovacieho signálu čiastočne

zahrnie do reprezentácie referenčného signálu:

$$\tilde{y}_{\text{tfm}} = \begin{cases} \frac{(y_{\text{tfm}} + x_{\text{tfm}})}{2}, & |y_{\text{tfm}}| < |x_{\text{tfm}}|, \\ y_{\text{tfm}}, & |y_{\text{tfm}}| \geq |x_{\text{tfm}}|, \end{cases} \quad (5.2)$$

kde y_{tfm} je reprezentácia poškodeného signálu, x_{tfm} reprezentácia referenčného signálu a indexy t , f a m odkazujú na čas, frekvenciu a modulačnú frekvenciu. Ak majú elementy testovaného signálu menšie absolútne hodnoty ako elementy referenčného, za dané hodnoty sa dosadí priemer hodnôt oboch elementov, čím sa rozdiely medzi elementmi zmenšia na polovicu. Tento postup sa riadi hypotézou, že chýbajúce komponenty poškodeného signálu sú menej rušivé ako prídavné komponenty.

Posledná operácia vzájomnej korelácie, ktorej výstup je percepčná miera kvality PSM (*perceptual quality measure*) sa vykonáva jednotlivo pre každý kanál, tie sú reprezentované maticami veľkosti $N \times M$. Koeficient lineárnej vzájomnej korelácie sa potom vypočíta ako [21]

$$r = \frac{\sum_{t,f=1}^{N,M} (x_{tf} - \bar{x})(y_{tf} - \bar{y})}{\sqrt{\sum_{t,f} (x_{tf} - \bar{x})^2 \sum_{t,f} (y_{tf} - \bar{y})^2}}, \quad (5.3)$$

kde N a M reprezentujú počet vzoriek a frekvenčných kanálov a \bar{x} , \bar{y} udávajú priemerné hodnoty. Následne sú korelačné koeficienty modulačných kanálov r_m váhované normalizovanými priemernými kvadrátmi hodnôt korešpondujúcich podmatíc a sčítané do finálnej miery kvality

$$\text{PSM} = \sum_m w_m r_m, \quad \text{kde } w_m = \frac{\sum_{t,f=1}^{N,M} (y_{\text{tfm}}^2)}{\sum_{t,f,m'=1}^{N,M,L} (y_{\text{tfm}'}^2)}, \quad (5.4)$$

kde L je počet modulačných kanálov. Váhovacie funkcie sú vypočítané z vnútorných reprezentácií testovacieho signálu, pričom sa ich hodnota zväčší pri vyššej aktivite v korešpondujúcom modulačnom kanáli. PSM môže nadobudnúť hodnoty v intervale $[-1, 1]$, kde 1 udáva totožnosť a čím menšia hodnota, tým väčšia je odchýlka voči referencii a tým viac je naznačená strata kvality signálu.

Druhotný ukazovateľ miery kvality PSM_t sa vypočíta postupnou vzájomnou koreláciou vnútorných reprezentácií s oknom dlhým 10 ms, následne váhovanou kľazovým priemerom vnútornej reprezentácie a piatym percentilom z tejto hodnoty. Z výsledku tohto procesu sa získa spojitost medzi okamžitou a celkovou vnímanou kvalitou zvuku. Aby bolo možné predpovedať objektívne hodnotenie podľa ODG stupnice (*Objective Difference Grade*) v tabuľke 5.1, hodnoty PSM_t sú do nej dosadené pomocou regresívnej funkcie [21].

Stupnica ODG je definovaná ako

$$\text{ODG} = \text{Grade}_{\text{test}} - \text{Grade}_{\text{ref}}, \quad (5.5)$$

Tab. 5.1: ODG a SDG stupnica hodnotenia kvality signálu

Hodnotenie	ODG, SDG	Zhoršenie
5,0	0,0	Nerozoznatelné
4,0	-1,0	Rozoznatelné, nie nepríjemné
3,0	-2,0	Trochu nepríjemné
2,0	-3,0	Nepříjemné
1,0	-4,0	Veľmi nepríjemné

kde $\text{Grade}_{\text{test}}$ je hodnotenie kvality testovaného a $\text{Grade}_{\text{ref}}$ hodnotenie kvality referenčného signálu.

5.1.3 PEAQ

Metóda percepčného ohodnotenia kvality zvuku (*Perceptual Evaluation of Audio Quality*) je založená na hodnotení odchýlky vnútornej reprezentácie medzi referenčným a testovaným zvukom. Výpočty prebiehajú v reálnom čase a je vhodná pre priebežné hodnotenie kvality rozhlasového vysielania [25].

Táto metóda je štandardizovaná v odporúčaní ITU–R BS.1387 [24]. Používanie oficiálnej verzie vyžaduje platenú licenciu, preto bude v tejto práci použitá bezplatná verzia nazvaná PQEvalAudio (PQEA), opísaná v [23] a dostupná na stiahnutie z <http://www-mmsp.ece.mcgill.ca/Documents/Software/Matlab/PEAQ/PQevalAudio-v1r0.tar.gz>.

Štandard BS.1387 má 2 verzie, základnú a rozšírenú. Základná verzia používa model ucha založený na FFT a rozšírená využíva okrem modelu založeného na FFT naviac aj model ucha s bankou filtrov. Výstupom algoritmu sú tzv. MOV (*model output variables*), z ktorých sa vypočíta výsledná známka ODG podľa tabuľky 5.1.

Spracovanie základného modelu pozostáva z FFT transformácie vstupného signálu, nasleduje modelovanie podľa vonkajšieho a stredného ucha, preskupenie do kritických pásiem, pridanie vnútorného šumu generovaného v uchu, frekvenčné a časové maskovanie a posledným krokom je výpočet výstupných premenných MOV. V základnom modeli je 11 MOV a sú to parametre hodnotiace zmeny v modulácii, hlasitosť deformácie, priemernú šírku pásma, frekvenciu počuteľných deformácií, odstup šumu od maskovacieho prahu, pravdepodobnosť detekcie chyby a harmonickú štruktúru chyby. Tieto premenné sú následne spracované kognitívnym modelom realizovaným neurónovou sieťou natrénovanou výsledkami subjektívnych testov [25].

PEAQ model vyžaduje, aby vstupný referenčný aj testovaný signál boli vzorkované na 48 kHz a bez vzájomného oneskorenia. Signály z použitej banky zvukov sú

vzorkované na 44,1 kHz a preto ich bolo spolu s rekonštruovanými signálmi potrebné prevzorkovať.

5.2 Subjektívne posluchové testy

Pre subjektívne testovanie sa použili rekonštrukcie signálov s parametrami, ktoré dosiahli najlepšie výsledky podľa Δ SDR.

Existujú dve kategórie subjektívnych metód merania kvality, metódy založené na relatívnej preferencii a na priradení číselnej hodnoty kvalite, tj. absolútne hodnotenie kvality.

Pri metódach relatívnej preferencie sú poslucháčom prehraté páry referenčného a testovaného úseku zvuku. Referenčný úsek je zvukový signál degradovaný pridaním šumu alebo filtráciou. Poslucháč zvolí z týchto úsekov zvuk, ktorý preferuje, tj. subjektívne kvalitnejší [25].

V testoch absolútneho hodnotenia kvality je poslucháčovi prehratý úsek zvuku, ktorého kvalitu má ohodnotiť na číselnej, typicky päťstupňovej stupnici. Nie je tu použitý žiadny referenčný signál. Najčastejšie sa používa metóda absolútneho ohodnotenia *ACR (Absolute Category Rating)*, pre testovanie kvalitných signálov je vhodnejšia metóda ohodnotenia degradácie *DCR (Degradation Category Rating)* alebo metóda ohodnotenia rozdielov *CCR (Comparison Category Rating)* [25].

5.2.1 MUSHRA

V tejto práci bude subjektívne hodnotenie kvality rekonštrukcie signálov prevedené pomocou metódy *MUSHRA (Multi Stimulus test with Hidden Reference and Anchor)*, viac-stimulový test so skrytou referenciou a kotvou. Táto metodika sa zameriava najmä na hodnotenie kvality pri prenose zvuku prostredníctvom internetu alebo satelitného vysielania, teda v systémoch, kde sa očakáva určité zníženie kvality zvuku a poslucháč to môže odhaliť. Poslucháč dostane niekoľko nahrávok spolu so známou referenciou. Medzi nahrávkami je aj tzv. kotva, najviac zhoršený úsek zvuku zo všetkých testovaných zvukov. Kotva slúži na lepšie vymedzenie hodnotiacej stupnice, ktorá sa bude sústrediť medzi ňou a referenciou [25].

Jedna skúška testu sa môže vykonať maximálne s pätnástimi stimulmi, kde jeden stimul je známa referencia, druhý je náhodne skrytá referencia, tretí je náhodne skrytá kotva a ostatné sú testované signály. Medzi najviac používané kotvy patrí referencia filtrovaná dolnou priepustou s medzným kmitočtom 3,5 kHz. Preto má kotva najhoršie posluchové vlastnosti a poslucháč ju ohodnotí najnižšou známkom. Ak je kotva rozpoznaná a ostatné podnety poslucháč porovnáva aj s referenciou, získa sa jemnejšie hodnotenie testu. Žiadny stimul nesmie presiahnuť dĺžku dvadsať

Tab. 5.2: Stupnica hodnotenia kvality signálu podľa metódy MUSHRA

Hodnotenie	Kvalita
100 – 80	Vynikajúca
80 – 60	Dobrá
60 – 40	Dostačujúca
40 – 20	Zlá
20 – 0	Neprijateľná

sekúnd [25]. Hodnotiaca stupnica je plynulá s piatimi definovanými bodmi, v praxi sa používajú celé čísla od 0 do 100.

Získané hodnoty sa vzťahujú len na testovanú vzorku poslucháčov, táto vzorka sa označuje ako výber. Cieľom vedúceho testu je z výberu dosiahnuť záver pre celú populáciu. Preto sa bude pracovať s intervalom spoľahlivosti, vnútri ktorého s určitou vysokou pravdepodobnosťou leží hodnota populačného priemeru. Najskôr sa vypočíta priemerná hodnota každého stimulu (výberový priemer)

$$u_{jk} = \frac{1}{N} \sum_{i=1}^N u_{ijk}, \quad (5.6)$$

kde u_{jk} je priemerná hodnota stimulu, N je počet hodnotiacich poslucháčov a u_{ijk} je absolútne hodnotenie stimulov jednotlivých subjektov. Podobne sa dá vypočítať priemerná hodnota u jednotlivých testov j a testovacích podmienok k .

