



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MATEMATIKY**

INSTITUTE OF MATHEMATICS

**OPTIMALIZACE REPREZENTACE DAT PRO SLEDOVÁNÍ  
OBJEKTU SENZOROVOU SÍTÍ**

OPTIMIZATION OF DATA REPRESENTATION FOR TARGET TRACKING USING SENSOR NETWORK

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Klára Kubalcová**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**doc. Mgr. Pavel Rajmic, Ph.D.**

**BRNO 2017**



# Zadání bakalářské práce

Ústav:	Ústav matematiky
Studentka:	<b>Klára Kubalcová</b>
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	<b>doc. Mgr. Pavel Rajmic, Ph.D.</b>
Akademický rok:	2016/17

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:

## **Optimalizace reprezentace dat pro sledování objektu senzorovou sítí**

### **Stručná charakteristika problematiky úkolu:**

Práce se bude zabývat analýzou a optimalizací metod pro výměnu dat v senzorové síti. Konkrétněji jde o decentralizovanou síť, která bude mít za úkol sledovat polohu pohybujícího se objektu. Všechny senzory musejí mít v každém okamžiku dostatečné informace o poloze objektu a k jejímu zpřesnění je nutné vyměňovat relativně velké množství dat. Naštěstí o datech víme, že jsou tzv. řídká, což umožňuje přímou či nepřímou úsporu pomocí komprese dat. V práci bude tento fakt využit a nově navržené metody budou porovnány se staršími.

### **Cíle bakalářské práce:**

Studentka se seznámí s modelováním senzorových sítí a výměny dat a distribuovanými výpočty v nich. Studentka navrhne a implementuje algoritmus, který na základě předložených trénovacích dat najde v definivaném smyslu optimální reprezentaci pro sensorová data. Studentka dále svou metodu porovná se staršími, mj. z hlediska energetické náročnosti výpočtu, energetické náročnosti spojení a výměny dat a tuto analýzu zpracuje v textu své práce. Součástí práce budou programy v jazyku MATLAB.

### **Seznam doporučené literatury:**

HRBÁČEK, R., RAJMIC, P., VESELÝ, V. a ŠPIŘÍK, J. Řídké reprezentace signálů: komprimované snímání, *Elektrorevue*, 2011. ISSN 1213-1539

HRBÁČEK, R., RAJMIC, P., VESELÝ, V. a ŠPIŘÍK, J. Řídké reprezentace signálů: úvod do problematiky, *Elektrorevue*, 2011. ISSN 1213-1539

HRBÁČEK, R., RAJMIC, P. a VESELÝ, V. Reprezentace signálů: od bází k framům, *Elektrorevue*, 2010. ISSN 1213-1539

ELAD, M. Sparse and redundant representations. New York: Springer, 2010.

HLINKA, O., HLAWATSCH, F. and DJURIC, P., Consensus-based Distributed Particle Filtering With Distributed Proposal Adaptation. IEEE Transactions on Signal Processing. USA: IEEE, 62(12): pp. 3029-3041. ISSN 1053-587X.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

---

prof. RNDr. Josef Šlapal, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty





## **Abstrakt**

Tato bakalářská práce se zabývá hledáním optimální reprezentace dat pro sledování objektu senzorem sítí. Je zde popsán model decentralizované senzorové sítě a využití tzv. slovníku k reprezentaci naměřených dat. Dále je zde teoreticky rozebrán algoritmus K-SVD sloužící k učení slovníku a na základě modelových signálů jsou naučeny slovníky sloužící k reprezentaci dat. Tyto slovníky jsou mezi sebou porovnány.

## **Summary**

The aim of this bachelor thesis is to find optimal data representation for target tracking using sensor network. There is described a model of decentralized sensor network and also the application of so called dictionary to represent the measured data. Also, there is theoretically introduced the K-SVD algorithm that is used for dictionary learning and there are learnt dictionaries for data representation based on the model signals. These dictionaries are compared with each other.

## **Klíčová slova**

OMP, K-SVD, senzorová síť, atom, slovník, řídká reprezentace

## **Keywords**

OMP, K-SVD, sensor network, atom, dictionary, sparse representation

KUBALCOVÁ, K. *Optimalizace reprezentace dat pro sledování objektu senzorem sítí*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 29 s. Vedoucí doc. Mgr. Pavel Rajmic, Ph.D.

Prohlašuji, že svou bakalářskou práci na téma „Optimalizace reprezentace dat pro sledování objektu sensorovou sítí“ jsem vypracovala 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 autorka uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Klára Kubalcová



Děkuji především svému vedoucímu práce doc. Mgr. Pavlu Rajmicovi Ph. D. za odborné vedení práce, cenné rady a návrhy a časté konzultace. Dále také děkuji svojí rodině za trpělivost a podporu.

Klára Kubalcová



# OBSAH

Úvod	2
<b>1 Matematický úvod</b>	<b>3</b>
1.1 Norma a řídkost . . . . .	3
1.1.1 Základní definice . . . . .	3
1.1.2 Normy matic . . . . .	4
1.1.3 Příklady . . . . .	4
1.2 Řídká řešení systému lineárních rovnic . . . . .	4
1.2.1 Zavedení problému . . . . .	4
1.2.2 Metody nalezení vektoru koeficientů . . . . .	5
1.3 Singulární rozklad . . . . .	6
<b>2 Senzorová síť</b>	<b>7</b>
2.1 Model sensorové sítě . . . . .	7
2.2 Lokální věrohodnostní funkce . . . . .	8
2.3 Globální věrohodnostní funkce . . . . .	9
2.4 Využití slovníku k výpočtu globální věrohodnostní funkce . . . . .	11
<b>3 Hledání optimální reprezentace dat</b>	<b>13</b>
3.1 Algoritmus K-SVD . . . . .	13
3.1.1 K-means a řídká reprezentace signálu . . . . .	13
3.1.2 Popis algoritmu . . . . .	13
3.2 Vstupní údaje . . . . .	14
3.2.1 Signály . . . . .	14
3.2.2 Startovací slovníky . . . . .	14
<b>4 Rozbor výsledků</b>	<b>18</b>
4.1 Spočtené slovníky . . . . .	18
4.2 Porovnání s jinými metodami . . . . .	21
<b>5 Závěr</b>	<b>26</b>
<b>6 Seznam použitých zkratk a symbolů</b>	<b>28</b>
<b>7 Obsah příloženého DVD</b>	<b>29</b>

# ÚVOD

V sensorové síti, která sleduje pohybující se objekt, je důležité získávat co nejpřesnější informace, aby bylo možné co nejlépe určit měřené stavové veličiny objektu – především jeho polohu, případně také jeho rychlost nebo směr pohybu. Zároveň je také třeba přenášet informace co nejrychleji, aby je senzory byly schopné využít dříve, než se objekt přemístí a nebudou potřeba. Každá stavová veličina je ve stavovém prostoru popsána svojí hustotou pravděpodobnosti, která je typicky lokalizována. Tato hustota nabývá lokálního maxima nejen v okolí bodu příslušejícímu danému stavu, ale také na několika dalších pozicích, kde jsou zaznamenána falešná hlášení, a nelze jednoduše určit, které maximum náleží objektu. Protože lokálních maxim je jen několik, je výhodné pro splnění předpokladů přesnosti a rychlosti přenosu použít řídkou reprezentaci. Ta předpokládá, že každý sensor zná soubor základních signálů vycházejících z hustoty pravděpodobnosti, tzv. *slovník*, a pomocí sečtení několika těchto signálů, tzv. *atomů*, vynásobených koeficienty určenými na základě předchozího stavu, se určí nový signál a z něj pak i nový stav, ve kterém se nachází sledovaný objekt. Je tedy potřeba přenést jen několik hodnot – koeficientů.

Tato bakalářská práce se zabývá hledáním slovníku, který by nejlépe reprezentoval informace (data) při co nejmenším počtu použitých atomů. Tento slovník můžeme zvolit buď fixní, a nebo jej natrénovat pomocí modelových signálů. V této práci jsme zvolili druhou možnost, a k natrénování slovníku použili algoritmus K-SVD. Tato metoda ovšem není konvexní, výsledek je ovlivněn tím, jak ji inicializujeme. Proto jsme na inicializaci K-SVD zvolili různé startovací slovníky – jak typově, tak počtem atomů.

Práce se skládá ze čtyř kapitol. První kapitola popisuje základní matematické pojmy jako je norma,  $k$ -řidkost, nebo singulární rozklad, které jsou v práci použity. Její druhá část je pak věnována řídkým řešením soustavy lineárních rovnic a algoritmu OMP, který slouží k určení tohoto řešení. V druhé kapitole se seznámíme se sensorovou sítí, přenosem dat v této síti, a využitím slovníku. Třetí kapitola je věnována algoritmu K-SVD a jednotlivým vstupním údajům, především startovacím slovníkům. Ve čtvrté kapitole jsou pak rozebrány výsledky, které byly získány implementací algoritmu v prostředí Matlab, a porovnány s výsledky u fixních slovníků.



# 1. MATEMATICKÝ ÚVOD

V této kapitole se budeme zabývat základními matematickými definicemi a znalostmi, které se týkají norem, řídkých vektorů a řešení nedourčené soustavy lineárních rovnic, mimo jiné také pomocí algoritmu OMP, který se bude později využívat. Co se týče značení, v celé práci budeme vektory uvažovat sloupcově a značit je malými tučnými písmeny  $\mathbf{x}$ . Složky vektoru značíme malými písmeny s dolním indexem nebo do závorky uvedeme číslo prvku  $x_i = \mathbf{x}(i)$ ,  $\mathbf{x} = [x_1, \dots, x_n]^T = [x_1; \dots; x_n]^T$ . Matice budeme značit velkými tučnými písmeny  $\mathbf{A}$ , jejich prvky malými písmeny s indexy, kde první označuje řádek a druhý sloupec  $a_{ij}$ . Pokud se bude matic označovat indexem, budeme její prvky značit pomocí závorky jako  $\mathbf{A}_1(i, j)$ . Označení  $\mathbf{A}_{:j}$  nebo  $\mathbf{A}_1(:, j)$  používáme pro  $j$ -tý řádek matice  $\mathbf{A}$  nebo  $\mathbf{A}_1$ ; obdobně  $\mathbf{A}_{i\cdot}$  nebo  $\mathbf{A}_1(i, :)$  označuje  $i$ -tý sloupec.  $\mathbf{A}^*$  označuje hermitovskou transpozici matice  $\mathbf{A}$ . K definici nebo přiřazení budeme používat znak  $:=$ .

## 1.1. Norma a řídkost

### 1.1.1. Základní definice

**Definice 1.** [1] Nechť  $X$  je vektorový prostor,  $\mathbf{x}, \mathbf{y} \in X$  a funkce  $f : X \rightarrow \mathbb{R}_0^+$  je funkce splňující:

1.  $f(\mathbf{x}) = 0 \iff \mathbf{x} = 0$
2.  $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$
3.  $f(a \cdot \mathbf{x}) = |a| \cdot f(\mathbf{x})$

Pak funkci  $f$  nazveme normou ve vektorovém prostoru  $X$  a prostor  $(X, f)$  nazveme normovaný. Funkci  $f(\mathbf{x})$  budeme dále značit jako  $\|\mathbf{x}\|$ .

**Definice 2.**  $l_p$ -normu vektoru  $\|\mathbf{x}\|_p$  v prostoru  $\mathbb{C}^n$  definujeme jako

$$\|\mathbf{x}\|_p := \sum_{i=1}^n |x_i|^p \text{ pro } 0 < p < 1, \quad (1.1)$$

$$\|\mathbf{x}\|_p := \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} \text{ pro } 1 \leq p < \infty, \quad (1.2)$$

$$\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})| \quad (\text{počet nenulových prvků}) \quad (1.3)$$

$$\|\mathbf{x}\|_\infty := \max_i |x_i|. \quad (1.4)$$

Pro případy, kdy  $0 \leq p < 1$  není splněna podmínka 3. z definice 1 pro normu, i tak ale budeme i pro tyto případy používat označení  $l_p$ -norma. Nejčastěji používaná norma je pro případ  $p = 2$ , kdy dostáváme tzv. euklidovskou normu, tedy  $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$ . Při práci s řídkými maticemi je však nejčastěji využita norma  $\|\mathbf{x}\|_0$ , případně  $\|\mathbf{x}\|_1$ .

**Definice 3.** Vektor  $\mathbf{x} \in \mathbb{C}^n$  nazveme  $k$ -řídkým  $\iff \|\mathbf{x}\|_0 = k$ .

## 1.2. ŘÍDKÁ ŘEŠENÍ SYSTÉMU LINEÁRNÍCH ROVNIC

Skutečné vektory ale obvykle nemívají přímo nulové hodnoty, proto mluvíme o  $k$ -řídském vektoru i v případě, že má nejvýše  $n-k$  hodnot blízkých nule. Definujeme proto tzv. chybu aproximace:

**Definice 4.** Chyba nejlepší aproximace vektoru  $\mathbf{x} \in \mathbb{C}^n$   $k$ -řídským vektorem v  $l_p$ -normě je definována jako:

$$\sigma_k(\mathbf{x})_p := \sigma_k^n(\mathbf{x})_p := \inf_{\mathbf{z} \in \Sigma_k^n} \|\mathbf{x} - \mathbf{z}\|_p,$$

kde  $\Sigma_k^n := \Sigma_k := \{\mathbf{x} \in \mathbb{C}^n : \|\mathbf{x}\|_0 \leq k\}$ . [2]

### 1.1.2. Normy matic

V některých případech potřebujeme určit velikost nejen vektoru, ale i matice. K tomu nám slouží maticová norma.

