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

MODERNÍ RESTAURACE AUDIOSIGNÁLU S CHYBĚJÍCÍMI ÚSEKY
MODERN RESTORATION OF AUDIO CONTAINING MISSING PORTIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Pavel Skyva
AUTHOR

VEDOUCÍ PRÁCE — doc. Mgr. Pavel Rajmic, Ph.D.
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Audio inženýrství**
Ústav telekomunikací

Student: Pavel Skyva ***ID:*** 173739
Ročník: 3 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Moderní restaurace audiosignálu s chybějícími úseky

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte model znehodnocení audiosignálu s chybějícími vzorky. Analyzujte tento problém a seznamte se s obvyklými metodami pro jeho řešení, jejich vlastnostmi. Rovněž se seznamte s moderními metodami zaměřenými na tzv. řídkosti signálů a navrhněte základní metodu restaurace. Implementujte v MATLABu proximální algoritmus řešící základní úlohu. Dále navrhněte možná vylepšení úlohy/algoritmu, implementujte a otestujte na simulovaných i reálných nahrávkách, včetně objektivního a subjektivního hodnocení kvality restaurace.

DOPORUČENÁ LITERATURA:

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

[2] Špiřík, J., Rajmic, P., Veselý, V. Reprezentace signálů: od bází k framům, Elektrorevue, 2010. ISSN 1213-1539

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 8.6.2017

Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.
Konzultant:

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

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

ABSTRAKT

Tato bakalářská práce se zabývá moderní metodou restaurace audiosignálů s chybějícími úseky. Rekonstrukce je založena především na řídkých reprezentací signálů. V práci je popsán způsob hledání řídkého řešení pomocí proximálního Douglas-Rachfordova algoritmu a jeho následné užití pro výpočet zrekonstruovaného signálu. Výpočetní program je implementován v prostředí MATLAB a k výpočtům využívá toolbox LTFAT. Výsledky rekonstrukce jsou hodnoceny objektivní metodou odstupu signálu od šumu (SNR).

KLÍČOVÁ SLOVA

Autoregrese, Douglas-Rachfordův algoritmus, LTFAT, MATLAB, proximální algoritmus, restaurace audiosignálu, řídké reprezentace.

ABSTRACT

This bachelor thesis deals with modern method of audio signal restoration. The reconstruction is primarily based on sparse signal representations. In thesis is described the way of searching sparse coefficients using proximal Douglas–Rachford algorithm and following computation of reconstructed signal using this coefficients. The algorithm of reconstruction is implemented in the MATLAB software with toolbox LTFAT included. Results of the reconstruction are compared using objective evaluation method Signal-to-Noise ratio (SNR).

KEYWORDS

Audio restoration, autoregression, Douglas-Rachford algorithm, LTFAT, MATLAB, proximal algorithm, sparse represenations.

SKYVA, P. *Moderní restaurace audiosignálu s chybějícími úseky.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 44 s. Vedoucí bakalářské práce doc. Mgr. Pavel Rajmic, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Moderní restaurace audiosignálu s chybějícími úseky“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne …… .....

podpis autora

PODĚKOVÁNÍ

Velmi rád bych poděkoval svému vedoucímu bakalářské práce panu doc. Mgr. Pavlu Rajmicovi, Ph.D. za odborné vedení, vstřícný a trpělivý přístup při konzultacích a za cenné návrhy a rady.

V Brně dne …… .....

podpis autora

Obsah

ÚVOD **8**

1 ZNAČENÍ A ZÁKLADNÍ DEFINICE **9**

1.1 Normy vektorů 9

1.2 Řídkost 10

1.3 Pseudoinverzní matice 10

2 BÁZE A FRAMY **11**

2.1 Báze vektorového prostoru 11

2.1.1 Ortogonální a ortonormální báze 11

2.2 Framy 11

2.2.1 Gaborovy framy 11

3 ŘÍDKÉ REPREZENTACE SIGNÁLŮ **14**

3.1 Metody pro hledání řídkých řešení 14

3.1.1 $\boldsymbol{\ell_1}$-relaxace 15

4 PROXIMÁLNÍ ALGORITMY **16**

4.1 Omezený a neomezený tvar optimalizační úlohy 16

4.2 Proximální operátor 16

4.3 Douglas-Rachfordův algoritmus 16

5 FORMULACE PROBLÉMU, MOŽNÁ ŘEŠENÍ **18**

5.1 Některé dřívější metody restaurace audiosignálu 18

5.1.1 Autoregresní metoda 18

5.1.2 Sinusoidální model 20

5.2 Moderní metoda restaurace založená na řídkosti signálů 21

5.2.1 Konkrétní podoba Douglas-Rachfordova algoritmu 22

5.2.2 Návrh vylepšení metody restaurace založené na řídkosti signálů 24

6 IMPLEMENTACE ALGORITMU V PROSTŘEDÍ MATLAB **26**

6.1 Implementace navrženého vylepšení 29

7 TESTOVÁNÍ ALGORITMU, HODNOCENÍ REKONSTRUKCE **30**

7.1 Metoda objektivního hodnocení úspěšnosti rekonstrukce 30

7.2 Vliv parametrů slovníku na rekonstrukci 30

7.2.1 Vliv šířky okna 30

7.2.2 Vliv posunu okna 32

7.3 Hodnocení úspěšnosti rekonstrukce pomocí SNR 33

7.3.1 Vliv vylepšení algoritmu na úspěšnost rekonstrukce 37

ZÁVĚR **39**
LITERATURA **40**
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK **42**
SEZNAM PŘÍLOH **43**
A OBSAH PŘILOŽENÉHO CD **44**

ÚVOD

Tato bakalářská práce se zabývá možnými způsoby rekonstrukce audiosignálu s chybějícími úseky. Ukážeme si zde dvě obvyklé metody řešení tohoto problému a jednu moderní metodu zaměřenou na tzv. řídkosti signálů. Ta pro nás bude stěžejní. Cílem je vytvořit kompletní algoritmus pro rekonstrukci signálů, navrhnout jeho vylepšení a otestovat daný algoritmu i s vylepšením na reálných nahrávkách.

V prvních dvou kapitolách si stanovíme základní značení a definujeme pojmy potřebné k porozumění dané problematice. Ve třetí části si řekneme něco o řídkých reprezentacích signálů. Následuje kapitola o proximálních algoritmech, pomocí kterých se řídké reprezentace hledají. Pátá kapitola je věnována samotnému problému chybějících úseků signálu a jejich rekonstrukci. V této kapitole nalezneme kompletní teoretické zpracování výpočetního algoritmu a návrh vylepšení tohoto algoritmu. V šesté kapitole se věnujeme konkrétní podobě realizace výpočtů v prostředí MATLAB. Poslední kapitola je věnována testování algoritmu a vyhodnocení úspěšnosti rekonstrukce.

1 ZNAČENÍ A ZÁKLADNÍ DEFINICE

Skalární veličiny značíme kurzívou, např. takto m, N. Vektory budeme značit tučným řezem, jako např. $\mathbf{x}, \mathbf{y}$. Nebude-li zadáno jinak, pak budeme všechny konečněrozměrné vektory uvažovat jako sloupcové. Jednotlivé prvky vektoru budeme indexovat od jedničky, tzn. $\mathbf{x} = [x_1, \dots, x_n]^{\mathrm{T}}$.

Definice 1.1. *Nosičem vektoru* $\mathbf{x}$ *myslíme množinu jeho indexů, v nichž má vektor nenulové hodnoty. Tuto množinu značíme* $\mathrm{supp}(\mathbf{x})$. *Tedy* $\mathrm{supp}(\mathbf{x}) = \{i | x_i \neq 0\}$.

Pro názornost, máme-li signál $\mathbf{x} = [x_1, \dots, x_n]^{\mathrm{T}} = [0, 0, 5, 0, 3, 7, 0, 9]^{\mathrm{T}}$, pak $\mathrm{supp}(\mathbf{x}) = \{5, 3, 7, 9\}$ a $|\mathrm{supp}(\mathbf{x})| = 4$.

Matice budeme značit rovněž tučně, avšak velkými písmeny, tedy $\mathbf{A}, \boldsymbol{\psi}$. Jednotlivé prvky matic budou značeny jejich adekvátními malými písmeny, tj. $a_{i,j}, \psi_{i,j}$. Řádek i matice $\mathbf{A}$ označíme $a_{i,\cdot}$, obdobně sloupec j jako $a_{\cdot,j}$. Symbolem * bude značena tzv. hermitovská transpozice, matice $\mathbf{A}^*$ vznikne složením transpozice matice a komplexního sdružení každého jejího prvku.

Vektorové prostory budeme značit takto: $\mathbb{V}, \mathbb{R}^n, \mathbb{C}^N$.

1.1 Normy vektorů

Norma vektoru je funkce, která každému vektoru přiřadí číslo vyjadřující jeho velikost.

Definice 1.2. ℓ_p*-norma vektoru* $\mathbf{x} \in \mathbb{C}^N$ *je definována jako*

$$\begin{aligned} \|\mathbf{x}\|_p &:= \left(\sum_{i=1}^{N} |x_i|^p \right)^{\frac{1}{p}} \quad pro\ 1 \leq p < \infty, \\ \|\mathbf{x}\|_p &:= \sum_{i=1}^{N} |x_i|^p \quad pro\ 0 < p < 1, \\ \|\mathbf{x}\|_\infty &:= \max_i |x_i|, \\ \|\mathbf{x}\|_0 &:= |\mathrm{supp}(\mathbf{x})|. \end{aligned} \tag{1.1}$$

Nejčastěji používanými normami jsou normy

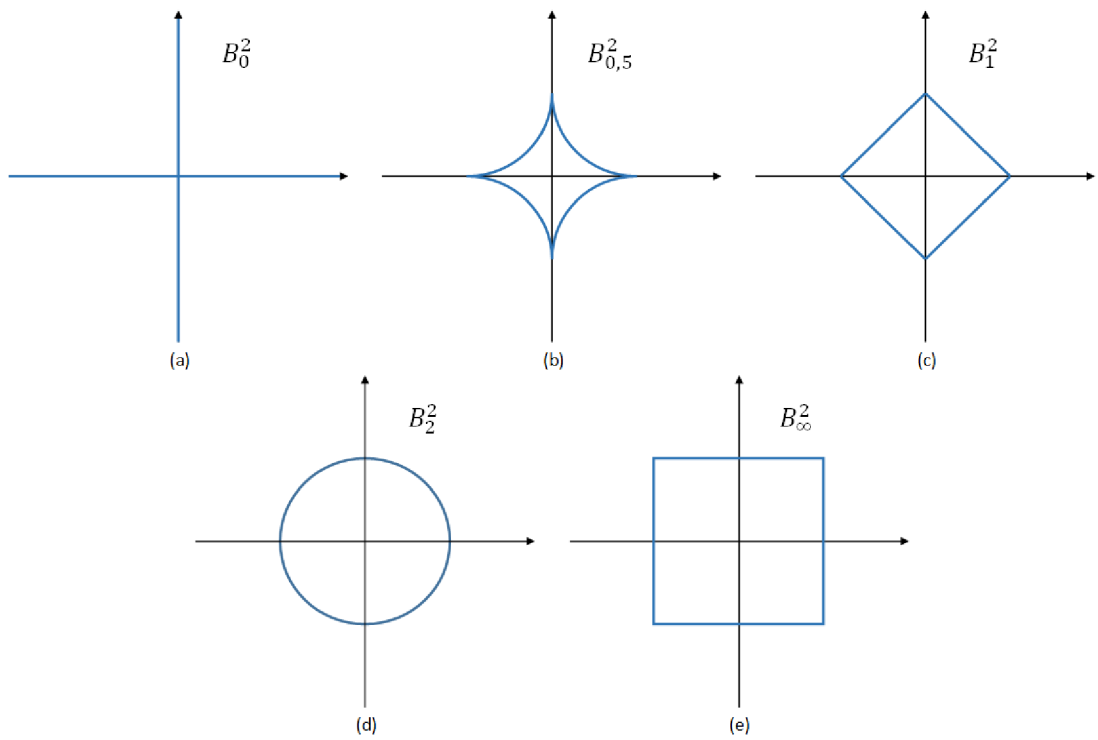
- $\|\cdot\|_0$, která představuje počet nenulových prvků vektoru, $\|\mathbf{x}\|_0 = |\mathrm{supp}(\mathbf{x})|$,
- $\|\cdot\|_1$, která představuje součet absolutních hodnot prvků vektoru, $\|\mathbf{x}\|_1 = \sum_i |x_i|$,
- $\|\cdot\|_2$, která představuje tzv. eukleidovskou normu, $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$.

Eukleidovská norma, se často značí zjednodušeně jako $\|\cdot\|$. Pro lepší představu si zobrazíme jednotkovou kouli v několika normách.

Definice 1.3. *Jednotková koule* B_p^N *je v normě* ℓ_p *definována jako*

$$B_p^N := \{\mathbf{x} \in \mathbb{C}^N | \|\mathbf{x}\|_p \leq 1\} \tag{1.2}$$

Zobrazení jednotkových koulí v různých normách je na obrázku 1.1.



Obr. 1.1: Zobrazení hranic jednotkových koulí (a) B_0^2, (b) $B_{0,5}^2$, (c) B_1^2, (d) B_2^2, (e) B_∞^2.

1.2 Řídkost

Definice 1.4. *Vektor* $\mathbf{x} \in \mathbb{C}^N$ *je k-řídký, pokud platí*

$$\|\mathbf{x}\|_0 \leq k. \tag{1.3}$$

Vektor tedy nazveme k-řídkým, pokud obsahuje nejvýše k nenulových složek.

1.3 Pseudoinverzní matice

Jedná se o zobecnění inverze matice pro neregulární matice. Pseudoinverzní matici k matici $\mathbf{A}$ budeme značit $\mathbf{A}^+$. Explicitně bychom mohli matici $\mathbf{A}^+$ vyjádřit takto

$$\mathbf{A}^+ = (\mathbf{A}^\mathrm{T}\mathbf{A})^{-1}\mathbf{A}^\mathrm{T}. \tag{1.4}$$

Definice 1.5. *Pro pseudoinverzní matici* $\mathbf{A}^+$ *k matici* $\mathbf{A}$ *platí*

$$\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A},$$

$$\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+,$$

$$(\mathbf{A}\mathbf{A}^+)^* = \mathbf{A}\mathbf{A}^+,$$

$$(\mathbf{A}^+\mathbf{A})^* = \mathbf{A}^+\mathbf{A}.$$

Jsou-li splněny tyto čtyři podmínky, pak je pro lineární systém $\mathbf{A}\mathbf{x} = \mathbf{y}$ řešením s minimální energií řešení $\mathbf{x} = \mathbf{A}^+\mathbf{y}$.

