

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ  
ÚSTAV MATEMATIKY  
FACULTY OF MECHANICAL ENGINEERING  
INSTITUTE OF MATHEMATICS

APLIKACE LINEÁRNÍ ALGEBRY A OPTIMALIZACE  
VE ZPRACOVÁNÍ ZVUKOVÝCH SIGNÁLŮ  
APPLICATIONS OF LINEAR ALGEBRA AND OPTIMIZATION IN SOUND SIGNAL PROCESSING

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

ANEŽKA KOLBÁBKOVÁ

VEDOUcí PRÁCE  
SUPERVISOR

Mgr. PAVEL RAJMIC, Ph.D.

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav matematiky

Akademický rok: 2011/2012

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

student(ka): Anežka Kolbábková

který/která studuje v **bakalářském studijním programu**

obor: **Matematické inženýrství (3901R021)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

### **Aplikace lineární algebry a optimalizace ve zpracování zvukových signálů**

v anglickém jazyce:

### **Applications of linear algebra and optimization in sound signal processing**

Stručná charakteristika problematiky úkolu:

Signály z reálného světa se často skládají z relativně malého počtu stavebních bloků vůči celkové délce. Toto zjištění je základem pro moderní tzv. řídké reprezentace signálů. Metodami lineární algebry a optimalizace lze nejen takové řídké vyjádření audiosignálů získat, ale také efektivně využít k praktickým úkolům jako jsou potlačování šumu, komprese, převzorkování, doplňování chybějících dat apod.

Smyslem práce je seznámit se se současným stavem problematiky, nastudovat algoritmy pro hledání řídkých reprezentací a vše empiricky ověřit na simulovaných i reálných zvukových datech.

Cíle bakalářské práce:

Nastudovat základní principy a algoritmy hledání tzv. řídkých reprezentací.

Zaměřit se na typické situace u zvukových signálů (krátkodobá spektra).

V software MATLAB nebo MAPLE empiricky ověřit teorii na konkrétních datech.

Shrnout a interpretovat výsledky.

Seznam odborné literatury:

ELAD, M., Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer, 2010.

HRBÁČEK, R.; RAJMIC, P.; VESELÝ, V.; ŠPIŘÍK, J. Řídké reprezentace signálů: Úvod do problematiky. Elektrovue - Internetový časopis, 2011, roč. 2011, č. 50, s. 1-10. ISSN: 1213-1539.

BALAZS, P., DOERFLER, M. et al., Theory, Implementation and Application of Nonstationary Gabor Frames, J. Comput. Appl. Math., In Press. (2011)

ADLER, A. et al., A Constrained Matching Pursuit Approach to Audio Declipping. Acoustics, Speech and Signal Processing, IEEE International Conference on (ICASSP 2011), IEEE, Prague, Czech Republic, 2011.

Vedoucí bakalářské práce: Mgr. Pavel Rajmic, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

V Brně, dne 12.11.2011

L.S.

---

prof. RNDr. Josef Šlapal, CSc.  
Ředitel ústavu

---

prof. RNDr. Miroslav Doupovec, CSc.  
Děkan fakulty

## **Abstrakt**

Práce je zaměřena na řídké reprezentace v oblasti audio signálů. Zahrnuje obecný teoretický úvod do základní problematiky řídkých reprezentací a simulaci teorie na vygenerovaných tónech „klavíru“ v programu Matlab. Teorii ověřuje na těchto vygenerovaných tónech i na reálných signálech.

## **Summary**

This thesis is focused on sparse representation of audio signals. It consists of theoretical introduction to basic issues of sparse representation and also simulation on artificial tones of “piano” in program Matlab. The theory is verified on this generated tones and also on real signals.

## **Klíčová slova**

řídký signál, zpracování signálů, komprimované snímání, hladové algoritmy

## **Keywords**

sparse signal, signal processing, compressed sensing, greedy algorithm

KOLBÁBKOVÁ, A. *Aplikace lineární algebry a optimalizace ve zpracování zvukových signálů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 27 s. Vedoucí Mgr. Pavel Rajmic, Ph.D.

Prohlašuji, že svou bakalářskou práci *Aplikace lineární algebry a optimalizace ve zpracování zvukových signálů* jsem vypracovala samostatně pod vedením Mgr. Pavla Rajmice, Ph.D. a pro vypracování jsem použila zdroje uvedené v seznamu literatury na konci práce.

Anežka Kolbábková

Ráda bych poděkovala vedoucímu mé bakalářské práce Mgr. Pavlu Rajmicovi, Ph.D. za odbornou pomoc a cenné rady při psaní práce.

Anežka Kolbábková

# OBSAH

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Základy problematiky řídkých signálů</b>	<b>3</b>
2.1	Základy . . . . .	3
2.1.1	Jaké se používají funkce $J(x)$ . . . . .	3
2.1.2	Slovník . . . . .	4
2.1.3	Doplňující definice a označení . . . . .	5
2.1.4	Označení . . . . .	6
<b>3</b>	<b>Metody řešení úloh s řídkými signály</b>	<b>7</b>
3.1	Relaxační metody . . . . .	7
3.1.1	Postačující podmínky jednoznačnosti řešení . . . . .	8
3.1.2	Lineární programování . . . . .	11
3.1.3	IRLS . . . . .	11
3.2	Hladové algoritmy . . . . .	12
3.2.1	OMP . . . . .	12
3.2.2	MP . . . . .	13
3.2.3	Normalizace . . . . .	13
3.3	Prahování . . . . .	14
<b>4</b>	<b>Komprimované snímání</b>	<b>15</b>
<b>5</b>	<b>Aplikace poznatků v programu Matlab</b>	<b>17</b>
5.1	Simulace tónů a tvorba slovníku . . . . .	17
5.1.1	Simulace tónů klavíru v Matlabu . . . . .	17
5.1.2	Tvorba slovníku . . . . .	18
5.1.3	Simulace zadaného signálu . . . . .	19
5.1.4	Metody OMP a MP aplikované na signál . . . . .	19
5.2	Komprimované snímání . . . . .	19
5.3	Rekonstrukce skutečného signálu . . . . .	22
5.4	Srovnání rekonstrukce OMP, MP a IRLS . . . . .	22
<b>6</b>	<b>Závěr</b>	<b>25</b>
<b>7</b>	<b>Seznam použitých zkratk a symbolů</b>	<b>27</b>

# 1. ÚVOD

Při řešení inženýrských problémů i v obyčejném životě se často setkáváme s úlohami, které vedou na systém nedourčených lineárních rovnic, tedy s menším počtem rovnic, než máme neznámých. V množině řešení tohoto systému hledáme to, které má co nejvíce nulových hodnot, tedy řídké. Výhod a využití zpracovávání signálů z hlediska řídkosti je mnoho. Například uveďme úspornou kompresi dat a numerickou stabilitu problému. Metody řešení se využívají pro zašuměné a neúplné signály. Podrobnější přiblížení problematiky odšumování a neúplných signálů můžeme získat v [1].

První algoritmus vytvořený pro hledání řídkých řešení MP byl zveřejněn v roce 1993 pány Mallatem a Zhangem. Jedná se tedy o problematiku s krátkou historií, přesto se jí již ve světě věnuje mnoho vědců a dokázali, že má smysl se algoritmy na hledání řídkých signálů zabývat. Stále se přichází s novými metodami, které jsou lepší a rychlejší než ty základní.

Zadání této práce slibuje vytvořit náhled do řešení řídkých signálů se zaměřením na základní principy a algoritmy v této oblasti. Konkrétně zkoumáme krátkodobá spektra zvukových signálů. Pro empirické ověření poznatků máme využít na vygenerovaných i reálných datech program Matlab.

Tento dokument je strukturován do několika oddílů. Začneme popisem základů problematiky řídkých signálů, následně v kapitole 3 popíšeme metody zpracování řídkých signálů a uvedeme některé podmínky pro použití metod. Teorii komprimovaného snímání obsahuje kapitola 4. Ke konkrétní aplikaci teorie přímo na zvukové signály se dostaneme v kapitole 5, která obsahuje popis vytvořených programů, provedená zkoumání a vyvozené důsledky vyplývající z práce na programu.

Na konec v závěru shrneme práci v oddílu 6. Práce dále obsahuje v seznamu použitých zkratek a symbolů a poslední uvádí seznam literatury, potřebné zdroje informací pro napsání tohoto textu.



# 2. ZÁKLADY PROBLEMATIKY ŘÍDKÝCH SIGNÁLŮ

## 2.1. Základy

V úvodu zaznělo, že se zabýváme systémem nedourčených lineárních rovnic a hledáme nejřidší řešení dané situace, tedy s hodně nízkým počtem nenulových proměnných. V této kapitole položíme základy pro porozumění metod a s nimi spojených pojmů. I v dalších dvou kapitolách budeme uvažovat situace, které mohou nastat obecně a zaměření na zvukové signály se konkrétně projeví až v kapitole 5. Pro lepší orientaci budeme matice značit velkými tučnými písmeny, pro vektory budeme používat malá tučná. Jiné symboly, které budou třeba pro matematické vyjádření zvýrazníme kurzívou.

Základní úlohu popíšeme rovnicí  $\mathbf{Ax} = \mathbf{b}$ . Matici  $\mathbf{A}$  typu  $n \times m$  pro  $n < m$ , nazveme slovníkem soustavy a její sloupce atomy,  $\mathbf{x}$  jsou přípustná řešení problému a vektor  $\mathbf{b}$  daná data úlohy, pro které hledáme nejřidší vektor  $\mathbf{x}$ . Abychom se vyhnuli různým obtížím s existencí řešení, budeme nadále předpokládat, že slovník  $\mathbf{A}$  má plnou hodnotu.

Řešení nemusí existovat, pokud nesouhlasí span matice  $\mathbf{A}$  a vektoru  $\mathbf{b}$ , nebo může mít systém nekonečně mnoho řešení. Mezi nimi hledáme tedy konkrétně ta, která mají mnoho koeficientů nulových. Řídkost vektoru se váže přímo k určitému slovníku, říkáme, že vektor  $\mathbf{x}$  je řídký v slovníku  $\mathbf{A}$ . Při použití jiného slovníku vektor  $\mathbf{x}$  být řídký vůbec nemusí.

Popsaný problém při daných podmínkách má nekonečně mnoho řešení a naším úkolem je zjistit to, které je nejřidší. Očividně budeme postupovat v duchu optimalizačních metod. Zúžení problému dostaneme po zvýšení požadavků na danou soustavu. Jeden ze způsobů, jak toho dosáhnout je regularizace problému popsaneho rovnicí:

$$(P_J) \min_{\mathbf{x}} J(\mathbf{x}) \text{ vzhledem k systému } \mathbf{b} = \mathbf{Ax} \quad (2.1)$$

V rovnici (2.1) představuje funkce  $J(\mathbf{x})$  člen zjišťující vhodnost vektoru  $\mathbf{x}$  a podle této podmínky si žádáme  $\mathbf{x}$  s nejmenší funkční hodnotou  $J(\mathbf{x})$ .

### 2.1.1. Jaké se používají funkce $J(x)$

Jako funkci  $J(\mathbf{x})$  většinou používáme  $l_p$  normu vektoru  $\mathbf{x}$ . Definice této funkce je obecně známá, avšak abychom se vyhnuli případným nedorozuměním,  $l_p$  normu zadefinujeme. Pro část definice budeme potřebovat pojem nosič vektoru  $\mathbf{x}$ .