**Definice 5.** Takzvanou přirozenou (spektrální, operátorovou) normu matice  $\mathbf{A}$  definujeme jako:

$$\|\mathbf{A}\| = \max_{\mathbf{x}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \text{ pro všechna } \mathbf{x} \in X, \text{ kde } \mathbf{x} \neq \mathbf{0}.$$

Je vidět, že hodnota maticové normy závisí na použité vektorové normě. Při použití  $l_p$ -normy budeme příslušnou maticovou normu značit jako  $\|\mathbf{A}\|_p$ . Pro  $l_1$ -normu a  $l_\infty$ -normu platí [4]:

$$\|\mathbf{A}\|_1 = \max_j \sum_i |a_{ij}| \quad (\text{maximum ze sloupcových součtů})$$

$$\|\mathbf{A}\|_\infty = \max_i \sum_j |a_{ij}| \quad (\text{maximum z řádkových součtů}).$$

V tomto textu budeme nejčastěji používat tzv. Frobeniovu normu, kterou značíme jako  $\|\mathbf{A}\|_F$  [2]:

$$\|\mathbf{A}\|_F := \sqrt{\sum_i \sum_j a_{ij}^2}.$$

### 1.1.3. Příklady

Mějme vektor  $\mathbf{z} = [3; 0; 6; 2; 0]$ . Lze jednoduše ověřit, že  $\|\mathbf{z}\|_0 = 3$ ,  $\|\mathbf{z}\|_1 = 11$ ,  $\|\mathbf{z}\|_2 = 7$ ,  $\|\mathbf{z}\|_\infty = 6$ . Také lze vidět, že tento vektor je 3-řídský. Nyní mějme matici  $\mathbf{A} = \begin{pmatrix} 1 & 0 & 3 \\ 4 & 2 & 0 \\ 5 & 3 & 0 \end{pmatrix}$ .

U této matice můžeme spočítat, že  $\|\mathbf{A}\|_1 = 10$ ,  $\|\mathbf{A}\|_\infty = 8$ , a  $\|\mathbf{A}\|_F = 8$ .

## 1.2. Řídká řešení systému lineárních rovnic

### 1.2.1. Zavedení problému

Uvažujme soustavu (neboli systém) lineárních rovnic  $\mathbf{y} = \mathbf{A}\mathbf{x}$ . Uvažujme také, že tato soustava je nedourčená, tedy počet řádku  $m$  matice  $\mathbf{A}$  je menší než počet jejích sloupců  $n$ .

Řešení  $\mathbf{x}$  této soustavy je nekonečně mnoho, hledáme tedy řešení v nějakém smyslu optimální. V našem případě je to řešení co nejdříve, úlohu proto převedeme do tvaru:

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \mathbf{y} = \mathbf{A}\mathbf{x}. \quad (1.5)$$

Vektor  $\mathbf{y} \in \mathbb{C}^m$  budeme nazývat *signál*, několik signálů pak *data*, matice  $\mathbf{A} \in \mathbb{C}^{m \times n}$  se nazývá *slovník*, jednotlivé sloupce matice  $\mathbf{A}$  budeme nazývat *atomy slovníku*,  $\mathbf{x} \in \mathbb{C}^n$  *vektor koeficientů* nebo pouze *koeficienty*. Pro případ, kdy budeme mít několik signálů  $\mathbf{y}_1, \dots, \mathbf{y}_K$  a k nim příslušné koeficienty  $\mathbf{x}_1, \dots, \mathbf{x}_K$ , zavedeme matice  $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_K]$  a  $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_K]$ . Je vidět, že platí  $\mathbf{Y} = \mathbf{A}\mathbf{X}$ .

### 1.2.2. Metody nalezení vektoru koeficientů

Pokud máme regulární soustavu lineárních rovnic  $\mathbf{y} = \mathbf{A}\mathbf{x}$ , lze vektor  $\mathbf{x}$  určit přesně pomocí inverzní matice a platí  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$ . Pro matici singulární, nedourčené, nebo přeúřčené soustavy však inverzní matice neexistuje, proto je potřeba použít k nalezení  $\mathbf{x}$  jiné metody. Jednou z možností je použití pseudoinverzní matice:

**Definice 6.** Mějme matici  $\mathbf{A} \in \mathbb{C}^{m \times n}$ . Pak matici  $\mathbf{A}^+$  splňující

1.  $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$
2.  $\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+$
3.  $(\mathbf{A}^+)^* = \mathbf{A}\mathbf{A}^+$
4.  $(\mathbf{A}^+\mathbf{A})^* = \mathbf{A}^+\mathbf{A}$

nazveme *pseudoinverzní maticí* k matici  $\mathbf{A}$ . [5]

Pokud je matice  $\mathbf{A}$  regulární, platí  $\mathbf{A}^+ = \mathbf{A}^{-1}$ . Pro nedourčenou soustavu je pseudoinverzní matice maticí s nejnižší Frobeniovou normou, pro přeúřčenou soustavu pak nemusí existovat matice  $\mathbf{B}$  taková, že  $\mathbf{B}\mathbf{y} = \mathbf{x}$ , ale platí, že  $\mathbf{A}\mathbf{A}^+\mathbf{y}$  je nejbližší aproximací  $\mathbf{y}$  ve smyslu nejmenších čtverců ze všech  $\mathbf{A}\mathbf{x}$ . [2, 3]

Existují ale i jiné způsoby nalezení vektoru koeficientů. Mezi ně patří například relaxační algoritmy, které místo  $l_0$ -normy v 1.5 používají  $l_1$ -normu. Také mezi ně patří hladové algoritmy (*greedy algorithms*), které v každé iteraci nachází jiný důležitý atom. To má tu výhodu, že lze určit, jak řídký má být vektor koeficientů. Mezi tyto algoritmy patří například MP (*matching pursuit*) nebo vylepšené OMP (*orthogonal matching pursuit*). [5, 2]

Poslední zmíněný algoritmus OMP budeme později používat, proto je nyní popsán a podrobněji pak shrnutý v tabulce 1.1. Nejprve jsou vloženy vstupní údaje – slovník  $\mathbf{A}$ , signál  $\mathbf{y}$ , maximální požadovaná řídkost  $K_{max}$  a maximální povolená chyba aproximace  $\epsilon$ . Po prvotní inicializaci se v každé iteraci určí atom, s jehož využitím se nejvíce minimalizuje velikost rezidua  $\|\mathbf{y} - \mathbf{A}_{\Omega_i}\mathbf{x}_i\|_2^2$ , kde  $\Omega_i$  je nosič vektoru koeficientů  $\mathbf{x}_i$ . Pokud program, dosáhne požadované řídkosti nebo požadované přesnosti, skončí. [5, 7]

### 1.3. SINGULÁRNÍ ROZKLAD

Tabulka 1.1: Algoritmus OMP

Algoritmus OMP
<b>Vstupy:</b> $\mathbf{y}$ ... signál, $\mathbf{A}$ ... slovník, $K_{max}$ ... hranice řídkosti, $\epsilon$ ... chyba aproximace.
<b>Inicializace:</b> $i = 0$ ... čítač iterací, $\mathbf{r}_0 = \mathbf{y}$ ... residuum, $\mathbf{A}_p = \mathbf{A}$ ... pomocný slovník, $\Omega_0 = \emptyset$ ... nosič vektoru.
<b>Hlavní cyklus</b> Dokud $\ \mathbf{r}_i\ _2^2 > \epsilon$ a $i \leq K_{max}$ : Zvedni čítač $i = i + 1$ . Najdi index $j$ odpovídající maximální hodnotě $\mathbf{A}_p^T \mathbf{r}$ . Aktualizuj nosič $\Omega_i = \Omega_{i-1} \cup j$ . Najdi řešení $\mathbf{x}_i = \arg \min_{\mathbf{u}} \ \mathbf{y} - \mathbf{A}_{\Omega_i} \mathbf{u}\ _2$ . Aktualizuj residuum $\mathbf{r}_i = \mathbf{y} - \mathbf{A}_{\Omega_i} \mathbf{x}_i$ . $j$ -tý sloupec $\mathbf{A}_p$ nahraď nulovým vektorem. Konec podmínky.
<b>Výstup:</b> $\mathbf{x}_k$ ... vektor koeficientů.

### 1.3. Singulární rozklad

**Definice 7.** [4] Každou matici  $\mathbf{A}$  typu  $(m, n)$  lze zapsat ve tvaru

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

kde  $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_m)$  je ortonormální matice řádu  $m$ ,  $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_p)$  je diagonální matice typu  $(m, n)$  a  $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$  je ortonormální matice řádu  $n$ . Také platí  $p = \min(m, n)$  a  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ . Počet nenulových prvků  $\sigma_1, \dots, \sigma_p$  je roven hodnotě matice  $\mathbf{A}$ .

Vyjádření  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  pak nazveme *singulárním rozkladem matice  $\mathbf{A}$* , čísla  $\sigma_1, \dots, \sigma_p$  *singulárními čísly*.

Singulární rozklad budeme využívat v kapitole 3.

## 2. SENZOROVÁ SÍŤ

V této kapitole se budeme zabývat namodelováním sensorové sítě a způsobem přenosu informací mezi jednotlivými senzory. Sensorová síť – respektive síť sensorů – slouží k přenosu informací o poloze, rychlosti, směru, případně jiných stavových veličinách pohybujícího se objektu. Každý sensor dokáže tyto veličiny určit jen s určitou mírou nejistoty. Poté se informace od jednotlivých sensorů vyhodnotí, všechny senzory obdrží výsledek, aby mohly co nejlépe určit dané veličiny v dalším časovém okamžiku a celý proces se opakuje.

### 2.1. Model sensorové sítě

Uvažujme objekt s  $t$  časově proměnnými stavovými veličinami, mezi které patří například pozice, rychlost nebo směr pohybu objektu. Vektor stavů příslušejících k objektu označíme jako  $\mathbf{x}_n \in \mathbb{R}^t$ , kde  $n \in \mathbb{N}_0$  označuje diskrétní čas. Dále uvažujme síť sensorů, které označíme písmenem  $s$ ;  $s = 1, \dots, S$ . V každém čase  $n$  každý sensor  $s$  vyhodnotí údaje z okolí a zjistí signifikantní měření na  $M_n^{(s)}$  místech. Tato měření označíme jako  $\mathbf{z}_{n,m}^{(s)}$ , kde  $m = 1, \dots, M_n^{(s)}$ . Z nich může nejvýše jedno patřit ke zkoumanému objektu. Ostatní budeme nazývat falešná hlášení (*clutter*). Dále mějme  $\mathcal{M}_n^{(s)} := \{1, \dots, M_n^{(s)}\}$ ,  $\mathbf{z}_n^{(s)} := [\mathbf{z}_{n,1}^{(s)T}, \dots, \mathbf{z}_{n,M_n^{(s)}}^{(s)T}]$  a  $\mathbf{z}_n := [\mathbf{z}_n^{(1)T}, \dots, \mathbf{z}_n^{(S)T}]$ .

Protože zkoumanému objektu přísluší nejvýše jedno z  $M_n^{(s)}$  měření, zavedeme parametr  $a_n^{(s)}$ , takový, že:

$$a_n^{(s)} := \begin{cases} m \in \mathcal{M}_n^{(s)} & \text{pokud v čase } n \text{ zjistil sensor } s \text{ objekt měřením } \mathbf{z}_{n,m}^{(s)}, \\ 0 & \text{pokud v čase } n \text{ sensor } s \text{ nedetekoval sledovaný objekt.} \end{cases} \quad (2.1)$$

V modelu také uvažujeme, že každý sensor  $s$  dokáže komunikovat s množinou sensorů  $\mathcal{N}^{(s)} \subseteq \{1, \dots, S\} \setminus s$ . Tuto množinu budeme nazývat *množinou sousedů* senzoru  $s$ .

