



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

# KOMPRIMOVANÉ VZORKOVÁNÍ PRO EFEKTIVNÍ SLEDOVÁNÍ OBJEKTU SENZOROVOU SÍTÍ

COMPRESSIVE SAMPLING FOR EFFECTIVE TARGET TRACKING IN A SENSOR NETWORK

## DIPLOMOVÁ PRÁCE

MASTER'S THESIS

## AUTOR PRÁCE

AUTHOR

Bc. Ondřej Klimeš

## VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. Pavel Rajmic, Ph.D.

BRNO 2019



## Zadání diplomové práce

Ústav:	Ústav matematiky
Student:	<b>Bc. Ondřej Klimeš</b>
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	<b>doc. Mgr. Pavel Rajmic, Ph.D.</b>
Akademický rok:	2018/19

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 diplomové práce:

### **Komprimované vzorkování pro efektivní sledování objektu sensorovou sítí**

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

Práce bude vývojem a optimalizací metod pro výměnu dat v sensorové síti. Přesněji půjde o tzv. decentralizovanou síť, která musí data zpracovávat svépomocí, bez účasti dodatečného výpočetního centra. Síť senzorů 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 ale máme apriorní, statisticky formulované informace o charakteru dat, a víme, že jsou tzv. řídká, což umožňuje úsporu komunikačních nároků pomocí komprese. K tomu bude využit moderní přístup označovaný komprimační vzorkování (compressive sampling) či komprimované snímání (compressed sensing), což znamená, že komprese probíhá nepřímo.

Téma je řešeno ve spolupráci s TU Vídeň, tým prof. Franze Hlawatsche.

**Cíle diplomové práce:**

Seznámit se s pravděpodobnostním modelováním sensorových sítí, výměny dat a distribuovanými výpočty v nich.

Seznámit se se simulačním programem pro Matlab. Navrhnout metody pro výměnu řídkých dat v síti pomocí komprimačního vzorkování.

Implementovat algoritmy v Matlabu a začlenit je do existujícího simulátoru.

Navrhnout metodu porovnání přístupů (z hlediska energetické náročnosti výpočtu, z hlediska energetické náročnosti spojení a výměny dat, apod.).

Zhodnotit výsledky.

**Seznam doporučené literatury:**

RAJMIC, P. a DAŇKOVÁ, M. Úvod do řídkých reprezentací signálů a komprimovaného snímání. Skriptum, Vysoké učení technické v Brně, 2014. ISBN 978-80-214-5169-8.

ŠPIŘÍK, J., RAJMIC, P. a VESELÝ, V. Reprezentace signálů: od bází k framům, Elektrevue, 2010. ISSN 1213-1539. ELAD, M. Sparse and redundant representations. Springer, 2010.

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

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

---

prof. RNDr. Josef Šlapal, CSc.  
ředitel ústavu

---

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



## **Abstrakt**

Diplomová práce se zabývá sledováním objektu. Ke sledování je použita decentralizovaná sensorová síť využívající distribuovaný částicový filtr s věrohodnostním konsenzem. Tento konsenzus je založen na řídké reprezentaci lokální věrohodnostní funkce pomocí vhodných slovníků. V této práci se porovnávají hojně používaný Fourierův slovník s námi navrženým slovníkem tzv. B-splajnů. Zároveň je díky řídkosti distribuovaných dat možné implementovat metodu komprimovaného snímání. Výsledky jsou porovnávány z hlediska přesnosti sledování a komunikační náročnosti. Součástí práce jsou také skripty a funkce v jazyce MATLAB.

## **Summary**

The master's thesis deals with target tracking. For this a decentralized sensor network using distributed particle filter with likelihood consensus is used. This consensus is based on a sparse representation of local likelihood function in a suitable chosen dictionary. In this thesis two dictionaries are compared: the widely used Fourier dictionary and our proposed B-splines. At the same time, thanks to the sparsity of distributed data, it is possible to implement compressed sensing method. The results are compared in terms of tracking error and communication costs. The thesis also contains scripts and functions in MATLAB.