#### Definice 2.1

*Nechť vektor  $\mathbf{x} \in \mathbb{R}^m$ . Pak množinu indexů, ve kterých  $\mathbf{x}$  má nenulové hodnoty nazýváme nosičem vektoru  $\mathbf{x}$  a značíme  $\text{supp}(\mathbf{x})$ .*

Po zavedení pojmu nosiče vektoru pokročíme k definici normy.

## Definice 2.2

Pro vektor  $\mathbf{x} \in \mathbb{C}^N$  definujeme normu<sup>1</sup>  $l_p$  takto:

$$\|\mathbf{x}\|_p := \left( \sum_{i=1}^N |\mathbf{x}_i|^p \right)^{1/p} \quad \text{pro } 1 \leq p < \infty$$

$$\|\mathbf{x}\|_p := \sum_{i=1}^N |\mathbf{x}_i|^p \quad \text{pro } 0 < p < 1$$

$$\|\mathbf{x}\|_\infty := \max_i |\mathbf{x}_i|$$

$$\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})|.$$

Dobrou představu o normách a jejich vlastnostech můžeme získat v [5]. Definujme si pořádně řídkost vektoru pomocí  $l_0$  normy.

## Definice 2.3

Nechť pro vektor  $\mathbf{x} \in \mathbb{C}^m$  platí rovnice (2.2) a  $k \in \mathbb{Z}_0^+$ . Pak  $\mathbf{x}$  nazveme  $k$ -řídký.

$$\|\mathbf{x}\|_0 \leq k, \tag{2.2}$$

### 2.1.2. Slovník

Při řešení problému se vždy používá slovník, je proto velmi důležitá jeho správná volba. Existují vlastnosti matic, které specifikují jejich vhodnost pro tento úkol. Více si o nich povíme v podkapitole 3.1. Na tomto místě shrneme nejčastěji používané typy slovníků.

- Matice vhodné pro řešení problém se mohou tvořit ze dvou ortogonálních bází, uveďme například jednotkovou matici  $\mathbf{I}$ , Fourierovu transformaci či wavelety. Kombinací dostaneme slovník, například  $\mathbf{A} = [\mathbf{F}, \mathbf{W}]$ .
- Deterministické matice s předem danými parametry vyhovující RIP<sup>2</sup> se ještě nikomu nepodařilo sestavit. Jinak je to s maticemi tvořenými z různých rozdělení pravděpodobnosti. Podívejme se na pár případů:
  - Matice s prvky vygenerovanými z normálního rozdělení s parametry  $\mu = 0$ ,  $\sigma^2 = 1/m$ . Takovéto matici říkáme Gaussovská.
  - Bernoullijské matice, kde prvky matice jsou náhodné veličiny nabývají hodnot  $\pm 1/\sqrt{m}$  se stejnou pravděpodobností.

Tyto dva výše uvedené typy matic splňují RIP s určitou pravděpodobností podle speciálních podmínek, které záleží na parametru  $m$  [5].

---

<sup>1</sup>Pro  $p \in (0, 1)$  se označíme „normu“  $l_p$  jako  $\|\mathbf{x}\|_p := \sum_{i=1}^N |\mathbf{x}_i|^p$ , podle vlastností se o normu v pravém slova smyslu nejde. Ani  $l_0$  nesplňuje celou definici normy, ale budeme pro jednoduchost používat i pro tyto případy označení  $l_p$  norma.

<sup>2</sup>Podrobnosti k RIP uvedeme v kapitole 3.1

- V některých programech, které jsou napsány pro tento text pracujeme se slovníkem tvořeným maticí DCT. Také je možné použít jako slovník DFT.
- Pro speciální úlohy si můžeme vytvořit slovník vlastní, podle potřebných parametrů. Konkrétní příklad takového slovníku uvedeme v kapitole 5.

### 2.1.3. Doplnující definice a označení

Na tomto místě definujeme některé pojmy potřebné v dalším textu.

#### Definice 2.4

Nechť  $\mathbf{B}$  je báze a pro  $\forall \mathbf{v}_i, \mathbf{v}_j \in \mathbf{B}$ ,  $i \neq j$  platí

$$(\mathbf{v}_i, \mathbf{v}_j) = 0. \quad (2.3)$$

Pak  $\mathbf{B}$  nazýváme ortogonální.

#### Definice 2.5

Nechť  $\mathbf{B}$  je ortogonální báze a navíc pro  $\forall \mathbf{v}_i, \mathbf{v}_j \in \mathbf{B}$ ,  $i \neq j$  platí

$$(\mathbf{v}_i, \mathbf{v}_i) = 1. \quad (2.4)$$

Pak  $\mathbf{B}$  nazýváme ortonormální.

#### Definice 2.6

Jednotková koule  $B_p^m$  v normě  $l_p$  je definovaná předpisem

$$B_p^m = \{\mathbf{x} \in \mathbb{C}^m, \|\mathbf{x}\|_p \leq 1\}.$$

#### Definice 2.7 [6]

Chyba nejlepší aproximace  $k$ -řádkého vektoru  $\mathbf{x} \in \mathbb{C}^m$  v normě  $l_p$  je definována jako

$$\sigma_k(\mathbf{x})_p = \inf_{\mathbf{z} \in \Sigma_k} \|\mathbf{x} - \mathbf{z}\|_p. \quad (2.5)$$

#### Definice 2.8

Nechť se matice  $\mathbf{A}^H$  je typu  $m \times n$  s komplexními prvky nazývá hermitovskými sdružená  $k$  matici  $\mathbf{A}$  typu  $n \times m$ , také s komplexními prvky, pokud je transponovaná a místo prvku  $a_{ij}$  je komplexně sdružený prvek  $\forall i, j$ ;  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ .

#### Definice 2.9

Nechť matice  $\mathbf{A}$  je typu  $n \times n$ , prvky matice jsou komplexní čísla. Pokud splňuje

$$\mathbf{A}^H \cdot \mathbf{A} = \mathbf{I},$$

pak řekneme že matice  $\mathbf{A}$  je unitární.

**Definice 2.10 [9]**

Nechť  $\mathbf{A}$  je matice typu  $n \times n$  a matice  $\mathbf{B}$  typu  $m \times m$ . Pak definujeme Kroneckerův součin jako

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} b_{1,1}\mathbf{A} & \dots & b_{1,m}\mathbf{A} \\ \vdots & \ddots & \vdots \\ b_{m,1}\mathbf{A} & \dots & b_{m,m}\mathbf{A} \end{pmatrix}$$

kde  $b_{i,j} \in \mathbf{B}$  pro  $1 \leq i \leq m$ ,  $1 \leq j \leq m$ .

**Definice 2.11**

Reálnou matici  $\mathbf{A}^+$  nazveme pseudoinverzní k matici reálných čísel  $\mathbf{A}$  typu  $m \times n$ , pokud splňuje rovnice

$$\begin{aligned} \mathbf{A}\mathbf{A}^+\mathbf{A} &= \mathbf{A} \\ \mathbf{A}^+\mathbf{A}\mathbf{A}^+ &= \mathbf{A}^+ \\ (\mathbf{A}\mathbf{A}^+)^T &= \mathbf{A}\mathbf{A}^+ \\ (\mathbf{A}^+\mathbf{A})^T &= \mathbf{A}^+\mathbf{A} \end{aligned}$$

Pseudoinverzní matice existuje vždy a je jednoznačně určena.

**2.1.4. Označení**

- Máme podmnožinu  $T \subset \{1, \dots, m\}$  a vektor  $\mathbf{x} \in \mathbb{C}^m$ , z vektoru  $\mathbf{x}$  odvodíme vektor  $\mathbf{x}_T \in \mathbb{C}^m$  tak, že indexy prvků  $\mathbf{x}$ , které neodpovídají prvkům  $T$  vynulujeme.
- Komplement  $T$  označíme jako  $T^c = \{1, \dots, m\} \setminus T$ .
- Kardinalitu  $T$  budeme značit  $|T|$ .

# 3. METODY ŘEŠENÍ ÚLOH S ŘÍDKÝMI SIGNÁLY

Nyní máme vysvětlené základní pojmy a můžeme se pustit do metod řešení. Dělíme je do několika skupin podle způsobu přístupu k (3.1). Hlavní typy algoritmů, kterými se budeme zabývat jsou relaxační metody, hladové algoritmy a thresholding.

Všechny postupy jsou pro netriviálnost iterativní a tedy jsou založeny na hledání výsledku, který je velmi málo odlišný od přesného řešení našeho základního problému (3.1) (najít nejjednodušší řešení soustavy lineárních rovnic). Jednoduše řečeno zabýváme se numerickými metodami.

$$(P_0) \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k systému } \mathbf{b} = \mathbf{Ax} \quad (3.1)$$

Pro nalezení optimálního řešení máme několik přístupů. Podívejme se blíže na problém (3.1). Neznámou  $\mathbf{x}$  máme jak v první části — minimalizování nosiče vektoru  $\mathbf{x}$  — tak i v řešení samotné soustavy rovnic a tedy je možné zaměřit na  $\text{supp}(\mathbf{x})$ . Je nezbytné si uvědomit, že když jednou najdeme nosič vektoru  $\mathbf{x}$  je snadné k dojit nenulovým hodnotám  $\mathbf{x}$  řešením MNC. Podstata  $\mathbf{x}$  i iteračních algoritmů je diskrétní a díky tomu existují a často jsou používány hladové algoritmy.

Zaměříme-li se na druhou část (3.1), tedy  $\mathbf{b} = \mathbf{Ax}$ , míříme k relaxačním metodám. S tímto přístupem opomeneme část první a považujeme neznámou za vektor  $\mathbf{x} \in \mathbb{R}^m$ . Pro tento případ je možné použít vlastnosti spojité optimalizace pro řešení (3.1) pokud vyhlazujeme penalizační funkci [4].