Pravděpodobnost, že jedno z měření náleží objektu, je známá a označíme ji  $P_t^{(s)}$ . Každý sensor  $s$  vyprodukuje falešná hlášení, jejichž rozdělení odpovídá Poissonovu rozdělení s parametrem  $\mu^{(s)}$  nezávislým na čase  $n$ . Tak můžeme určit sdruženou pravděpodobnostní funkci proměnných  $a_n^{(s)}$  a  $M_n^{(s)}$ :

$$p(a_n^{(s)}, M_n^{(s)}) := \begin{cases} C_1(M_n^{(s)})P_t^{(s)}, & a_n^{(s)} \in \mathcal{M}_n^{(s)} \\ C_1(M_n^{(s)})(1 - P_t^{(s)})\mu^{(s)}, & a_n^{(s)} = 0, \end{cases} \quad (2.2)$$

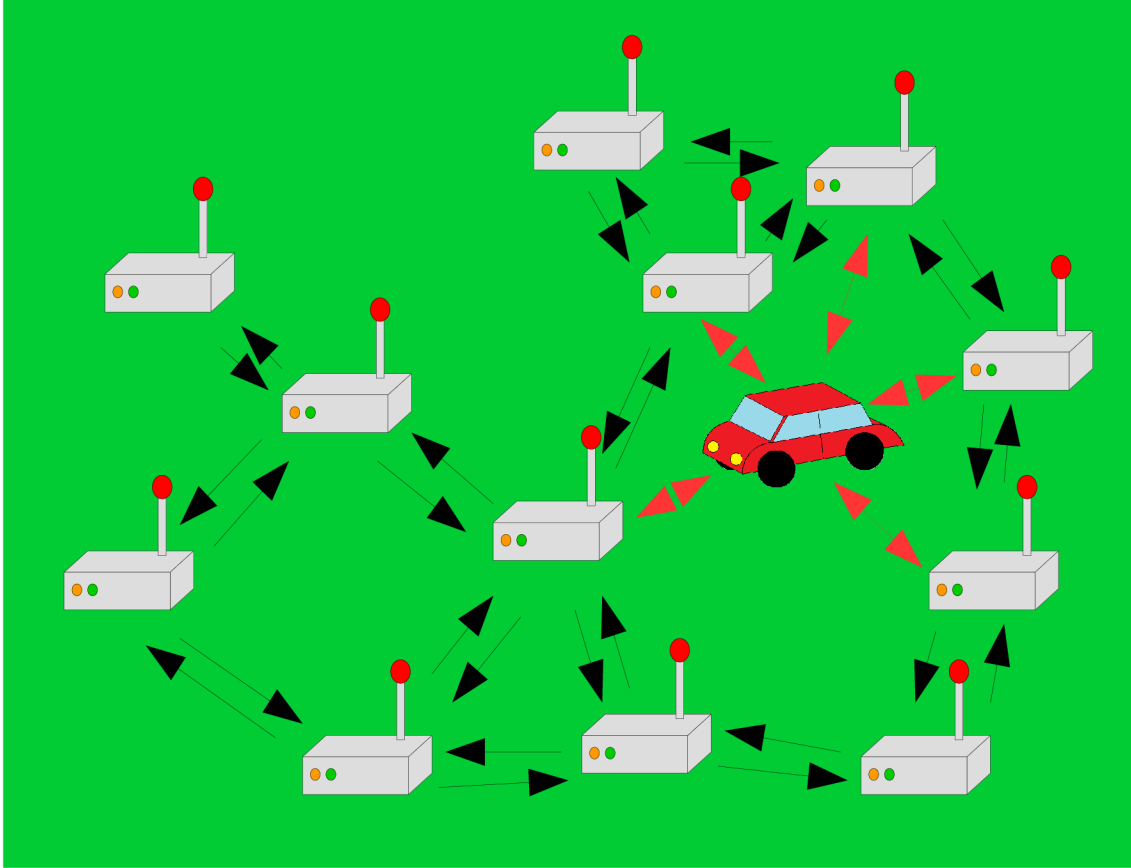
kde  $C_1(M_n^{(s)})$  je parametr závisející pouze na počtu měření  $M_n^{(s)}$ .

Podmíněná hustota pravděpodobnosti pro každý sensor  $s$  v čase  $n$  na pozici  $m$  je pak

$$f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)}) = \begin{cases} f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = 0), & \text{pokud } a_n^{(s)} = m \in \mathcal{M}_n^{(s)} \\ f_c(\mathbf{z}_{n,m}^{(s)}), & \text{pokud } a_n^{(s)} = 0, \end{cases} \quad (2.3)$$

kde  $f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = 0)$  a  $f_c(\mathbf{z}_{n,m}^{(s)})$  jsou známé hustoty pravděpodobnosti. [6]

## 2.2. LOKÁLNÍ VĚROHODNOSTNÍ FUNKCE



Obrázek 2.1: Model sensorové sítě. Každý sensor komunikuje se skupinou sousedních sensorů, což značí černé šipky. Sledovaný objekt (auto) je detekován jen několika senzory, což je označené červenými šipkami

### 2.2. Lokální věrohodnostní funkce

Na senzoru  $s$  máme  $M_n^{(s)}$  měření  $\mathbf{z}_{n,m}^{(s)}$ . Nyní z těchto měření chceme určit lokální věrohodnostní funkci  $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ , která popisuje závislost mezi vektorem měření  $\mathbf{z}_n^{(s)}$  a stavem měřeného objektu v čase  $n$ . Jelikož jednotlivá měření  $\mathbf{z}_{n,m}^{(s)}$  jsou nezávislá, platí

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)}) = \prod_{m \in \mathcal{M}_n^{(s)}} f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)}). \quad (2.4)$$

V této rovnici je ovšem použita rovnost (2.3), která se větví na dva případy. Proto je budeme dělit i v tomto případě. Nejprve si nadefinujeme  $f_c(\mathbf{z}_n^{(s)}|M_n^{(s)}) := \prod_{m \in \mathcal{M}_n^{(s)}} f_c(\mathbf{z}_{n,m}^{(s)})$ .

Poté

- pokud  $a_n(s) = 0$ , pak

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)} = m, M_n^{(s)}) = f_c(\mathbf{z}_n^{(s)}|M_n^{(s)}). \quad (2.5)$$

- pokud  $a_n(s) = m \in \mathcal{M}_n^{(s)}$ , tak jako  $m'$  označíme měření, která nepatří ke sledovanému objektu, tedy  $m' \in \mathcal{M}_n^{(s)} \setminus \{m\}$  a příslušná měření jsou tedy značena jako  $\mathbf{z}_{n,m'}^{(s)}$ . A platí

$$\begin{aligned}
f(\mathbf{z}_n^{(s)} | \mathbf{x}_n, a_n^{(s)} = m, M_n^{(s)}) &= \\
&= f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m) \prod_{m' \in \mathcal{M}_n^{(s)} \setminus \{m\}} f_c(\mathbf{z}_{n,m'}^{(s)}) = \\
&= f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m) \frac{f_c(\mathbf{z}_n^{(s)} | M_n^{(s)})}{f_c(\mathbf{z}_{n,m}^{(s)})}.
\end{aligned} \tag{2.6}$$

Podle věty o úplné pravděpodobnosti platí:

$$f(\mathbf{z}_n^{(s)} | \mathbf{x}_n) = \sum_{a_n^{(s)}=0}^{M_n^{(s)}} f(\mathbf{z}_n^{(s)} | \mathbf{x}_n, a_n^{(s)}, M_n^{(s)}) p(a_n^{(s)}, M_n^{(s)}), \tag{2.7}$$

a dosazením předchozích rovností (2.2), (2.6) a (2.5) dostaneme

$$\begin{aligned}
f(\mathbf{z}_n^{(s)} | \mathbf{x}_n) &= \\
&= f_c(\mathbf{z}_n^{(s)} | M_n^{(s)}) C_1(M_n^{(s)}) (1 - P_t^{(s)}) \mu^{(s)} + C_1(M_n^{(s)}) P_t^{(s)} \sum_{m \in \mathcal{M}_n^{(s)}} f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m) = \\
&= C_2(\mathbf{z}_n^{(s)}) \left[ (1 - P_t^{(s)}) \mu^{(s)} + P_d^{(s)} \sum_{m \in \mathcal{M}_n^{(s)}} \frac{f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m)}{f_c(\mathbf{z}_{n,m}^{(s)})} \right],
\end{aligned} \tag{2.8}$$

kde  $C_2(\mathbf{z}_n^{(s)}) := C_1(M_n^{(s)}) f_c(\mathbf{z}_n^{(s)} | M_n^{(s)})$ . První člen součtu  $C_2(\mathbf{z}_n^{(s)}) (1 - P_t^{(s)}) \mu^{(s)}$  nazveme *základem (floor)*, další členy  $C_2(\mathbf{z}_n^{(s)}) P_d^{(s)} \frac{f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m)}{f_c(\mathbf{z}_{n,m}^{(s)})}$ ,  $m = 1 \dots M_n^{(s)}$  pak nazveme *mody (modes)*. Základ a mody jsou nezáporné, protože  $C_2(\mathbf{z}_n^{(s)})$ ,  $P_d^{(s)}$ ,  $(1 - P_t^{(s)})$ ,  $\mu^{(s)}$  i  $\frac{f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m)}{f_c(\mathbf{z}_{n,m}^{(s)})}$  jsou nezáporné výrazy. [6]

### 2.3. Globální věrohodnostní funkce

Aby senzor  $s$  dokázal co nejlépe určit veličiny patřící ke sledovanému objektu  $\mathbf{x}_{n+1}$  v následujícím čase, měl by počítat i s výsledky měření od jiných senzorů, které dostaneme s využitím tzv. algoritmu konsensu průměrů – *average consensus algorithm*.

Nejprve je však potřeba sestavit globální věrohodnostní funkci (*global likelihood function* – *GLF*)  $f(\mathbf{z}_n | \mathbf{x}_n)$ . Protože měření  $\mathbf{z}_n^{(s)}$  na jednom senzoru v čase  $n$  není závislé na měření na jiných senzorech v tomtéž čase, platí

$$f(\mathbf{z}_n | \mathbf{x}_n) = \prod_{s=1}^S f(\mathbf{z}_n^{(s)} | \mathbf{x}_n). \tag{2.9}$$

### 2.3. GLOBÁLNÍ VĚROHODNOSTNÍ FUNKCE

Tuto formu ale dále upravíme, protože pro další výpočty bude vhodnější nahradit součin funkcí jejich součtem. Proto (2.9) převedeme do logaritmického tvaru:

$$\begin{aligned} f(\mathbf{z}_n|\mathbf{x}_n) &= \exp(\ln f(\mathbf{z}_n|\mathbf{x}_n)) = \exp\left(\ln \prod_{s=1}^S f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)\right) = \\ &= \exp\left(\sum_{s=1}^S \ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)\right) = \exp(Sl(\mathbf{x}_n)), \end{aligned} \quad (2.10)$$

kde

$$l(\mathbf{x}_n) = \frac{1}{S} \sum_{s=1}^S \ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n). \quad (2.11)$$

O  $l(\mathbf{x}_n)$  můžeme mluvit jako o průměru logaritmů  $S$  lokálních věrohodnostních funkcí  $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ . Protože se jedná o funkce, nelze přímo použít algoritmus shody průměrů, protože ten pracuje s čísly, a ne s funkcemi. Proto je třeba (2.10), respektive (2.11), upravit a aproximovat pomocí čísel.

V (2.8) jsou dva druhy členů – základ a mody. Protože mody jsou nezáporné a základ je konstantní, platí

$$\min_{\mathbf{x}_n \in \mathbb{R}^d} \{f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)\} = C_2(\mathbf{z}_n^{(s)})(1 - P_t^{(s)})\mu^{(s)} \quad (2.12)$$

Protože tento výraz je kladný a logaritmus je prostá funkce, můžeme tento výraz zlogaritmovat a zavést proměnou  $\lambda^{(s)}$ , takovou, že

$$\lambda^{(s)} = \min_{\mathbf{x}_n \in \mathbb{R}^d} \{\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)\} = \ln \left( C_2(\mathbf{z}_n^{(s)})(1 - P_t^{(s)})\mu^{(s)} \right). \quad (2.13)$$

Potom můžeme napsat

$$\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) = \lambda^{(s)} + \Lambda^{(s)}(\mathbf{x}_n) \quad (2.14)$$

a dosazením do (2.11) dostaneme

$$l(\mathbf{x}_n) = \frac{1}{S} \sum_{s=1}^S (\lambda^{(s)} + \Lambda^{(s)}(\mathbf{x}_n)) = \frac{1}{S} \sum_{s=1}^S \lambda^{(s)} + \frac{1}{S} \sum_{s=1}^S \Lambda^{(s)}(\mathbf{x}_n) = \bar{\lambda} + \bar{\Lambda}, \quad (2.15)$$

kde

$$\bar{\lambda} := \frac{1}{S} \sum_{s=1}^S \lambda^{(s)} \quad (2.16)$$

a

$$\bar{\Lambda}(\mathbf{x}_n) := \frac{1}{S} \sum_{s=1}^S \Lambda^{(s)}(\mathbf{x}_n). \quad (2.17)$$

[6]

Určit  $\bar{\lambda}$  ovšem nelze pouze jednoduchým dosazením do vzorce. Každý sensor  $s$  zná svoji hodnotu  $\lambda^{(s)}$ , ale dokáže komunikovat pouze se svými sousedy, nelze proto jednoduše získat informace od všech sensorů. Proto použijeme algoritmus shody průměrů. Tento algoritmus v každé iteraci určí vnitřní stav senzoru  $u_i^{(s)}$  a při vhodně zvolených parametrech



(především vah) platí  $\lim_{i \rightarrow \infty} u_i^{(s)} = \bar{\lambda}$ . V reálném použití se samozřejmě počítá s konečným počtem iterací  $n_{it}$ .