2 BÁZE A FRAMY

2.1 Báze vektorového prostoru

Bází vektorového prostoru rozumíme množinu lineárně nezávislých vektorů, jejichž lineární kombinací jsme schopni vyjádřit jakýkoliv vektor v daném vektorovém prostoru. Máme-li bázi $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$, pak můžeme každý prvek $\mathbf{x} \in \mathbb{V}$ vyjádřit jako

$$\mathbf{x} = \sum_{i=1}^{n} c_i \mathbf{b}_i = \mathbf{B}\mathbf{c}. \tag{2.1}$$

2.1.1 Ortogonální a ortonormální báze

S ortogonálními a ortonormálními bázemi se v praxi pracuje nejjednodušeji. Pro ortogonální či ortonormální bázi platí, že jakékoliv dva její bázové vektory jsou na sebe vždy kolmé. Potom tedy v bázi $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ pro její libovolné dva vektory platí

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0, \langle \mathbf{b}_i, \mathbf{b}_i \rangle \neq 0. \tag{2.2}$$

V ortonormální bázi navíc pro kterýkoliv její prvek $\mathbf{b}_i$ platí $\|\mathbf{b}_i\| = 1$, neboli $\mathbf{B}^* = \mathbf{B}^{-1}$.

2.2 Framy

Frame je rozšíření báze vektorového prostoru. Na rozdíl od bází, kde je počet bázových vektorů v konečněrozměrném prostoru roven dimenzi vektorového prostoru, frame je tvořen množinou vektorů, jejichž počet může být větší než je dimenze prostoru. Stále však pomocí těchto vektorů můžeme vyjádřit jakýkoliv vektor v daném vektorovém prostoru. Z uvedeného však vyplývá, že vektory tvořící frame již nutně musí být lineárně závislé. Užití framů může mít výhody i nevýhody. Framy jsou méně omezené než báze, a proto se používají pro svoji flexibilitu. Nevýhodou jsou náročnější výpočtové postupy a riziko numerické nestability [1].

Definice 2.1. *Spočetná množina vektorů* $\{\phi_k\}_{k\in\mathbb{I}}$ *ve vektorovém prostoru* $\mathbb{V}$ *se nazývá frame, pokud existují takové konstanty* $0 < \mathrm{A} \leq \mathrm{B} < \infty$, *že*

$$A\|\mathbf{x}\|^2 \leq \sum_{k\in\mathbb{I}} |\langle \mathbf{x}, \phi_k \rangle|^2 \leq B\|\mathbf{x}\|^2, \forall \mathbf{x} \in \mathbb{V}. \tag{2.3}$$

Prvky framu ϕ_k se obvykle nazývají atomy. Konstanty A, B se nazývají meze framu. Je-li $A = B$, pak takový frame nazýváme těsný frame. Speciálním případem je potom tzv. 1-těsný frame, často také nazývaný těsný Parsevalův frame, pro nějž dokonce platí $A = B = 1$.

2.2.1 Gaborovy framy

Jako Gaborovy framy označujeme speciální skupinu framů, které jsou dnes hojně využívány především pro časově-kmitočtovou analýzu signálů. S využitím těchto framů jsme schopni lokalizovat časovou a frekvenční informaci funkcí. Tyto informace pak

obvykle využíváme k vystavění tzv. spektrogramu signálu. Ve většině případů se jedná o dvojrozměrný obrázek s nepravými barvami. Příklad spektrogramu je na obrázku 2.1.

Gaborova analýza v $L^2(\mathbb{R})$ je založena na operátorech translace a modulace a zaměřuje se na reprezentaci funkce $f \in L^2(\mathbb{R})$ jako superpozici posunutých a modulovaných verzí fixní funkce $g \in L^2(\mathbb{R})$. Soubor takových funkcí

$$\{\mathrm{e}^{\mathrm{j}2\pi mbx} g(x - na)\}_{m,n\in\mathbb{Z}} = \{E_{mb} T_{na} g\}_{m,n\in\mathbb{Z}} \tag{2.4}$$

pro zvolené dvě hodnoty a a b se nazývá Gaborův (či gaborovský) systém, podle Dennise Gabora, který se jako první věnoval jejich matematické analýze [1]. Funkce E_{mb} reprezentuje modulované verze funkce g a funkce T_{na} představuje posunuté verze funkce g. Často se také stejná problematika označuje jako STFT (krátkodobá Fourierova transformace).

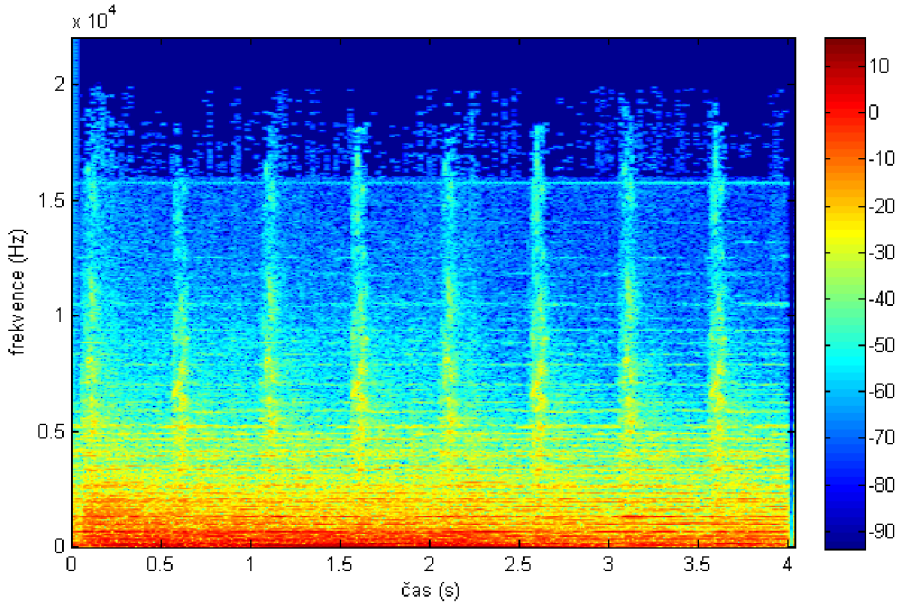
Základní otázkou je, jak zvolit funkci $g \in L^2(\mathbb{R})$ a parametry translace a modulace $a, b > 0$ tak, aby Gaborův systém (2.2) tvořil v prostoru $L^2(\mathbb{R})$ frame [2]. D. Gabor uvažoval posloupnost funkcí $\{E_{mb} T_{na} g\}_{m,n\in\mathbb{Z}}$, kde $ab = 1$ a g je Gaussova funkce

$$g(x) = \mathrm{e}^{\frac{-x^2}{2}}. \tag{2.5}$$

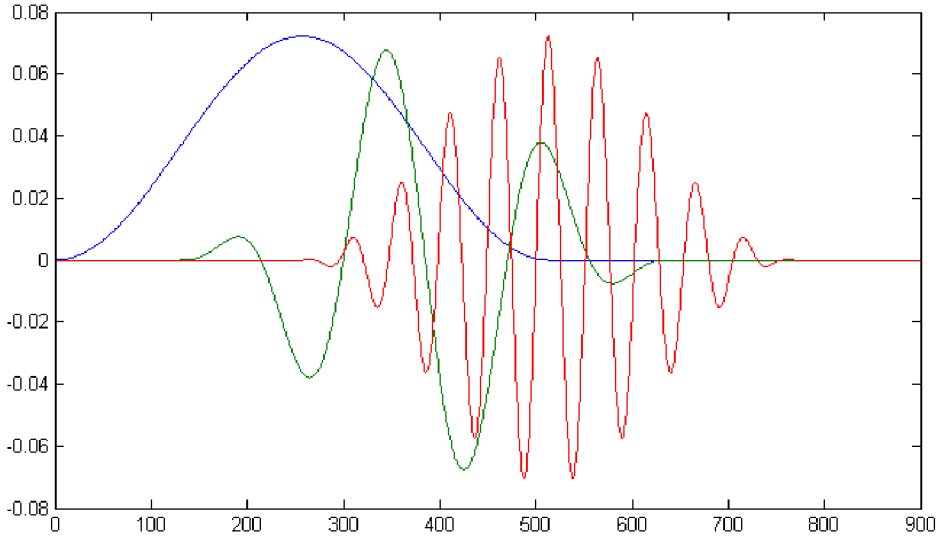
Funkce g se nazývá okénková funkce [3]. Parametr $na, n \in \mathbb{Z}$ ovlivňuje pouze translaci, parametr $mb, m \in \mathbb{Z}$ ovlivňuje pouze modulaci. Explicitně pak lze tyto funkce napsat jako

$$E_{mb} T_{na} g = \mathrm{e}^{j2\pi mbx} g(x - na). \tag{2.6}$$

Největší problém však je najít takovou funkci g, která vytvoří frame v celém $L^2(\mathbb{R})$. Jednou z možností je užít jako okénkovou funkci tzv. Hannovo okno. Příklad posunutých a modulovaných Hannových oken je na obrázku 2.2.



Obr. 2.1: Ukázka spektrogramu úryvku hudebního signálu.



Obr. 2.2: Ukázka několika prvků Gaborova framu, který je tvořen posunutými a modulovanými okénkovými funkcemi. Okénkovou funkcí je zde Hannovo okno (modře). Zobrazena je pouze reálná část daných prvků.

3 ŘÍDKÉ REPREZENTACE SIGNÁLŮ

Při řešení nedourčených systémů lineárních rovnic pro nás mohou být ze všech možných řešení výhodná taková, která jsou tzv. řídká (mají co nejvíce neznámých současně nulových). Budeme-li řešit soustavu $\mathbf{y} = \mathbf{A}\mathbf{x}$, pak máme takový zájem, aby $\|\mathbf{x}\|_0$ bylo minimální. Náš problém bychom mohli formulovat takto

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \mathbf{y} = \mathbf{A}\mathbf{x}, \tag{3.1}$$

kde známe vektor $\mathbf{y} \in \mathbb{C}^m$ a matici $\mathbf{A} \in \mathbb{C}^{N \times M}$, kdy $M > N$ (obvykle dokonce $M \gg N$). Matice $\mathbf{A}$ se nazývá slovník, jejím sloupcům se říká atomy. $\|\mathbf{x}\|_0$ je ℓ_0-norma rovna počtu nenulových prvků vektoru $\mathbf{x}$.

$$\|\mathbf{x}\|_0 := |\mathrm{supp}(\mathbf{x})|. \tag{3.2}$$

Výpočetní složitost problému (3.1) odpovídá složitosti třídy NP, což znamená, že výsledku je možné dosáhnout v čase vyšším než polynomiálním [4].Pro získání výsledku v rozumném čase musíme, i s rizikem úbytku přesnosti, přistoupit k některé z aproximativních metod.

3.1 Metody pro hledání řídkých řešení

Metody pro hledání řídkých řešení bychom mohli vzhledem k principu, na kterém se zakládají, rozdělit do dvou skupin. První skupina metod vychází z ℓ_1-relaxace (tyto algoritmy se nazývají relaxační), druhou skupinu jsou tzv. hladové („greedy“) algoritmy [5].

Relaxačními algoritmy jsou například:

- BP (Basis Pursuit)
- Modifikovaný LARS (Least Angle Regression, homotopy method)
- IRLS (Iterative Reweighted Least Squares, někdy také jako FOCUSS – FOCal Underdetermined Systém Solver)
- Dantzig Selector

Jsou-li dodrženy podmínky ℓ_1-relaxace, pak jsou tyto algoritmy schopné dostat se k přesnému nebo alespoň relativně blízkému řešení.

Jako hladové algoritmy označujeme například:

- MP (Matching Pursuit)
- OMP (Orthogonal Matching Pursuit)

Hlavní princip těchto algoritmů spočívá v tom, že v každé iteraci najdou jeden (nebo více) „nejvýznamnějších“ atomů, který (případně které) se v dalším průběhu algoritmu už nemohou zbavit podílu na konečném řešení. Výhodou těchto metod je nízká složitost, nevýhodou pak, že není zaručeno dosažení globálního optima [5].

Mimo tyto dvě kategorie algoritmů stojí ještě algoritmy založené na prahování (thresholdingu) a tzv. hybridní algoritmy, které využívají jednotlivé výhody z obou skupin metod.

3.1.1 ℓ_1-relaxace

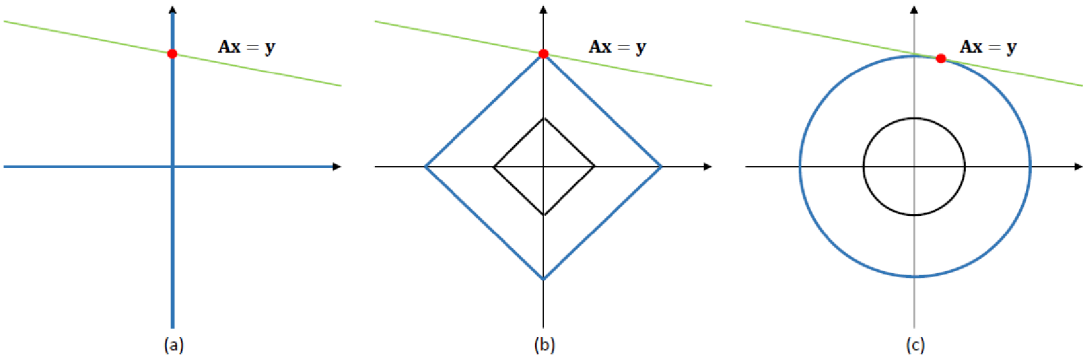
Abychom byli schopni problém (3.1) řešit pomocí algoritmů konvexní optimalizace, přistupujeme k jeho relaxaci. Vzhledem k tomu, že normy ℓ_p jsou konvexní pro $p \geq 1$, jako „nejbližší“ konvexní norma k normě ℓ_0 se nám nabízí ℓ_1-norma. ℓ_1-relaxací bychom měli být schopni dosáhnout alespoň přibližného řešení problému. Relaxovaný problém (3.1) by poté vypadal takto

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{3.3}$$

V některých případech se řešení problémů (3.1) a (3.3) dokonce shodují. Uvažujeme-li však možnou odchylku řešení δ (zašuměná data/nepřesnosti), daný problém bychom mohli napsat jako

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2 \leq \delta. \tag{3.3δ}$$

Problém (3.3δ) je také znám pod zkratkou LASSO (Least Absolute Shrinkage and Selection Operator) [1].



Obr. 3.1: „Nafukující se“ koule v normách (a) ℓ_0 , (b) ℓ_1 a (c) ℓ_2 při styku s nadrovinou určenou rovnicí $\mathbf{Ax} = \mathbf{y}$.