Studentovo rozdelenie je rozdelenie pravdepodobnosti, ktoré je pri väčšom výbere totožné s normálnym rozdelením. Premenná S_{jk} je výberová smerodajná odchýlka a $t_{0,05}$ označuje kritickú hodnotu Studentovho rozdelenia t pri hladine významnosti $\alpha = 0,05$ [25]

$$S_{jk} = \sqrt{\sum_{i=1}^N \frac{(u_{jk} - u_{ijk})^2}{\sqrt{N-1}}}. \quad (5.7)$$

Výsledky výberového priemeru je nutné prezentovať 95% intervalom spoľahlivosti s krajnými bodmi $[u_{jk} - \delta_{jk}, u_{jk} + \delta_{jk}]$, kde δ_{jk} je polovica šírky intervalu spoľahlivosti

$$\delta_{jk} = t_{0,05} \frac{S_{jk}}{\sqrt{N}}. \quad (5.8)$$

Hodnota $t_{0,05}$ sa odčíta zo štatistických tabuliek. Týmto sa získa 95% istota, že hodnota populačného priemeru leží vo vnútri vypočítaného intervalu spoľahlivosti.

Pre testovanie bola použitá aplikácia pre vyhodnocovanie kvality reštaurácie zvukových signálov popísaná v [26]. V tomto programe je možnosť výberu viacero

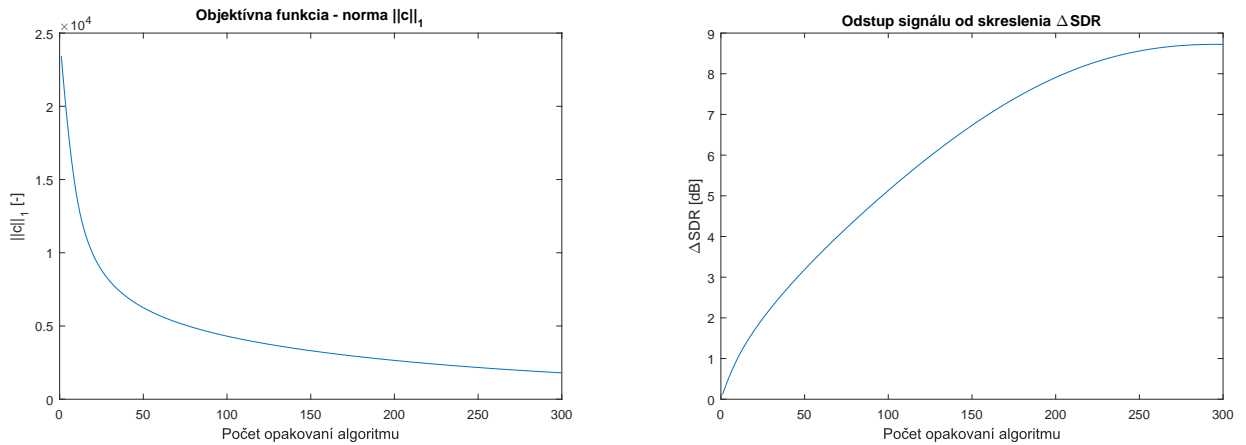
rôznych metód, ako sú ABX, MOS alebo MUSHRA. Po výbere testu sa do programu dajú načítať zvukové súbory, ktoré budú testované v skúškach (stranách) testu. Zvukové súbory budú náhodne usporiadané, okrem referencie, ktorá bude v poradí vždy ako prvý stimul. Ohodnotiť jednotlivé stimuly sa dá posuvníkmi, ktoré určujú ohodnotenie jednotlivých vzoriek a vedľa neho je tlačidlo pre prehratie stimulu. Uložené výsledky sa vypíšu do .csv súboru.

Testovanie prebiehalo pre tri signály hudobných nástrojov, konkrétne huslí, klarinetu a orchestra. Každý signál bol otestovaný pre 3 kvantizačné hladiny vybraté tak, aby bolo možné výsledky interpretovať pre rôznu mieru poškodenia rozličných signálov. V každej skúške testu bolo hodnotených 7 podnetov – referenčný nepoškodený signál, kvantizovaný signál na daný počet kvantizačných hladín (kotva) a päť zreštaurovaných signálov DR, CP a SPADEQ algoritmami. Zúčastnených poslucháčov bolo 9, všetci boli otestovaní v rovnakých podmienkach z hľadiska posluchového prostredia a použitých zariadení. Test bol prevedený na notebooku Lenovo Y50-70, externej zvukovej karte Focusrite Scarlett Solo 2nd Gen a slúchadlách Beyerdynamic DT 770 Pro. Dĺžka testovania sa u jednotlivých poslucháčov pohybovala od 15 do 50 minút.

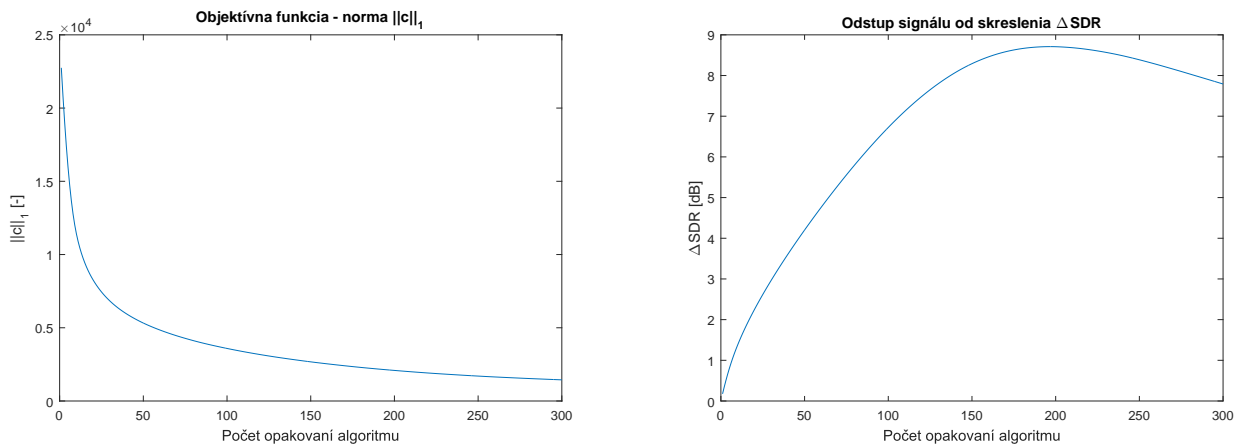
5.3 Nastavovanie parametrov algoritmov

Pri testovaní boli postupne empiricky skúšané rôzne nastavenia parametrov – počet kvantizačných hladín, veľkosť okna, posun framu, počet kmitočtových kanálov a počet iterácií. U DR algoritmu navyše aj hodnota prahu mäkkého prahovania, pri CP algoritme váhový parameter ζ a u SPADEQ algoritmov ϵ – prah pre funkciu na prerušenie algoritmu. Parametre boli nastavované tak, aby pre daný počet iterácií algoritmu boli hodnoty ukazovateľa ΔSDR najvyššie, aj keď minimalizácia normy $\|\mathbf{c}\|_1$ stále prebiehala. V tejto časti bol pre väčšinu grafických znázornení z testovania vybraný signál huslí, rekonštruovaný DR algoritmom, ak nie je uvedené inak. Počet kmitočtových kanálov framu bol nastavený na dvojnásobok hodnoty veľkosti okna.

Typický priebeh hodnôt ΔSDR spolu s normou $\|\mathbf{c}\|_1$ je možné vidieť na obrázku 5.1. Na ďalšom obrázku 5.2 sa dá všimnúť, že po približne 200. opakovaní algoritmu začína ΔSDR s ďalšími iteráciami klesať. Jedná sa o testovanie toho istého signálu so všetkými vstupnými parametrami nastavenými rovnako, až na prah mäkkého prahovania. Hodnota prahu mäkkého prahovania bola zvolená príliš vysoká pre daný počet iterácií, aj keď norma $\|\mathbf{c}\|_1$ má stále klesajúcu tendenciu o hodnoty rádovo v jednotkách.



Obr. 5.1: Priebeh hodnôt normy $\|c\|_1$ a ΔSDR pri správne nastavenom prahu mäkkého prahovania

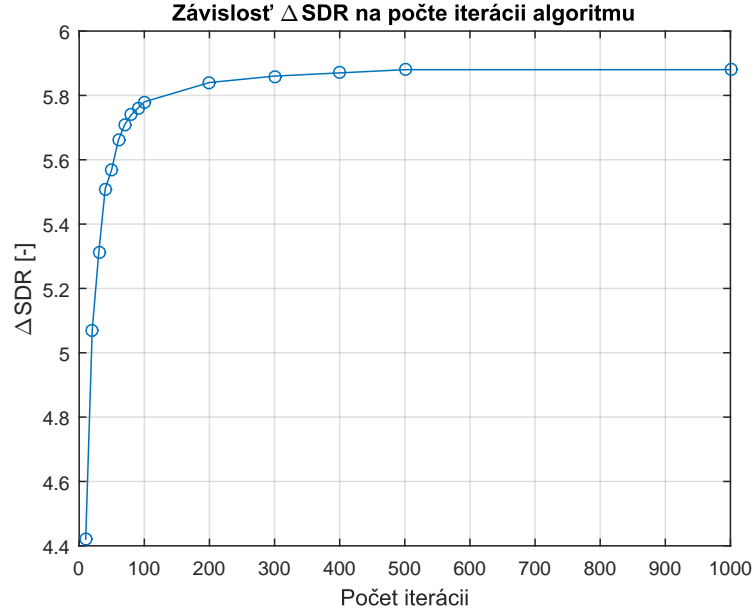


Obr. 5.2: Priebeh hodnôt normy $\|c\|_1$ a ΔSDR pri príliš veľkom prahu mäkkého prahovania

5.3.1 Nastavenie počtu iterácií

Najskôr sa niektoré parametre nastavili na určité hodnoty a potom z priebehu objektívnej funkcie a ΔSDR boli upravované tak, aby bol dosiahnutý čo najlepší výsledok hodnôt ΔSDR . Pre začiatok boli zvolené parametre počet kvantizačných hladín 8, veľkosť okna 8192 a posun okna 0,25-násobok veľkosti okna. Parameter, ktorý sa menil, bol počet iterácií. Cieľom bolo zistiť, koľko iterácií algoritmu je pre rekonštrukciu signálu potrebných, aby bol výsledok čo najpriaznivejší pri čo najnižšej výpočtovej náročnosti. Postupne boli nastavené hodnoty: 10 – 100 iterácií so zvyšovaním o 10 iterácií, 100 – 500 so zvyšovaním o 100 iterácií, a 1000.