V algoritmu se střídají dva kroky. V prvním kroku každý senzor  $s$  pošle svůj poslední stav  $u_{i-1}^{(s)}$  všem svým sousedům  $s' \in \mathcal{N}^{(s)}$  a dostane od nich jejich stavy. V druhém kroku aktualizuje svůj stav jako

$$u_i^{(s)} = \sum_{s' \in \mathcal{N}^{(s)} \cup \{s\}} w_{s,s'} u_{i-1}^{(s')}, \quad i = 1, 2, \dots \quad (2.18)$$

kde  $w_{s,s'}$  jsou váhy splňující  $\sum_{s' \in \mathcal{N}^{(s)} \cup \{s\}} w_{s,s'} = 1$ . Populární volba vah jsou tzv. Metropolitní váhy.

Algoritmus je inicializován jako  $u_0^{(s)} = \lambda^{(s)}$ . [6, 9]

## 2.4. Využití slovníku k výpočtu globální věrohodnostní funkce

Nyní jsme určili základní konstantu  $\bar{\lambda}$  a zbývá určit hodnotu  $\bar{\Lambda}(\mathbf{x}_n)$ . Upravením (2.14) dostaneme

$$\Lambda^{(s)}(\mathbf{x}_n) = \ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n) - \lambda^{(s)}. \quad (2.19)$$

Následně  $\Lambda^{(s)}(\mathbf{x}_n)$  aproximuje s využitím funkcí  $\mathbf{d}_1(\mathbf{x}_n), \dots, \mathbf{d}_B(\mathbf{x}_n)$ , které nazveme *atomy*, a vektoru koeficientů  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_B]$ :

$$\Lambda^{(s)}(\mathbf{x}_n) \approx \sum_{b=1}^B \alpha_b^{(s)} \mathbf{d}_b(\mathbf{x}_n). \quad (2.20)$$

Dosazením (2.20) do (2.17) dostaneme

$$\bar{\Lambda}(\mathbf{x}_n) \approx \frac{1}{S} \sum_{s=1}^S \sum_{b=1}^B \alpha_b^{(s)} \mathbf{d}_b(\mathbf{x}_n) = \sum_{b=1}^z a_b \mathbf{d}_b(\mathbf{x}_n), \quad (2.21)$$

kde

$$a_b = \sum_{s=1}^S \alpha_b^{(s)} \quad \text{pro } b = 1, \dots, B. \quad (2.22)$$

Poté  $a_b$  aproximujeme pomocí algoritmu konsensu průměrů pro všechna  $b$  jako  $y_{m_{it}}^{(b,s)}$ :

$$a_b \approx y_{m_{it}}^{(b,s)} \quad \text{pro } b = 1, \dots, B, \quad (2.23)$$

kde  $m_{it}$  je počet iterací algoritmu a algoritmus je inicializován jako  $y_0^{(b,s)} = \alpha_b^{(s)}$ . Dosazením do (2.21) dostaneme

$$\bar{\Lambda}(\mathbf{x}_n) \approx \sum_{b=1}^z y_{m_{it}}^{(b,s)} \mathbf{d}_b(\mathbf{x}_n). \quad (2.24)$$

Celou globální věrohodnostní funkci pak dosazením jednotlivých vztahů vyjádříme jako

$$f(\mathbf{z}_n | \mathbf{x}_n) \approx \hat{f}(\mathbf{z}_n^{(s)} | \mathbf{x}_n) := \exp \left( S \left( u_{m_{it}}^{(s)} + \sum_{b=1}^B y_{m_{it}}^{(b,s)} \mathbf{d}_b(\mathbf{x}_n) \right) \right). \quad (2.25)$$

## 2.4. VYUŽITÍ SLOVNÍKU K VÝPOČTU GLOBÁLNÍ VĚROHODNOSTNÍ FUNKCE

Z důvodu rychlého přenosu dat je lokální věrohodnostní funkce vzata pouze v několika částicích  $\mathbf{x}_n^{(s,j)}$ ,  $j = 1, \dots, J$ . Tyto částice jsou zvoleny na základě předchozího výpočtu GLF. GLF pak vyjádříme jako

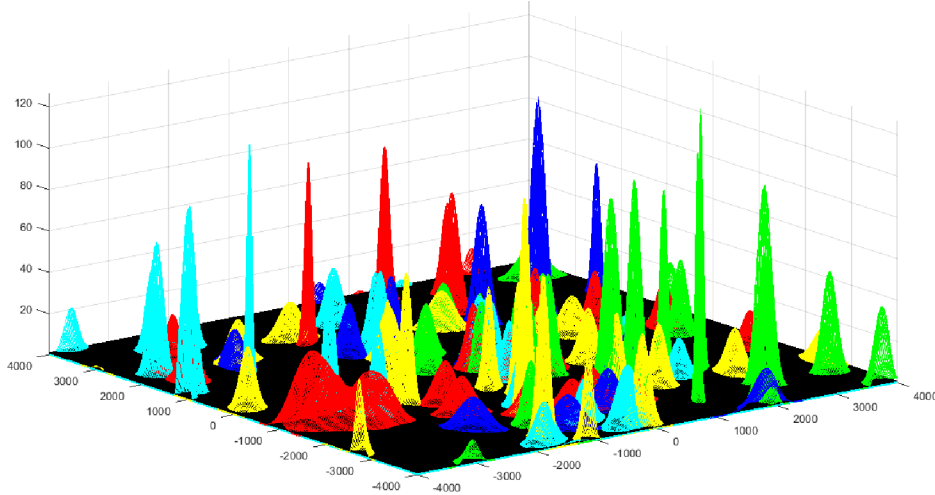
$$\hat{f}(\mathbf{z}_n^{(s)} | \mathbf{x}_n^{(s,j)}) := \exp \left( S \left( u_{nit}^{(s)} + \sum_{b=1}^L y_{mit}^{(b,s)} \mathbf{d}_b(\mathbf{x}_n^{(s,j)}) \right) \right). \quad (2.26)$$

[6, 10]

V této práci se budeme zabývat hledáním konkrétních atomů, které ale nebudou vyjádřeny pomocí funkcí, ale pomocí vektorů nebo matic (dimenze závisí na počtu sledovaných stavových veličin  $d$ ), a z těchto atomů pak sestavíme slovník. Prvky atomů lze považovat za funkční hodnoty. V tomto případě nelze určit částice kdekoliv, ale pouze v bodech odpovídajících prvkům atomu, respektive při nalezení dané částice  $j$  na senzoru  $s$  v čase  $n$  se tato částice posune do nejbližšího bodu odpovídajícího prvku atomu. Tím vznikne slovník

$$\mathbf{D} = [\mathbf{d}_1 \dots \mathbf{d}_B] = \begin{pmatrix} \mathbf{d}_1(\mathbf{x}_n^{(s,1)}) & \mathbf{d}_2(\mathbf{x}_n^{(s,1)}) & \dots & \mathbf{d}_B(\mathbf{x}_n^{(s,1)}) \\ \mathbf{d}_1(\mathbf{x}_n^{(s,2)}) & \mathbf{d}_2(\mathbf{x}_n^{(s,2)}) & \dots & \mathbf{d}_B(\mathbf{x}_n^{(s,2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{d}_1(\mathbf{x}_n^{(s,J)}) & \mathbf{d}_2(\mathbf{x}_n^{(s,J)}) & \dots & \mathbf{d}_B(\mathbf{x}_n^{(s,J)}) \end{pmatrix},$$

Slovník  $\mathbf{D}$  se určí z mnoha vygenerovaných signálů  $\Lambda^{(s)}(\mathbf{x}_n^{(s,j)})$  algoritmem K-SVD popsaným v kapitole 3. Podobu  $\Lambda^{(s)}(\mathbf{x}_n)$  můžeme vidět na obrázku 2.2, kde je zobrazena pro 5 různých senzorů.



Obrázek 2.2: Zobrazení  $\Lambda^{(s)}(\mathbf{x}_n)$  pro různé senzory. Každá barva označuje jiný senzor. Zdroj [6]

## 3. HLEDÁNÍ OPTIMÁLNÍ REPREZENTACE DAT

Aby senzory zvládly efektivně určit stav pozorovaného objektu, měly by pracovat co nejrychleji, a tedy přenášet jen malé množství dat. K tomu je vhodná řídká reprezentace, protože pokud budou znát všechny senzory daný slovník, pomocí kterého jsou data reprezentována, stačí jim přenést pouze několik koeficientů. Proto se v této kapitole zaměříme na hledání vhodného slovníku pro reprezentaci dat k vytvoření globální věrohodostní funkce, který by data nejpřesněji aproximoval. Tento slovník, respektive jeho parametry, se pokusíme nalézt s využitím algoritmu vhodným na učení slovníku z dostupných trénovacích signálů, konkrétně použijeme algoritmus K-SVD, protože ve srovnání uvedeném v [8] vykazoval nejlepší výsledky oproti jiným algoritmům.

### 3.1. Algoritmus K-SVD

V K-SVD se střídají dva kroky – nejprve zafixujeme slovník a libovolnou metodou (v této práci pomocí OMP) se nalezne vektor koeficientů, a poté zafixujeme koeficienty a s využitím singulárního rozkladu upravíme slovník. Název algoritmu K-SVD je spojením dvou názvů – *K-means*, což je algoritmus používaný ve shlukové analýze a v našem případě k určení vektoru koeficientů, a *SVD*, což je zkratka pro singulární rozklad.[8],[7]

#### 3.1.1. K-means a řídká reprezentace signálu

Ve shlukové analýze se naleznou vektory  $\mathbf{c}_1, \dots, \mathbf{c}_K$  takové, že každý vzorek lze aproximovat jedním z těchto vektorů, případně násobením těchto vektorů. To je obdobné jako při hledání řídké reprezentace, kdy je ale každý signál aproximován několika vektory. Algoritmus K-means se pak obdobně jako K-SVD skládá ze dvou kroků – v prvním kroku s neměnnými vektory  $\mathbf{c}_1, \dots, \mathbf{c}_K$  nalezneme nejlepší reprezentace pro každý vzorek, a ve druhém kroku pomocí těchto reprezentací mění vektory. [8]

#### 3.1.2. Popis algoritmu

Cílem tohoto algoritmu je najít nejlepší slovník hledáním minima účelové funkce

$$\min_{\mathbf{D}, \mathbf{X}} \{ \|\mathbf{Y} - \mathbf{DX}\|_F^2 \} \quad \text{vzhledem k} \quad \forall i, \|\mathbf{x}_i\|_0 \leq K_{\max}. \quad (3.1)$$

Předpokládejme, že slovník  $\mathbf{D}$  je pevně daný a hledáme jen matici koeficientů  $\mathbf{X}$  Pak můžeme (3.1) převést na  $N$  problémů

$$\min_{\mathbf{x}_i} \{ \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_F^2 \} \quad \text{vzhledem k} \quad \|\mathbf{x}_i\|_0 \leq K_{\max} \text{ pro } i = 1, \dots, N. \quad (3.2)$$

Při použití v algoritmu pak budeme volit řídkost pevně danou, protože velikost matic  $\mathbf{U}$ ,  $\mathbf{\Sigma}$  a  $\mathbf{V}$  vzniklých singulárním rozkladem závisí na přesné řídkosti vektoru koeficientů – při omezení řídkosti shora se může stát, že různé signály budou mít různou řídkost a pak

### 3.2. VSTUPNÍ ÚDAJE

i nové atomy budou různě dlouhé a nebude možné sestavit nový slovník. Účelové funkce se pak pozmění na

$$\min_{\mathbf{x}_i} \{ \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_{\mathbb{F}}^2 \} \quad \text{vzhledem k} \quad \|\mathbf{x}_i\|_0 = K_{\max} \quad \text{pro } i = 1, \dots, N, \quad (3.3)$$

kde řídkost  $K$  je ale třeba volit rozumně, tak aby byla dosažitelná (nelze mít větší řídkost než jaký je počet trénovacích signálů).

Přesnost nalezeného řešení můžeme posoudit velikostí chyby – velikostí matice reziduí  $\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathbb{F}}^2$  – pokud by byla rovna nule, pak slovník  $\mathbf{D}$  a koeficienty  $\mathbf{X}$  přesně reprezentují data  $\mathbf{Y}$ . V prvním kroku nalezneme řídkou reprezentaci  $\mathbf{X}$  dat  $\mathbf{Y}$ . V dalším kroku se snažíme minimalizovat chybu, kterou vyjádříme ve tvaru

$$\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\mathbb{F}}^2 = \|\mathbf{Y} - \sum_{b=1}^K \mathbf{d}_b \mathbf{x}^b\|_{\mathbb{F}}^2 = \left\| \left( \mathbf{Y} - \sum_{b \neq k} \mathbf{d}_b \mathbf{x}^b \right) - \mathbf{d}_k \mathbf{x}^k \right\|_{\mathbb{F}}^2, \quad (3.4)$$

kde  $\mathbf{x}^k$  je  $k$ -tý řádek  $\mathbf{X}$ .