4 PROXIMÁLNÍ ALGORITMY

Mnoho úloh na zpracování signálů lze formulovat jako úlohy konvexní optimalizace, které bychom obecně zapsali takto

$$\arg\min_{x} f_1(x) + f_2(x), \tag{4.1}$$

kde f_1, f_2 jsou konvexní funkce z $\mathbb{R}^N$. Hlavní problém, který vzniká při řešení této úlohy, vyplývá ze skutečnosti, že obvykle některá z funkcí není diferencovatelná [6]. Algoritmy označujeme jako proximální proto, že každá z funkcí z (4.1) v procesu vystupuje jako součást proximálního operátoru. Princip algoritmu spočívá za daných podmínek v iterativní minimalizaci funkcionálů, kdy konverguje k optimu úlohy.

4.1 Omezený a neomezený tvar optimalizační úlohy

Výše zmíněný problém (3.3) je v tzv. omezeném tvaru. Abychom mohli řešit tento problém pomocí proximálního algoritmu, je třeba upravit problém tak, aby byl ve tvaru neomezeném odpovídajícím (4.1). To provedeme pomocí indikátorové funkce.

Definice 4.1. *Je-li C neprázdná množina $\mathbb{R}^n$, indikátorová funkce množiny C je*

$$\iota_C : \mathbf{x} \mapsto \begin{cases} 0, pro\ \mathbf{x} \in C, \\ \infty, pro\ \mathbf{x} \notin C. \end{cases} \tag{4.2}$$

Poté místo (3.3) získáme rovnici

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \iota_C(\mathbf{x}), \tag{4.3}$$

která už je v požadovaném neomezeném tvaru.

4.2 Proximální operátor

Definice 4.2. *Nechť funkce $f \in \Gamma_0(\mathbb{R}^N)$. Pro každé $\mathbf{x} \in \mathbb{R}^N$ má minimalizační problém*

$$\arg\min_{\mathbf{y} \in \mathbb{R}^N} f(\mathbf{y}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 \tag{4.4}$$

jednoznačné řešení, které označíme jako $\mathrm{prox}_f \mathbf{x}$. *Operátor* $\mathrm{prox}_f : \mathbb{R}^N \to \mathbb{R}^N$ *je proximální operátor funkce f.*

Proximální operátor má mnoho výhod, díky kterým je vhodný pro iterativní minimalizační algoritmy [6].

4.3 Douglas-Rachfordův algoritmus

Máme-li již definovanou obecnou podobu proximálního operátoru, můžeme přistoupit k samotnému výpočetnímu postupu optimalizační úlohy (4.1). Pro naši aplikaci bude nejvhodnější tzv. Douglas-Rachfordův algoritmus. Ten nám na rozdíl od jiných metod, jako je např. metoda dopředně-zpětného dělení, umožní řešení problému i v situaci, kdy není ani jeden z funkcionálů minimalizace funkcí diferencovatelnou.

Algoritmus připouští pro $\gamma \in]0, +\infty[$ minimálně jedno řešení, které je charakterizováno podmínkou

$$\begin{cases} \mathbf{x} = \mathrm{prox}_{\gamma f_2}\mathbf{y}, \\ \mathrm{prox}_{\gamma f_2}\mathbf{y} = \mathrm{prox}_{\gamma f_1}\left(2\mathrm{prox}_{\gamma f_2}\mathbf{y} - \mathbf{y}\right). \end{cases} \tag{4.5}$$

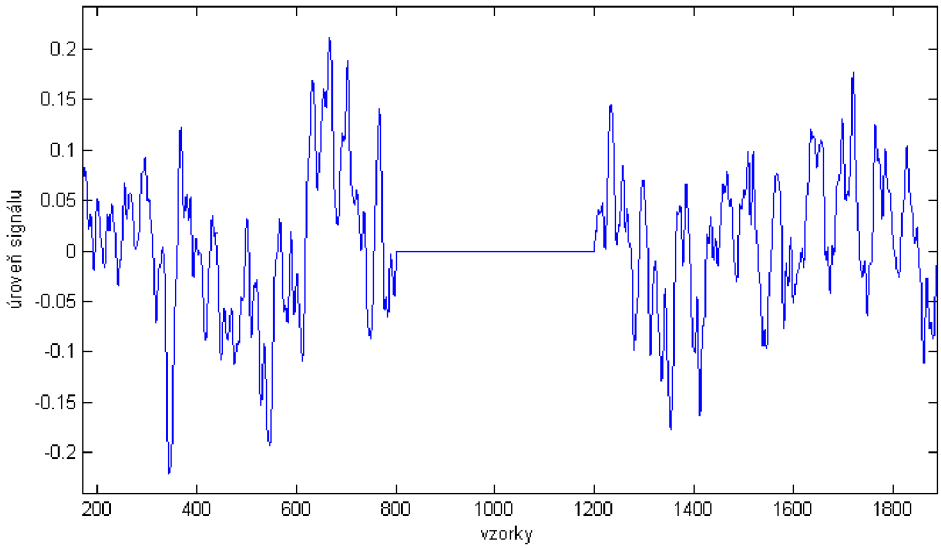
Máme-li počáteční vektor $\mathbf{y}_0 \in \mathbb{R}^N$ a zvolíme $\varepsilon \in]0, 1[$, pak bude algoritmus v každé iteraci $n = 0,1, \ldots$ provádět výpočet

$$\begin{aligned} \mathbf{x}_n &= \mathrm{prox}_{\gamma f_2}\mathbf{y}_n, \\ \mathbf{y}_{n+1} &= \mathbf{y}_n + \lambda_n\left(\mathrm{prox}_{\gamma f_1}(2\mathbf{x}_n - \mathbf{y}_n) - \mathbf{x}_n\right), \end{aligned} \tag{4.6}$$

kde $\lambda_n \in [\varepsilon, 2 - \varepsilon]$. Jak již bylo řečeno, po každé iteraci konverguje algoritmus k řešení problému (4.1).

5 FORMULACE PROBLÉMU, MOŽNÁ ŘEŠENÍ

Budeme mít k dispozici monofonní audiosignál $\mathbf{y} \in \mathbb{R}^N$, který však nebude kompletní. Několik jeho vzorků nám nebude známo. Budeme předpokládat, že chybějící hodnoty tvoří jeden celistvý úsek. V této kapitole si ukážeme několik možných algoritmů k dopočítání chybějících vzorků. Ukázka signálu s chybějícím úsekem je na obrázku 5.1.



Obr. 5.1: Příklad audiosignálu s chybějícími vzorky (nahrazeny nulami).

5.1 Některé dřívější metody restaurace audiosignálu

Jednou z nejjednodušších metod doplnění chybějících dat je zopakování posledních M vzorků.

$$\{y(l-M), y(l-M+1), \dots, y(l-1)\} \tag{5.1}$$

kde l je index prvního chybějícího vzorku. Při užití této metody často dochází k nesouladu na jednom nebo i obou koncích doplňovaného úseku [7].

Jiné dříve užívané metody doplňování chybějících úseků signálu byly založeny především na autoregresních modelech, případně na tzv. sinusoidálních modelech restaurace audiosignálu.

5.1.1 Autoregresní metoda

Jedná se o metodu, která předpokládá, že lze signál popsat parametrickým modelem, který je popsán koeficienty. Je založena na tom, že lze hodnotu pozorovaného signálu $\mathbf{y}$ v daném okamžiku předpovídat lineárně z předcházejících hodnot,

$$y_i = \sum_{j=1}^{k} a_j y_{i-j} + u_i, \tag{5.2}$$

kde a_j jsou tzv. autoregresní koeficienty, neboli váhy příslušné předcházejícím hodnotám, a u_i je náhodná chyba modelu (odchylka). Číslo k udává řád modelu, přičemž současná hodnota už nezávisí na hodnotách starších než y_{i-k}.

Pro vytvoření autoregresního modelu je potřeba mít kolem chybějící části signálu dostatečné množství vzorků, ze kterých model zkonstruujeme. Nyní si představíme metodu využívající známé vzorky z obou stran chybějící části. Místo jednoho autoregresního modelu tak budou použity dva modely, pro signál nalevo a napravo od chybějícího úseku.

Na obrázku 5.2 je znázornění signálu **y**, kde chybí vzorky

$$\{y(l), y(l+1), \dots, y(l+M-1)\}. \tag{5.3}$$

M je počet chybějících vzorků. N_L a N_R bude počet vzorků nalevo a napravo od chybějící části, které se budou podílet na vytvoření modelu pro obnovu signálu. Chybějící signál je možné odhadnout pouze z předcházejících vzorků, tzv. dopředná predikce, pouze z následujících vzorků, tzv. zpětná predikce, nebo ze vzorků z obou stran chybějící části, tzv. dopředno-zpětná predikce. Odhadovaný signál z předchozích vzorků $\hat{y}_L(i)$ potom je

$$\{y(l-1), y(l-2), \dots, y(l-N_L)\}. \tag{5.4}$$

Vzorky signálu odhadnuté z následujících vzorků signálu $\hat{y}_R(i)$ jsou

$$\{y(l+M), y(l+M+1), \dots, y(l+M+N_R-1)\}. \tag{5.5}$$

Využijeme-li k rekonstrukci dopředno-zpětné predikce, můžeme celkovou odhadovanou část signálu $\hat{y}_{LR}(i)$ vyjádřit součtem

$$\hat{y}_{LR}(i) = w_{L,M}(i-l)\hat{y}_L(i) + w_{R,M}(i-l)\hat{y}_R(i), l \leq i < l+M \tag{5.6}$$

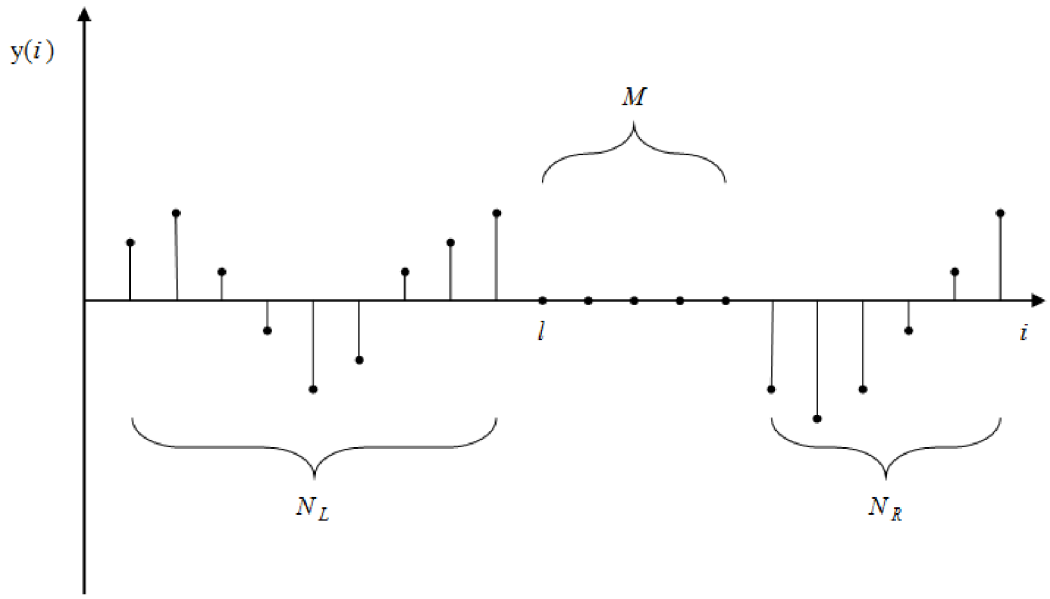
kde $w_{L,M}$ $w_{R,M}$ jsou váhovací funkce levého a pravého odhadu. Obvykle se jako váhovací funkce pro tyto účely využívá funkce raised-cosine.

Váhovací posloupnost levé strany pak vypadá následovně

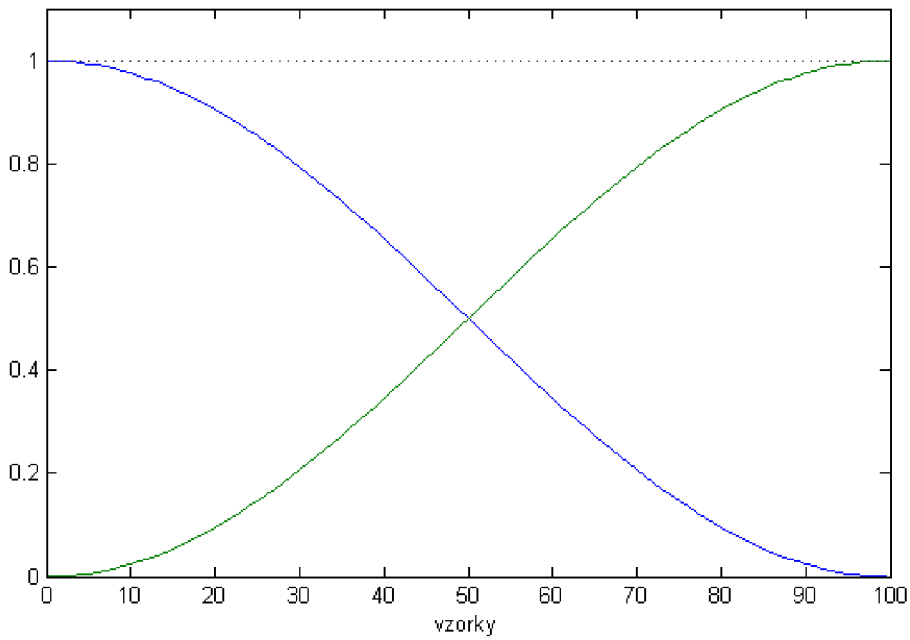
$$w_{L,M} = \begin{cases} \frac{1}{2}\left(1 - \cos\left(2\pi \frac{M+i}{2M}\right)\right), \text{pro } l \leq i < M, \\ 0, \text{jinak.} \end{cases} \tag{5.7}$$

Váhovací posloupnost pravé strany vypadá takto

$$w_{R,M} = \begin{cases} \frac{1}{2}\left(1 - \cos\left(2\pi \frac{i}{2M}\right)\right), \text{pro } 0 \leq i < M, \\ 0, \text{jinak.} \end{cases} \tag{5.8}$$



Obr. 5.2: Znázornění signálu s chybějícími vzorky M a vzorky N_L a N_R podílejícími se na autoregresním modelování.



Obr. 5.3: Váhové posloupnosti raised-cosine pro levou i pravou stranu. Jejich součet dává jedničku.