Na obrázku 5.3 je možné vidieť, že pri 500 iteráciách je výsledná hodnota ΔSDR rovnaká ako po 1000 iteráciách, pre 400 iterácií je nižšia o 0,01 dB a pri 300 opa-



Obr. 5.3: Závislosť ΔSDR na počte opakovaní algoritmu pre signál huslí

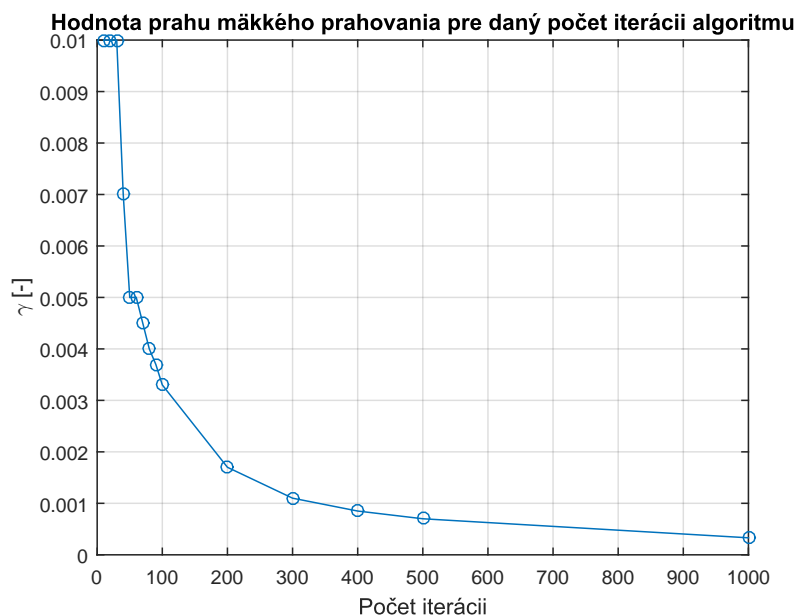
kovaniach algoritmu o 0,02 dB. Pri nižšom počte iterácií bol rozdiel oproti 1000 iteráciám väčší. Na základe toho bol pre ďalšie testovanie vybraný počet iterácií 500, ako najlepšia voľba z hľadiska kvality rekonštrukcie voči náročnosti na výpočet.

5.3.2 Nastavenie parametrov γ a ζ

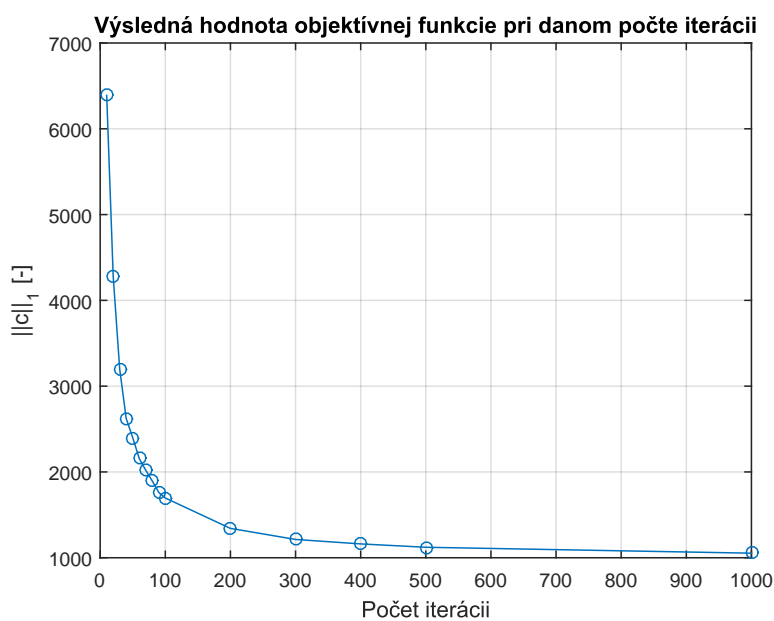
Z rovnakého testovania ako v sekcii 5.3.1 boli vybrané hodnoty prahu mäkkého prahovania podľa dosiahnutého ΔSDR , zobrazené na obr. 5.4 a výsledná hodnota normy $\|\mathbf{c}\|_1$ pre daný počet iterácií. Nastavenie hodnoty prahu má tendenciu byť nepriamo úmerné k počtu iterácií algoritmu, napr. pre 1000 iterácií bola nastavená hodnota prahu 0,00033, pričom pre 100 iterácií to bolo presne 10-krát väčšie číslo. Pre iné signály boli hodnoty prahu nastavené na iné hodnoty, pričom tendencia nepriamej úmery hodnoty prahu k počtu iterácií platila pri všetkých testovaných signáloch.

Graf výslednej hodnoty objektívnej funkcie 5.5 ukazuje, že s väčším počtom iterácií sa hľadané riešenie stáva čím viac redším. Od určitého počtu iterácií ešte klesá objektívna funkcia, no hodnoty ΔSDR sa menia len minimálne, ako je znázornené na obr. 5.3.

Hodnoty parametra ζ pre CP algoritmus mali podobnú tendenciu priebehu ako γ v grafe 5.4. Tieto parametre pre každý signál a počet kvantizačných hladín sú uvedené v tabuľkách v elektronickej prílohe.



Obr. 5.4: Zistené hodnoty parametra γ pre rôzny počet iterácií algoritmu

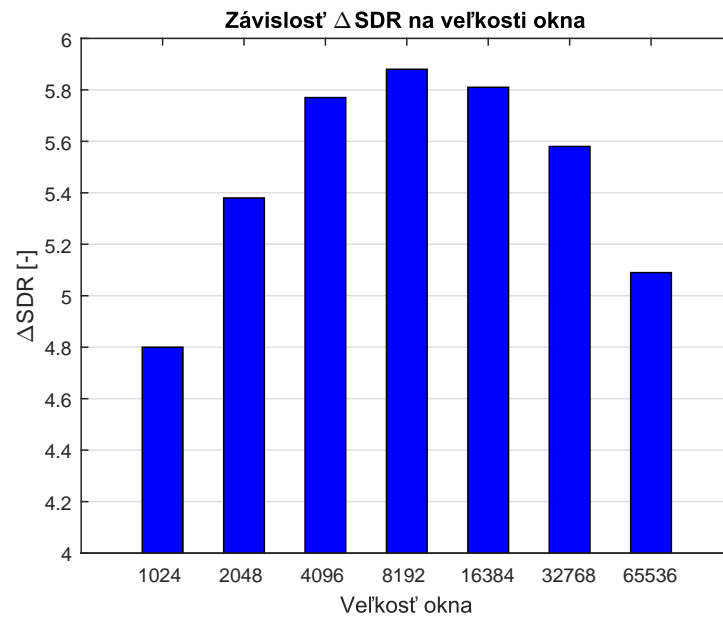


Obr. 5.5: Hodnoty $\|c\|_1$ pre daný počet iterácií

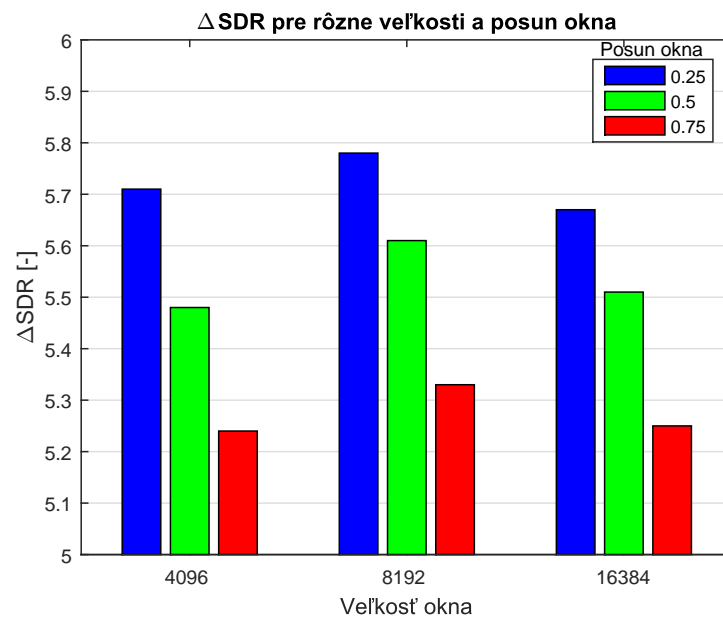
5.3.3 Nastavenie veľkosti a posunu okna

Ďalšie testované parametre boli veľkosť a posun okna. Najskôr bol testovaný vplyv veľkosti okna na rekonštrukciu. Nastavené parametre boli: počet kvantizačných hladín 8, počet iterácií 500 a posun okna 0,25-násobok veľkosti okna. Veľkosť okna bola testovaná na hodnoty 1024 – 65536, vždy so zvýšením o dvojnásobok predošlej

hodnoty.