Označíme si  $\mathbf{E}_k := \mathbf{Y} - \sum_{b \neq k} \mathbf{d}_b \mathbf{x}^b$ . Pokud bychom provedli singulární rozklad, dokázali bychom velmi rychle snížit chybu aproximace, ale nebyla by zaručena požadovaná řídkost. Proto se nejprve  $\mathbf{E}_k$  zúží na  $\mathbf{E}_k^R$  odstraněním sloupců, pro které nabývá prvek ve vektoru koeficientů nulové hodnoty. Poté se provede singulární rozklad  $\mathbf{E}_k^R = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  a přiřadí se do  $\mathbf{d}_k$  první sloupec  $\mathbf{U}$  a do  $\mathbf{x}^k$  první sloupec  $\mathbf{V}$  vynásobený  $\sigma_1$ . Podrobněji je algoritmus popsán v tabulce 3.1. [8]

## 3.2. Vstupní údaje

### 3.2.1. Signály

Měření senzorů poskládané z jednotlivých částic mají přibližně tvar dvojrozměrného normálního rozdělení. Protože známe tuto charakteristiku, můžeme se jí řídit při hledání vhodných startovacích slovníků, kterými inicializujeme algoritmus K-SVD. Můžeme určit vrchol měření nebo hranici, pod kterou jsou hodnoty blízké nule a ve startovacím slovníku je zanedbat. Při našem hledání slovníku jsme nepoužili data přímo od senzorů, ale data vygenerovaná pomocí konkrétního slovníku ne spojitě, ale ve formě matic řádu 101. To má tu výhodu, že víme, jak by měl vypadat optimální slovník. Tato vygenerovaná data jsou podobná jako data na obrázku 2.2. V úvahu jsme vzali dvě zkoumané veličiny – pozici sledovaného objektu vzhledem k ose  $x$ , a jeho pozici vzhledem k ose  $y$ . Naše signály jsou proto dvojrozměrné a tvoří matici. Trénovacích signálů bylo 2000, kontrolních 250. Aby s nimi mohl algoritmus K-SVD pracovat, je třeba je přeskládat tak, že jednotlivé sloupce signálu jsou seřazeny za sebe, tedy z tvaru  $\mathbf{y}_m = [\mathbf{y}_m(:, 1), \mathbf{y}_m(:, 2), \dots, \mathbf{y}_m(:, 101)]$  do tvaru  $\hat{\mathbf{y}}_m = [\mathbf{y}_m(:, 1)^T, \mathbf{y}_m(:, 2)^T, \dots, \mathbf{y}_m(:, 101)^T]^T$ . Na obrázku 3.1 vidíme příklad vygenerovaného signálu.

### 3.2.2. Startovací slovníky

Hledání optimálního slovníku je minimalizační úloha, která ale není konvexní, tedy každé nalezené lokální minimum nemusí být globálním minimem. Zároveň to znamená, že při

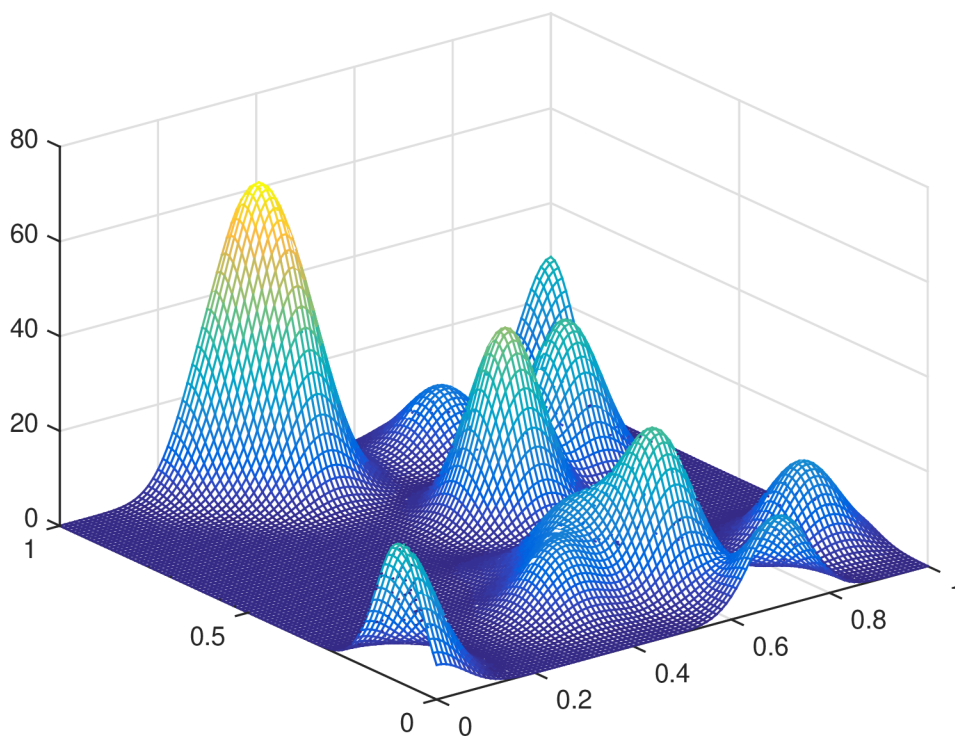
Algoritmus K-SVD
<p><b>Vstupy:</b>  <math>\mathbf{Y}</math> ... matice signálů  <math>\mathbf{D}</math> ... startovací slovník,  <math>K_{\max}</math> ... hranice řídkosti,  <math>J_{\max}</math> ... počet iterací</p>
<p><b>Inicializace:</b>  <math>j = 1</math> ... čítač iterací,  <math>\mathbf{D}^{(1)} = \mathbf{D}</math></p>
<p><b>Hlavní cyklus</b>                  Dokud <math>j \leq J_{\max}</math>:                  Pomocí libovolného algoritmu nalezni matici koeficientů <math>\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n]</math>                  k signálům <math>\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_n]</math> přibližným řešením problému  <math>\min_{\mathbf{x}_i} \{\ \mathbf{y}_i - \mathbf{D}^{(j)} \mathbf{x}_i\ _2^2\}</math> vzhledem k <math>\forall i, \ \mathbf{x}_i\ _0 = K_{\max}</math>.                  Pro každý atom <math>\mathbf{d}_i^{(j)}</math> slovníku <math>\mathbf{D}^{(j)}</math>:  <ul style="list-style-type: none"> <li>• Nalezni vektor všech pozic, na kterých je <math>\mathbf{x}^k</math> nenulové – <math>\omega_k := \{t   \mathbf{x}^k(t) \neq 0\}</math>.</li> <li>• Spočti chybovou matici <math>\mathbf{E}_k</math> jako <math>\mathbf{E}_k = \mathbf{Y} - \sum_{b \neq k} \mathbf{d}_b^{(j)} \mathbf{x}_k</math>.</li> <li>• Z <math>\mathbf{E}_k</math> vezmi v úvahu jen sloupce odpovídající <math>\omega_k</math> – získá se <math>\mathbf{E}_k^R</math>.</li> <li>• Proveď singulární rozklad matice <math>\mathbf{E}_k^R = \mathbf{U} \Sigma \mathbf{V}^T</math>.</li> <li>• Přiřaď <math>\mathbf{d}_k^{(j)}</math> první sloupec <math>\mathbf{U}</math></li> <li>• <math>\mathbf{x}_k</math> nahraď prvním sloupcem <math>\mathbf{V}</math> vynásobeným <math>\sigma_1</math>.</li> </ul>                 Zvedni čítač <math>j = j + 1</math>.  <math>\mathbf{D}^{(j+1)} = \mathbf{D}^{(j)}</math>.                  Konec podmínky.</p>
<p><b>Výstup:</b>  <math>\mathbf{D}^{(J_{\max})}</math> ... výsledný slovník.</p>

nastartování v jiném bodě, respektive jiným startovacím slovníkem, nalezne K-SVD jiný výsledný slovník. Proto jsme vyzkoušeli různé startovací slovníky, které pro přehlednost budeme značit **A–F**:

- **A** – Původní slovník, pomocí kterého byla nagenеровána data; je vytvořený pomocí splajnů. Očekáváme, že výsledný slovník bude podobný tomuto slovníku
- **B** – Tento slovník obsahuje buď nuly, nebo řadu jedniček na pozicích, kde očekáváme nenulovou hodnotu.
- **C** – Slovník obsahující samé jedničky.
- **D** – Atomy v tomto slovníku obsahují jen jedinou nenulovou hodnotu – 1 – na místě, kde očekáváme nejvyšší hodnotu v atomu.
- **E, F** – Dva různé náhodně vygenerované slovníky.

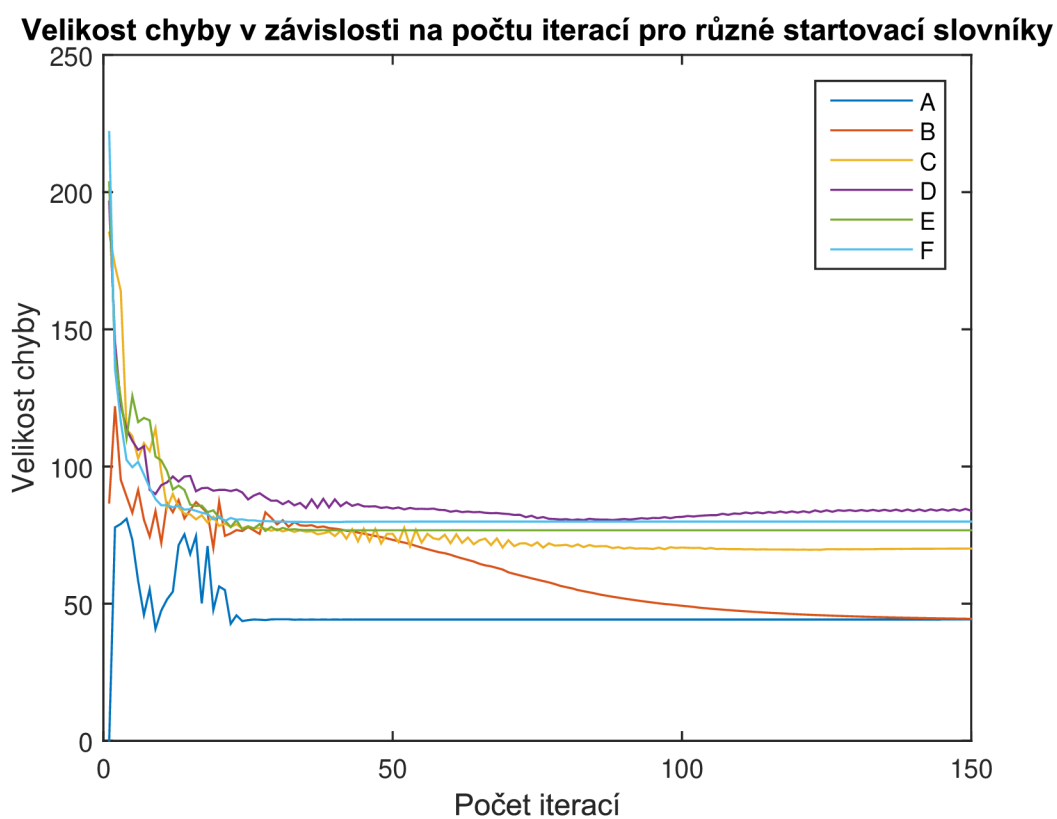
Při testování na jednodušších datech se ukázalo, jak je vidět na obrázku (3.2), že nejlépe metoda skončila při použití slovníků A a B, proto byly tyto slovníky použity i na složitějších datech. Pro porovnání výsledků jsme zvolili různé počty atomů – 25, 36, 121 a

### 3.2. VSTUPNÍ ÚDAJE



Obrázek 3.1: Jeden z trénovacích signálů složitějších dat.

441 – a různou řídkost – 10, 20, 30 a 40 – v závislosti na velikosti slovníku. Slovník, který jsme získali algoritmem K-SVD při použití například startovacího slovníku  $\mathbf{A}$ , budeme nazývat slovník generovaný startovacím slovníkem  $\mathbf{A}$



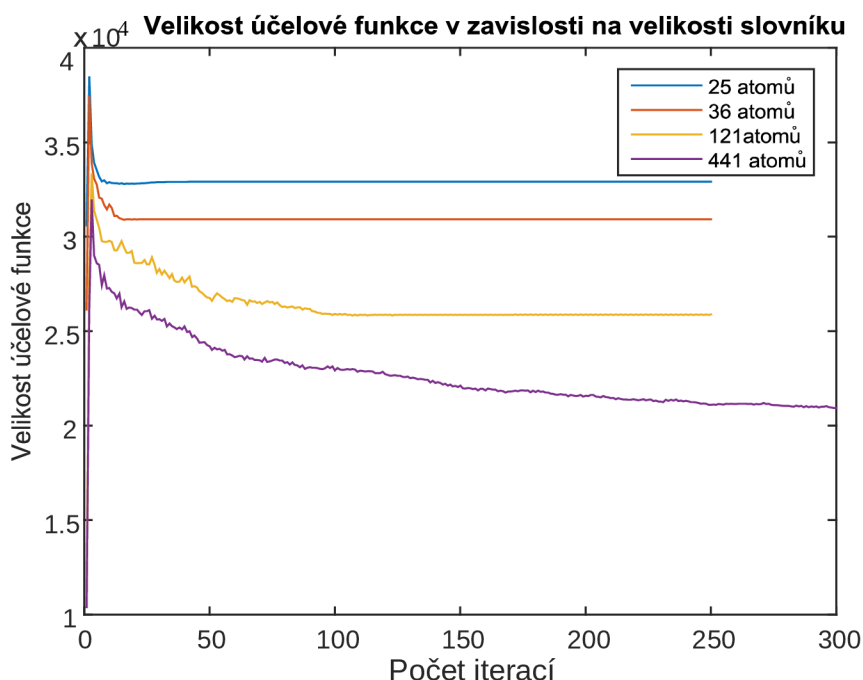
Obrázek 3.2: Graf závislosti chyby aproximace na počtu iterací pro různé startovací slovníky.