## 3.1. Relaxační metody

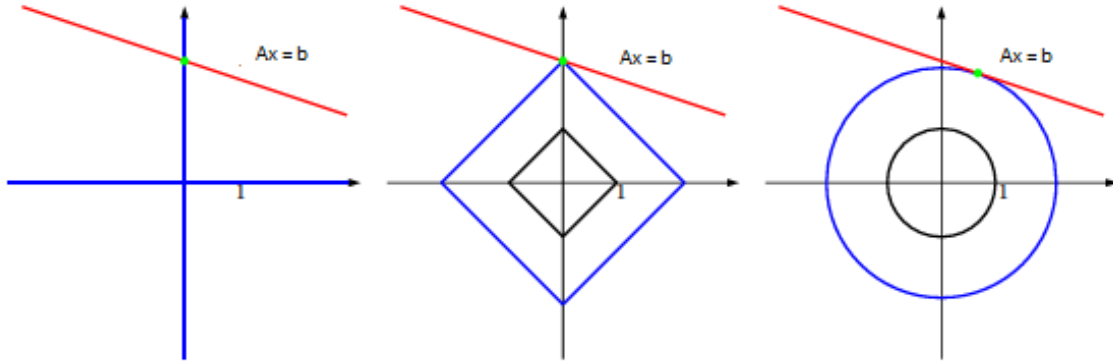
Tato skupina je časově náročnější výpočet než následující hladové algoritmy, přirozeně však dosahují kvalitnějších výsledků. Relaxační metody se liší podle toho, na kterou konkrétní funkci  $J(\mathbf{x})$  jsou vztaženy, vztahujeme je tedy nejčastěji na  $l_p$  normy. Vyjádříme obdobu (2.1) jako

$$(P_p) \min_{\mathbf{x}} \|\mathbf{x}\|_p \text{ vzhledem k systému } \mathbf{b} = \mathbf{Ax}.$$

Problém (3.1) bychom rádi řešili s  $l_0$  normou, což bohužel není v reálném čase možné, protože není konvexní a nelze na ni použít metody konvexního programování. Museli bychom projít všechny možné řešení soustavy a to je velmi časově náročné. Spokojíme se tedy s tím, že se pokusíme najít řešení, které je přibližně přesné, podle tolerance, kterou si zvolíme. Při pokusu využití na řešení problému nejmenší konvexní normu (tedy  $l_1$  normou) zjišťujeme, že tato možnost je za určitých podmínek uspokojivá. Problém (3.1) tedy aproximujeme (3.2) a řešíme relaxačními metodami.

$$(P_1) \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k systému } \mathbf{b} = \mathbf{Ax} \quad (3.2)$$

Pokud víme, že v naší soustavě jsou data nepřesná, neboli zašuměná, lze použít problém LASSO, pro který platí trošku upravená rovnice, ve které je zahrnuta odchylka od přesného řešení  $\delta$ :



Obrázek 3.1: Dotyk nafukující se koule v normách  $l_0$ ,  $l_1$ ,  $l_2$  a nadroviny řešení  $\mathbf{Ax}=\mathbf{b}$

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k systému } \|\mathbf{Ax} - \mathbf{b}\|_2 \leq \delta \quad (3.3)$$

Pro úplnost uvedeme více zástupců relaxačních metod. Řadí se mezi ně algoritmy LARS, IRLS, lineární programování a také BP, jedna z nejstarších. Více si všimneme lineárního programování a především metody IRLS.

### 3.1.1. Postačující podmínky jednoznačnosti řešení

Jak už jsme zmínili, na slovníku soustavy hodně záleží úspěšnost rekonstrukce. Snažíme se převést problém (3.1) na (3.2). Pro posouzení ekvivalence řešení  $l_0$ - a  $l_1$ -minimalizací byly zavedeny NSP a RIP, než se k nim dostaneme, musíme si nejprve ujasnit některé další pojmy. Budou nás zajímat pouze  $l_0$ - a  $l_1$ -minimalizace,  $l_2$ -minimalizace je pro naše výpočty nevhodná, protože nehledá nejřidší řešení, ale řešení s minimální energií. Tato skutečnost je zřejmá z obrázku 3.1, kde vidíme, že na rozdíl od prvních dvou typů koulí, dotyk nadroviny řešení a koule  $l_2$  normy je mimo osy a tedy, jak bylo řečeno výše, nejedná se o řídké řešení.

Když metodou najdeme řídké řešení, zajímá nás, jestli je už nejřidší možné, nebo existuje ještě řidší. Abychom mohli toto posoudit, zavedeme pojem *spark* ( $\mathbf{A}$ ).

#### Definice 3.1

Nechť  $\mathbf{A}$  je matice s reálnými prvky typu  $n \times m$ , a vektor  $\mathbf{z} \in \ker^1 \mathbf{A}$ ,  $\mathbf{z} \neq 0$ , pak

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{z}} \|\mathbf{z}\|_0. \quad (3.4)$$

*Spark* ( $\mathbf{A}$ ) je tedy nejmenší počet lineárně závislých atomů matice  $\mathbf{A}$ .

#### Věta 3.1

Pokud soustava  $\mathbf{Ax} = \mathbf{b}$  má řešení  $\mathbf{x}$  splňující

$$\|\mathbf{x}\|_0 < \frac{\text{spark}(\mathbf{A})}{2} \quad (3.5)$$

pak je  $\mathbf{x}$  nutně nejřidší řešení a žádné jiné řešení se stejnou řídkostí neexistuje.

<sup>1</sup>Jádro lineárního zobrazení určeného maticí  $\mathbf{A}$ .

Vypadá to jednoduše, ale výpočet  $\text{spark}(\mathbf{A})$  je přibližně stejně náročný jako řešení rovnice (3.1) a tedy potřebujeme najít jiný způsob ověření minimální řídkosti.

### Definice 3.2

Nechť  $\mathbf{A}$  je matice s reálnými prvky typu  $n \times m$ ,  $\mathbf{a}_j$  je její  $j$ -tý sloupec. Pak největší absolutní normalizovaný skalární součin dvou různých atomů matice  $\mathbf{A}$  nazveme vzájemná koherence matice  $\mathbf{A}$ .

$$\mu(\mathbf{A}) = \max_{i \leq j, k \leq m, j \neq k} \frac{|\mathbf{a}_j^T \mathbf{a}_k|}{\|\mathbf{a}_j\|_2 \cdot \|\mathbf{a}_k\|_2} \quad (3.6)$$

Vzájemná koherence udává tedy míru lineární nezávislosti mezi sloupci matice. Nulovou koherenci mají pouze unitární matice, soustava nedourčených rovnic má koherenci nenulovou. Snažíme se přiblížit nulové koherenci co nejvíce, tedy aby měla podobné vlastnosti jako unitární matice viz definice v podkapitole 2.1.3. Objem výpočtů vzájemné koherence je daleko nižší než  $\text{spark} \mathbf{A}$ , vzájemnou koherenci ho můžeme ohraničit, konkrétně podle následujících dvou vět. Snažíme se nacházet slovníky, které jsou co nejvíce nekoherentní.

### Věta 3.2 [5]

Pro libovolnou matici  $\mathbf{A}$  platí

$$\text{spark}(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{A})} \quad (3.7)$$

### Věta 3.3 [5]

Pokud má soustava  $\mathbf{A}\mathbf{x} = \mathbf{b}$  řešení  $\mathbf{x}$  splňující

$$\|\mathbf{x}\|_0 < \frac{1}{2} \left( 1 + \frac{1}{\mu(\mathbf{A})} \right), \quad (3.8)$$

pak  $\mathbf{x}$  je nutně nejřidší možné a jediné takové. Navíc toto řešení lze dosáhnout  $l_1$ -minimalizací.

### NSP

Null Space Property se překládá jako vlastnost nulového prostoru. Pokud slovník splňuje NSP, je zajištěna jednoznačnost a existence  $k$ -řídkého řešení a je možné ho nalézt  $l_1$ -minimalizací. Také uvedeme větu, která určuje omezení chyby aproximace řešení  $l_1$ -minimalizace.

### Definice 3.3 [5]

Řekneme, že matice  $\mathbf{A}$  s komplexními prvky typu  $n \times m$  splňuje NSP řádu  $k$  s konstantou  $\gamma \in (0,1)$ , pokud platí rovnice (3.9)  $\forall T \subset \{1, \dots, m\}$ ,  $|T| \leq k$  a pro  $\forall \eta \in \ker \mathbf{A}$ .

$$\|\eta_T\|_1 \leq \gamma \|\eta_{T^c}\|_1, \quad (3.9)$$

### Věta 3.4 [5]

Nechť matice  $\mathbf{A}$  splňuje NSP řádu  $k$  s konstantou  $\gamma \in (0,1)$ . Dále necht  $\mathbf{x} \in \mathbb{C}^m$ ,  $\mathbf{b} = \mathbf{A}\mathbf{x}$  a  $\mathbf{x}^* \in \mathbb{C}^m$  je řešení  $l_1$ -minimalizace. Pak platí nerovnice

$$\|\mathbf{x} - \mathbf{x}^*\|_1 \leq \frac{2(1+\gamma)}{1-\gamma} \sigma_k(\mathbf{x})_1 \quad (3.10)$$

### RIP

Vlastnost zeslabené izometrie, anglicky Restricted Isometry Property, je výhodnější než NSP a také je stabilnější pod vlivem šumu.

### Definice 3.4

Nechť  $\mathbf{A}$  je matice s komplexními prvky typu  $n \times m$  pak nejmenší číslo  $\delta_k$  splňující rovnici

$$(1 - \delta_k) \leq \frac{\|\mathbf{A}\mathbf{z}\|_2^2}{\|\mathbf{z}\|_2^2} \leq (1 + \delta_k) \quad (3.11)$$

$\forall \mathbf{z} \in \Sigma_k^m$  nazveme konstanta omezené izometrie. Řekneme, že matice  $\mathbf{A}$  splňuje RIP řádu  $k$  s konstantou  $\delta_k$ , pokud  $\delta_k \in (0,1)$ .

Matice splňující RIP s konstantou  $\delta_k$  musí mít všechny podmatice o  $k$  sloupcích přibližně ortogonální. Povoluje se tedy malá odchylka  $\delta_k$  od přesné izometrie<sup>2</sup>. Do definice jsme zahrnuli všechny podmatice o  $k$  sloupcích, protože dopředu nevíme, které sloupce slovníku se na signálu  $b$  podílí a které ne.

Chybu aproximace  $l_1$ -optimalizací můžeme určit i pomocí RIP a tady existuje vztah mezi NSP a RIP. Toto propojení popisuje následující věta.

### Věta 3.5 [6]

Nechť matice  $\mathbf{A}$  s komplexními prvky typu  $n \times m$  splňuje RIP řádu  $K = k+h$  s konstantou  $\delta_K \in (0,1)$ . Potom  $\mathbf{A}$  splňuje NSP řádu  $k$  s konstantou

$$\gamma = \sqrt{\frac{k}{h} \frac{1 + \delta_K}{1 - \delta_K}}. \quad (3.12)$$

Tato podmínka není příliš vhodná pro praktické výpočty, proto zavedeme silnější podmínku, která odchylku od řešení ohraničí shora.

### Věta 3.6 [5]

Nechť  $\mathbf{A}$  splňuje RIP řádu  $2k$  s konstantou  $\delta_{2k} < \sqrt{2} - 1$ , pak pro nějakou konstantu  $C$  platí nerovnosti

$$\|\mathbf{x} - \mathbf{x}^*\|_2 \leq C \frac{\sigma_k(\mathbf{x})_1}{\sqrt{k}}, \quad (3.13)$$

$$\|\mathbf{x} - \mathbf{x}^*\|_1 \leq C \sigma_k(\mathbf{x})_1. \quad (3.14)$$

Podívejme se také jak se změní úloha, když pracujeme s šumem (zde se omezíme pouze na bílý šum).

---

<sup>2</sup>Izometrie je lineární zobrazení, které zachovává délku vektorů (je unitární).



### Věta 3.7 [5]

Nechť matice  $\mathbf{A}$  s komplexními prvky, typu  $n \times m$  vyhovuje RIP řádu  $2k$  s konstantou

$$\delta_{2k} < \frac{2}{3 + \sqrt{7/4}}.$$

Dále necht' měření jsou zatížena šumem:  $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}$ ,  $\|\mathbf{e}\|_2 \leq \epsilon$ ,  $\mathbf{x}^*$  je řešením úlohy

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \text{vzhledem k} \quad \|\mathbf{A}\mathbf{z} - \mathbf{b}\|_2 \leq \epsilon \quad (3.15)$$

Potom

$$\|\mathbf{x} - \mathbf{x}^*\|_2 \leq C_1 \frac{\sigma_k(\mathbf{x})_1}{\sqrt{k}} + C_2 \epsilon$$

platí  $\forall \mathbf{x} \in \mathbb{C}^m$ , pro kladné konstanty  $C_1, C_2$  závisující jen na  $\delta_{2k}$ .