Obr. 5.6: Závislosť Δ SDR na veľkosti okna pre signál huslí pre 8 hladín



Obr. 5.7: Výsledky testovania pre rôzne hodnoty veľkosti a posunu okna

Z grafu 5.6 vyplýva, že rekonštrukcia prebehla v tomto prípade najlepšie pre veľkosť okna 8192. Veľkosť okna bola pre každý signál testovaná pre 3 rôzne hodnoty počtu kvantizačných hladín a vo väčšine prípadov dosiahla rekonštrukcia najvyššiu hodnotu Δ SDR pre jednu veľkosť okna, preto boli zvyšné rekonštrukcie z ostatných

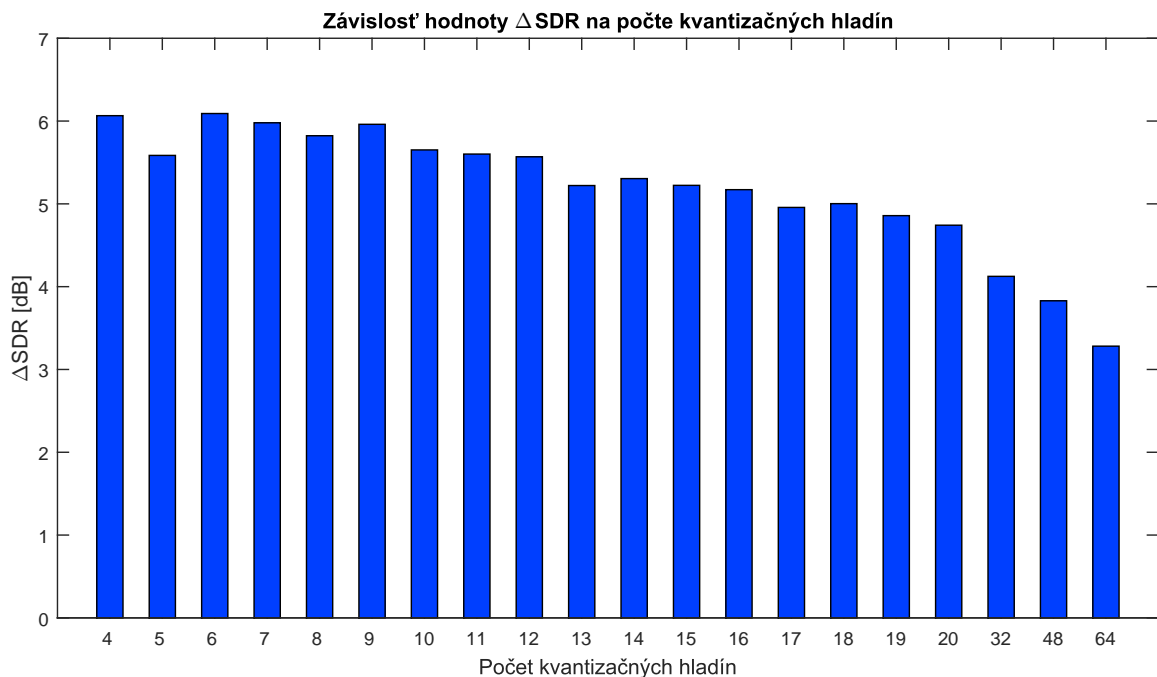
počtov hladín prevedené s touto veľkosťou okna. Pre rôzne signály a algoritmy boli najvýhodnejšie veľkosti okna 8192, 16384, 32768 alebo 65536 vzoriek.

Pre testovanie posunu okna boli vybrané hodnoty 0,25-, 0,5- a 0,75-násobok veľkosti okna. Otestované boli veľkosti okna 4096, 8192 a 16384. Z grafu 5.7 vyplýva, že najlepší výsledok je dosiahnutý s posunom 0,25-násobku veľkosti okna 8192. Δ SDR bola pre všetky veľkosti okna vyššia o 0,16 – 0,23 dB ako u 0,5-násobného posunu, a o ďalších 0,24 – 0,27 dB oproti 0,75-násobku veľkosti okna.

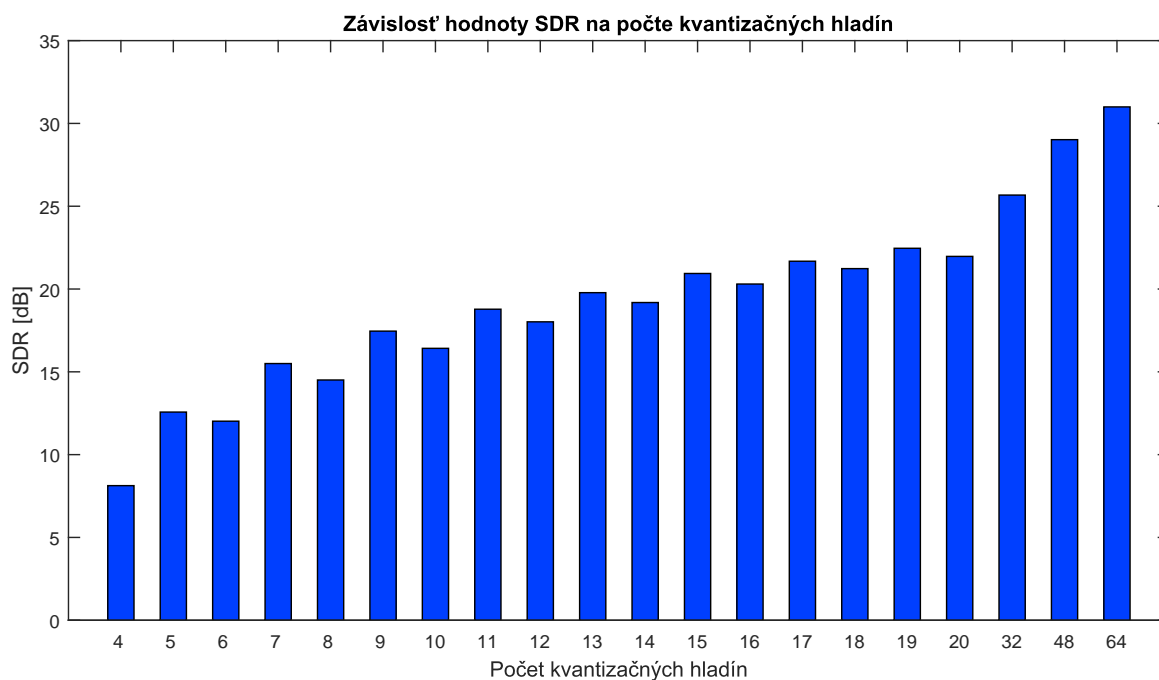
S rôznymi hodnotami posunu okna sa mení aj výpočtovú náročnosť. Kým pri posune okna s hodnotou 0,75-násobku veľkosti okna trval priebeh algoritmu pre 500 iterácií priemerne 125 – 167 s, pri 0,5-násobku to bolo 183 – 217 s a pri 0,25-násobku 350 – 425 s.

5.3.4 Nastavenie počtu kvantizačných hladín

Nasledujúce testovanie bolo zamerané na zistenie počtu kvantizačných hladín pre poškodenie signálu tak, aby po rekonštrukcii bola hodnota odstupu signálu od skreslenia čo najvyššia. Testovaný počet hladín bol 4-20 s pridávaním po jednej hladine, a 32, 48 a 64. Vstupné parametre boli počet iterácií 500 a posun okna 0,25-násobok veľkosti okna pre hodnotu 8192.



Obr. 5.8: Výsledky rekonštrukcie signálu huslí DR algoritmom pre rôzny počet kvantizačných hladín



Obr. 5.9: Hodnoty SDR po rekonštrukcii signálu huslí DR algoritmom pre rôznyi počet kvantizačných hladín

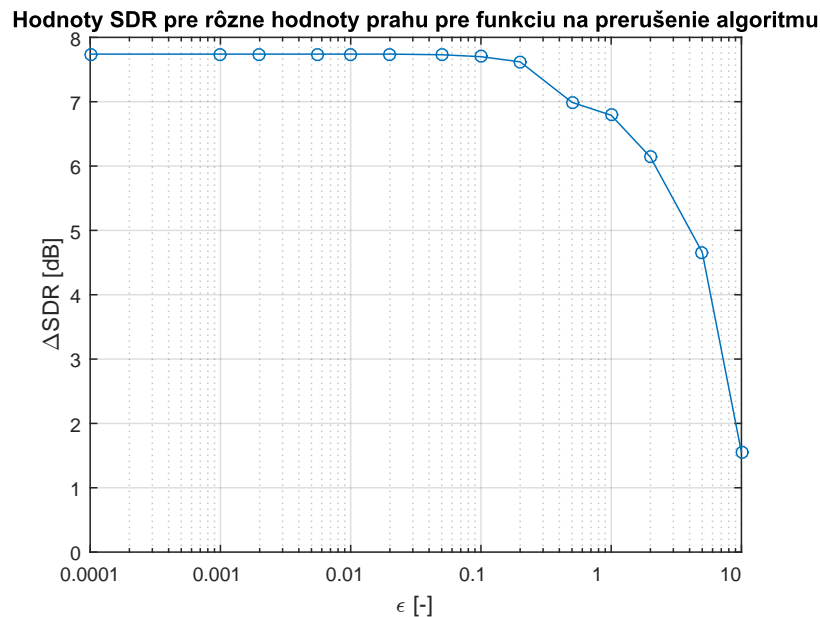
Z obrázka 5.8 vyplýva, že najúspešnejšia rekonštrukcia podľa hodnoty ΔSDR vychádza pre signál huslí rekonštruovaný DR algoritmom zo 6 hladín, hodnota ΔSDR nadobudla v tomto prípade hodnotu 6,09 dB. Najmenšiu hodnotu ΔSDR dosiahla rekonštrukcia signálu kvantizovaného na 64 hladín. Hodnoty SDR po rekonštrukcii signálu huslí DR algoritmom je možné vidieť na obr. 5.9. Pri zvyšovaní počtu kvantizačných hladín poškodeného signálu sa zvyšuje aj dosiahnutá hodnota SDR rekonštruovaného signálu, z čoho vyplýva, že rekonštrukcia bola úspešná pre všetky testované hodnoty kvantizačných hladín.

5.3.5 Nastavenie parametrov pre SPADEQ algoritmy

Pre SPADEQ algoritmy je potrebné nastaviť parameter ϵ , prah pre funkciu na prerušenie algoritmu tak, aby výsledná hodnota ΔSDR rekonštrukcie bola čo najvyššia pri čo najnižšej výpočetnej náročnosti algoritmu. Algoritmus bude iterovať dovtedy, kým norma rozdielu medzi signálom po projekcii na množinu prípustných riešení a signálom pred krokom projekcie nedosiahne hodnotu nižšiu ako ϵ . Čím nižšia je hodnota tohto parametra, tým bude signál zrekonštruovaný lepšie.

Prednastavené parametre boli veľkosť okna 8192 vzoriek, posun okna 0,25-násobok veľkosti okna, počet kvantizačných hladín 8, parametre s a r mali hodnotu 1 a počet iterácií bol nastavený podľa maximálneho množstva koeficientov

vstupujúcich do funkcie tvrdého prahovania.