## 4. ROZBOR VÝSLEDKŮ

### 4.1. Spočtené slovníky

Pro porovnávání jednotlivých výsledků v závislosti na použitém startovacím slovníku jsme použili velikost účelové funkce (neboli velikost chyby)  $\|\mathbf{Y} - \mathbf{DX}\|_F^2$ . Na následujících grafech lze vidět, že pro spočtení některých slovníků bylo zapotřebí menší množství iterací, aby algoritmus K-SVD konvergoval, a ostatní iterace nebyli počítány.

Na obrázku 4.1 je vidět, že se zvětšujícím se počtem atomů se zvyšuje i přesnost aproximace signálů  $\mathbf{Y}$  pomocí slovníku  $\mathbf{A}$  a koeficientů  $\mathbf{X}$ , a to i přesto, že s vyšším počtem atomů se zvyšuje počet prvků vstupujících do normy při výpočtu velikosti účelové funkce. Na obrázcích 4.2 a 4.3 jsou grafy ukazující, že s větší povolenou řídkostí klesá velikost reziduí (neboli účelové funkce).



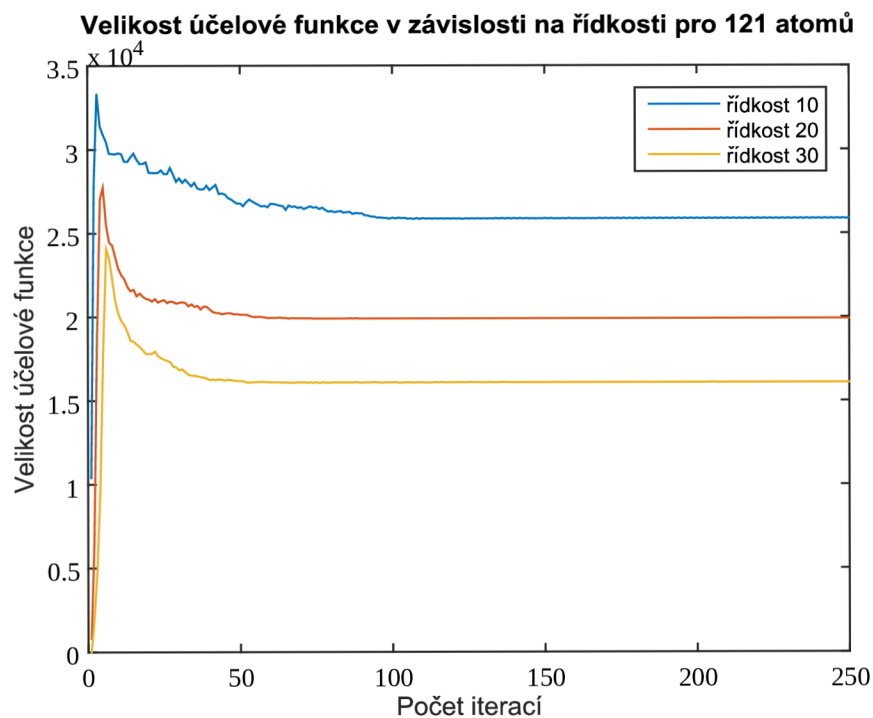
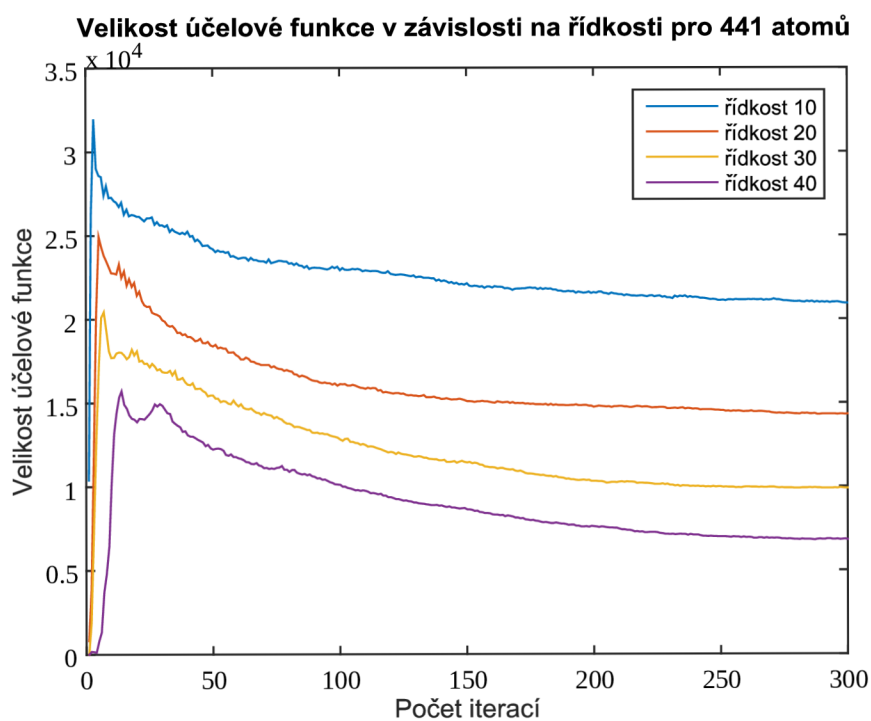
Obrázek 4.1: Graf závislosti velikosti účelové funkce na počtu atomů slovníku  $\mathbf{A}$ .

Na obrázcích 4.4, 4.5 a 4.6 vidíme opět graf závislosti velikosti účelové funkce na počtu atomů a grafy závislosti velikosti účelové funkce na řídkosti, tentokrát pro slovník, jehož výpočet byl inicializovaný startovacím slovníkem  $\mathbf{B}$ .

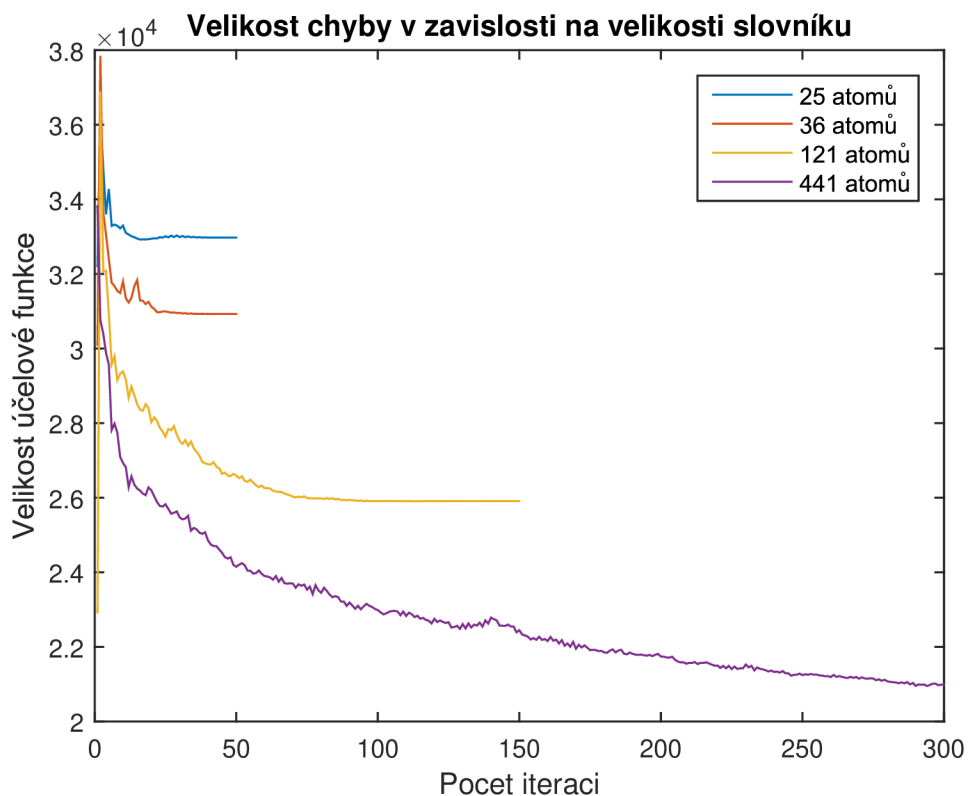
V porovnání obou slovníků na obrázku 4.7 je vidět, že velikost účelové funkce je podobná při inicializaci algoritmu K-SVD slovníkem  $\mathbf{A}$  i slovníkem  $\mathbf{B}$ .

Z porovnání velikostí účelových funkcí nelze rozhodnout, který startovací slovník je vhodnější zvolit, ale je vidět, že je vhodné použít co největší množství atomů. Nicméně víme, že signály byly nagenеровány ze slovníku  $\mathbf{A}$  se 121 atomy (příklad atomu je na obrázku 4.8) a natrénovaný slovník by se mu měl co nejvíce přiblížit. Když si však prohlédneme atomy natrénovaných slovníků, například na obrázku 4.9 a 4.10, zjistíme, že zdaleka

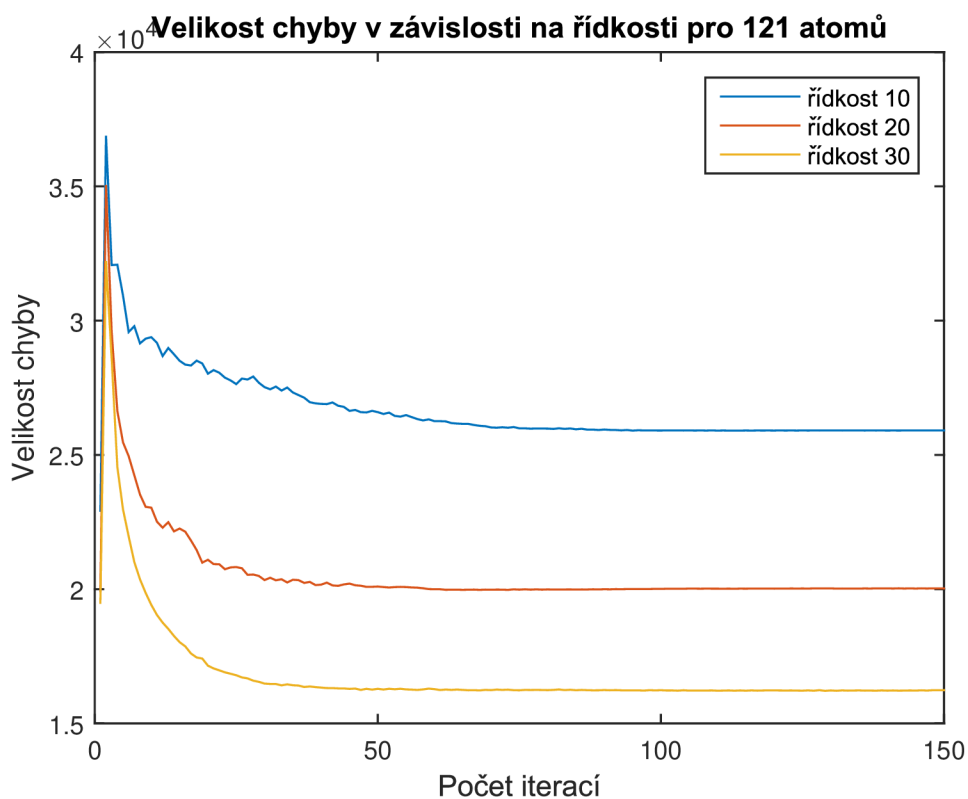


Obrázek 4.2: Graf závislosti velikosti účelové funkce na řidkosti pro 121 atomů slovníku **A**.Obrázek 4.3: Graf závislosti velikosti účelové funkce na řidkosti pro 441 atomů slovníku **A**.