Celková chyba se pak skládá ze části, která závisí na řídkosti řešení a zvláště na část podmíněnou energií rušení.

Matice vyhovující RIP jsme uvedli na straně 4 ve výčtu matic vhodných pro použití jako slovník.

### 3.1.2. Lineární programování

Zmíníme se jen velice stručně. Lineární programování řeší problém nejřidšího řešení  $\mathbf{x}$  na základě

$$\min_{\mathbf{x} \in M} \mathbf{c}^T \mathbf{x} \quad \text{vzhledem k} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \leq 0. \quad (3.16)$$

Kde  $M$  je množina všech přípustných řešení a vektor  $\mathbf{c} \in \mathbb{R}^n$ . Někdy je výhodné situaci (3.2) převést na problém k řešení pro lineární programování. Tento algoritmus se vlastně jmenuje BP, který jsme už zmínili v přehledu metod patřících do relaxačních metod.

### 3.1.3. IRLS

IRSL znamená Iterative Reweighed Least Squares. Z názvu tušíme, že se jedná o postup využívající uvnitř MNČ.

#### Slovní popis algoritmu IRLS

Naším úkolem je najít optimální  $\mathbf{x}$  pro problém  $(P_p)$ . Řešíme úlohu pro konkrétní matici  $\mathbf{A}$ , vektor řešení  $\mathbf{b}$ ,  $t$  vyjadřující požadovanou přesnost a konstantu  $p \in (0, 1)$ .

Pro výpočty budeme potřebovat sloupcový vektor jedniček  $\mathbf{x}$  délky  $m$  a jednotkovou matici  $\mathbf{X}$  typu  $m \times m$ . Začneme iteraci výpočtem

$$\mathbf{x}_k = (\mathbf{X}_{k-1}^2 \mathbf{A}^T (\mathbf{A} \mathbf{X}_{k-1}^2 \mathbf{A}^T)^+ \mathbf{b})$$

Pokročíme k

$$\mathbf{X}_k(j, j) = |\mathbf{x}_k(j)|^{1-p/2}$$

a zjistíme, jestli naše řešení  $\mathbf{x}_k$  už má dostatečnou přesnost

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2 < t,$$

podle situace buď následuje další iterace, nebo vektor  $\mathbf{x}_k$  prohlásíme hledaným řešením.

## 3.2. Hladové algoritmy

Postupy jsou založeny na vyhledání jednoho (či více atomů), které v dané iteraci mají největší důležitost vzhledem k řešenému problému. Dále se takto vybrané atomy už nezabývají významu na výsledném řešení.

### Podmínky pro nalezení řešení

K hladovým algoritmům potřebujeme zavést několik předpokladů. Slovník  $\mathbf{A}$  má  $\text{spark}(\mathbf{A}) > 2$  [5], [4], hodnota (3.1) v optimálním řešení rovno jedné. Vektor  $\mathbf{b}$  se tím pádem předpokládá skalárním násobkem jednoho atomu matice  $\mathbf{A}$ . Pro tyto předpoklady má (3.1) jednoznačné řešení.

Hladové algoritmy v  $j$ -té iteraci vynásobí postupně všechny atomy matice  $\mathbf{A}$  s vektorem  $\mathbf{b}$  a vytvoří  $j$  složek vektoru  $\epsilon(j)$  :

$$\epsilon(j) = \min_{\mathbf{z}_j} \|\mathbf{a}_j \mathbf{z}_j - \mathbf{b}\|_2^2. \quad (3.17)$$

Pro  $\mathbf{z}_j$  použijeme rovnici [4]

$$\mathbf{z}_j = \mathbf{a}_j^T \mathbf{b}^T / \|\mathbf{a}_j \mathbf{z}_j - \mathbf{b}\|_2^2.$$

Iterace provádíme tak dlouho, až najdeme řešení, které uspokojuje námi zvolenou dovolenou odchylku od přesného řešení (3.1).

### 3.2.1. OMP

Algoritmus OMP má specifickou strategii při hledání řešení problému rovnice (3.1), která je založena na ekvivalenci úloh hledání nejmenšího  $\epsilon(j)$  a největšího násobku mezi residuem  $\mathbf{r}^{k-1}$  a normovanými atomy matice  $\mathbf{A}$ . V metodě OMP podle výše popsaných podmínek rovnice (3.1) přejde na rovnici (3.2) [4].

$$\epsilon(j) = \|\mathbf{r}^{k-1}\|_2^2 - \frac{(\mathbf{a}_j^T \mathbf{r}^{k-1})^2}{\|\mathbf{a}_j\|_2^2} \quad (3.18)$$

### Slovní popis metody OMP

Máme řešit (3.1) pro zadaný slovník  $\mathbf{A}$  a vektor výstupního signálu  $\mathbf{b}$ , také je zadána požadovaná přesnost  $t$ .

Definujeme si na počátku nulový sloupcový vektor  $\mathbf{x}$  o délce  $m$ , do residua  $\mathbf{r}$  přiřadíme vektor  $\mathbf{b}$  a vytvoříme prázdný nosič  $\mathbf{S}^k$  vektoru  $\mathbf{x}$ .

V každé iteraci pak nejprve nalezneme minimální složku  $\epsilon(j)$  podle rovnice 3.18 a přidáme její index do stávajícího nosiče  $\mathbf{S}^k$ , hledaného vektoru  $\mathbf{x}$ . Protože není nutné v dalším kroku pracovat s celou maticí  $\mathbf{A}$  (zbytečně by zpomalila výpočet), použijeme z ní pouze „výřez“. Ten nazveme  $\mathbf{A}_{\mathbf{S}^k}$  o rozměrech  $n \times |\mathbf{S}^k|$ , tedy délka atomu  $\times$  velikost nosiče.

Dále vektor  $\mathbf{x}_{\mathbf{S}^k}$  vytvoříme ze složek vektoru  $\mathbf{x}$  odpovídajících nosiči  $\mathbf{S}^k$ . Nové residuum  $\mathbf{r}^k$  spočteme z rovnice

$$\mathbf{r}^k = \mathbf{b} - \mathbf{A}_{\mathbf{S}^k} \mathbf{x}_{\mathbf{S}^k}.$$

Matice  $\mathbf{A}_{\mathbf{S}^k}$  je tím pádem ortogonální k  $\mathbf{r}^k$ . Iterace je zakončena porovnáním normy residua a předem zvolené tolerance  $\|\mathbf{r}^k\| < t$ , pokud podmínka není splněna tak se přirozeně přistoupí k další iteraci a postup se opakuje, než nerovnost vyhovuje.

### 3.2.2. MP

Metoda MP je založena na podobném principu jako OMP. Je starší, jednodušší, avšak jak to bývá, také méně přesná. Pro tuto metodu použijeme rovnici (3.1) s dosazením za  $z_j$  optimální tvar [4]

$$\mathbf{z}_j^* = \frac{(\mathbf{a}_j^T \mathbf{r}^{k-1})}{\|\mathbf{a}_j\|_2^2}. \quad (3.19)$$

Po dosazení (3.19) do (3.17) dostaneme rovnici (3.20), se kterou dále operujeme.

$$\epsilon(j) = \mathbf{a}_j \frac{(\mathbf{a}_j^T \mathbf{r}^{k-1})}{\|\mathbf{a}_j\|_2^2} - \mathbf{r}^{k-1} \quad (3.20)$$

### Slovní popis metody MP

Opět řešíme (3.1) pro zadaný slovník  $\mathbf{A}$  a vektor  $\mathbf{b}$  a zvolíme si požadovanou přesnost  $t$ .

Inicializujeme vektor  $\mathbf{x}$  a residuum  $\mathbf{r}$  stejně jako u OMP.

Začneme iteraci vyhledáním minimální složky  $j_0$  vektoru  $\epsilon(j)$ , který jsme získali řešením rovnice (3.20). Vektor  $\mathbf{x}^{k-1}$  získaný v minulé iteraci (případně pro  $k = 1$  v inicializační části postupu) nastavíme jako  $\mathbf{x}^k$  a do složky  $\mathbf{x}_{j_0}^k$  přiřadíme hodnotu součtu  $\mathbf{x}_{j_0}^k + \mathbf{z}_j^*$ . Spočítáme nové residuum dosazením do rovnice

$$\mathbf{r}^k = \mathbf{r}^{k+1} - \mathbf{z}_{j_0}^* \mathbf{a}_{j_0}.$$

Pokud  $\|\mathbf{r}^k\| < t$  iterace zastavíme a vektor  $\mathbf{x}$  je konečným řešením.

### 3.2.3. Normalizace

Pro metody OMP i MP lze obecně zavést normalizaci slovníku. Tato úprava není nezbytná, ale je vhodná a použijeme ji ve všech algoritmech. Snažíme se najít matici, ze které bude hladovými algoritmy nalezen stejný nosič jako u původní matice  $\mathbf{A}$ . Tato operace se nazývá normalizace a spočívá ve vynásobení základní matice  $\mathbf{A}$  diagonální maticí  $\mathbf{W}$  speciálního tvaru. Ta obsahuje na diagonále na  $j$ -té pozici hodnotu  $1/\|\mathbf{a}_j\|_2^2$ . Podle definice uvedené a dokázané v [4], můžeme normalizovat  $\mathbf{A}$  na matici  $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{W}$ . Proč se s tímto postupem vůbec zdržovat? S maticí  $\tilde{\mathbf{A}}$  se lépe a rychleji pracuje, než s původní maticí  $\mathbf{A}$ .

Pokud v algoritmu použijeme normalizovanou matici  $\tilde{\mathbf{A}}$ , je nutné spočtený vektor  $\tilde{\mathbf{x}}$  „odnormalizovat“, abychom dostali stejné řešení, jako kdybychom použili nenormalizovanou matici  $\mathbf{A}$ . Zřejmě se proces odnormalizace provádí  $\mathbf{x}^k = \mathbf{W}\tilde{\mathbf{x}}^k$ . Abychom viděli názorně, jak se normalizace funguje, uvedeme rovnici převzatou z [4] :

$$\mathbf{b} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}^k = \mathbf{A}\mathbf{W}\tilde{\mathbf{x}}^k = \mathbf{A}\mathbf{x}^k \Rightarrow \mathbf{x}^k = \mathbf{W}\tilde{\mathbf{x}}^k \quad (3.21)$$

Normalizaci pro program Matlab uvádíme na příloženém CD jako funkci s názvem `Normovani.m`.

### 3.3. Prahování

Tresholding, česky prahování, je založeno na lehce odlišném přístupu než hladové algoritmy MP a OMP. Jeho jednoduchost spočívá ve vybrání  $k$  indexů nejmenších složek  $\epsilon(j)$  a z nich minimalizací  $\| \mathbf{Ax} - \mathbf{b} \|_2^2$  získání vektoru  $\mathbf{x}$  s nenulovými složkami na pozicích odpovídajících nosiči.