Obr. 5.10: Hodnoty ΔSDR zrekonštruovaného signálu huslí algoritmom S-SPADEQ_{DR} v závislosti na parametri ϵ

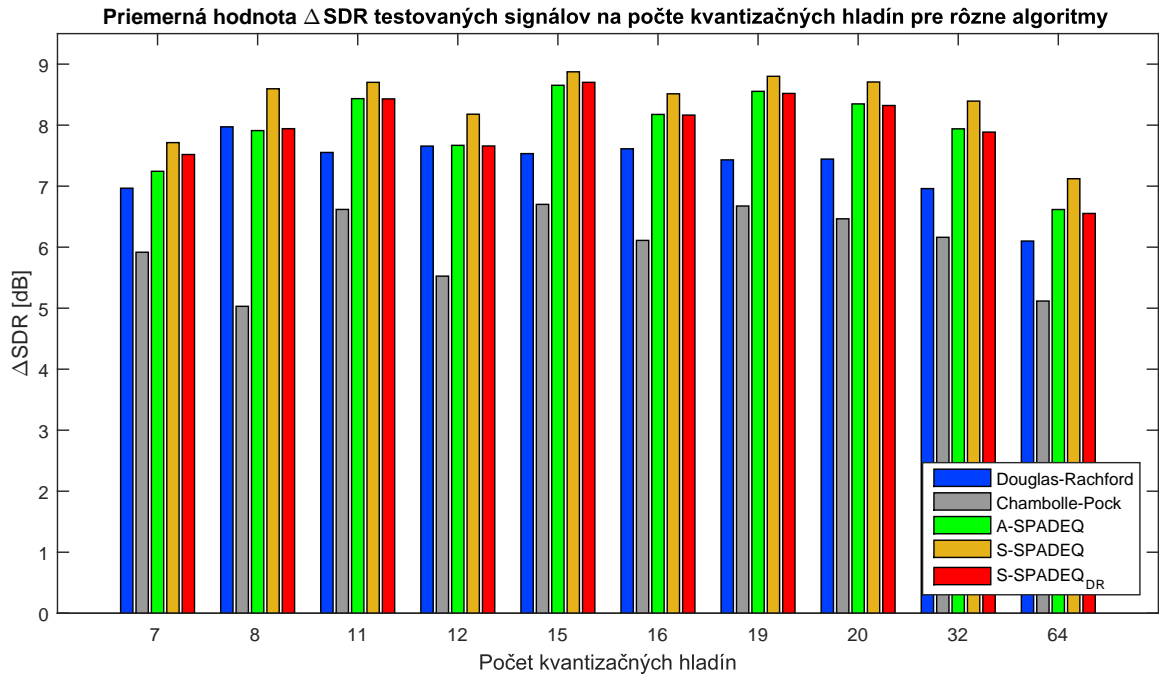
Výsledné hodnoty ΔSDR podľa hodnoty ϵ sú uvedené na obr. violinspadeq. Najvyššiu hodnotu ΔSDR dosahuje algoritmus pre $\epsilon < 0,02$, a to 7,47 dB. Pri $\epsilon = 0,05$ to bolo o 0,06 dB menej, pri vyšších hodnotách prahu klesala hodnota ΔSDR ešte viac. Pre ďalšie testovanie bude hodnota ϵ nastavená na 0,02, pre najlepšiu pomer dosiahnutej rekonštrukcie a vypočetnej náročnosti.

5.4 Výsledky testovania

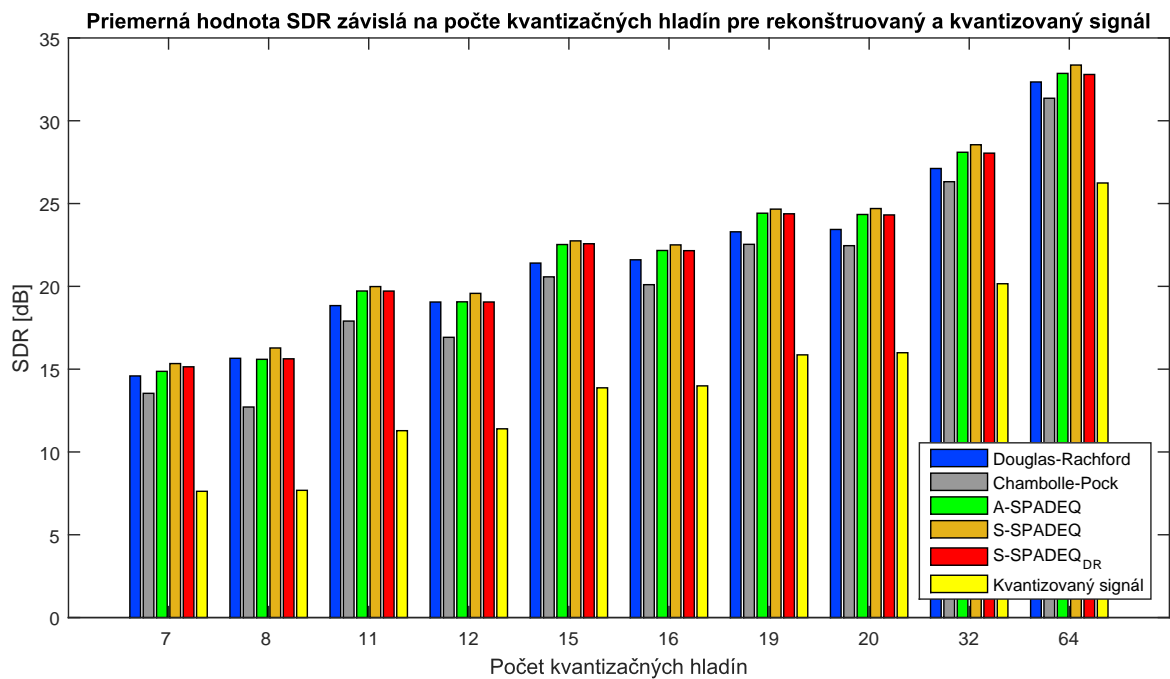
Táto sekcia obsahuje výsledky testovania pre všetky testované algoritmy a signály. Výsledky budú spriemerované zo všetkých signálov pre daný výber počtu kvantizačných hladín. V niektorých grafoch budú k hodnotám, ktoré dosiahli zrekonštruované signály pridané na porovnanie aj hodnoty kvantizovaných alebo pôvodných nepoškodených signálov. Podrobné grafické znázornenia budú uvedené v elektronickej prílohe.

5.4.1 Výsledky podľa hodnôt ΔSDR a SDR

Pre všetky signály bol posun okna nastavený na 0,25-násobok hodnoty veľkosti okna, počet iterácií 500 pre DR a CP, a také hodnoty parametrov γ a ζ , aby bola dosiahnutá hodnota ΔSDR čo najvyššia, pričom sa tieto parametre menili pre každú



Obr. 5.11: Priemerná hodnota Δ SDR rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín



Obr. 5.12: Priemerná hodnota SDR rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín

zmenu počtu hladín, na ktoré bol signál kvantizovaný. Pre SPADEQ algoritmy bola hodnota prahu pre funkciu na prerušenie algoritmu nastavená na $\epsilon = 0,02$ a algoritmy iterovali dovtedy, kým nebola nedosiahnutá táto hodnota, pričom potom sa algoritmus prerušil; v opačnom prípade pokračoval, kým do funkcie tvrdého prahovania nevstúpil maximálny možný počet koeficientov. Pre rôzne signály sa u DR, CP a SPADEQ algoritmov menili aj veľkosti okna, pričom boli zisťované rovnakým postupom, akým sa dosiahli hodnoty zobrazené na obr. 5.6. Najvýhodnejšie zistené veľkosti okna pre každý algoritmus a signál sú uvedené v tabuľke v elektronickej prílohe.

Na obr. 5.11 je vyobrazená priemerná hodnota ΔSDR rekonštrukcie zo všetkých testovaných signálov pre daný počet kvantizačných hladín. Najlepší algoritmus je z hľadiska hodnôt ΔSDR S-SPADEQ. Pri malom počte kvantizačných hladín sa k výsledkom SPADEQ algoritmov približuje DR, no od 15 hladín dosahujú SPADEQ algoritmy najlepšie hodnoty. Priemerné hodnoty SDR pre všetky testované signály podľa kvantizačných hladín sú znázornené na obr. 5.12.

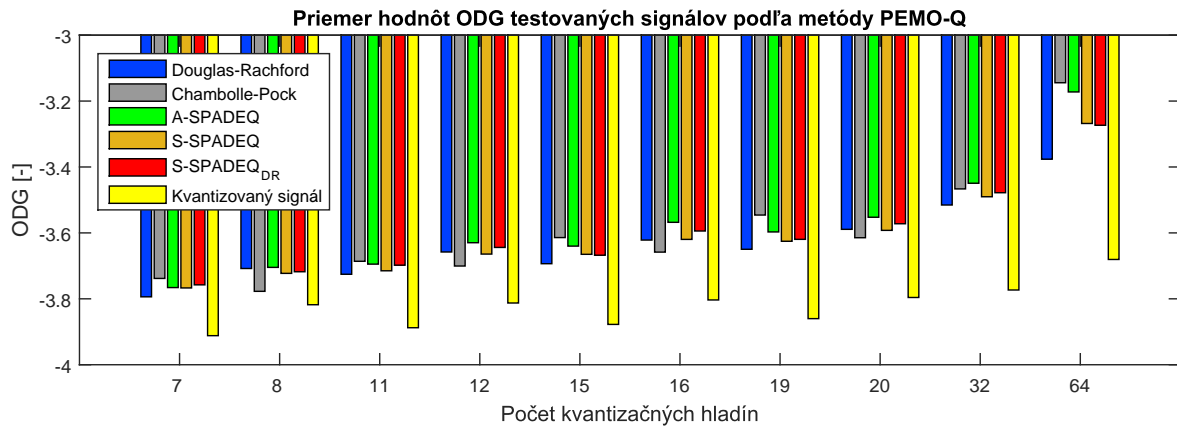
Podľa dosiahnutých priemerných hodnôt ΔSDR sa dá vyvodiť, že najkvalitnejšiu rekonštrukciu by mali vykonávať SPADEQ algoritmy. Spomedzi SPADEQ algoritmov nadobudla najvyššie hodnoty ΔSDR S-SPADEQ verzia.

5.4.2 Výsledky podľa metódy PEMO-Q

Vyhodnotenie rekonštrukcie pre signál huslí podľa ukazovateľa ODG získaného metódou PEMO-Q je zobrazené na obr. ???. Pre rôzny počet kvantizačných hladín sa ODG pohybuje medzi -4 a -3,5, čo značí, že algoritmus vyhodnotil subjektívne zhoršenie kvality všetkých rekonštruovaných signálov ako „veľmi nepríjemné“ oproti referenčnému nepoškodenému signálu. Graf tiež zobrazuje hodnoty ODG kvantizovaného signálu ako horšie než rekonštruovaných signálov, čo značí, že rekonštrukcia signálu bola úspešná.