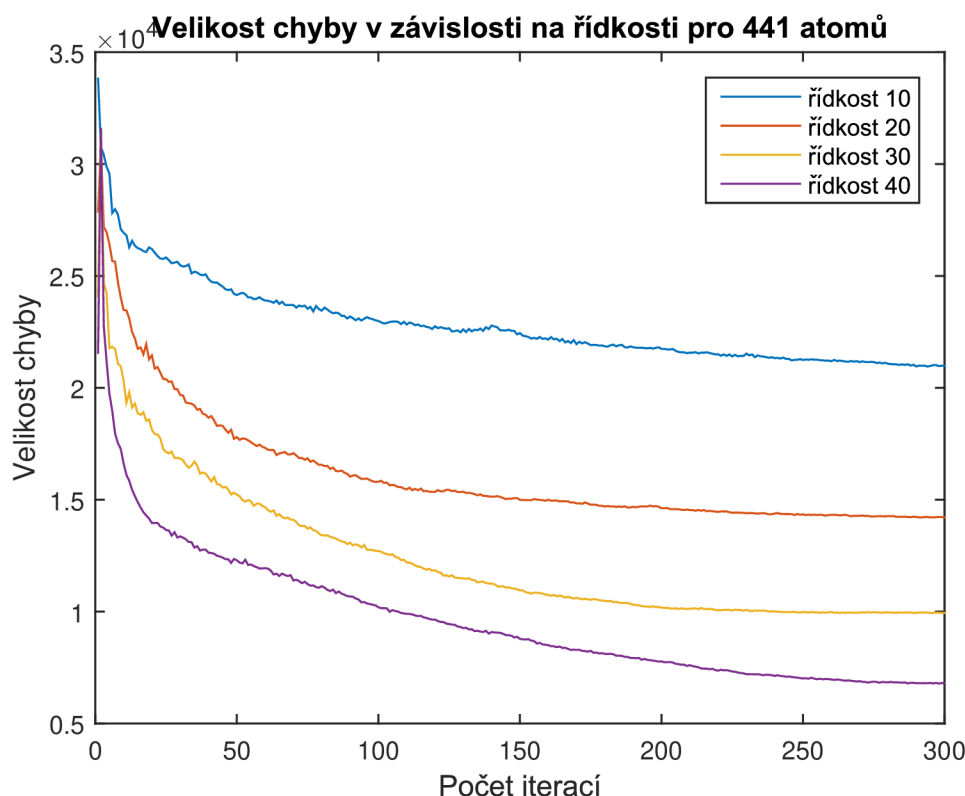
#### 4.1. SPOČTENÉ SLOVNÍKY



Obrázek 4.4: Graf závislosti velikosti účelové funkce na počtu atomů slovníku **B**.



Obrázek 4.5: Graf závislosti velikosti účelové funkce na řídkosti pro 121 atomů slovníku **B**.



Obrázek 4.6: Graf závislosti velikosti účelové funkce na řídkosti pro 441 atomů slovníku **B**.

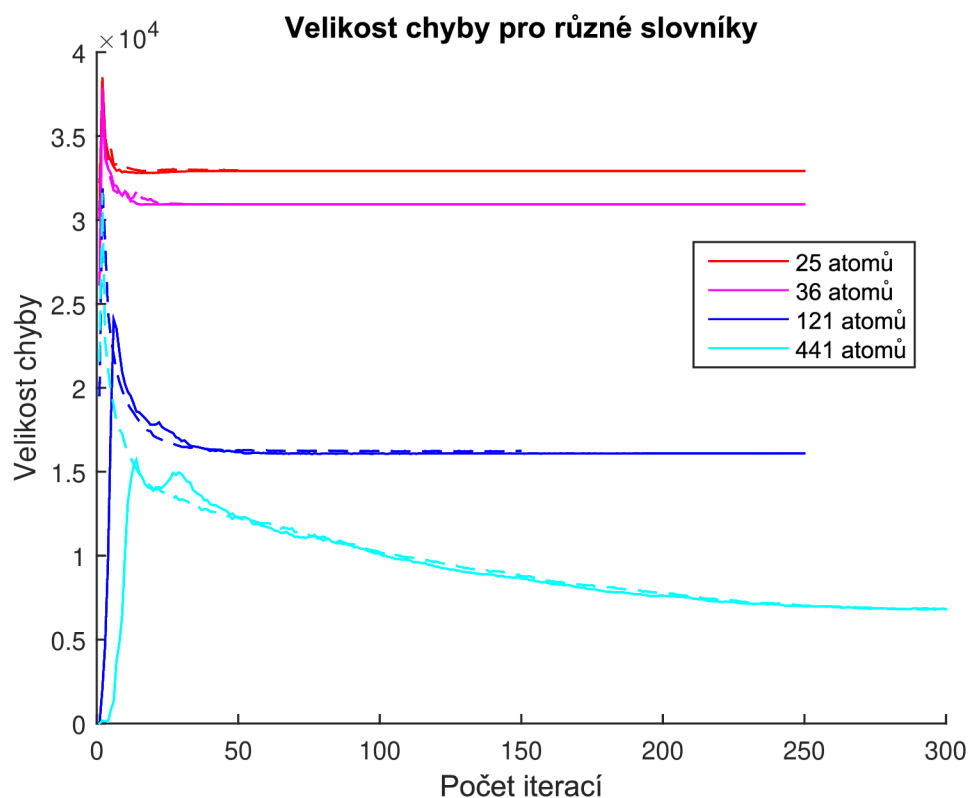
neodpovídají očekávání. To je nejspíše způsobeno přetrénováním, tedy natrénovaný slovník se přizpůsobil konkrétním trénovacím signálům, a ne jejich společným trendům. Tomu by se možná zabránilo použitím více trénovacích signálů. To bylo původně také zamýšleno, ale výrazně by se tím zvýšily nároky na paměť a výkon počítače, proto nebylo s větší množinou možné pracovat.

## 4.2. Porovnání s jinými metodami

Metodu učení slovníku algoritmem K-SVD budeme porovnávat s metodou použití fixního slovníku – tedy za slovník bereme libovolnou matici, jedná se například o startovací slovník. Při porovnávání z hlediska energetické náročnosti, je metoda učení slovníku podstatně náročnější, než při zvolení fixního slovníku. Zatímco u fixního slovníku není třeba složitých výpočtů, algoritmus K-SVD je energeticky náročný a učení slovníku může trvat v závislosti na různých faktorech (například velikost slovníku, požadovaná řídkost, počet a složitost trénovacích signálů, výkon počítače) několik dní. Konkrétně naučit se slovník nastartovaný slovníkem **A** se 441 atomy a řídkostí 40 při provedení 300 iterací trvalo 6 dní na počítači, který byl vybaven procesorem Intel Xeon E7-4820 běžícím na frekvenci 2GHz, dále operační paměť 32 GB RAM a operačním systémem 64bit Windows 10.

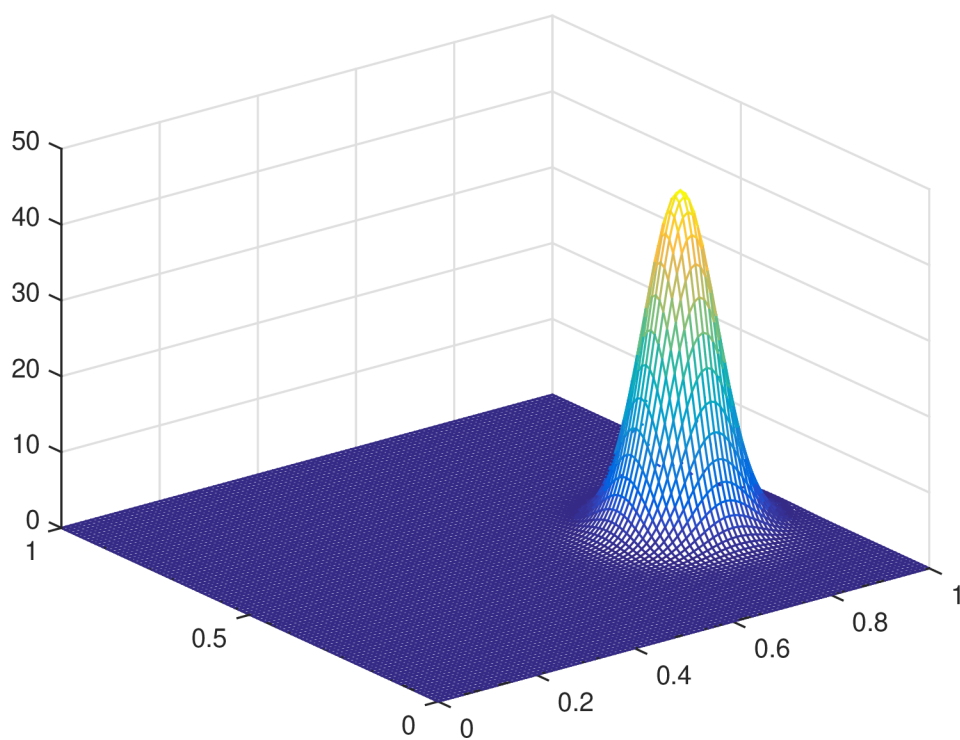
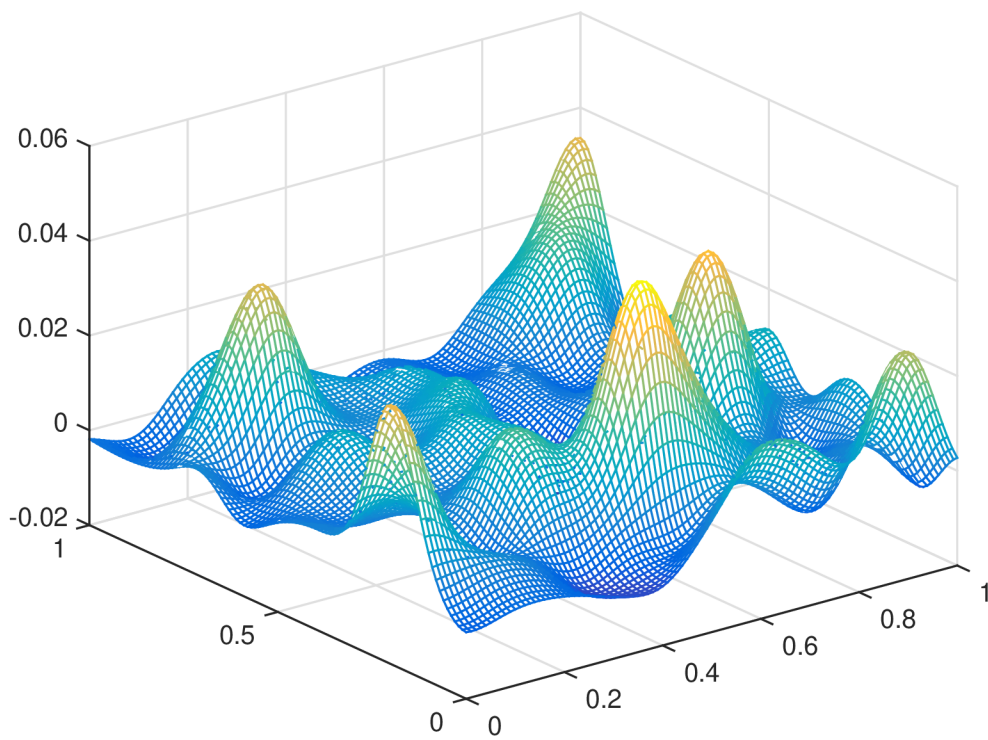
K porovnání fixního a natrénovaného slovníku jsme použili 250 kontrolních signálů, tedy takových, na kterých nebyl slovník trénován. Za fixní slovník jsme zvolili startovací slovník **A**. Nejprve jsme našli matici koeficientů ve smyslu nejmenších čtverců s povolenou řídkostí 25 (20 v případě slovníku s 25 atomy) a poté jsme slovník zredukovali na atomy odpovídající nejvýznamnějším koeficientům, našli novou matici koeficientů ve

## 4.2. POROVNÁNÍ S JINÝMI METODAMI



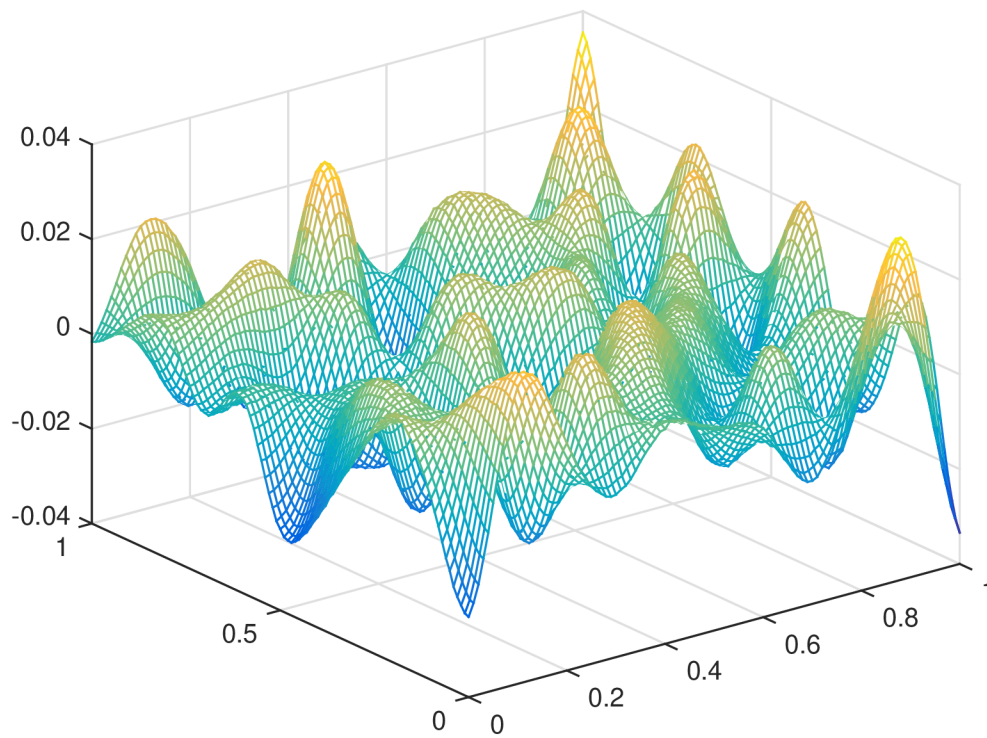
Obrázek 4.7: Porovnání různých natrénovaných slovníků. Plná čára označuje slovníky generované slovníkem **A**, přerušovaná slovníky generované slovníkem **B**.

smyslu nejmenších čtverců a určili velikost účelové funkce. Na obrázku 4.11 je porovnání pro fixní a naučený slovník s omezenou řídkostí na 5 nebo 10 atomů. Lze vidět, že nejlépe dopadl slovník generovaný slovníkem **A** s nízkou chybou pro obě zvolené řídkosti, u něj tedy stačí přenést méně koeficientů než u jiných porovnávaných slovníků. Slovník generovaný startovacím slovníkem **B** dokázal data dobře aproximovat s povolenou řídkostí 10, ale při omezení řídkosti na 5 byla jeho chyba podobná jako chyba startovacího slovníku **A**.