5.1.2 Sinusoidální model

Další z metod rekonstrukce signálu je metoda založená na sinusoidálním modelování signálu. Spočívá v identifikaci a separaci jednotlivých tzv. sinusoidálních komponent. Tyto komponenty jsou v čase definovány amplitudou, frekvencí a fází. Pro rozklad signálu na jednotlivé komponenty se většinou využívají metody založené na Fourierově transformaci. Výsledný rekonstruovaný signál však není modelován přímo pomocí těchto

komponent, nýbrž pomocí hyperparametrů komponent. Poměrně dobrých výsledků tato metoda dosahuje pro stacionární pseudoperiodické signály [8].

5.2 Moderní metoda restaurace založená na řídkosti signálů

Nyní si ukážeme metodu doplnění chybějících vzorků audiosignálu za pomoci řídkosti. Ve vhodné časově-kmitočtové reprezentaci lze většinu přirozených audiosignálů považovat za řídké [1].

$$\mathbf{y} \approx \mathbf{D}\mathbf{x} \text{ a } \|\mathbf{x}\|_0 \ll N, \tag{5.9}$$

kde $\mathbf{D}$ je slovník, ve kterém jsou atomy daného slovníku v jednotlivých sloupcích. N je počet vzorků signálu (jeho délka). Na obrázku 5.4 je znázorněna syntéza audiosignálu plným slovníkem a také situace s neúplným signálem $\mathbf{y}^{\mathrm{r}}$ a tomu odpovídajícím redukovaným slovníkem $\mathbf{D}^{\mathrm{r}}$. Zatímco rekonstrukce signálu syntézou pomocí plného slovníku $\hat{\mathbf{y}} = \mathbf{D}\hat{\mathbf{x}}$ je triviální záležitostí, podstatně náročnější je odhad souřadnic $\hat{\mathbf{x}}$. Formálně odhadujeme $\hat{\mathbf{x}}$ pomocí redukovaného slovníku $\mathbf{D}^{\mathrm{r}}$ a vzorků $\mathbf{y}^{\mathrm{r}}$, které známe

$$\hat{\mathbf{x}} = f(\mathbf{y}^{\mathrm{r}}, \mathbf{D}^{\mathrm{r}}). \tag{5.10}$$

Podstatné tedy je, jaký způsobem odhadneme f. Jeden z možných způsobů je takovýto

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\mathbf{x}, \tag{5.11}$$

případně

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \|\mathbf{y}^{\mathrm{r}} - \mathbf{D}^{\mathrm{r}}\mathbf{x}\|_2 \leq \delta. \tag{5.12}$$

Jak jsme si již řekli v kapitole 3, tyto úlohy jsou složitosti třídy NP. Lze je však řešit aproximativními metodami, jako např. hladovými algoritmy nebo jinými. My při jejich řešení přistoupíme k ℓ_1-relaxaci, budeme tedy řešit konvexní optimalizační úlohu. Relaxovaný problém (5.11) potom vypadá následovně

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\mathbf{x}. \tag{5.13}$$

Abychom mohli tento problém řešit pomocí některého z proximálních algoritmů, musíme jej upravit dle postupu uvedeného v části 4.1 na neomezený tvar odpovídající

$$\arg\min_{x} f_1(x) + f_2(x). \tag{5.14}$$

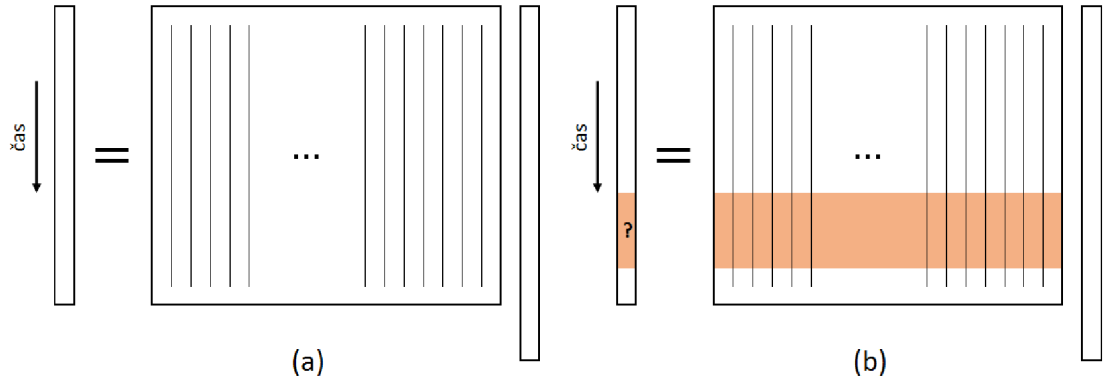
Výsledný neomezený tvar pak vypadá takto

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \iota_{\{\mathbf{c}:\mathbf{y}^{\mathrm{r}}=\mathbf{D}^{\mathrm{r}}\mathbf{c}\}}(\mathbf{x}), \tag{5.15}$$

kde pro indikátorovou funkci $\iota_{\{\mathbf{y}^{\mathrm{r}},\mathbf{D}^{\mathrm{r}}\}}(\mathbf{x})$ platí

$$\iota_{\{\mathbf{c}:\mathbf{y}^{\mathrm{r}}=\mathbf{D}^{\mathrm{r}}\mathbf{c}\}}(\mathbf{x}) = \begin{cases} 0, \text{ pokud } \mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\mathbf{x}, \\ \infty, \text{ pokud } \mathbf{y}^{\mathrm{r}} \neq \mathbf{D}^{\mathrm{r}}\mathbf{x}. \end{cases} \tag{5.16}$$

Jelikož není ani jedna z funkcí v (5.15) diferencovatelná, hodí se pro náš výpočet Douglas-Rachfordův algoritmus definovaný v části 4.3.



Obr. 5.4: (a) syntéza signálu plným slovníkem $\hat{\mathbf{y}} = \mathbf{D}\hat{\mathbf{x}}$, (b) syntéza za užití redukovaného slovníku $\mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\hat{\mathbf{x}}$.

5.2.1 Konkrétní podoba Douglas-Rachfordova algoritmu

Při aplikaci Douglas-Rachfordova algoritmu na úlohu (5.15) získá algoritmus následující podobu:

Algoritmus 5.1. *Počátečním vektorem je* $\mathbf{x}_0 = (\mathbf{D}^{\mathrm{r}})^{\mathrm{T}}\mathbf{y}^{\mathrm{r}}$.

Pro zvolené $\varepsilon \in]0, 1[$ *a* $\gamma > 0$ *provede algoritmus v každé iteraci* $n = 0,1, \ldots$ *s parametrem* $\lambda_n \in [\varepsilon, 2 - \varepsilon]$ *výpočty*

$$\begin{aligned} \mathbf{y}_n &= \mathrm{prox}_{\gamma f_2}\mathbf{x}_n, \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \lambda_n\big(\mathrm{prox}_{\gamma f_1}(2\mathbf{y}_n - \mathbf{x}_n) - \mathbf{y}_n\big), \end{aligned} \tag{5.17}$$

Iterace ukončíme, jakmile je splněno nějaké vhodné kritérium.

Dle definice 4.2 si vyjádříme podobu proximálních operátorů funkcí $f_1(\mathbf{x})$ a $f_2(\mathbf{x})$. Proximální operátor funkce $f_2(\mathbf{x}) = \iota_{\{\mathbf{c}:\mathbf{y}^{\mathrm{r}}=\mathbf{D}^{\mathrm{r}}\mathbf{c}\}}(\mathbf{x})$ vypadá takto

$$\begin{aligned} \mathbf{y}_n = \mathrm{prox}_{\gamma f_2}\mathbf{x}_n &= \arg\min_{\mathbf{x}} \ \gamma\iota_{\{\mathbf{c}:\mathbf{y}^{\mathrm{r}}=\mathbf{D}^{\mathrm{r}}\mathbf{c}\}}(\mathbf{x}_n) + \frac{1}{2}\|\mathbf{y}_n - \mathbf{x}_n\|_2^2 = \\ &= \arg\min_{\mathbf{x}} \ \iota_{\{\mathbf{c}:\mathbf{y}^{\mathrm{r}}=\mathbf{D}^{\mathrm{r}}\mathbf{c}\}}(\mathbf{x}_n) + \frac{1}{2}\|\mathbf{y}_n - \mathbf{x}_n\|_2^2 = \\ &= \mathrm{proj}_{\{\mathbf{c}:\mathbf{y}^{\mathrm{r}}=\mathbf{D}^{\mathrm{r}}\mathbf{c}\}}(\mathbf{x}_n). \end{aligned} \tag{5.18}$$

Jde o ortogonální projekci na afinní prostor. Víme-li, že platí tvrzení

$$\mathrm{proj}_{\{\mathbf{c}:\mathbf{A}\mathbf{c}=0\}}(\mathbf{z}) = \mathbf{z} - \mathbf{A}^{+}\mathbf{A}\mathbf{z} \tag{5.19}$$

[9], pak bude výsledný tvar našeho proximálního operátoru takovýto

$$\mathbf{y}_n = \mathrm{prox}_{\gamma f_2}\mathbf{x}_n = \mathbf{x}_n - (\mathbf{D}^{\mathrm{r}})^{+}(\mathbf{D}^{\mathrm{r}}\mathbf{x}_n - \mathbf{y}^{\mathrm{r}}), \tag{5.20}$$

kde $\mathbf{D}^{\mathrm{r}} = \mathbf{I}^{\mathrm{r}}\mathbf{D}$. Matice $\mathbf{I}^{\mathrm{r}}$ je neúplnou jednotkovou maticí, která nám doplní nuly do příslušných řádků slovníku $\mathbf{D}$ a vytvoří tak redukovaný slovník $\mathbf{D}^{\mathrm{r}}$. Navíc, je-li $\mathbf{D}$ tzv. těsný Parsevalův frame ($\mathbf{D}\mathbf{D}^{*} = \mathbf{I}$), můžeme tvrdit, že $(\mathbf{D}^{\mathrm{r}})^{+} = \mathbf{D}^{*}(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}$ a daný proximální operátor se nám zjednoduší na tvar

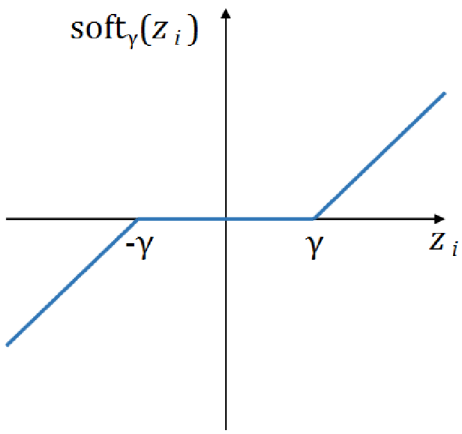
$$\mathbf{y}_n = \mathrm{prox}_{\gamma f_2}\mathbf{x}_n = \mathbf{x}_n - \mathbf{D}^{*}(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}\mathbf{I}^{\mathrm{r}}\mathbf{D}\mathbf{x}_n + \mathbf{D}^{*}(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}\mathbf{y}^{\mathrm{r}}. \tag{5.21}$$

Přičemž operand $\mathbf{D}^*(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}\mathbf{y}^{\mathrm{r}}$ si lze předpočítat. $\mathbf{D}^*(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}$ představuje analýzu signálu, $\mathbf{I}^{\mathrm{r}}\mathbf{D}$ představuje syntézu.

Obdobně získáme podobu proximálního operátoru funkce $f_1(\mathbf{x}) = \|\mathbf{x}\|_1$,

$$\begin{aligned}\mathrm{prox}_{\gamma f_1}(2\mathbf{y}_n - \mathbf{x}_n) &= \arg\min_{\mathbf{x}} \ \gamma\|\mathbf{x}_n\|_1 + \frac{1}{2}\|\mathbf{y}_n - (2\mathbf{y}_n - \mathbf{x}_n)\|_2^2 = \\ &= \arg\min_{\mathbf{x}} \ \|\gamma\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{y}_n - (2\mathbf{y}_n - \mathbf{x}_n)\|_2^2 = \\ &= \mathrm{soft}_\gamma(2\mathbf{y}_n - \mathbf{x}_n).\end{aligned} \tag{5.22}$$

Jedná se o tzv. měkké prahování (soft thresholding) s prahem γ, kdy operátor prahuje každou složku vektoru $(2\mathbf{y}_n - \mathbf{x}_n)$ zvlášť. Pro lepší představu je na obrázku 5.5 znázorněno, jak měkké prahování funguje.



Obr. 5.5: Ukázka měkkého prahování vektoru **z**. Je-li velikost z_i větší než práh γ, pak je z_i o tuto hodnotu snížena. Naopak pokud je z_i menší než práh γ, je tento koeficient vynulován.

5.2.2 Návrh vylepšení metody restaurace založené na řídkosti signálů

Využíváme-li k odhadu řídkého vektoru koeficientů $\hat{\mathbf{x}}$ při syntéze $\mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\mathbf{x}$ redukovaný slovník $\mathbf{D}^{\mathrm{r}}$, jsou vlivem redukce některé jeho atomy zastoupeny s menší energií, než jak by tomu bylo při užití plného slovníku $\mathbf{D}$. Návrhem tedy je pokusit se tyto změny způsobené redukcí slovníku kompenzovat. Úlohu, kterou bychom potom řešili, můžeme formulovat takto

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\mathbf{W}\mathbf{x}, \tag{5.23}$$

kde $\mathbf{W}$ je určitou váhovací maticí, jejímž úkolem je kompenzovat pokles energie redukovaných prvků slovníku. Tato váhovací matice je čtvercovou diagonální maticí s nenulovými a nezápornými prvky v diagonále.