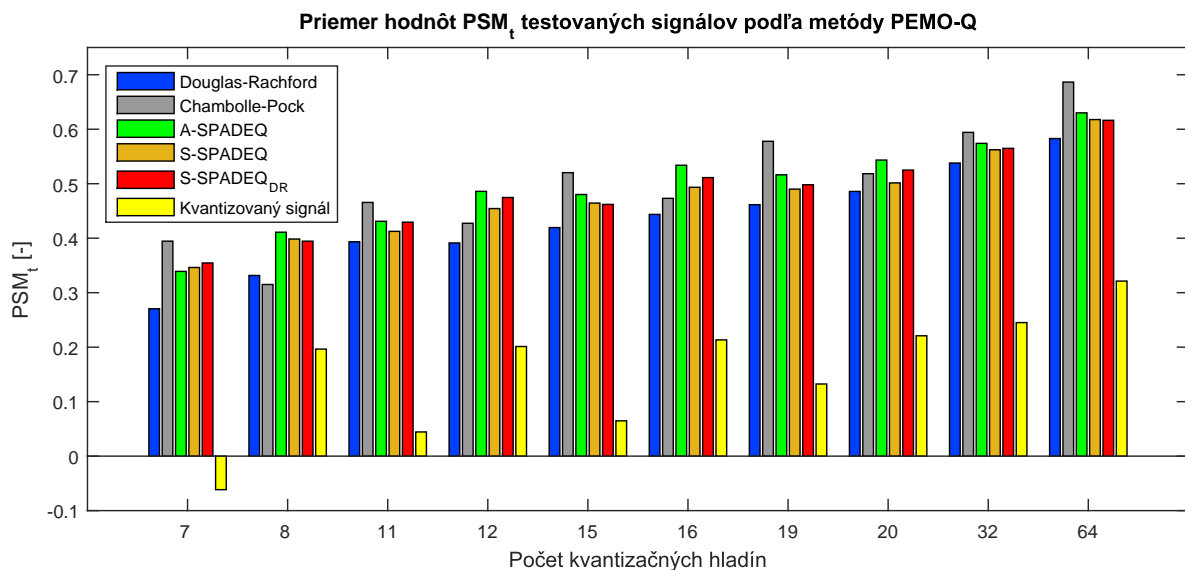
Priemerné hodnoty ODG pre rekonštrukcie zo všetkých testovaných signálov sú uvedené na obr. 5.13. Pre rôzny počet kvantizačných hladín sa ODG pohybuje medzi -4 a -3, čo značí, že algoritmus vyhodnotil subjektívne zhoršenie kvality väčšiny rekonštruovaných signálov ako „veľmi nepríjemné“ oproti referenčnému nepoškodenému signálu, jedine pre 64 hladín je priemerné zhoršenie kvality určené o stupeň lepšie ako „nepríjemné“. Graf tiež zobrazuje hodnoty ODG kvantizovaného signálu ako horšie než rekonštruovaných signálov, čo značí, že rekonštrukcia signálu bola úspešná.

Druhé vyhodnotenie podľa metódy PEMO-Q bude zobrazenie ukazovateľa PSM_t . Priemerné hodnoty PSM_t zo všetkých signálov sú zobrazené na obr. 5.14. Kvantizovaný signál dosahuje zápornú hodnotu PSM_t pre 7 kvantizačných hladín, čo značí



Obr. 5.13: Priemerná hodnota ODG(PEMO-Q) rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín

veľkú odchýlku od pôvodného nepoškodeného signálu. Ostatné hodnoty sú kladné, a najvyššiu hodnotu dosahuje pre 64 hladín. Takisto aj rekonštruované signály dosahujú najlepších hodnôt pre 64 kvantizačných hladín a pohybujú sa medzi 0,58 – 0,69.

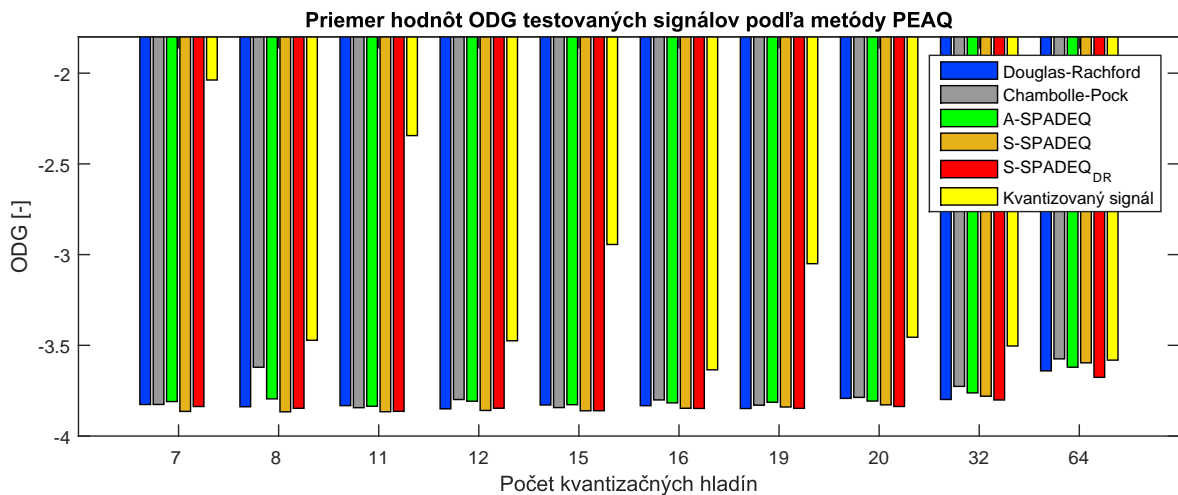


Obr. 5.14: Priemerná hodnota PSM_t rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín

Podľa ukazovateľov ODG a PSM_t získaných metódou PEMO-Q sa dá vyvodit záver, že najlepšie subjektívne vnímané reštaurované signály by mali byť rekonštruované pomocou CP alebo A-SPADEQ algoritmov.

5.4.3 Výsledky podľa metódy PEAQ

Vyhodnotenie reštaurácie signálov podľa metódy PEAQ je uvedené na obr. 5.15. ODG pre poškodený signál dosahuje vyšších hodnôt ako pre reštaurovaný signál, čo nekorešponduje s výsledkami uvedenými v sekcii 5.4.2. Žiadny priemer hodnôt ODG rekonštruovaných signálov nedosahuje lepšie hodnoty ako -3,5, čo označuje „veľmi nepríjemné“ zhoršenie kvality signálu. Hodnotenie metódou PEAQ bude zrejme chybné, a môže to byť skutočnosťou, že táto metóda nebola vytvorená pre hodnotenie signálu poškodeného kvantizáciou, prípadne jej odlišnou metodikou vyhodnocovania ODG oproti metóde PEMO-Q.

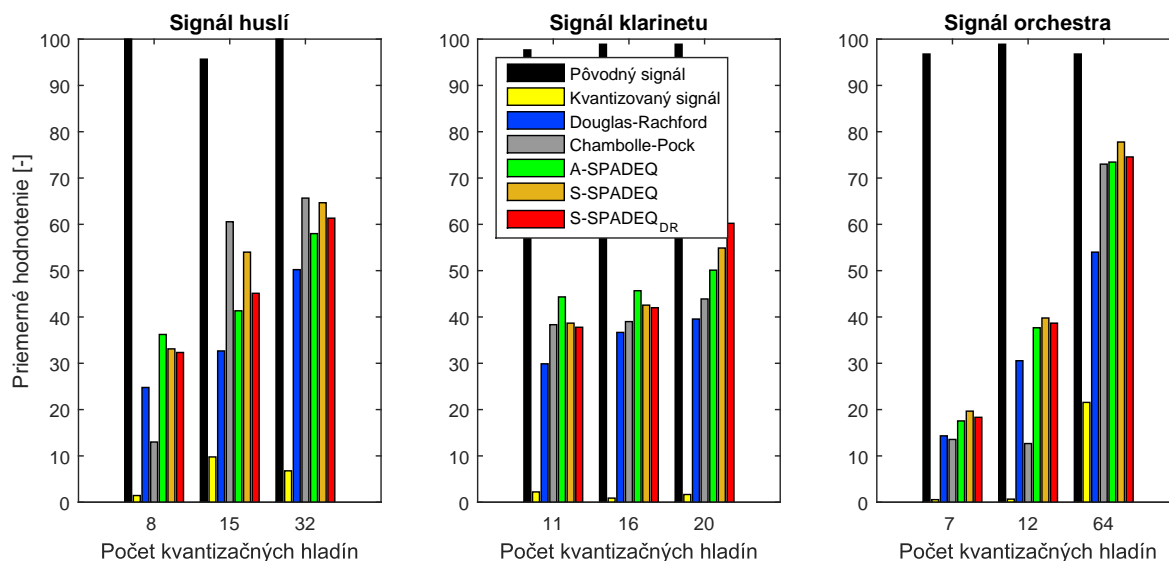


Obr. 5.15: Priemerná hodnota SDR rekonštrukcie testovaných signálov podľa použitého algoritmu pre rôzny počet kvantizačných hladín

5.4.4 Výsledky podľa testu MUSHRA

Výsledky subjektívneho testovania sú uvedené v grafe 5.16. Poslucháči nemali veľké problémy pri rozpoznávaní referenčného signálu, priemerné hodnotenie bolo pri všetkých testovaných signáloch viac ako 95 bodov. Takisto bola vo všetkých prípadoch rozpoznaná kotva, ktorej priemerné hodnoty boli v ôsmich prípadoch menšie ako 10, teda „neprijateľnej“ kvality, a v prípade signálu orchestra kvantizovaného na 64 hladín dosiahla kotva „zlú“ kvalitu s hodnotením 23 bodov.

Pri signále huslí bol pre malý počet hladín subjektívne najlepšie zrekonštruovaný signál pomocou A-SPADEQ algoritmu, o pár bodov menej boli hodnotené S-SPADEQ a S-SPADEQ_{DR}. Pre 15 a 32 hladín bol najlepšie hodnotený CP algoritmus, ktorý v oboch prípadoch získal „dobré“ hodnotenie kvality, pričom pre 32 hladín bol iba o jeden bod horšie hodnotený S-SPADEQ algoritmus.