Alternativní postup:

Z vektoru  $\epsilon(j)$ , který obdržíme ze zjednodušené rovnice

$$\epsilon(j) = \| \mathbf{b} \|_2^2 - (\mathbf{a}_j^T \mathbf{b})^2$$

vybereme indexy  $k$  nejmenších složek a zaznamenejme je do nosiče  $\mathbf{S}$ . Hledané řešení  $\mathbf{x}$  pak je „nejmenší“ vektor splňující  $\| \mathbf{Ax} - \mathbf{b} \|_2^2$  s nenulovými složkami podle nosiče  $\mathbf{S}$ .

## 4. KOMPRIMOVANÉ SNÍMÁNÍ

Komprimované snímání je jednou z aplikací v řídkých signálech. Využívá se například pro rychlé snímání záznamů z magnetické rezonance. Kompresi dat se mimo řídké signály realizuje nasbíráním všech dat, uděláním vhodné transformace, vyhodnocením koeficientů získaných transformací a nepoužití většiny z nich z důvodu zanedbání malých hodnot.

Komprimované snímání postupuje jinak. Pokud budeme předpokládat signál přibližně řídký v určité reprezentaci, můžeme signál snímat lineárně a neadaptivně. Také nemusíme spoustu dat „vyhodit z okna“, ale dostaneme právě tolik informací, kolik opravdu využijeme. Bázi budeme značit  $\Psi$  a signál jako  $\mathbf{z}$ , které se spočte jako  $\mathbf{z} = \Psi\mathbf{x}$ . Vektor  $\mathbf{x}$  je v této bázi  $k$ -řídký. Zavedeme snímací matici  $\mathbf{P}$  s reálnými prvky typu  $s \times m$ ,  $s \ll m$ , pro signál  $\mathbf{b}$  pak platí

$$\mathbf{b} = \mathbf{P}\mathbf{z} = \mathbf{P}\Psi\mathbf{x}. \quad (4.1)$$

Měřicí matici  $\mathbf{P}$  je vhodné uvažovat ve tvaru  $\mathbf{P} = \mathbf{R}\Phi$ , matice  $\mathbf{R}$  vznikne výběrem  $s$  řádků z  $\mathbf{I}$  o rozměrech  $m \times m$  a  $\Phi$  o rozměrech  $m \times m$  specifikujeme později. Součin  $\mathbf{R}$  a  $\Phi$  je vlastně jako náhodný výběr řádků z  $\Phi$  a řídí se rovnoměrným rozdělením pravděpodobnosti. Slovník je u komprimovaného snímání tvaru  $\mathbf{A} = \mathbf{R}\Phi\Psi$ , přirozeně se snažíme opět řešit rekonstrukci pomocí  $l_1$ -minimalizace s obměnou rovnice (3.2) :

$$\mathbf{x}^1 = \arg \min \|\mathbf{x}\|_1 \quad \text{vzhledem k} \quad \mathbf{b} = \mathbf{P}\Psi\mathbf{x} \quad (4.2)$$

Existuje několik podmínek pro vybrání  $\Phi$ , pro jednu uvedenou dále budeme potřebovat speciální případ vzájemné koherence  $\mu$ , pro matici složenou ze dvou ortonormálníchází  $\Psi$  a  $\Phi \rightarrow [\Psi, \Phi]$  a tedy

$$\mu([\Psi, \Phi]) = \max_{1 \leq i, j \leq m} |\psi_i^T \phi_j| \quad (4.3)$$

hodnota této úpravy koherence padne do intervalu  $\frac{1}{\sqrt{m}}$  a 1 [5]. Podmínka pro zaručení přesné rekonstrukce z  $s$  měření je vyjádřena následující větou.

### Věta 4.1 [5]

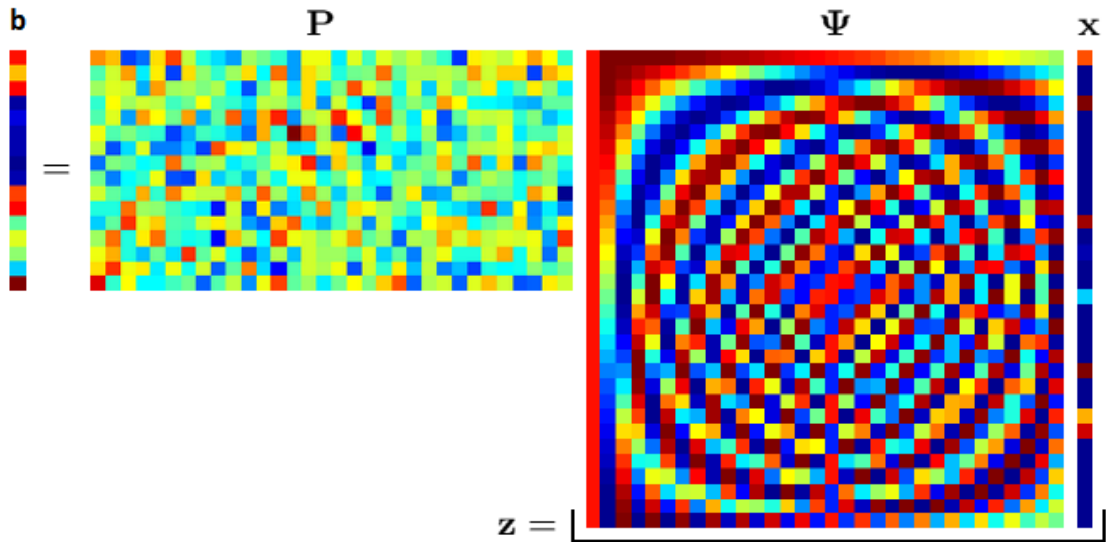
*Nechť je dán signál  $\mathbf{z}$ , který má v matici  $\Psi$   $k$ -řídkou reprezentaci  $\mathbf{x}$ . Pak řešení  $l_1$ -minimalizace (4.2), kde  $\mathbf{y}$  jsou měření, je zároveň s vysokou pravděpodobností nejřidší možné, jestliže je zvolíme počet vzorků  $s$*

$$s \geq C \cdot \mu^2([\Phi, \Psi]) \cdot k \cdot m \cdot \ln m \quad (4.4)$$

pro nějakou konstantu  $C > 0$ .

Pak počet měření a řídkost na sobě záleží pouze lineárně [5] a z koherence mezi ortobázemi kvadraticky ovlivňuje nutný počet měření. Opět se tedy budeme snažit najít dvojice ortobází s minimální koherencí. Úplně minimální koherence dosahuje  $[\mathbf{I}, \mathbf{F}]$ .

Důležitou úlohu ve volbě  $\Phi$  hraje RIP, budeme uvažovat možnosti  $\mathbf{P} = \mathbf{R}\Phi$  s maticí  $\Phi$  pouze gaussovskou nebo bernouliovskou (popsali jsme je na straně 4). Volba padla konkrétně na tyto typy matic, protože jsou invariantní vzhledem k unitární transformaci. Pokud tedy najdeme matici  $\mathbf{P}$ , která splňuje RIP, pak tuto vlastnost přebírá i  $\mathbf{P}\Phi$  také a v konečné fázi i slovník  $\mathbf{A}$  splňuje RIP a nezáleží ortonormální bázi, ve které je signál



Obrázek 4.1: [7] Situace matic při komprimovaném snímání. Snímáme vektor  $\mathbf{z}$ , ten sám o sobě není řídký, ale je v nějaké bázi.

řídký.

Na obrázku 4.1 vidíme na levé straně rovnice naměřený vektor  $\mathbf{y}$  o velikosti  $s \times 1$ , který se rovná součinu měřicí matice  $\mathbf{P} = \mathbf{R}\Phi$  a vektoru  $\mathbf{z}$ , řídkýchov bázi  $\Psi$ . Vektor  $\mathbf{z}$  můžeme pozorovat, je roven součinu  $\Psi$  a  $\mathbf{x}$ . V roli  $\Psi$  na obrázku 4.1 vystupuje ortonormální zpětná báze DCT.

# 5. APLIKACE POZNATKŮ V PROGRAMU MATLAB

Přišel čas abychom vyzkoušeli a ověřili teorii, která byla zaznamenána v předchozích kapitolách. Hlavní část kapitoly 5 popisuje programy `HlavniProgram.m`, `porovnani.m` a `Realne_signaly.m`, jejichž skripty lze nalézt na přiloženém CD a popis experimentů, na nich prováděných. Smyslem vytvoření programu je, jak už jsme si řekli, ověření teorie řídkých signálů. V programu `HlavniProgram.m` budeme testovat konkrétně na vygenerovaných tónech<sup>1</sup> s frekvencemi odpovídajícími tónům klavíru.

Klavír je strunný úderový hudební nástroj, jako takový slouží pro přehrávání tónů určitých frekvencí. V reálném světě klavírní tóny mají výšku, délku, sílu a barvu. Výška je dána frekvencí, délka dobou znění tónu, síla amplitudou (v tomto textu budeme za sílu tónu zmiňovat jako hlasitost) a barva záleží na spektrálním složení zvuku, je specifická pro určitý každý druh hudebního nástroje. Pro programy v tomto textu se s barvou tónu nebudeme zabývat, jen s frekvencí, dobu trvání a hlasitostí.

Klavírní skladby představují příklad řídkých signálů v reálném světě. Tedy je možné na ně aplikovat teorii řídkých signálů.

Abychom nezůstávali pouze u umělých signálů, zkusíme v programu `Realne_signaly.m` aplikovat metody MP a OMP na skutečné zvukové signály a vyhodnotíme, jak moc se liší jejich úspěšnost rekonstrukce od umělých signálů.

Testy uvedené v podkapitole 5.4 vychází z programu `srovnani.m`, který byl sestaven pro porovnávání metod OMP, MP a IRLS v závislosti na více druzích „zátěže“.

Použitou verzí programu Matlab, ve které byly napsány skripty je R2010b.

## 5.1. Simulace tónů a tvorba slovníku

### 5.1.1. Simulace tónů klavíru v Matlabu

Výpočet tónů provádíme v podle frekvencí, jako počáteční vezmeme 440 Hz a postupujeme geometrickou řadou s kvocientem  $\sqrt[12]{2}$  až k 880 Hz. Pracujeme tedy s 13 –ti po sobě jdoucími pultóny. Frekvence 440 Hz náleží komornímu *a*.

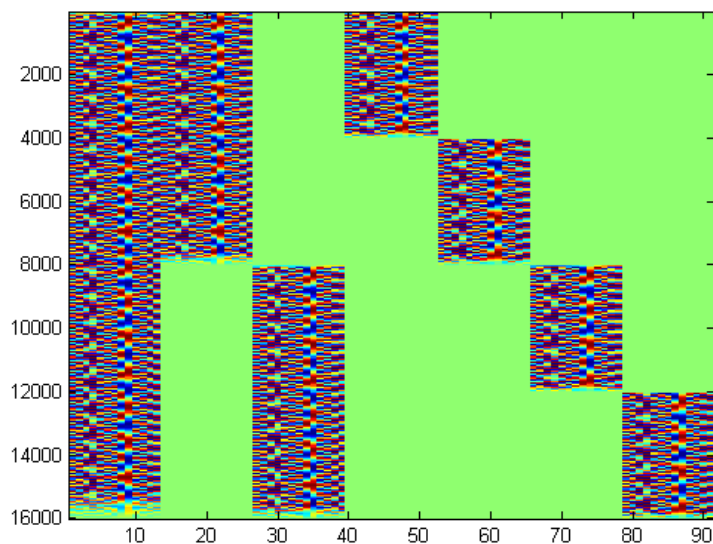
Abychom dobře porozuměli programu je třeba několika větami zmínit pojem vzorkovací frekvence. Signál se kterým pracuje počítač není spojitý, ale diskrétní. Snímání signálu počítač provádí s určitou frekvencí, tato frekvence se nazývá vzorkovací.