Jako nejjednodušší způsob řešení tohoto nového problému se jeví užití substituce $\mathbf{c} = \mathbf{W}\mathbf{x}$, po jejímž zavedení dostaneme danou rovnici ve tvaru

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{W}^{-1}\mathbf{c}\|_0 \text{ vzhledem k } \mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\mathbf{c}, \tag{5.24}$$

přičemž hledaný řídký vektor koeficientů je přímo odhad $\hat{\mathbf{c}}$. Abychom byli schopni tuto rovnici řešit, musíme opět přistoupit k ℓ_1-relaxaci. Dostaneme tak

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{W}^{-1}\mathbf{c}\|_1 \text{ vzhledem k } \mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\mathbf{c}, \tag{5.25}$$

kdy znovu užijeme indikátorové funkce pro převedení dané rovnice na neomezený tvar tak, aby odpovídala tvaru

$$\arg\min_{c} f_1(\mathbf{c}) + f_2(\mathbf{c}). \tag{5.26}$$

Indikátorová funkce bude vypadat takto

$$\iota_{\{\mathbf{c}:\mathbf{y}^{\mathrm{r}}=\mathbf{D}^{\mathrm{r}}\mathbf{c}\}}(\mathbf{c}) = \begin{cases} 0, \text{ pokud } \mathbf{y}^{\mathrm{r}} = \mathbf{D}^{\mathrm{r}}\mathbf{c}, \\ \infty, \text{ pokud } \mathbf{y}^{\mathrm{r}} \neq \mathbf{D}^{\mathrm{r}}\mathbf{c}. \end{cases} \tag{5.27}$$

Neomezený tvar rovnice (5.25) pak vypadá následovně

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \|\mathbf{W}^{-1}\mathbf{c}\|_1 + \iota_{\{\mathbf{c}:\mathbf{y}^{\mathrm{r}}=\mathbf{D}^{\mathrm{r}}\mathbf{c}\}}(\mathbf{c}). \tag{5.28}$$

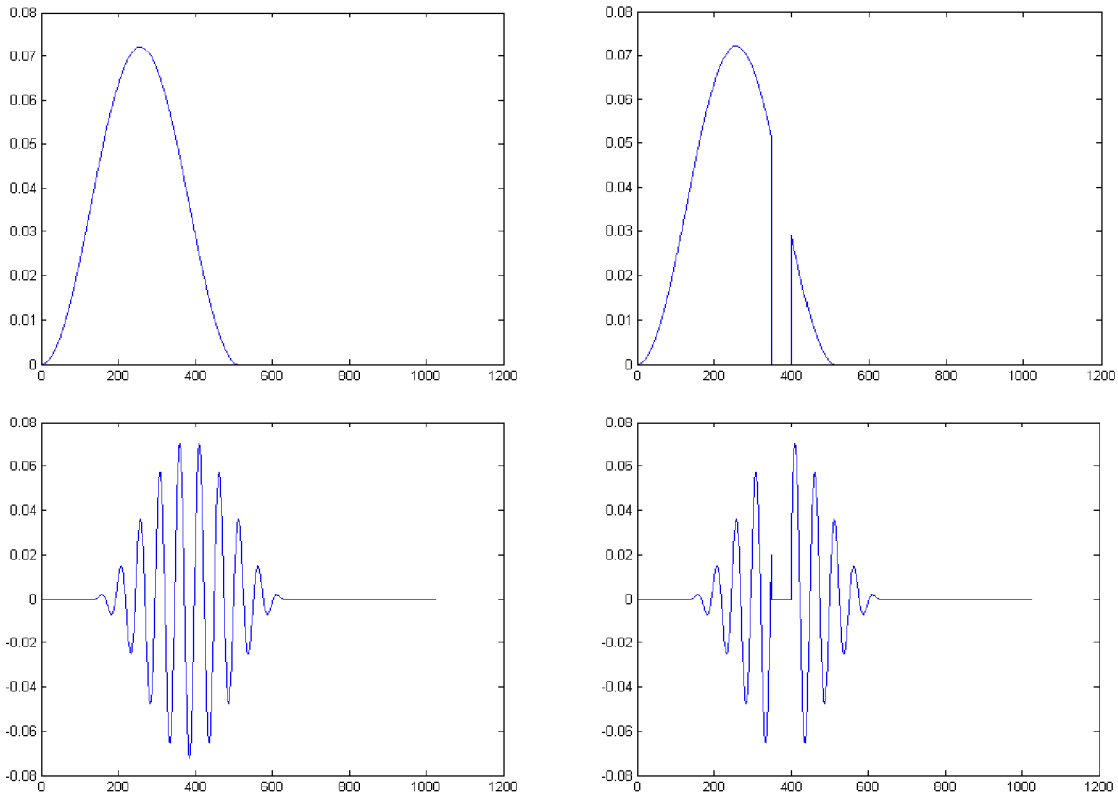
Srovnáme-li nyní podobu rovnice (5.28) s rovnicí (5.15), vidíme, že se příliš neliší. Opět můžeme k jejímu řešení využít Douglas-Rachfordova algoritmu. Jak lze také vidět, ortogonální projekce zůstane stejná jako v předchozím případě. Můžeme tak okamžitě napsat výsledný tvar proximálního operátoru $\mathrm{prox}_{\gamma f_2}\mathbf{c}_n$, který vypadá takto

$$\mathbf{y}_n = \mathrm{prox}_{\gamma f_2}\mathbf{c}_n = \mathbf{c}_n - \mathbf{D}^*(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}\mathbf{I}^{\mathrm{r}}\mathbf{D}\mathbf{c}_n + \mathbf{D}^*(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}\mathbf{y}^{\mathrm{r}}. \tag{5.29}$$

Jediné, co se nám změní, tak je tvar proximálního operátoru prvního funkcionálu, ten má nyní podobu $f_1(\mathbf{c}) = \|\mathbf{W}^{-1}\mathbf{c}\|_1$. Proximální operátor $\mathrm{prox}_{\gamma f_1}$ tak bude vypadat následovně

$$\begin{aligned} \mathrm{prox}_{\gamma f_1}(2\mathbf{y}_n - \mathbf{c}_n) &= \arg\min_{\mathbf{c}} \ \gamma\|\mathbf{W}^{-1}\mathbf{c}_n\|_1 + \frac{1}{2}\|\mathbf{y}_n - (2\mathbf{y}_n - \mathbf{c}_n)\|_2^2 = \\ &= \arg\min_{\mathbf{c}} \ \|\gamma\mathbf{W}^{-1}\mathbf{c}_n\|_1 + \frac{1}{2}\|\mathbf{y}_n - (2\mathbf{y}_n - \mathbf{c}_n)\|_2^2 = \\ &= \mathrm{soft}_{\gamma\mathrm{diag}(\mathbf{W}^{-1})}(2\mathbf{y}_n - \mathbf{c}_n). \end{aligned} \tag{5.30}$$

Stále se jedná o měkké prahování, avšak hodnoty prahu se budou měnit pro každou složku vektoru $(2\mathbf{y}_n - \mathbf{c}_n)$.



Obr. 5.6: Ukázka atomů úplného a redukovaného slovníku. Vlevo jsou atomy neredukované. Vpravo jsou atomy redukované. Zobrazena je pouze reálná část atomů.

6 IMPLEMENTACE ALGORITMU V PROSTŘEDÍ MATLAB

K realizaci navrženého algoritmu pro rekonstrukci audiosignálu využijeme program MATLAB. Důležitým doplňkem pro nás bude toolbox LTFAT (The Large Time/Frequency Analysis Toolbox) vyvíjený ve spolupráci univerzit CAHR (Technical University of Denmark), ARI (Austrian Academy of Sciences) a LATP (Universite de Provence). Jedná se o toolbox pro práci s časově-frekvenční analýzou a syntézou. Poskytuje velké množství lineárních transformací včetně Gaborovy a waveletové trasformace spolu s konstrukcí okénkových funkcí a manipulací s koeficienty objektů daných transformací [10]. Důležité pro nás je především to, že jsme tak schopni vytvořit Gaborův fram. Toolbox umožňuje vytvoření matice Gaborových koeficientů. Z hlediska výpočetní náročnosti je práce s maticemi velmi nevýhodná. Při počítání s maticemi trvá výpočet poměrně dlouho. Proto je pro nás výhodnější nepracovat s framem v podobě matice, ale v podobě objektu dané transformace. K analýze a syntéze signálu potom využíváme funkce `frana.m` a `frsyn.m`, které nám LTFAT nabízí.

Frame vytvoříme pomocí příkazu:

```
frame('dgtreal', g, a, M)
```

Vytvořený frame bude objektem diskrétní Gaborovy transformace pro reálné signály (proto `'dgtreal'`) s oknem `g`, posunem okna `a` a počtem kanálů `M`. V našem případě vypadá příkaz takto:

```
F = frame('dgtreal',{'hann',velOkna},posunOkna,2048);
```

Typ okna `'hann'` představuje okno Hannovo s šířkou zadanou proměnnou `velOkna`. Tomu, jaké hodnoty velikosti a posunu okna zvolit se budeme věnovat později. Abychom na náš problém mohli aplikovat algoritmus navržený v bodě 5.2.1, musí být námi vytvořený frame tzv. 1-těsný. Tuto podmínku snadno dodržíme užitím příkazu:

```
F = frametight(F);
```

Ten vrací těsný frame vytvořený z framu zadaného v jeho parametru. Byl-li frame těsný už předtím, provede pouze jeho normalizaci. Pomocí následujícího příkazu si zobrazíme hranice námi vytvořeného framu:

```
[A,B] = framebounds(F)
```

Měli bychom dostat výsledek $A = B = 1$, čímž jsme si potvrdili, že se jedná o těsný Parsevalův frame.

Průběh Douglas-Rachfordova algoritmu je naprogramován ve funkci `DR.m`, realizace kódu je patrná z výpisu 6.1. Vstupními parametry funkce jsou: námi vytvořený frame Gaborovy transformace `F`, počáteční vektor `x0`, γ, λ, maximální počet iterací `maxIter`, první chybějící vzorek `zacDir`, poslední chybějící vzorek `konDir` a tolerance pro ukončení iterací. Výstupem je hledaný řídký vektor koeficientů, vektor hodnot objektivní funkce a počet vykonaných iterací.

Výpis kódu 6.1: DR.m – Douglas-Rachfordův algoritmus

```
function [xn,historie,i] = DR(F,x0,gama,lambda,maxIter,zacDir,
konDir,tolerance)

xn = x0;
historie = zeros(maxIter,1); %vektor pro ukládání hodnot
objektivní funkce
i=1;

while 1
    yn = proxf2(F,xn,x0,zacDir,konDir);
    historie(i) = norm(yn,1); %výpočet hodnoty objektivní
funkce pro danou iteraci
    xn = xn + lambda*(proxf1(yn,xn,gama) - yn);
    if i>1 && (abs(historie(i-1)-historie(i)))/historie(i-
1)<tolerance
        break %ukončení iterací pokud je rozdíl posledních dvou
hodnot objektivní funkce vzhledem k poslední hodnotě menší než
daná tolerance
    end
    if i==maxIter
        break %ukončení algoritmu pokud je dosaženo
maximálního počtu iterací
    end
    i=i+1;
end
end
```

Výpočet ortogonální projekce, tedy proximálního operátoru

$$\mathrm{prox}_{\gamma f_2}\mathbf{x}_n = \mathbf{x}_n - \mathbf{D}^*(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}\mathbf{I}^{\mathrm{r}}\mathbf{D}\mathbf{x}_n + \mathbf{D}^*(\mathbf{I}^{\mathrm{r}})^{\mathrm{T}}\mathbf{y}^{\mathrm{r}}, \tag{6.1}$$

je v rámci funkce proxf2.m realizován následovně:

```
function [ out ] = proxf2(F,xn,x0,zacDir,konDir)
out = xn - frana(F,doplniNuly(frsyn(F,xn),zacDir:konDir)) + x0;
end
```

Funkce doplniNuly.m zastupuje ve výpočtu matici $\mathbf{I}^{\mathrm{r}}$. Slouží tedy k doplnění nul na zadané pozice.

Výpočet měkkého prahování, tedy proximálního operátoru

$$\mathrm{prox}_{\gamma f_1}(2\mathbf{y}_n - \mathbf{x}_n) = \mathrm{soft}_{\gamma}(2\mathbf{y}_n - \mathbf{x}_n), \tag{6.2}$$

je naprogramován ve funkci proxf1.m takto:

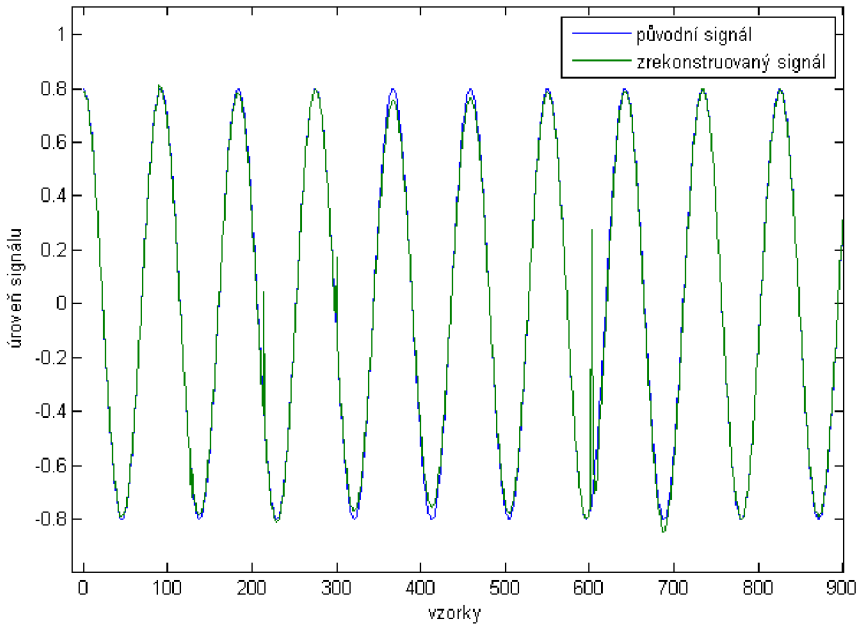
```
function [ out ] = proxf1(yn,xn,gama)
out = mprahovani(2*yn - xn,gama);
end
```

Realizace funkce `mprahovani.m` je následovná:

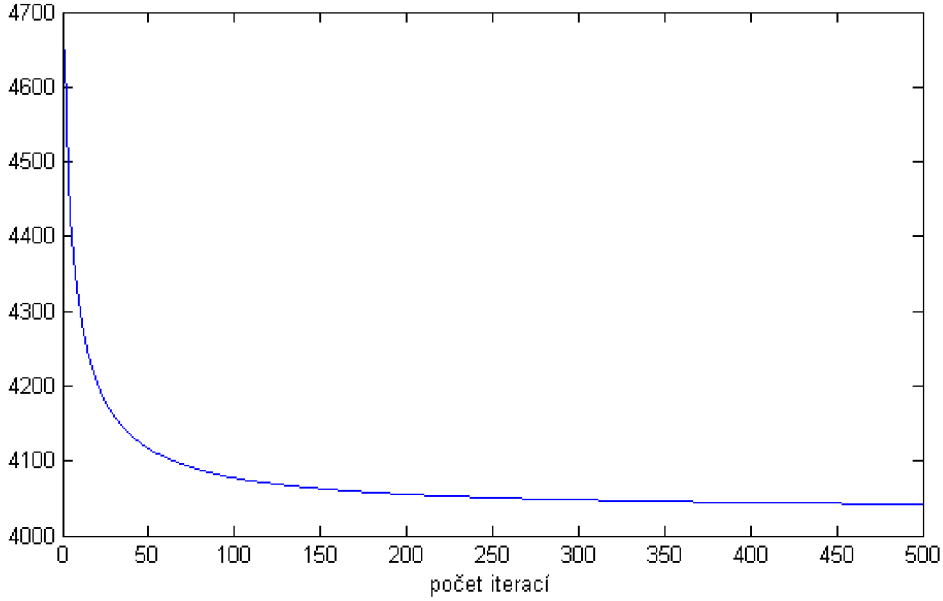
```
function [ out ] = mprahovani( x, gama )
out = sign(x).*max(abs(x)-gama,0);
end
```