Obr. 5.16: Priemerné hodnotenie stimulov z testu MUSHRA pre rôzne signály a počty kvantizačných hladín

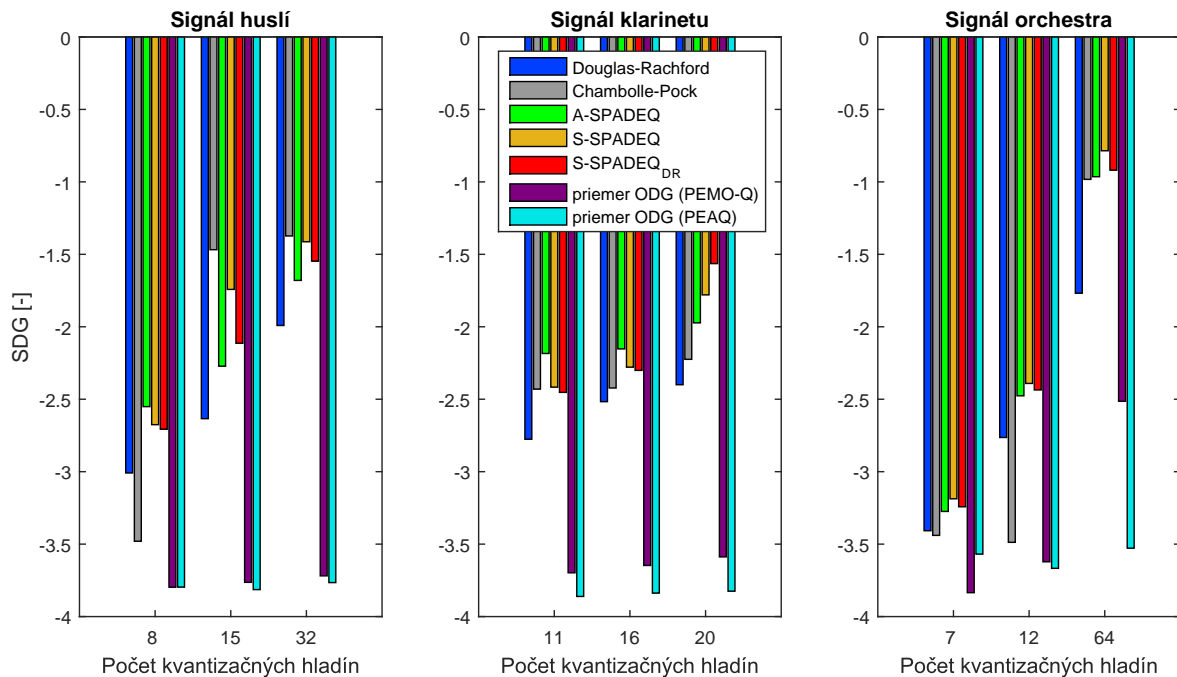
Reštaurovaný signál klarinetu bol pre všetky testované kvantizačné hladiny hodnotený najlepšie pre SPADEQ algoritmy, konkrétne A-SPADEQ pre 11 a 16 hladín a S-SPADEQ_{DR} pre 20 hladín, ktorého rekonštrukcia ako jediného algoritmu bola hodnotená ako „dobrá“. Algoritmy DR a CP boli ohodnotené navzájom podobne, vždy o 5 – 20 bodov horšie ako najlepší zo SPADEQ algoritmov.

Pre signál orchestra boli reštaurácie všetkých algoritmov pre 7 kvantizačných hladín hodnotené ako „neprijateľné“. Pre 12 hladín boli SPADEQ algoritmy hodnotené najlepšie, všetky získali okolo 40 bodov s rozdielmi menšími ako jeden bod. Pre 64 hladín boli rekonštruované signály všetkými algoritmi okrem DR hodnotené ako „dobrej“ kvality.

5.5 Porovnanie dosiahnutých výsledkov ukazovateľa ODG a MUSHRA testu

Keďže metóda MUSHRA používa inú hodnotiacu stupnicu ako metódy pracujúce s ODG, hodnoty jednotlivých rekonštrukcií z MUSHRA testu bolo najprv potrebné vhodne transformovať do ODG stupnice. MUSHRA hodnoty boli transformované lineárne, do úvahy sa brali hodnoty medzi referenčným signálom a kotvou, tzn. hodnotenie referenčného stimulu pre každú skúšku testu bolo prevedené na hodnotu 0,0 ODG stupnice a hodnotenie kotvy na najnižšiu známku, -4,0. Hodnoty ODG z metód PEMO-Q a PEAQ boli získané spriemerovaním hodnôt pre všetky algoritmy

pre daný signál a počet kvantizačných hladín. Porovnanie výsledkov je zobrazené na obr. 5.17.



Obr. 5.17: Porovnanie priemerných hodnotení stimulov z testu MUSHRA transformovaných na SDG stupnicu a hodnôt ODG z metód PEMO-Q a PEAQ

Z grafu 5.17 je zrejmé, že vo všetkých prípadoch rekonštruované signály boli hodnotené subjektívne kvalitnejšie, než na hodnoty, ktoré boli predpovedané pomocou metód PEMO-Q a PEAQ. Najväčší rozdiel bol pre signál huslí pre 32 kvantizačných hladín, kde PEMO-Q a PEAQ vyhodnotili zhoršenie kvality rekonštrukcie ako „veľmi nepríjemné“, pričom poslucháči v teste ohodnotili zhoršenie v priemere ako „roznateľné, nie nepríjemné“ alebo „trochu nepríjemné“.

Vo väčšine prípadov bola rekonštrukcia pre rôzne signály pre väčší počet kvantizačných hladín hodnotená lepšie ako pre menší počet, čo by aj malo odpovedať skutočnosti. Jediný prípad, kde toto neplatí, bol pre signál orchestra pre 7 a 12 hladín rekonštruovaný CP algoritmom. Subjektívne hodnotenie dosiahlo známku „nepríjemného“ zhoršenia kvality, a značí, že CP algoritmus je pre rekonštrukciu signálu kvantizovaného na malý počet hladín (menej ako 12) nevhodný.

Rozdiely medzi hodnotami ODG získanými z metód PEMO-Q a PEAQ a transformovaným MUSHRA hodnotením môžu byť spôsobené tým, že PEMO-Q ani PEAQ neboli vytvorené primárne pre ohodnotenie rekonštrukcií signálov poškodených kvantizáciou. Z výsledkov v grafe 5.17 sa dá vyvodiť, že ani jedna z týchto metód nie je vhodná na predpovedanie subjektívneho hodnotenia zhoršenia kvality,

minimálne pre hodnoty počtu kvantizačných hladín testované v tejto práci.

Z porovnania výsledkov posluchového testu MUSHRA a hodnotenia ODG z metód PEMO-Q a PEAQ sa dá skonštatovať, že pre hocikjaký počet kvantizačných hladín je výhodné signál reštaurovať niektorým zo SPADEQ algoritmov. Výsledky verzií pre analyzujúci model signálu aj verzií pre syntezujúci model sú veľmi podobné a jednoznačná odpoveď na to, ktorý z nich je najlepší pre väčšinu prípadov sa nedá určiť. Pre počet kvantizačných hladín 15, 32 a 64 je možné použiť aj CP algoritmus, ktorý v týchto prípadoch dosiahol podobné hodnotenie ako SPADEQ algoritmy. Rekonštrukcie DR algoritmom dosahovali vo všetkých testovaných signáloch najhoršie alebo druhé najhoršie hodnotenia, a dá sa vyvodiť záver, že tento algoritmus nie je pre reštauráciu kvantizovaných signálov veľmi vhodný.

6 Záver

Táto diplomová práca sa zaoberá problematikou reštaurácie zvukového signálu po poškodení kvantizáciou.

V prvej kapitole je popísaný obecný opis javov kvantizácie a dekvantizácie a využitie týchto javov v spracovaní signálov. Ďalej sú uvedené rôzne existujúce metódy pre dekvantizáciu zvukových signálov, ich princípy fungovania a dosiahnuté výsledky z testovania.

Druhá kapitola rozoberá teóriu riedkych reprezentácií, vzťahujúcu sa k dekvantizácii. Najskôr je uvedené použité značenie a základné poznatky ako sú normy vektorov, bázy vektorových priestorov a framy. Tiež je tu opísaná Gaborova transformácia. V ďalšej časti tejto kapitoly je popísaná problematika riedkosti signálu. Uvedené sú úlohy, s ktorými sa bude pracovať, a ich riešenia využívajúce minimalizáciu ℓ_1 normy. Následne sú vymenované rôzne druhy proximálnych algoritmov a podmienky, s ktorými sa hľadajú vyhovujúce riešenia. V poslednej časti tejto kapitoly je popis proximálnych operátorov a syntezujúceho a analyzujúceho modelu signálu.

Obsahom tretej kapitoly je implementácia riešenia dekvantizácie v podobe algoritmov založených na konvexnej (Douglas–Rachford, Chambolle–Pock) a nekonvexnej (SPADEQ) optimalizácii pre analyzujúci aj syntezujúci model signálu. Opísané sú princípy týchto algoritmov, ich jednotlivé kroky a vstupné a výstupné parametre.

V ďalšej kapitole je popis programového riešenia vytvoreného v aplikácii MATLAB. Popísané sú jednotlivé súbory skriptu pre dekvantizáciu a ich použitie v programe.

Posledná kapitola obsahuje opis metód testovania rekonštrukcie signálu podľa objektívnych aj subjektívnych testov a výsledky testovania. Objektívne ukazovatele, podľa ktorých boli algoritmy otestované, zahŕňajú (Δ) SDR, PEMO-Q, PEAQ a subjektívny posluchový test bol vykonaný metódou MUSHRA. V prvej časti tejto kapitoly sú opísané metodiky jednotlivých ukazovateľov. Ďalšia časť je zameraná na hľadanie najvhodnejších hodnôt vstupných parametrov jednotlivých algoritmov pre získanie čo najlepšieho výsledku Δ SDR. Pre každý signál boli nájdené vhodné parametre pre algoritmy a takto rekonštruované signály sa použili v ďalších častiach kapitoly aj na testovanie podľa ostatných ukazovateľov.

Podľa ukazovateľa Δ SDR vykonáva rekonštrukciu signálov s najväčšou úspešnosťou algoritmus S–SPADEQ, a celkovo SPADEQ algoritmy boli úspešnejšie ako DR a CP, pričom DR algoritmus podával najhoršie výsledky. Ukazovatele PEMO-Q a najmä PEAQ ukázali nejednoznačnosť predošlých výsledkov, keďže najúspešnejšie algoritmy podľa týchto ukazovateľov boli CP a A–SPADEQ. Veľký rozdiel hodnotení podľa PEMO-Q a PEAQ voči subjektívnym hodnotám z testu MUSHRA ukazuje, že

ani jedna z týchto metód nie je vhodná na predpovedanie subjektívneho hodnotenia reštaurovaných signálov, ktoré boli poškodené kvantizáciou.