Podmínku pro velikost vzorkovací frekvence definuje Nyquist—Shannonův [10] vzorkovací teorém který říká, že pokud vezmeme vzorkovací frekvenci  $Fs \geq 2 \cdot f_{max}$ , je možná perfektní rekonstrukce signálu<sup>2</sup>. Touto podmínkou se předchází jevu, který se nazývá aliasing<sup>3</sup>. Vzorkovací frekvence se uvádí v jednotkách  $s^{-1}$  nebo v Hz. Velikost vzorkovací frekvence se používá pro každý přístroj jiná. Pro kompaktní disky je to například  $Fs = 44,1$  kHz. V programu budeme většinou pracovat se vzorkovací frekvencí  $Fs = 16000$  vzorků za vteřinu, která je odpovídající pro zvukové signály. Pouze v kapitole 5.3 se setkáme s

<sup>1</sup>Pojmem tón se označuje každý zvuk s konstantní frekvencí.

<sup>2</sup> $f_{max}$  je největší frekvence obsažená v signálu.

<sup>3</sup>Jev, který nastává při vzorkování spojitého signálu, když k němu dojde, navzorkovaný signál je zničený a rekonstrukce není možná.



Obrázek 5.1: Matice  $\mathbf{D}_i$ , vytvořená Matlabem příkazem `imagesc()`. Zelená barva představuje nulové hodnoty, barevné bloky pak matice  $\mathbf{T}_1$ ,  $\mathbf{T}_2$  a  $\mathbf{T}_3$ , podle velikostí je zřejmé, která část  $\mathbf{D}_i$  patří které době.

jinou hodnotou vzorkovací frekvence.

Jako generátor frekvencí o určité délce slouží funkce `klavir.m`. Vstupují do něj proměnné vzorkovací frekvence  $Fs$  a doba. Funkce vytvoří interval  $t$ , který odpovídá žádané době. Do matice vytvoří 13 sloupců  $\mathbf{D}$  o počtu řádků odpovídajících intervalu a do sloupců vloží sinusovky vygenerované funkcí `generuj_sin.m`.

Když stiskneme klávesu skutečného klavíru, zazní tón, který nemá konstantní hlasitost. Na začátku a konci je trošku ztišený, proto pro matici s tóny ve sloupcích potřebujeme této situaci přizpůsobit. Aplikujeme na ni funkci `Obalka.m`, ve které předpokládáme, že ztišení probíhá lineárně. Ztišení na začátku a konci nemá stejné trvání, i to je zohledněno ve výše zmíněné funkci.

### 5.1.2. Tvorba slovníku

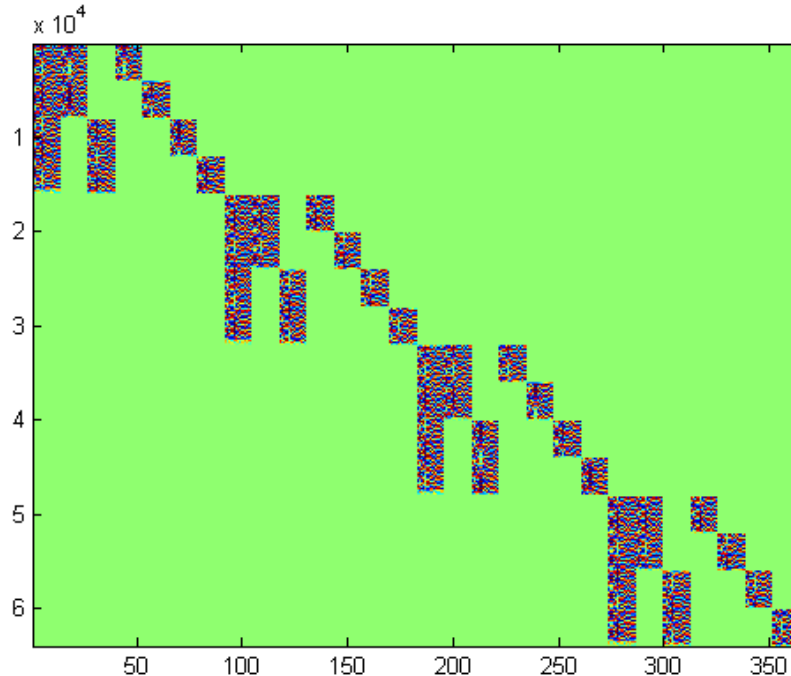
Simulované melodie klavíru jsou zcela jistě řídké ve slovníku s atomy tvořenými jednotlivými tóny. Z funkce `Melodie.m` načteme informace o tónech a délkách zadaných do matice<sup>4</sup>. Zřejmě pro řešení úloh budeme potřebovat slovník a to celkem specifický.

Pro vysvětlení si zavedeme pomocnou matici  $\mathbf{T}$ . Jelikož pracujeme s tóny klavíru, naplníme sloupce jeden za druhým posloupností vypočtených vektorů podle výše zmíněných frekvencí. Matice  $\mathbf{T}$  má zjevně 13 sloupců, jako použitých frekvencí. Počet řádků odpovídá délce tónu, kterou právě potřebujeme. V programu budeme pracovat pouze s tóny trvajících 1 s, 0,5 s a 0,25 s, kterým budeme říkat celé, půlové a čtvrté. Pro tyto tři případy si vytvoříme matice  $\mathbf{T}_1$ ,  $\mathbf{T}_2$  a  $\mathbf{T}_3$  funkcí `klavir.m`, popsanou v minulém odstavci. Pro tvorbu celkového slovníku budeme uvažovat co nejjednodušší situaci. Vytvoříme matici  $\mathbf{D}_i$ <sup>5</sup> typu  $16000 \times 91$ , která má na určitých místech, zřejmých z obrázku 5.1 vložené matice  $\mathbf{T}_1$ ,  $\mathbf{T}_2$  a  $\mathbf{T}_3$  a jinde má nulové prvky. Matici  $\mathbf{I}$  o velikosti počtu zadaných tónů

<sup>4</sup>Zadání je konkrétně ve funkci `klavir.m`, kterou nalezneme na příloženém CD

<sup>5</sup>Zkratka pro diagonální blok, protože tak bude fungovat ve slovníku.





Obrázek 5.2: Slovník  $\mathbf{A}$  pro čtyři tóny, vysvětlení částí obrázku je stejné jako u 5.1

a  $\mathbf{D}_i$  vynásobíme Kroneckerovým součinem, který jsme definovali na straně 6 a dostaneme celkový slovník pro zadané tóny  $\mathbf{A}$  typu  $64000 \times 364$ . Můžeme si ho prohlédnout na obrázku 5.2.

### 5.1.3. Simulace zadaného signálu

V části skriptu `HlavniProgram.m` s názvem *Vytvoření vektoru  $\mathbf{b}$  ze slovníku  $\mathbf{A}$* , je vytvořen vektor  $\mathbf{x}$  podle na zadání. Provedením součinu slovníku a tohoto vektoru dostáváme zadaný signál  $\mathbf{b}$ . Tedy provedeme operaci

$$\mathbf{b} = \mathbf{A}\mathbf{x}.$$

### 5.1.4. Metody OMP a MP aplikované na signál

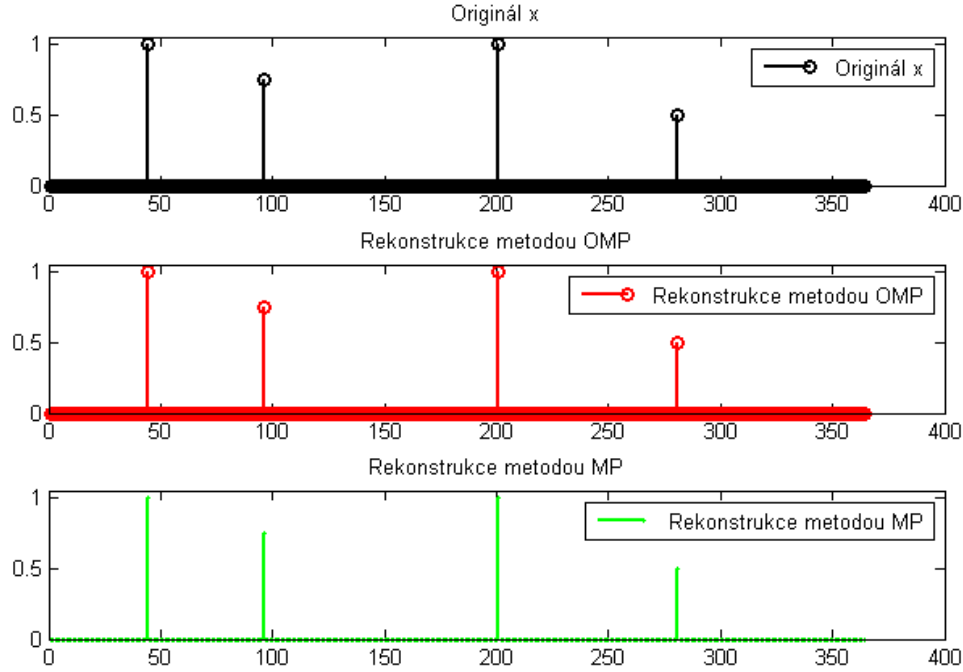
Skript `HlavniProgram.m` také vykreslí pro srovnání grafy vektorů  $\mathbf{x}$  rekonstruovaných metodami MP a OMP ze slovníku  $\mathbf{A}$ , signálu  $\mathbf{b}$  při toleranci  $t = 0,01$  a matici zadaných hodnot ve funkci `Melodie.m`. Výsledek můžeme vidět na obrázku 5.3. Odchylka výsledků vypočítaná vzorcem (5.2) dává pro OMP  $2,4990 \cdot 10^{-15}$  a pro MP hodnotu  $9,3856 \cdot 10^{-16}$ . Obě odchylky jsou tedy tak malé, že je můžeme považovat za nulové.

## 5.2. Komprimované snímání

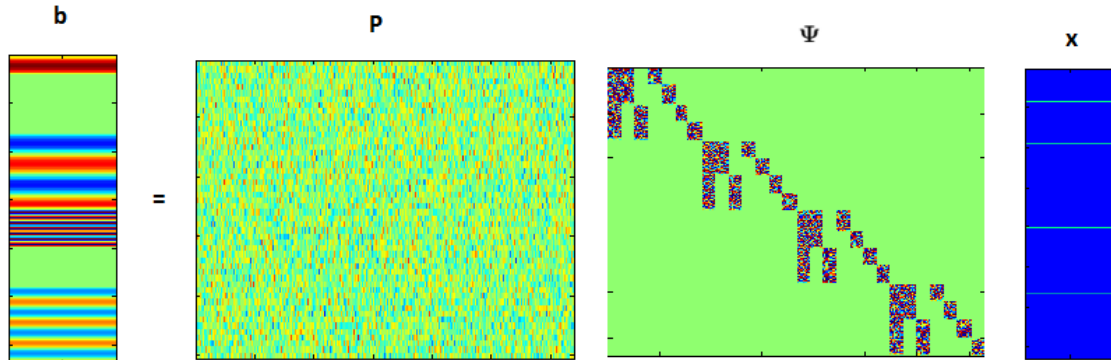
V kapitole 4 jsme si zavedli potřebné pojmy, nyní si ukážeme aplikaci. Základní rovnice vypadá takto:

$$\mathbf{b} = \mathbf{P}\Psi\mathbf{x}. \quad (5.1)$$

Pro naši situaci reprezentuje matici  $\Psi$  původní slovník  $\mathbf{A}$ , matici  $\mathbf{R}$  vynecháme, protože vygenerujeme gaussovskou matici  $\Phi$  rovnou typu  $s \times m$  a označíme ji jako  $\mathbf{P}$ , což je ekvivalentní postup a přirozeně rychlejší.