Po prvotním sestavení programu a aplikaci na sinusový signál s uměle vytvořenou chybějící částí jsme dostali až nečekaně nedokonalé výsledky, viz obr. 6.1. Problém byl především v tom, že zrekonstruovaný signál neodpovídal originálnímu ani v úsecích mimo chybějící část. Ty by však rekonstrukcí neměly být změněny. Lepších výsledku jsme dosáhli po několika krátkých úvahách:

- Pro co nejúspěšnější rekonstrukci je důležité, aby po obou stranách chybějící části byl dostatečně dlouhý úsek původního signálu.
- Dalším důležitým parametrem je počet iterací Douglas-Rachfordova algoritmu. Aby byla rekonstrukce co nejúspěšnější, je třeba vykonat dostatečné množství iterací. Dobrou představu o tom, kolik iterací je pro danou úlohu vhodné vykonat nám poskytuje graf historie objektivní funkce. Příklad takového grafu je na obrázku 6.2. Z vyobrazeného průběhu je však také patrné, že od určitého počtu iterací již dochází jen k malým změnám funkce. Včasné ukončení iterací nám tak může ušetřit mnoho výpočetního času i při dosažení téměř stejné úspěšnosti rekonstrukce, jaké bychom dosáhli při větším počtu iterací.
- Podstatného zlepšení jsme také dosáhli, končil-li algoritmus krokem s projekcí. V průběhu iterací algoritmu se měkké prahování stará především o hledání požadovaného řídkého řešení. To má však za následek nepřesnosti rekonstrukce v oblastech i mimo chybějící úsek. Těchto nepřesností se zbavíme právě užitím projekce. Proto bylo logické zařadit za poslední krok algoritmu, kterým bylo dle návrhu v části 5.2.1 měkké prahování, ještě jeden krok s ortogonální projekcí.



Obr. 6.1: Nedokonalá rekonstrukce při prvotní aplikaci algoritmu (chybějící část od 300. do 600. vzorku).



Obr. 6.2: Graf objektivní funkce v průběhu iterací Douglas-Rachfordova algoritmu.

6.1 Implementace navrženého vylepšení

V části 5.2.2 jsme navrhli vylepšení založené na kompenzaci poklesu energie redukovaných atomů slovníku za pomoci určité váhovací matice. Pro výpočet potřebných váhových koeficientů jsme proto vytvořili funkci `vypVahKoef.m` jejímž výstupem je vektor těchto koeficientů `w`. Už také víme, že navržené vylepšení se v průběhu algoritmu dotkne pouze kroku s měkkým prahování. Stačí nám tedy upravit kód funkce `mprahovani.m`. Její nová podoba je následující:

```
function [ out ] = mprahovaniSVyl( x, gama, w )
gama = gama*w;
out = sign(x).*max(abs(x)-gama(1:length(x)),0);
end
```

7 TESTOVÁNÍ ALGORITMU, HODNOCENÍ REKONSTRUKCE

7.1 Metoda objektivního hodnocení úspěšnosti rekonstrukce

K hodnocení úspěšnosti rekonstrukce použijeme metodu SNR, neboli odstup signálu od šumu. Ta nám udává poměr výkonu užitečného signálu P_s ku výkonu signálu neužitečného P_n [11]. V našem případě je užitečným signálem originální signál bez chybějící části. Neužitečným signálem je rozdíl originálního signálu a signálu zrekonstruovaného. SNR se udává v decibelech. Rekonstrukce je tím kvalitnější, čí větší hodnotu dostaneme. Obecný vzorec pro výpočet SNR vypadá následovně

$$SNR = 10\log\frac{P_s}{P_n} = 10\log\frac{\sum_{m=0}^{l-1} s^2[m]}{\sum_{m=0}^{l-1} n^2[m]}, \tag{7.1}$$

kde l je délka signálu, $s[m]$ je vzorek signálu a $n[m]$ je vzorek šumu [12]. S užitím definice 1.2. si tuto rovnici může pro náš případ přepsat do tvaru

$$SNR = 10\log\frac{\|\mathbf{y}\|_2^2}{\|\mathbf{y}-\hat{\mathbf{y}}\|_2^2} = 20\log\frac{\|\mathbf{y}\|_2}{\|\mathbf{y}-\hat{\mathbf{y}}\|_2}. \tag{7.2}$$

Dále víme, že díky projekci bude zrekonstruovaný signál mimo chybějící úsek stejný jako signál původní. Stačí nám tak počítat SNR pouze v oblasti chybějící části. Potom

$$SNR_{\mathrm{d}} = 20\log\frac{\|\mathbf{y}_{\mathrm{d}}\|_2}{\|\mathbf{y}_{\mathrm{d}}-\hat{\mathbf{y}}_{\mathrm{d}}\|_2}, \tag{7.3}$$

kde $\mathbf{y}_{\mathrm{d}}$ a $\hat{\mathbf{y}}_{\mathrm{d}}$ jsou vzorky pouze z oblasti chybějícího úseku.

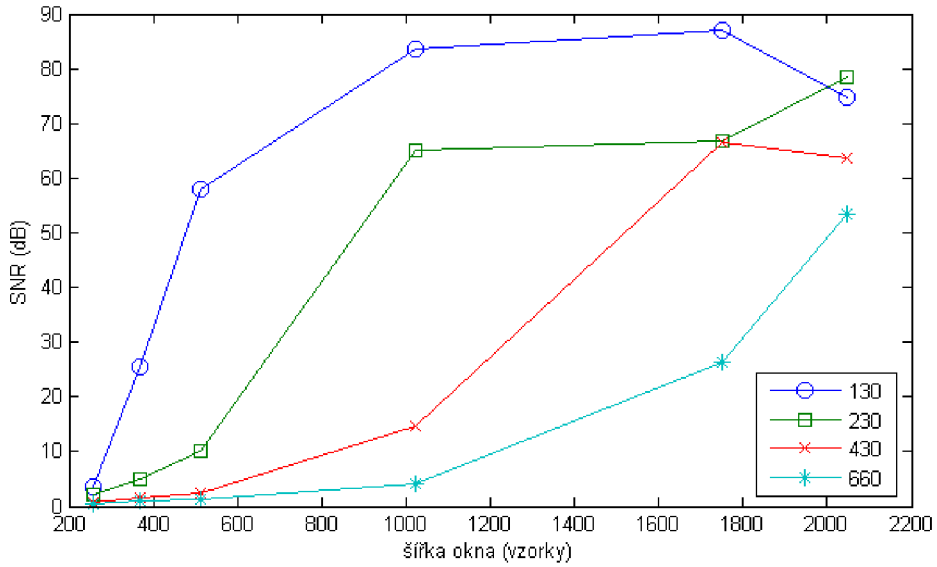
7.2 Vliv parametrů slovníku na rekonstrukci

Zajímalo nás, jaké zvolit hodnoty parametrů použitého slovníku, abychom dosáhli co nejlepší rekonstrukce. Počet kanálů jsme si stanovili 2048. Zkoumanými parametry pak byla šířka okna a jeho posun. Pozorování jsme provedli na sinusovém signálu a na krátkém vzorku signálu složeného, kterým byl úryvek ze skladby Hanse Zimmera Mountains. Tyto vzorky signálů byly vybrány proto, aby reprezentovali výsledky pro spektrálně zcela odlišné případy. Spektrum sinusového signálu je nejjednodušší jaké může být, oproti tomu spektrum části dané skladby je poměrně složité. Pro oba signály jsme vytvořili 4 případy s různou velikostí chybějící části. Délky chybějících úseků byly 130, 230, 430 a 660 vzorků. Úspěšnost rekonstrukce byla porovnávána pomocí SNR.

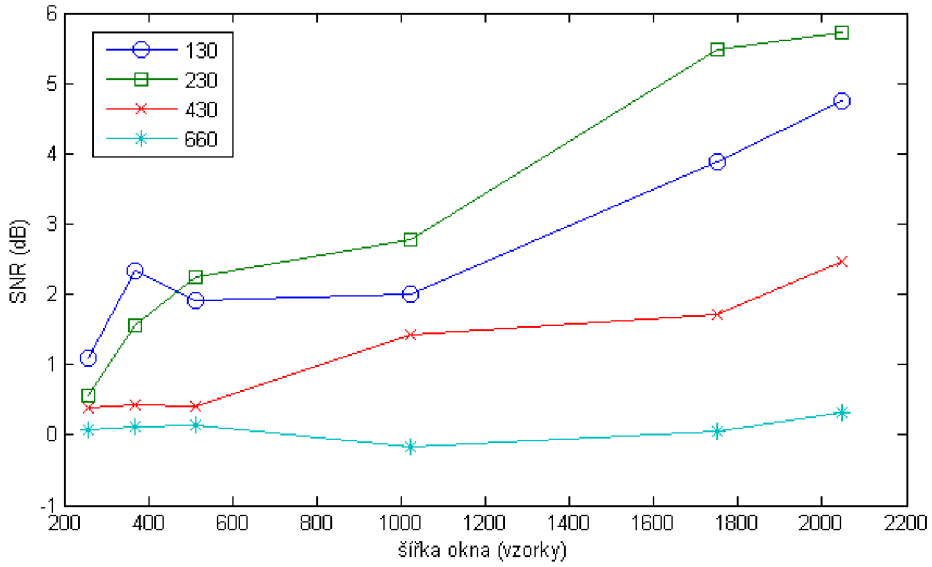
7.2.1 Vliv šířky okna

Pro každý již výše zmíněný případ velikosti chybějící části jsme spočítali SNR rekonstrukce při užití oken o velikostech 256, 370, 512, 1024, 1750 a 2048. Posun okna byl 128. Výsledky výpočtů pro sinusový signál jsou v grafu na obrázku 7.1. Výsledky pro signál složený jsou v grafu na obrázku 7.2.

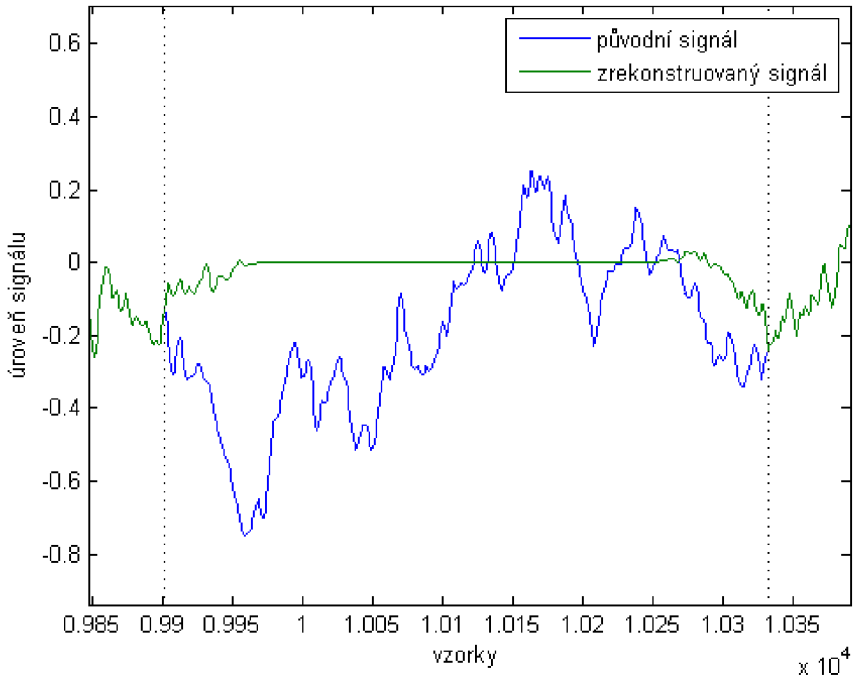
Předpokladem bylo, že čím širší okno bude, tím bude rekonstrukce úspěšnější. To se také potvrdilo. Z grafů je navíc patrné, že pro malé chybějící úseky dosáhneme vysoké úspěšnosti i s menšími šířkami oken. Rozhodně by však šířka okna neměla být menší než délka chybějící části signálu. Rekonstrukce je pak značně nedokonalá. Je to z toho důvodu, že okno, které se během výpočtu dostane translací do středu chybějící části, nemá k dispozici žádné vzorky signálu původního a vychází tedy pouze z nulových prvků chybějícího úseku. Ukázka takové nedokonalé rekonstrukce je na obrázku 7.3. Obecně by se dalo říct, že bychom měli usilovat o co možná největší šířku okna.



Obr. 7.1: Graf znázorňující úspěšnost rekonstrukce sinusového signálu v závislosti na šířce okna; modrá křivka pro chybějící úsek délky 130 vzorků, zelená pro 230, červená pro 430 a světle modrá pro 660 vzorků.



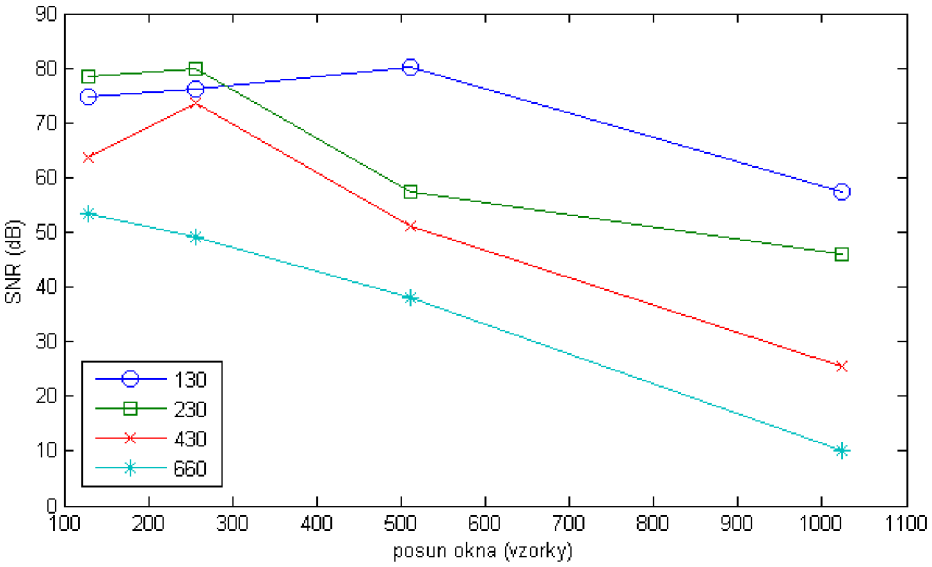
Obr. 7.2: Graf znázorňující úspěšnost rekonstrukce složeného signálu v závislosti na šířce okna; modrá křivka pro chybějící úsek délky 130 vzorků, zelená pro 230, červená pro 430 a světle modrá pro 660 vzorků.