Výsledky subjektívneho testovania metódou MUSHRA ukázali, že najlepšia rekonštrukcia sa dosiahne použitím SPADEQ algoritmov pre všetky hodnoty počtu kvantizačných hladín a pre vyšší počet hladín je výhodné na rekonštrukciu použiť aj CP algoritmus.

Literatúra

- [1] GRAY, R. M., NEUHOFF, D. L. Quantization. *IEEE Transactions on Information Theory*, vol. 44, no. 6, s. 2325-2383, 1998. ISSN 1557-9654
- [2] MAHER, R. C. On the nature of granulation noise in uniform quantization systems. *J. Audio Eng. Soc.*, vol. 40, no. 1/2, s. 12-20, 1992. Dostupné z: <http://www.montana.edu/rmaher/publications/maher_jaes_0192_12-20.pdf>
- [3] KRATOCHVÍL, T. *Nízkofrekvenční a audio elektronika, přednáška č. 7*. [Přednáška]. Brno: FEKT VUT v Brně, 2017.
- [4] TROUGHTON, P. T. Bayesian restoration of quantised audio signals using a sinusoidal model with autoregressive residuals. *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. WAS-PAA'99 (Cat. No.99TH8452)*, New Paltz, NY, USA, 1999, s. 159-162. DOI: 10.1109/ASPAA.1999.810874
- [5] BRAUER, C., GERKMANN, T., LORENZ, D. Sparse reconstruction of quantized speech signals. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, s. 5940-5944. DOI: 10.1109/ICASSP.2016.7472817
- [6] BRAUER, C., ZHAO, Z., LORENZ, D., FINGSCHEIDT, T. Learning to Dequantize Speech Signals by Primal-dual Networks: an Approach for Acoustic Sensor Networks. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, s. 7000-7004. DOI: 10.1109/ICASSP.2019.8683341
- [7] RENCKER, L., BACH, F., WANG, W., PLUMBLEY, M. D. Fast Iterative Shrinkage for Signal Declipping and Dequantization. *Proceedings of iTWIST'18, Paper-ID: 4, Marseille, France, November, 21-23, 2018*. 2018. Dostupné z: <<https://arxiv.org/pdf/1812.01540.pdf>>
- [8] RAJMÍČ, P., DAŇKOVÁ, M. *Úvod do řídkých reprezentací signálů a komprimovaného snímání*. Brno: Vysoké učení technické v Brně, 2014. ISBN 978-80-214-5169-8
- [9] KRAMÁŘ, D. *Restaurace signálu po průchodu limiterem s použitím psychoakustického modelu: bakalářská práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 70 strán. Vedúci práce Ing. Pavel Záviška.

- [10] ZÁVIŠKA, P., RAJMIC, P., SCHIMMEL, J. Psychoacoustically Motivated Audio Declipping Based on Weighted l_1 Minimization. *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. Budapest, Hungary, 2019, s. 338-342. DOI: 10.1109/TSP.2019.8769109
- [11] ZÁVIŠKA, P., RAJMIC, P., PRŮŠA, Z., VESELÝ, V. Revisiting Synthesis Model in Sparse Audio Declipper. *LVA ICA 2018. The 14th International Conference on Latent Variable Analysis and Signal Separation*. Guildford, United Kingdom, 2018, s. 429-445. DOI: 10.1109/TSP.2019.8769109
- [12] RUDIN, W. *Analýza v reálném a komplexním oboru*. Praha: Academia, druhé vydanie, 2003.
- [13] CHRISTENSEN, O. *Frames and Bases, An Introductory Course*. Boston: Birkhäuser, 2008. ISBN 9780817646776
- [14] BRUCKSTEIN, A. M., DONOHO, D. L., ELAD, M. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. *SIAM Review*, ročník 51, č. 1, s. 34–81, 2009. ISSN 0036-1445
- [15] BOYD, S. P., VANDENBERGHE, L. *Convex Optimization*. Cambridge University Press, 2004. ISBN 0521833787
- [16] COMBETTES, P. L., PESQUET, J. A Douglas–Rachford Splitting Approach to Nonsmooth Convex Variational Signal Recovery *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, s. 564-574, 2007. DOI: 10.1109/J-STSP.2007.910264
- [17] ZÁVIŠKA, P., RAJMIC, P., MOKRÝ, O. *Chambolle-Pock algorithm used for Declipping*. Brno: Vysoké učení technické v Brně, 2017.
- [18] KITIĆ, S., BERTIN, N., GRIBONVAL, R., Sparsity and cosparsity for audio declipping: a flexible non-convex approach. *LVA/ICA 2015 - The 12th International Conference on Latent Variable Analysis and Signal Separation*. Liberec, Czech Republic, 2015, s.8. DOI: 10.1007/978-3-319-22482-4_28
- [19] ZÁVIŠKA, P., MOKRÝ, O., RAJMIC, P. *S-SPADE Done Right: Detailed Study of the Sparse Audio Declipper Algorithms*. [Techreport]. Brno: University of Technology, Czech Republic, 2018. Dostupné z: <<https://arxiv.org/pdf/1809.09847.pdf>>

- [20] ZÁVIŠKA, P., RAJMÍČ, P., MOKRÝ, O., PRŮŠA, Z. A Proper Version of Synthesis-based Sparse Audio Declipper. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom, 2019, s. 591-595. DOI: 10.1109/ICASSP.2019.8682348
- [21] HUBER, R., KOLLMEIER, B. PEMO-Q—A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception. *IEEE Transactions on Audio, Speech, and Language Processing*. vol. 14, no. 6, 2006, s. 1902-1911. DOI: 10.1109/TASL.2006.883259
- [22] DAU, T., KOLLMEIER, B., KOHLRAUSCH, A. Modeling auditory processing of amplitude modulation. I. Detection and masking with narrow-band carriers. *The Journal of the Acoustical Society of America*. vol. 102, no. 6, 1997, s. 2892-2905. DOI: 10.1121/1.420344
- [23] KABAL, P. An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality. *IEEE Transactions on Audio, Speech, and Language Processing*. [Techreport]. Montreal: McGill University, Department of Electrical & Computer Engineering, 2003. Dostupné z: <<http://www-mmsp.ece.mcgill.ca/Documents/Reports/2002/KabalR2002v2.pdf>>
- [24] Recommendation ITU-R BS.1387 *Method for Objective Measurements of Perceived Audio Quality*. 1998. Dostupné z: <https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1387-1-200111-I!!PDF-E.pdf>
- [25] SCHIMMEL, J. *Akustika a zvukové systémy*. Učebný text, Brno: Vysoké učení technické v Brně, 2018.
- [26] ČERNÝ, V. *Nástroj pro subjektivní hodnocení kvality restaurace audio signálů: bakalářská práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 50 strán. Vedúci práce Ing. Pavel Závíška.

Zoznam symbolov, veličín a skratiek

DR	Douglas–Rachford
CP	Chambolle–Pock
SPADEQ	Sparse Audio Dequantizer
SDR	Signal to Distortion Ratio
(F)ISTA	(Fast) Iterative Shrinkage/Thresholding Algorithm
A/D	Analog-to-Digital
MCMC	Markov Chain Monte Carlo
RMS	Root Mean Square
CELP	Code-Excited Linear Predictive speech coding
AMR	Adaptive Multi-Rate
AMR-WB	Wideband Adaptive Multi-Rate
DFT	Discrete Fourier Transform
SNR	Signal to Noise Ratio
MSE	Mean Squared Error
PESQ	Perceptual Evaluation of Speech Quality
ADMM	Alternating Direction Method of Multipliers
STFT	Short-Time Fourier Transform
LASSO	Least Absolute Shrinkage and Selection Operator
NSP	Null Space Property
RIP	Restricted Isometry Property
MP	Matching Pursuit
OMP	Orthogonal Matching Pursuit
BP	Basis Pursuit
LARS	Least Angle Regression
IRLS	Iterative Reweighted Least Squares
FOCUSS	Focal Underdetermined System Solver
DC	Direct Current
PEMO-Q	Perception Model-Quality
SDG	Subjective Difference Grade
ODG	Objective Difference Grade
PSM	Perceptual Similarity Measure
PEAQ	Perceptual Evaluation of Audio Quality
MUSHRA	Multiple Stimuli with Hidden Reference and Anchor

A Elektronická príloha

V elektronickej prílohe sa v hlavnej zložke nachádzajú skripty s funkciami pre spustenie v programe MATLAB. Hlavný skript je nazvaný `main.m`, v ňom je možné nastaviť vstupné parametre ako napr. počet kvantizačných hladín, typ kvantizácie, veľkosť a posun okna framu, počet iterácii algoritmu alebo hodnota prahu mäkkého prahovania. Funkcia `quant.m` vytvorí kvantizátor na základe vstupných parametrov zadanych v hlavnom skripte, vstupný signál kvantizuje a vráti poškodený signál. Funkcie `douglas.m` a `chambolle.m` obsahujú kódy pre Douglas-Rachfordov a Chambolle-Pockov algoritmus, po ich priebehu vráti do hlavného skriptu rekonštruovaný signál. Funkcie `proj.m` a `proj_c.m` označujú projekcie na množinu prípustných riešení v časovej, resp. kmitočtovej oblasti. Pre segmentáciu signálu potrebnú pre realizáciu SPADEQ algoritmov je určená funkcia `spadeq_segment.m`, pričom z tejto funkcie sa volajú funkcie SPADEQ algoritmov `aspadeq.m`, `sspadeq.m` a `sspadeq_new.m`. Poslednou funkciou je `hard_thr.m`, ktorá slúži pre vykonanie tvrdého prahovania ako jedného z krokov SPADEQ algoritmov.

Zložka Grafy obsahuje grafické znázornenia hodnôt ukazovateľov ΔSDR , SDR , ODG a PSM_t pre testované signály pre rôzny počet kvantizačných hladín.

V zložke Tabuľky sa nachádzajú tabuľky s najlepšími hodnotami parametrov pre jednotlivé algoritmy pre každý signál a rôzne počty kvantizačných hladín.

Testované signály sú uložené v zložke Zvukové signály.

Kód bol vytvorený a testovaný vo verzii programu MATLAB R2015a.