Obrázek 5.3: Grafy znázorňují vektor  $\mathbf{x}$  získaný ze zadání a jeho výsledky jeho rekonstrukce metodami OMP a MP ze slovníku  $\mathbf{A}$  a signálu  $\mathbf{b}$ , při toleranci  $t = 0,01$ , na svislé ose je vynášena „hlasitost“



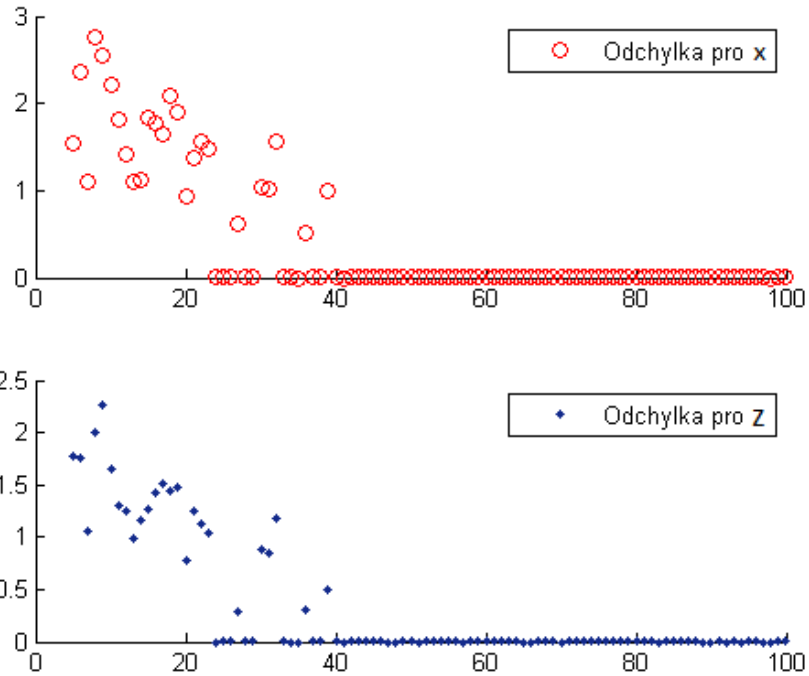
Obrázek 5.4: Ilustrace konkrétní situace (5.1).

Celý experiment provádíme opět pro zadání ze skriptu `Melodie.m`, tedy máme zadané pouze čtyři tóny. Konkrétní situaci vidíme na obrázku 5.7.

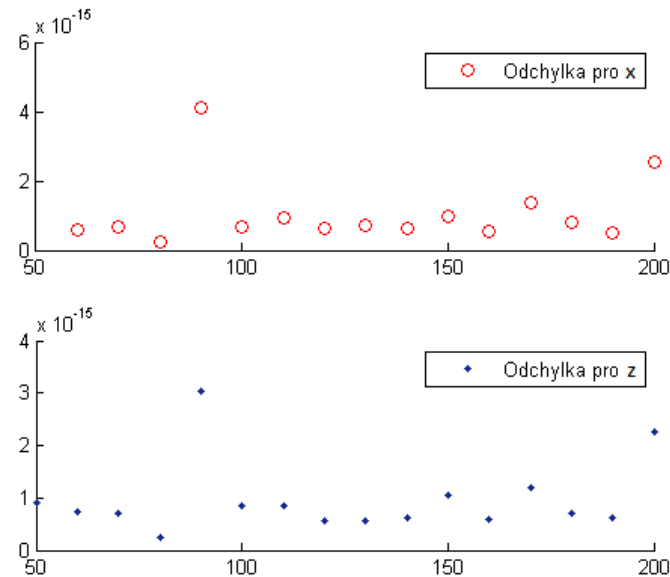
Abychom mohli porovnat úspěšnost komprimovaného snímání musíme provést následující. Vezmeme  $s$  a vygenerujeme prvky matice  $\mathbf{P}$  typu  $s \times m$  z normálního rozdělení o parametrech  $\mu = 0$  a  $\sigma^2 = 1/s$ . Spočteme vektor  $\mathbf{b}$  z rovnice 5.1. Hodnoty vektoru  $\mathbf{b}$  jsou výsledky „měření.“ Vektor  $\mathbf{b}$  dosadíme do metod OMP a MP, za slovník vezmeme součin matic  $\mathbf{P}$  a  $\Psi$ , jako toleranci ponecháme  $t = 0,01$ . Tím získáme vektor  $\mathbf{x}$ . Opět použijeme rovnici 5.2 pro odpovídající hodnoty a vykreslíme do grafu.

Rekonstruovali jsme pro porovnání vektor  $\mathbf{x}$ , následuje přirozeně ještě porovnání rekonstrukce skutečného signálu, vektoru  $\mathbf{z}$ . To provedeme násobením matic  $\Psi \mathbf{x}$ . Odchylku spočteme z rovnice 5.2 a zaneseme do grafu.

Tento postup provedeme pro  $s = \{5, 6, 7, \dots, 100\}$  a zaneseme do obrázku 5.5, poté ještě znova pro další množinu  $\{50, 60, 70, \dots, 200\}$ , výsledek vykreslíme na obrázek 5.6. Z obrázku 5.5 vidíme, že rekonstrukce je velmi úspěšná, z celkových 64000 hodnot vekt-



Obrázek 5.5: Odchylky od správného řešení při komprimovaném snímání pro vektory  $\mathbf{x}$  a  $\mathbf{z}$  vynášíme na svislou osu a na osu vodorovnou prvky  $z$  první množiny



Obrázek 5.6: Na vertikální ose jsou vyneseny odchylky od správného řešení při komprimovaném snímání pro vektory  $\mathbf{x}$  a  $\mathbf{z}$  pro druhou množinu, s hodnotami vyneseny na horizontální osu

toru  $\mathbf{b}$  nám stačí nejméně 40 vzorků, aby byla rekonstrukce  $\mathbf{x}$  i  $\mathbf{z}$  úspěšná! Jistě jsme vzali vektor  $\mathbf{x}$  opravdu hodně řídký, ale i pro méně řídké vektory by výsledek byl velmi optimistický. Obrázek 5.6 zde uvádíme proto, abychom jasně viděli jak velmi malé odchylky při komprimovaném snímání dostáváme.

### 5.3. Rekonstrukce skutečného signálu

Do skriptu `Realne_signaly.m` jsme načetli dva soubory typu `.wav`, nahrávku skutečného elektrického klavíru a také hlas papouška<sup>6</sup>. Použijeme zde jinou vzorkovací frekvenci o velikosti  $F_s = 22050$ .

Obě tyto nahrávky mají přibližně 5 s, tomu vytvoříme odpovídající slovník  $\mathbf{A}$  a provedeme rekonstrukci pomocí metod OMP a MP. Signály jsou reálné a proto nastavíme toleranci na velikost  $t = 76$ . Pokud bychom pracovali s tolerancí jako u vygenerovaných signálů, tedy kolem 0,01, kloudného výsledku bychom se nedočkali.

Výsledkem programu jsou hodnoty euklidovských norem rozdílu vektorů skutečných signálů  $\mathbf{b}$  a rekonstruovaných metodami MP a OMP. Konkrétně výsledek záznamu klavíru metodou MP měl ve výše popsané normě velikost 0,9007 a metodou OMP 0,8993. Pro druhou nahrávku jsme z výsledku metodou MP získali hodnotu 75,9801 a OMP 75,8672.

Na těchto příkladech je vidět, že metoda OMP je skutečně trochu přesnější než MP. Jak jsme předpokládali, rekonstrukce záznamu klavíru dopadla daleko lépe než hlasu papouška. Je to způsobeno charakterem signálu, klavírní zvuk obsahuje tóny podobné těm, které máme uložené ve slovníku. Jeden z faktorů ovlivňujících rekonstrukci je i nestejná délka trvání nahraných tónů klavíru a umělých tónů v atomech slovníku.

### 5.4. Srovnání rekonstrukce OMP, MP a IRLS

Srovnáme metody OMP a MP v závislosti na toleranci a OMP, MP a IRLS při zašuměných datech, skripty metod jsou uvedeny na CD.

Jelikož simulace úloh se zašuměným signálem a úlohy používající IRLS jsou velmi výpočtově náročné, provedeme testy na jednodušším programu.

Jako první případ srovnáme úspěšnost metod OMP a MP aplikovaných na úlohu:

- slovník tvořený DCT o velikosti  $50 \times 50$
- 5-řádkový vektor  $\mathbf{b}$ , vygenerovaný funkcí `RidkySignal.m`
- velikost tolerance  $t \in \{0,1; \dots; 1\}$
- bez zašuměných dat

Konkrétní příkazy nalezneme ve skriptu `Porovnaní.m`. Výstup z programu si můžeme prohlédnout na obrázku 5.7. Graf na něm popisuje míru úspěšnosti rekonstrukce vektoru  $\mathbf{x}$  metodami OMP a MP pro hodnoty tolerované nepřesnosti. Toleranci  $t$  vynášíme na vodorovnou osu a příslušné odchylky  $z$  získané z rovnice

$$z = \frac{\|\mathbf{x} - \mathbf{x}_i\|_2}{\|\mathbf{x}\|_2} \quad (5.2)$$

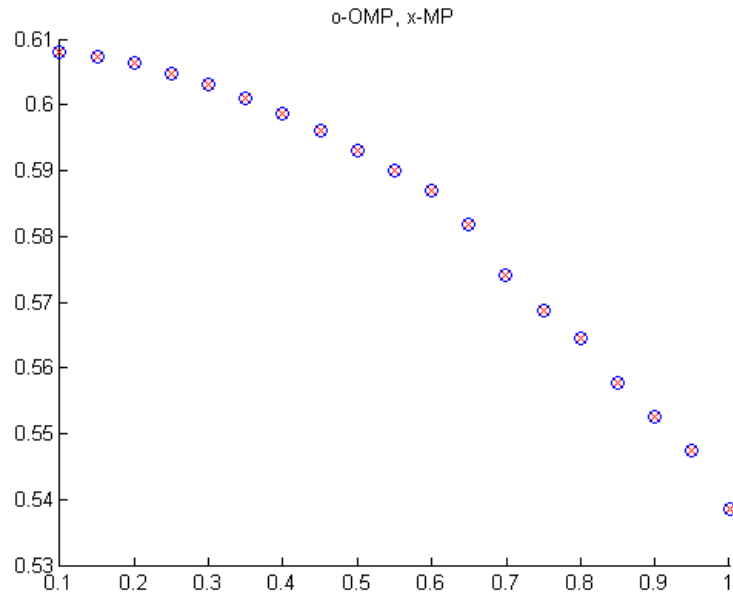
na osu svislou. Vektor  $\mathbf{x}$  představuje vektor optimálního řešení a  $\mathbf{x}_i$  konkrétní výsledek metody OMP, MP u grafu 5.9 IRLS.