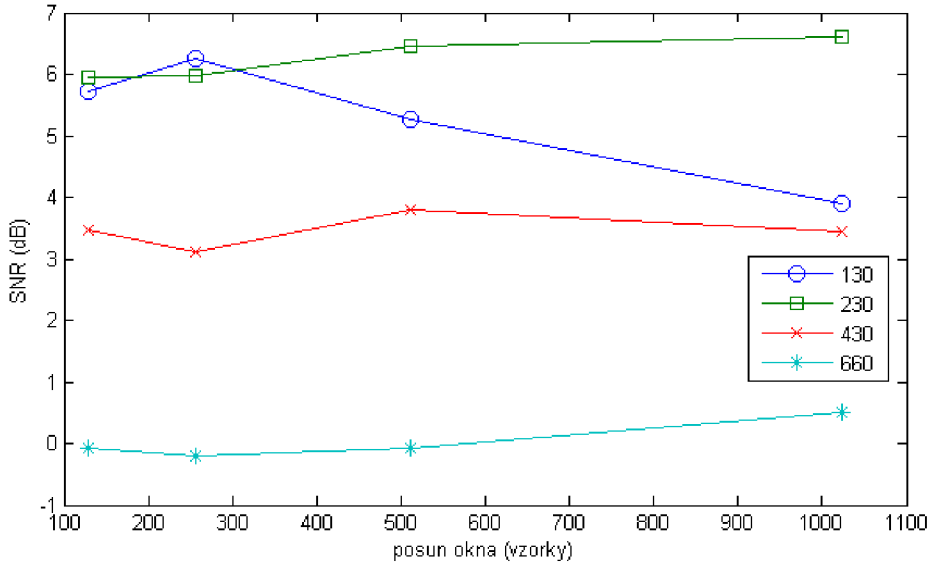
Obr. 7.3: Příklad rekonstrukce signálu s užitím příliš krátkého okna. Velikost chybějící části je 430 vzorků, šířka okna je 256.

7.2.2 Vliv posunu okna

Nyní jsme počítali a zobrazovali SNR pro různé velikosti posunu okna, ty byly 128, 256, 512 a 1024. Velikost okna byla 2048. Výsledky pro sinusový signál jsou v grafu na obrázku 7.4. Výsledky pro signál složený jsou na obrázku 7.5.



Obr. 7.4: Graf znázorňující úspěšnost rekonstrukce sinusového signálu v závislosti na velikosti posunu okna; modrá křivka pro chybějící úsek délky 130 vzorků, zelená pro 230, červená pro 430 a světle modrá pro 660 vzorků.

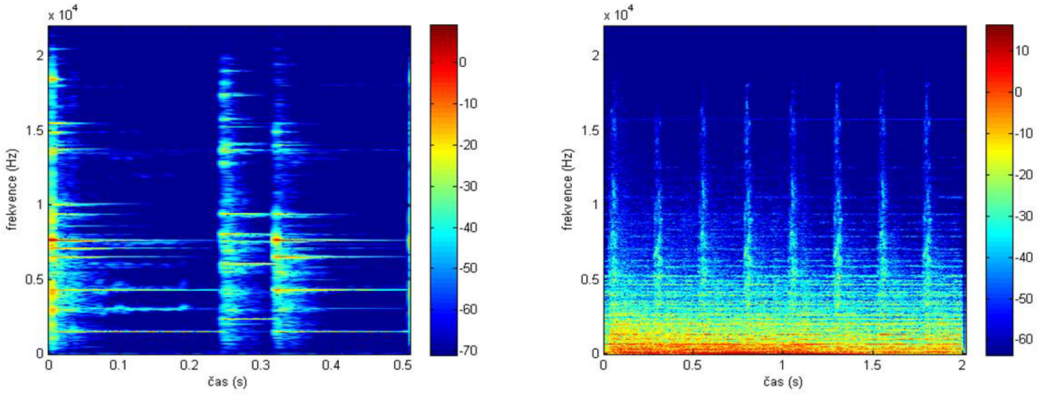


Obr. 7.5: Graf znázorňující úspěšnost rekonstrukce složeného signálu v závislosti na velikosti posunu okna; modrá křivka pro chybějící úsek délky 130 vzorků, zelená pro 230, červená pro 430 a světle modrá pro 660 vzorků.

Z obrázku 7.4 je patrné, že pro rekonstrukci sinusového signálu je vhodné, aby byl posun okna co nejmenší. Oproti tomu pro složený signál je na obrázku 7.5 vidět, že velikost posunu okna nemá na úspěšnost rekonstrukce velký vliv. Z toho lze usuzovat, že pro rekonstrukci většiny signálů nebude velikost posunu okna zásadním parametrem. Přesto by se dalo říct, že menšími hodnotami velikosti posunu okna nic nezkazíme. K vytvoření Gaborova framu musí být navíc posun okna vždy menší než šířka daného okna.

7.3 Hodnocení úspěšnosti rekonstrukce pomocí SNR

S využitím poznatků z části 7.2 se nyní pokusíme o co nejlepší rekonstrukci několika vybraných signálů zatím bez použití navrženého vylepšení. Prvním rekonstruovaným signálem bude signál sinusový. Dalším bude nahrávka zvonků. Jedná se o signál s poměrně řídkým spektrem, viz obrázek 7.6a. Poslední nahrávkou bude již výše používaný úryvek ze skladby Mountains. Spektrum tohoto signálu je na obrázku 7.6b.



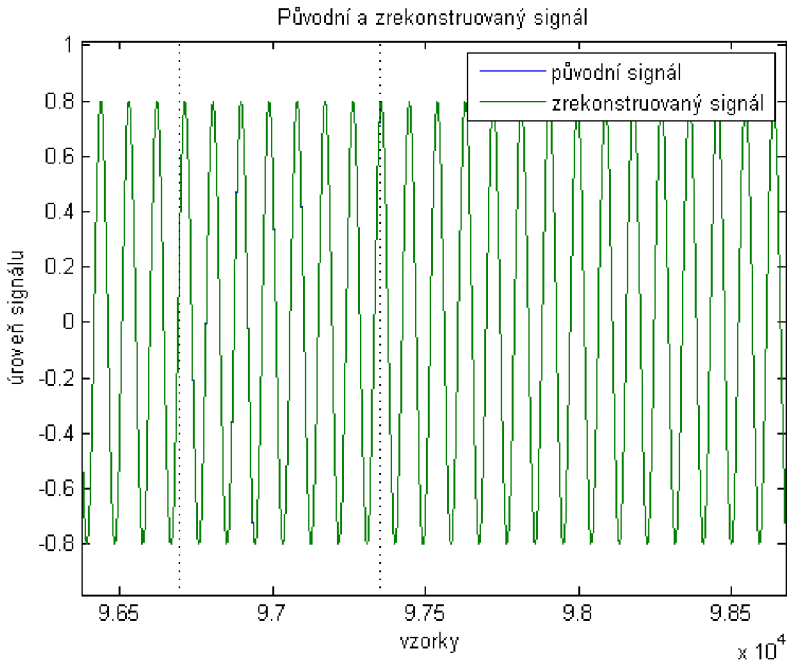
(a) (b)

Obr. 7.6: Spektrogramy pro nahrávku zvonků (a) a úryvek skladby Mountains (b).

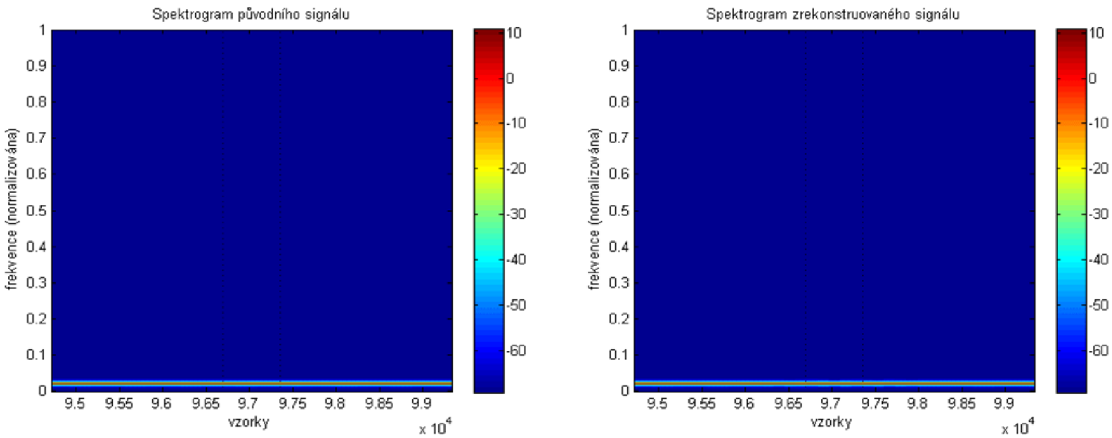
Výpočet SNR jsme provedli opět pro 4 různé případy délky chybějící části. Ty byly 130, 230, 430 a 660 vzorků. Pozice chybějící části signálu byla náhodná. Počet kanálů slovníku byl 2048, šířka okna byla 2048 a posun okna 128. Tolerance rozdílu objektivní funkce pro ukončení iterací byla 0,000005.

Tab. 7.1: Vyhodnocení testování rekonstrukce sinusového signálu pomocí SNR

sinusový signál	
velikost díry (vzorky)	SNR_d (dB)
130	80,3920
230	79,8075
430	77,5683
660	58,6783



Obr. 7.7: Časový průběh originálního a zrekonstruovaného sinusového signálu pro délku chybějící části 660 vzorků.

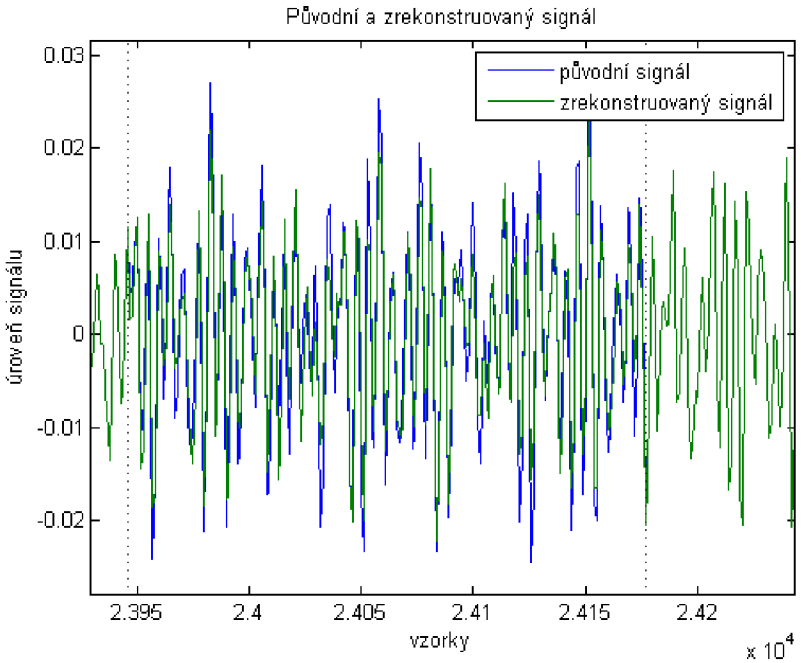


Obr. 7.8: Spektrogram originálního (vlevo) a zrekonstruovaného (vpravo) sinusového signálu pro délku chybějící části 660 vzorků. Zobrazena je pouze oblast chybějícího úseku a okolí.

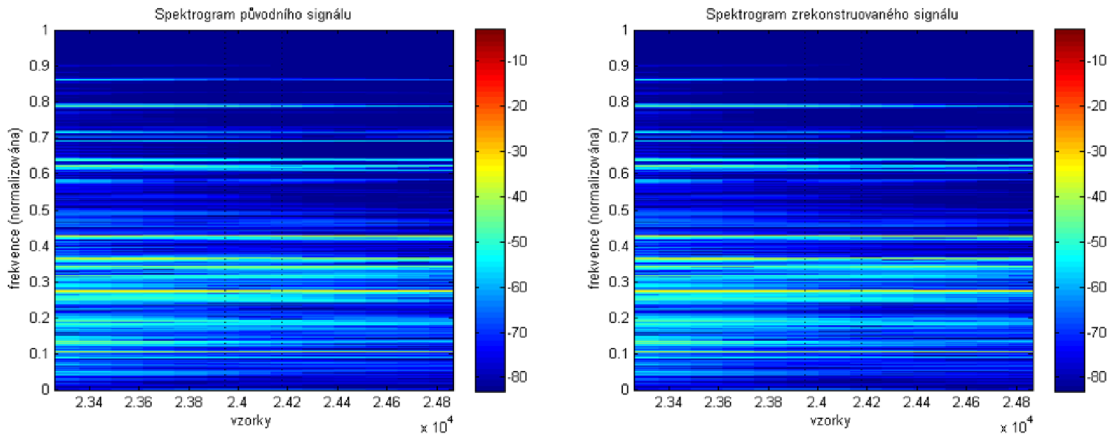
Jak je vidět z tabulky 7.1 a obrázků 7.7 a 7.8, pro sinusový signál jsme schopni dosáhnout vysoké úspěšnosti rekonstrukce. Čím menší je délka chybějící části, tím úspěšnější rekonstrukce je.

Tab. 7.2: Vyhodnocení testování rekonstrukce pomocí SNR pro nahrávku zvonků

zvonky	
velikost díry (vzorky)	SNR_d (dB)
130	21,0866
230	9,4258
430	15,9523
660	12,3731



Obr. 7.7: Časový průběh originálního a zrekonstruovaného signálu nahrávky zvonů pro délku chybějící části 230 vzorků.