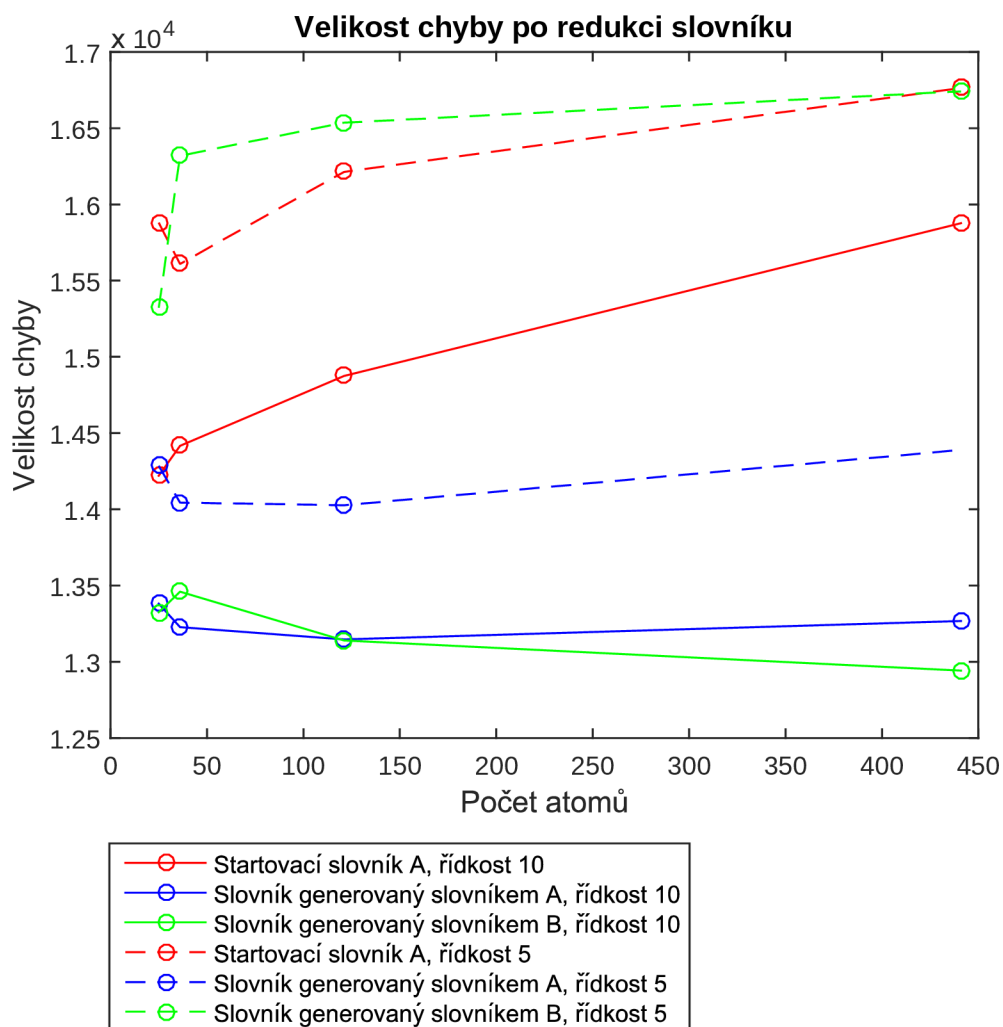
Obrázek 4.8: Příklad atomu startovacího slovníku **A**

Obrázek 4.9: Typický atom slovníku spočteného K-SVD se 121 atomy a řidkostí 30

## 4.2. POROVNÁNÍ S JINÝMI METODAMI



Obrázek 4.10: Typický atom slovníku spočteného K-SVD se 441 atomy a řidkostí 30



Obrázek 4.11: Porovnání velikostí chyby aproximace pro různé slovníky o různých velikostech – 25, 36, 121 nebo 441 atomů. Červená čára označuje fixní slovník, zelená a modrá natrénované.

## 5. ZÁVĚR

Úkolem této bakalářské práce bylo seznámení se s modelem decentralizované sensorové sítě, návrh algoritmu, který nalezne optimální reprezentaci pro trénovací signály, a porovnání s jinými metodami. K reprezentaci dat byla vybrána řídká reprezentace a naučení slovníku na základě trénovacích dat pomocí algoritmu K-SVD.

Algoritmus K-SVD je popsán v kapitole 3. V této kapitole jsou také popsány zvolené startovací slovníky a vybrány ty, s jejichž použitím je možné nalézt slovník, který aproximuje data s co nejmenší chybou. V kapitole 1 je pak popsán algoritmus OMP, který byl zvolen pro použití v první části K-SVD.

Algoritmus byl implementován v prostředí Matlab, a v kapitole 4 jsou popsány slovníky spočtené pomocí tohoto algoritmu. Jako startovací slovníky byly vybrány dva typy, každý o čtyřech různých velikostech, a řídkost byla v závislosti na velikosti slovníku omezena na 10, 20, 30, nebo 40 koeficientů. Při porovnání jednotlivých slovníků bylo zjištěno, že lépe aproximují data slovníky s více atomy, a také slovníky generované slovníkem tvarově podobným trénovacím signálům.

Ve srovnání s jinými metodami na konci kapitoly 4 bylo zjištěno, že při omezení řídkosti aproximují naučené slovníky kontrolní signály lépe, než fixní slovník. Při podrobnějším prozkoumání slovníků však bylo zjištěno, že z tvaru jednotlivých atomů naučených slovníku nelze říci, co vyjadřují, na rozdíl od zvoleného fixního slovníku.

Na příloženém DVD se nachází programy sloužící k učení slovníku, k vyobrazení jednotlivých slovníků, a jsou tam také uloženy trénovací a kontrolní signály a vypočtené slovníky včetně absolutní a relativní chyby, která vznikala mezi jednotlivými iteracemi K-SVD.

Závěr této studentské práce tedy je, že pro reprezentaci dat v sensorové síti je výhodné použít naučený slovník, obzvláště pokud byl algoritmus na učení slovníku nastartován splajnovým slovníkem, kde jednotlivé atomy mají tvar podobný jako části trénovacích signálů.



## LITERATURA

- [1] LOMTATIDZE, Alexander: *Lineární funkcionální analýza*. Učební text.
- [2] HRBÁČEK, Radek, RAJMIC, Pavel, VESELÝ, Vítězslav, a ŠPIŘÍK, Jan: *Řídké reprezentace signálů: úvod do problematiky*, Elektrevue, 2011. ISSN 1213-1539
- [3] *Reprezentace signálů: od bází k framům*, Elektrevue, 2010. ISSN 1213-1539
- [4] ČERMÁK, Libor, HLAVIČKA, Rudolf: *Numerické metody*, CERM, Brno:2008.
- [5] MACH, Václav: *Inpainting of missing audio signal samples*: doctoral thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, 2016. 125 p. Supervised by doc. Mgr. Pavel Rajmic, Ph.D.
- [6] RAJMIC, Pavel, MEYER, Florian, PAPPA, Giuseppe, HLAWATSCH, Franz *Distributed Object Tracking with Sparsity-Exploiting Clutter Rejection*. Nepublikovaná zpráva, 2008.
- [7] OZDOBINSKI, Roman *Učení slovníku pro rekonstrukci řídkého signálu*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 46 s. Vedoucí práce byl Ing. Václav Mach
- [8] AHARON, Michal, ELAD, Michael, BRUCKSTEIN, Alfred *K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation*[online]. 2006, roč. 54, č. 11, s. 4311- 4322, [cit. 2017-04-15]. DOI: 10.1109/TSP.2006.881199. Dostupné z: <[http://www.cs.technion.ac.il/~elad/publications/journals/2004/32\\_KSVD\\_IEEE\\_TSP.pdf](http://www.cs.technion.ac.il/~elad/publications/journals/2004/32_KSVD_IEEE_TSP.pdf)>.
- [9] XIAO Lin, BOYD Stephen, KIM Seung-Jean *Distributed Average Consensus with Least-Mean-Square Deviation*. [online]. [cit. 2017-05-01]. Dostupné z: <[http://www.web.stanford.edu/~boyd/papers/pdf/lmsc\\_mtns06.pdf](http://www.web.stanford.edu/~boyd/papers/pdf/lmsc_mtns06.pdf)>.
- [10] HLINKA, Ondrej, HLAWATSCH, Franz, a DJURIC, Pavel *Consensus-based Distributed Particle Filtering With Distributed Proposal Adaptation*. IEEE Transactions on Signal Processing. USA: IEEE, 62(12): pp. 3029-3041. ISSN 1053-587X.

## 6. SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

supp	nosič vektoru – počet nenulových prvků
MP	<i>Matching pursuit</i>
OMP	<i>Orthogonal matching pursuit</i>
SVD	<i>Singular value decomposition</i> – singulární rozklad
GLF	<i>Global likelihood function</i> – globální věrohodnostní funkce
K-SVD	algoritmus K-SVD

## 7. OBSAH PŘILOŽENÉHO DVD

Na přiloženém DVD se nachází několik funkcí, které lze spustit v prostředí Matlab:

- `ksvd.m` – mírně upravený algoritmus K-SVD poskytnutý doc. Mgr. Pavlem Rajmícem Ph.D.
- `omp.m` – algoritmus OMP poskytnutý doc. Mgr. Pavlem Rajmícem Ph.D.
- `norm_cols.m` – funkce, která normalizuje každý řádek vstupní matice. Volá se v K-SVD.
- `pokracovani_ksvd.m` – funkce, která slouží k volání `ksvd.m` a každých 10 iterací uloží spočtený slovník a příslušnou absolutní a relativní chybu. Lze ji použít i v případě, že chceme pokračovat v počítání dříve spočteného slovníku.
- `prevod_2D_3D.m` – většina uložených slovníků je převedena do 2D tvaru, který je nutný pro vstup do `ksvd.m`. Pokud je vstupem slovník nebo matice signálů ve tvaru 2D, převede je na 3D, a naopak.
- `ukazka_atomu.m` – Vstupem je 2D nebo 3D slovník a funkce zobrazí postupně všechny atomy tohoto slovníku.

Dále se zde nachází složka `vstupni_udaje` ve které se nachází následující položky:

- `data_kontrolni_sloupcove.mat` – po načtení v prostředí Matlab se objeví proměnná `signaly_kontrolni_sloupcove`, ve které se nachází kontrolní signály převedené do 2D podoby.
- `data_sloupcove.mat` – po načtení se objeví proměnná `signaly_sloupcove`, kde jsou uloženy trénovací signály ve formě 2D matice
- `slovník_A.mat`, `slovník_B.mat` – v těchto souborech se nachází startovací slovníky ve tvaru `A_X`, případně `B_X`, kde za `X` lze dosadit čtyři velikosti slovníků – 25, 36, 121 a 441.

Ve složce `spoctene_slovníky` se nachází slovníky vypočtené pomocí `ksvd.m`. Soubory jsou ve tvaru `slovník_X_Y_Z.mat`, kde `X` je 1, pokud se jedná o slovník generovaný slovníkem **A**, nebo `X` je 2, pokud se jedná o slovník generovaný slovníkem **B**. `Y` značí velikost slovníku (25, 36, 121 nebo 441) a `Z` označuje použitou řídkost (10, 20, 30 nebo 40). Příklad souboru je `slovník_1_121_30.mat`. Po načtení slovníku se objeví proměnné `slovník`, označující spočtený slovník, `abs_chyba`, kde jsou uloženy velikosti absolutní chyby v jednotlivých iteracích, a `rel_chyba`, kde jsou uloženy velikosti relativní chyby mezi jednotlivými iteracemi.