Z grafu je vidět, že rozdíly mezi rekonstrukcí MP a OMP nejsou významné.

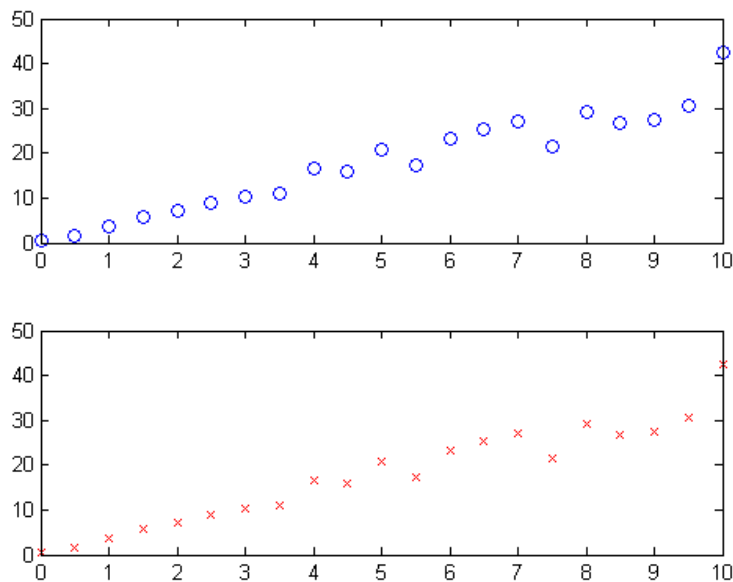
Následující test se zaměřuje na působení šumu na úspěšnost rekonstrukce. Výsledné

---

<sup>6</sup>Nahraný zvuk jsou slova papouška vlnkovaného (andulky), který říká: „Mám vás rád, usmívej se dítě nešťastný, život je krásný, jsem děsnej brebta“



Obrázek 5.7: Graf pro porovnání OMP a MP závislosti na toleranci.

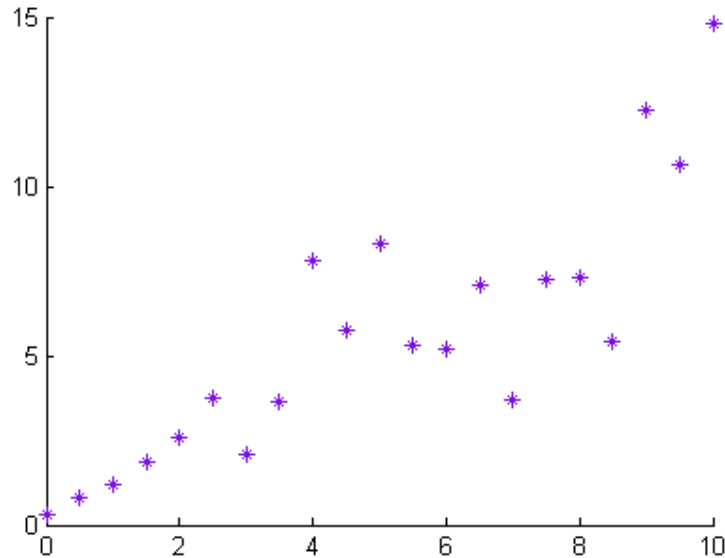


Obrázek 5.8: Graf pro porovnání působení šumu na rekonstrukci OMP (první graf),MP.

grafy můžeme vidět na obrázcích 5.8 a 5.9 Tentokráte máme na vodorovné ose velikost chybové konstanty, se kterou jsme zašuměli signál a na vodorovné máme opět odpovídající výsledky rovnice (5.2). Pracovali jsme při zadaných parametrech:

- slovník tvořený DCT o velikosti  $50 \times 50$
- 5-řádký vektor  $\mathbf{y}$
- velikost tolerance  $t = 0,1$
- velikost zašumění od 0 do 10 po krocích 0,5

Obrázek 5.9 zobrazuje úspěšnost rekonstrukce metodou IRLS v závislosti na přidaném šumu. Tento byl graf simulován za podmínek:



Obrázek 5.9: Graf pro porovnání působení šumu na rekonstrukci IRLS.

- slovník tvořený DCT o velikosti  $5 \times 5$
- 1-řádkový vektor  $\mathbf{y}$
- parametr pro funkci IRLS  $p = 1$
- velikost tolerance  $t = 10^{-14}$
- velikost zašumění od 0 do 10 po krocích 0,5

Zvolili jsme velmi malé parametry kvůli některým vlastnostem IRLS, zejména dlouhému trvání výpočtů. Ze zkušeností z výpočtů metod MP, OMP a IRLS můžeme prohlásit, že při stejně nastavených podmínkách se všem třem algoritmům zhoršuje úspěšnost rekonstrukce přibližně lineárně s téměř stejnou směrnici. Navíc odpovídající si hodnoty OMP a MP na vodorovné ose, jak je vidět na obrázku 5.9, jsou prakticky shodné.

## 6. ZÁVĚR

Cílem práce bylo nastudovat základní principy a metody hledání řídkých reprezentací, zaměřit se na typické situace v problematice zvukových signálů. Dále si empiricky vyzkoušet teoretické znalosti v programu Matlab na konkrétních datech.

V kapitole Základy problematiky řídkých signálů je popsán ústřední problém (2.1) a některé potřebné definice pro další text.

Druhá kapitola popisuje některé numerické metody, které do této problematiky spadají, rozděluje je podle přístupu k problematice na relaxační, hladové a prahování. Také určuje situace, pro které můžeme metody používat a je v ní zmíněn pojem normalizace matice a její význam.

Následuje část věnující se komprimovanému snímání. Obsahuje velmi stručný náhled do této problematiky.

Poslední kapitola popisuje simulaci tónů klavíru, ověřuje teorii uvedenou v předcházejících částech a se skládá z popisování testů provedených na programech `HlavniProgram.m`, `Realne_signaly.m` a `Porovnaní.m`, které jsou uvedené v přílohách a na CD, a také z jejich vyhodnocování.

Z provedených testů vidíme, že rekonstrukce metodami OMP a MP na skriptu `HlavniProgram.m` jsou pro naše případy v podstatě dokonalé. Při snížení řídkosti se přirozeně odrazí na výsledcích rekonstrukce, které se budou zhoršovat, ale podstatné odchylky se objeví až pro velké snížení řídkosti.

Pozorovali jsme úspěšnost rekonstrukce pro reálné nahrávky elektrického klavíru a zvuku hlasu papouška. Druhou nahrávku jsme zařadili, abychom ukázali, že úspěšnost rekonstrukce závisí na použitém slovníku.

Komprimované snímání je rychle se rozvíjející se disciplína, která nalezla uplatnění například při rekonstrukce obrazu z magnetické rezonance. Její výhodou je nižší počet měření, tedy nižší počet dat na úschovu a uspokojujivá možnost rekonstrukce skutečného signálu. Komprimované snímání jsme úspěšně vyzkoušeli ve skriptu `HlavniProgram.m` a kontrolovali jsme odchylky rekonstrukce vektoru  $\mathbf{x}$  a signálu  $\mathbf{z}$  v závislosti na počtu měření signálu.

Zkontrolovali jsme úspěšnost rekonstrukce metodami OMP a MP při různých hodnotách tolerance, a pro OMP, MP a IRLS jsme pozorovali zvětšující se odchylky při rostoucím zašumění signálu  $\mathbf{b}$ .

Všechny výpočty probíhaly v programu Matlab.

# LITERATURA

- [1] ADLER, A. et al., A Constrained Matching Pursuit Approach to Audio Declipping. Acoustics, Speech and Signal Processing, IEEE International Conference on (ICASSP 2011), IEEE, Prague, Czech Republic, 2011
- [2] BALAZS, P., DOERFLER, M. et al., Theory, Implementation and Application of Nonstationary Gabor Frames, J. Comput. Appl. Math., In Press. (2011)
- [3] BALZANO, L. Compressed Sensing Audio Demonstration. [Http://sunbeam.ece.wisc.edu/](http://sunbeam.ece.wisc.edu/) [online]. 19. září 2011 [cit. 2012-03-31]. Dostupné z: <http://sunbeam.ece.wisc.edu/csaudio/>
- [4] ELAD, M. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. New York: Springer, 2010. ISBN 97-1-4419-7010-7.
- [5] HRBÁČEK, R., RAJMIC P., VESELÝ V. a ŠPIŘÍK J.: Řídké reprezentace signálů: úvod do problematiky. Elektorevue. 2011, č. 50. ISSN 1213 - 1539. Dostupné z: <http://elektorevue.cz/cz/clanky/zpracovani-signalu/0/ridke-reprezentace-signalu--uvod-do-problematiky/>
- [6] HRBÁČEK, R.: Compressive sampling a simulace one-pixel camera: bakalářská práce. Brno: VUT, FEKT, Ústav telekomunikací, 2011. 48 s.
- [7] HRBÁČEK, R., RAJMIC, P., VESELÝ V. a ŠPIŘÍK J.: Řídké reprezentace signálů: komprimované snímání. Elektorevue - Internetový časopis [online]. 2011, č. 67 [cit. 2012-02-21]. ISSN 1213 - 1539. Dostupné z: <http://elektorevue.cz/cz/clanky/zpracovani-signalu/0/ridke-reprezentace-signalu--komprimovane-snimani/>
- [8] ŠPIŘÍK, J., RAJMIC a VESELÝ. Reprezentace signálů: od bází k framům. Elektorevue - Internetový časopis [online]. 2010, č. 111 [cit. 2012-03-15]. ISSN 1213 - 1539. Dostupné z: <http://elektorevue.cz/cz/clanky/zpracovani-signalu/0/ridke-reprezentace-signalu--komprimovane-snimani/>
- [9] OLSZAR, V. Rychlé řešení sedlobodových soustav pomocí spektrálního rozkladu [online]. Ostrava, 2006 [cit. 2012-04-10]. Dostupné z: <http://am.vsb.cz/theses/bakalari/2007/pdfs/ols050.pdf>. Bakalářská práce. VŠB.
- [10] Sampling rate. In: Wikipedia, the free encyclopedia [online]. 2012 [cit. 2012-04-19]. Dostupné z: [http://en.wikipedia.org/wiki/Sampling\\_rate](http://en.wikipedia.org/wiki/Sampling_rate)



## 7. SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

<b>BP</b>	Basis Pursuit
<b>DCT</b>	Diskrétní kosinová transformace
<b>DFT</b>	Diskrétní fourierova transformace
<b>F</b>	Fourierova báze
<b>F<sub>s</sub></b>	Vzorkovací frekvence
<b>I</b>	Jednotková matice
<b>IRLS</b>	Iterative Reweighted Least Squares
<b>LASSO</b>	Least absolute shrinkage and selection operator
<b>LARS</b>	Least Angle Regresion
<b>MP</b>	Matching pursuit
<b>MNČ</b>	Metoda nejmenších čtverců
<b>NSP</b>	Null Space Property — Vlastnost nulového prostoru
<b>OMP</b>	Orghogonal matching pursuit
<b>RIP</b>	Restricted Isometry Property — Vlastnost zeslabené izometrie
<b>W</b>	Wavelety