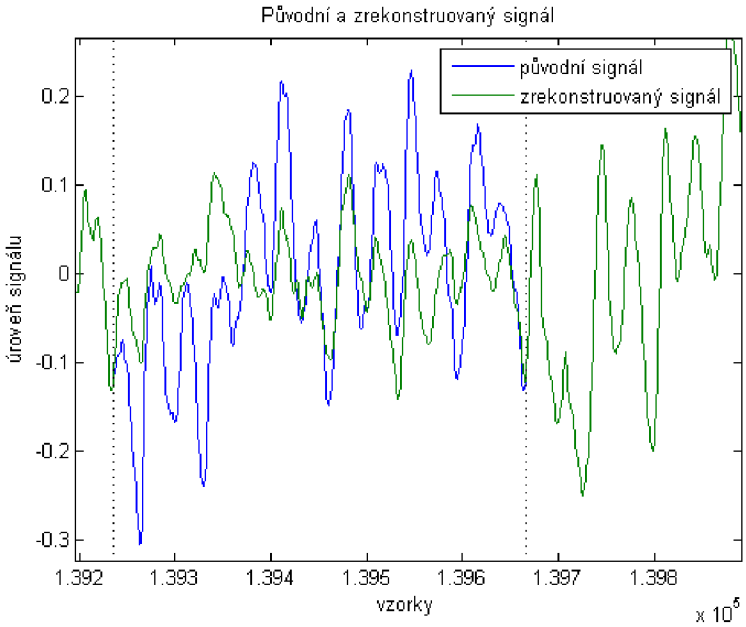


Obr. 7.8: Spektrogram originálního (vlevo) a zrekonstruovaného (vpravo) signálu zvonků pro délku chybějící části 230 vzorků. Zobrazena je pouze oblast chybějícího úseku a okolí.

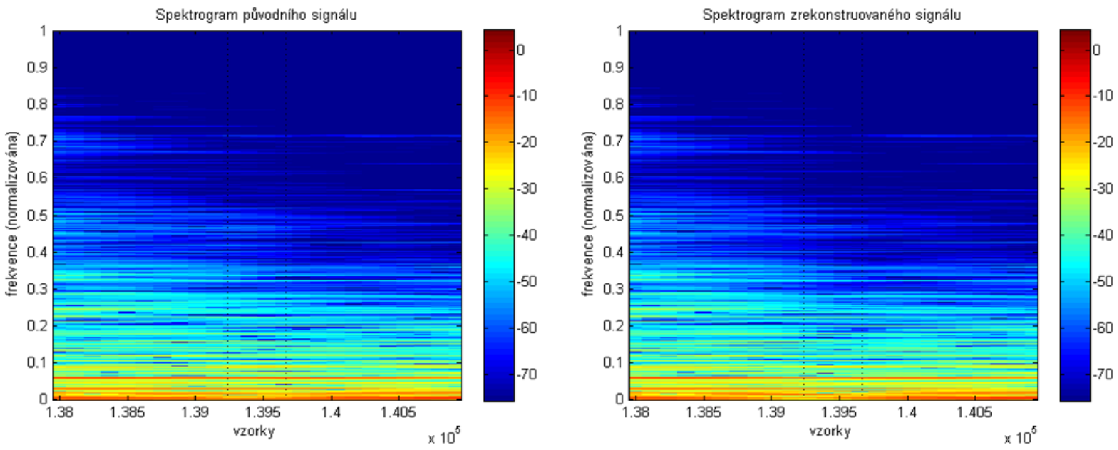
Vzhledem k řídkému charakteru signálu jsme stále schopni dosáhnout poměrně vysoké úspěšnosti rekonstrukce. Stále by se dalo říct, že čím je délka chybějící části menší, tím je úspěšnost rekonstrukce větší. Nemusí tomu tak však být vždy. Může se zde projevit vliv náhodné pozice chybějící části, kdy se může stát, že bude rekonstrukce většího chybějícího úseku úspěšnější než rekonstrukce menšího úseku v jiné části signálu.

Tab 7.3: Vyhodnocení testování rekonstrukce pomocí SNR pro část skladby Mountains

Mountains	
velikost díry (vzorky)	SNR_d (dB)
130	5,4908
230	0,6296
430	0,9902
660	0,8005



Obr. 7.7: Časový průběh originálního a zrekonstruovaného signálu úryvku skladby Mountains pro délku chybějící části 430 vzorků.



Obr. 7.8: Spektrogram originálního (vlevo) a zrekonstruovaného (vpravo) signálu úryvku skladby Mountains pro délku chybějící části 430 vzorků. Zobrazena je pouze oblast chybějícího úseku a okolí.

Vzhledem k poměrně složitému spektru a s tím souvisejícímu rozmanitému časovému průběhu signálu není rekonstrukce příliš úspěšná. Zásadní vliv na úspěšnost rekonstrukce má také pozice chybějícího úseku v rámci signálu.

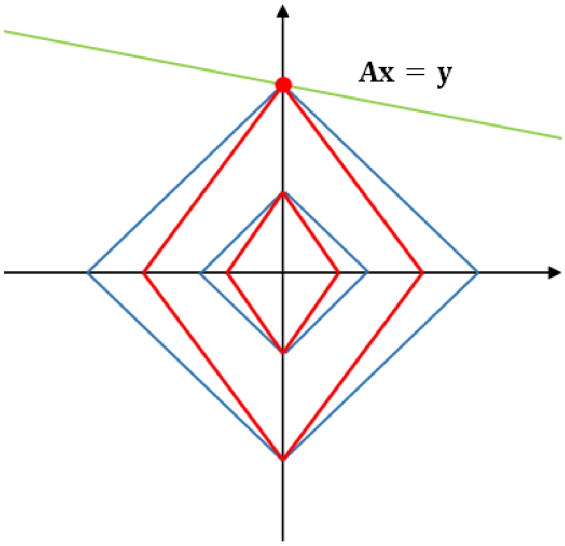
7.3.1 Vliv vylepšení algoritmu na úspěšnost rekonstrukce

Nyní porovnáme SNR rekonstrukce bez a s námi navrženým vylepšením. Porovnání jsme provedli na stejných signálech, jaké jsme využívali v předchozí části práce. Stejně tak jsme zachovali velikosti chybějících úseků a parametry použitého slovníku. Jen tolerance pro ukončení iterací byla nyní pro zkrácení výpočetního času 0,00005. V následující tabulce jsou zjištěné hodnoty SNR a jejich rozdíl.

Tab. 7.4: Srovnání úspěšnosti rekonstrukce bez a s vylepšením

sinusový signál			
velikost díry (vzorky)	SNR bez vylepšení (dB)	SNR s vylepšení (dB)	rozdíl SNR (dB)
130	78,2648	78,2663	0,0015
230	74,2842	74,2842	0,0000
430	65,2652	65,2652	0,0000
660	44,8297	44,8297	0,0000
zvonky			
velikost díry (vzorky)	SNR bez vylepšení (dB)	SNR s vylepšení (dB)	rozdíl SNR (dB)
130	24,8122	24,8183	0,0061
230	20,8694	20,8694	0,0000
430	19,7199	19,7456	0,0257
660	10,7478	10,7478	0,0000
Mountains			
velikost díry (vzorky)	SNR bez vylepšení (dB)	SNR s vylepšení (dB)	rozdíl SNR (dB)
130	5,8823	5,8823	0,0000
230	2,5687	2,5687	0,0000
430	0,7245	0,7245	0,0000
660	0,5891	0,5891	0,0000

Jak můžeme vidět, tak naše vylepšení ve skutečnosti žádné výrazné zlepšení nepřineslo. Ve většině případů dostaneme s vylepšením stejné řešení jako bez něj. Sice modifikujeme řídký vektor koeficientů, ale přesto se i s touto modifikací dostaneme ke stejnému řešení. Možné vysvětlení této situace znázorňuje obrázek 7.9. K nalezení řídkého řešení totiž využíváme ℓ_1-relaxaci. Na obrázku je vidět, že přesto, že jsme řídké koeficienty modifikovali, dostáváme se ke stejnému řešení jako bez dané úpravy.



Obr. 7.9: Zvětšující se koule v ℓ_1 normě. Modře případ bez modifikace koeficientů, červeně případ s modifikovanými koeficienty.

ZÁVĚR

Cílem této bakalářské práce bylo porozumět problému znehodnocení audiosignálu s chybějícími vzorky, seznámit se s obvyklými metodami jeho řešení a také se seznámit s moderní metodou restaurace audiosignálu zaměřenou na tzv. řídkosti signálů. Dále bylo za úkol navrhnout výpočetní algoritmus rekonstrukce a jeho vylepšení, otestovat daný algoritmus i s vylepšením a zhodnotit úspěšnost rekonstrukce na reálných signálech.

Jako zástupce obvyklých metod rekonstrukce signálu byla představena metoda založená na autoregresním modelování signálu a krátce také metoda založená na sinusoidálním modelování. Řekli jsme si, co je to řídkost signálů a provedli rozbor metody rekonstrukce využívající právě řídkosti. Navrhli jsme kompletní výpočetní algoritmus rekonstrukce. Vzniknul i návrh vylepšení tohoto algoritmu. Celý algoritmus i s vylepšení jsme realizovali v prostředí MATLAB za pomoci toolboxu LTFAT. Bylo provedeno zkoumání s cílem zjistit, s jakými parametry dosáhne algoritmus nejlepších výsledků. Tyto poznatky byly použity k rekonstrukci několika nahrávek s uměle vytvořenými chybějícími úseky o různých velikostech. Úspěšnost rekonstrukce jsme hodnotili pomocí SNR. Zjistili jsme, že jsme schopni s poměrně vysokou úspěšností rekonstruovat signály, které nejsou ve svém spektru příliš složité. Naopak ne příliš uspokojivé výsledky jsme dostali pro signály se složitým spektrem (rozmanitým časovým průběhem). V závěru jsme porovnávali úspěšnost rekonstrukce signálu při výpočtu bez vylepšení a s vylepšením. Zjistili jsme, že navržené vylepšení ve skutečnosti nepřináší žádné výrazné zlepšení rekonstrukce a pokusili jsme se přinést vysvětlení této skutečnosti.

LITERATURA

[1] RAJMIC, Pavel. *Řídké a nízkohodnostní reprezentace signálů s aplikacemi: Sparse and low-rank representations of signals with applications : zkrácená verze habilitační práce*. Brno: VUTIUM, 2015. ISBN 9788021451957.

[2] ŠPIŘÍK, Jan, Pavel RAJMIC a Vítězslav VESELÝ. Reprezentace signálů: od bází k framům. *Elektrorevue* [online]. b.r., **2010**(6), 7-8 [cit. 2016-12-07]. ISSN 1213 - 1539. Dostupné z: http://elektrorevue.cz/cz/download/ reprezentace-signalu--od-bazi-k-framum/

[3] ŠPIŘÍK, Jan. *Modul pro generování "atomů" pro přeparametrizovanou reprezentaci signálu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010, 52 s.

[4] Třídy složitosti a Turingovy stroje. *Algoritmy.net* [online]. b.r. [cit. 2016-12-07]. Dostupné z: https://www.algoritmy.net/article/5774/Tridy-slozitosti

[5] HRBÁČEK, Radek, Pavel RAJMIC, Vítězslav VESELÝ a Jan ŠPIŘÍK. Řídké reprezentace signálů: úvod do problematiky. *Elektrorevue* [online]. b.r., **2011**(5), 8 [cit. 2016-12-07]. Dostupné z: http://www.elektrorevue.cz/cz/download/ ridke-reprezentace-signalu--uvod-do-problematiky/

[6] COMBETTES, Patrick a Jean-Christophe PESQUET. *Proximal Splitting Methods in Signal Processing* [online]. 2009, , 3-10 [cit. 2016-12-07]. Dostupné z: https://arxiv.org/pdf/0912.3522v4.pdf

[7] MACH, Václav. *Inpainting of Missing Audio Signal Samples*. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2016. Doctoral thesis.

[8] LAGRANGE, Mathieu, Sylvain MARCHAND a Jean-Bernard RAULT. Long Interpolation of Audio Signals Using Linear Prediction in Sinusoidal Modeling. *J. Audio Eng. Soc. 53*. b.r., **2005**(10), 892.

[9] MEYER, Carl. *Matrix analysis and applied linear algebra*. Philadelphia: Society for Industrial and Applied Mathematics, 2000, xii, 718 s. Solutions manual (171 s.) 1 elektr. ISBN 0898714540.

[10] PRŮŠA, Zdeněk, Peter SøNDERGAARD, Nicki HOLIGHAUS, Christoph WIESMEYR a Peter BALAZS. *The Large Time-Frequency Analysis Toolbox 2.0* [online]. b.r., s. 419 [cit. 2017-05-10]. DOI: 10.1007/978-3-319-12976-1_25. Dostupné z: http://link.springer.com/10.1007/978-3-319-12976-1_25

[11] WHITE, Glenn a Gary LOUIE. *The audio dictionary*. 3rd ed., rev. and expanded. Seattle: University of Washington Press, 2005. ISBN 0-295-98498-8.

[12] VONDRÁŠEK, Martin. *Odhad SNR řečového signálu snímaného v hlučném prostředí*. České vysoké učení technické v Praze, Fakulta elektrotechnická, 2004. Dostupné také z: http://noel.feld.cvut.cz/speechlab/publications/032_diplomka04.pdf. Vedoucí práce Doc. Ing. Petr Pollák, CSc.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

LASSO	Least Absolute Shrinkage and Selection Operator
LTFAT	The Large Time/Frequency Analysis Toolbox
složitost třídy NP	nepolynomiální – výsledku je možné dosáhnout v čase vyšším než polynomiálním
SNR	Signal-to-noise Ratio
soft	měkké prahování
STFT	krátkodobá Fourierova transformace

SEZNAM PŘÍLOH

A Obsah přiloženého CD

A OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje

- `Pavel_Skyva-bakalarska_prace.pdf` – elektronická verze bakalářské práce.
- `restauraceSignalu.zip` – zkomprimovaný soubor obsahující všechny potřebné soubory pro funkci programu v MATLABu.

V souboru `restauraceSignalu.zip` naleznete všechny skripty a funkce a ve složce `signaly` také audiosoubory použité při rekonstrukcích.

Skript `rekonstrukceBezVyl.m` spouští program pro rekonstrukci signálu bez námi navrženého vylepšení. Výstupem tohoto programu je graf s časovým průběhem původního a zrekonstruovaného signálu, graf historie objektivní funkce v průběhu iterací, spektrogram signálu původního a spektrogram signálu zrekonstruovaného. Výstupem jsou také hodnoty SNR pro oblast chybějící části i pro celý signál.

Skript `rekonstrukceSVyl.m` má stejné výstupy jako výše zmíněný skript, avšak pro výpočet s vylepšením.

Skript `rekonstrukcePorovnani.m` spustí program, který vypočítá rekonstrukci signálu s vylepšením i bez něj a zobrazí jednotlivé hodnoty SNR pro oba případy a jejich rozdíl. Výstupem skriptu je také graf s časovými průběhy signálu původního, zrekonstruovaného bez vylepšení a signálu zrekonstruovaného s vylepšením.

Program byl realizován a spouštěn v programu MATLAB R2013a.