## **Klíčová slova**

Senzorová síť, sledování pohybu objektu, distribuovaný sekvenční bayesovský odhad, částicový filtr, řídkost, OMP, komprimované snímání, B-splajny

## **Keywords**

Sensor network, target tracking, distributed sequential Bayesian estimation, particle filter, sparsity, OMP, compressed sensing, B-splines

KLIMEŠ, O. *Komprimované vzorkování pro efektivní sledování objektu sensorovou sítí*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2019. 70 s. Vedoucí doc. Mgr. Pavel Rajmic, Ph.D.



Prohlašuji, že jsem diplomovou práci *Komprimované vzorkování pro efektivní sledování objektu senzorovou sítí* vypracoval samostatně pod vedením doc. Mgr. Pavla Rajmice, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Bc. Ondřej Klimeš



Děkuji především svému vedoucímu práce doc. Mgr. Pavlu Rajmicovi, Ph.D. za vstřícnost, cenné rady a časté konzultace. Dále děkuji Ing. Renemu Reppovi za poskytnuté programy a rady. Děkuji také rodičům a prarodičům za podporu během celých pěti let studia. V neposlední řadě chci poděkovat přítelkyni, spolužákům, spolubydlícím a všem přátelům za trpělivost a psychickou podporu.

Bc. Ondřej Klimeš



# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Matematický aparát</b>	<b>6</b>
2.1	Pravděpodobnost . . . . .	6
2.1.1	Základní definice . . . . .	6
2.1.2	Podmíněná pravděpodobnost . . . . .	7
2.1.3	Bayesova věta . . . . .	7
2.1.4	Nezávislost . . . . .	7
2.1.5	Náhodná veličina . . . . .	8
2.1.6	Náhodný vektor . . . . .	9
2.1.7	Podmíněná hustota . . . . .	11
2.1.8	Bayesovská statistika . . . . .	12
2.2	Grafy . . . . .	13
2.3	Aproximace funkcí . . . . .	14
2.3.1	Metoda nejmenších čtverců . . . . .	14
2.3.2	Aproximace funkcí dvou proměnných . . . . .	16
2.3.3	Slovníky pro aproximaci funkce dvou proměnných . . . . .	17
2.3.4	Řídké řešení soustav rovnic . . . . .	21
2.4	Komprimované snímání . . . . .	23
2.4.1	Způsoby komprimování dat . . . . .	24
2.4.2	Princip komprimovaného snímání . . . . .	24
2.4.3	Volba kompresní matice . . . . .	25
<b>3</b>	<b>Distribuovaný částicový filtr v sensorových sítích</b>	<b>27</b>
3.1	Senzorové sítě . . . . .	27
3.2	Sekvenční bayesovský odhad . . . . .	28
3.3	Částicové filtry . . . . .	30
3.4	Distribuovaný sekvenční bayesovský odhad . . . . .	32
3.5	Distribuované částicové filtry . . . . .	32
3.5.1	Přehled DPF algoritmů . . . . .	33
3.5.2	Věrohodnostní konsenzus . . . . .	33
3.5.3	Částicová reprezentace . . . . .	35
<b>4</b>	<b>Model sensorové sítě</b>	<b>37</b>
4.1	Měřící proces . . . . .	37
4.2	Odvození lokální věrohodnostní funkce . . . . .	40
4.3	Aproximace lokální věrohodnostní funkce . . . . .	41
4.4	Částicový filtr . . . . .	43
4.5	Promazávání slovníku . . . . .	44
4.6	Kompresní věrohodnostní konsenzus . . . . .	46
4.7	Algoritmus modelu sensorové sítě . . . . .	47

## OBSAH

<b>5</b>	<b>Simulace a výsledky</b>	<b>49</b>
5.1	Nastavení modelu . . . . .	49
5.2	Hodnotící kritéria . . . . .	51
5.2.1	Chyba sledování . . . . .	51
5.2.2	Úspěšnost sledování . . . . .	52
5.2.3	Komunikační náklady . . . . .	54
5.2.4	Výpočetní náročnost . . . . .	55
5.3	Výsledky . . . . .	56
5.3.1	Porovnání slovníků . . . . .	56
5.3.2	Použití komprimovaného snímání . . . . .	59
5.3.3	Efektivita promazávání slovníku B-splajnů . . . . .	60
<b>6</b>	<b>Závěr</b>	<b>62</b>
<b>7</b>	<b>Seznam použitých zdrojů</b>	<b>63</b>
<b>8</b>	<b>Seznam použitých zkratk a symbolů</b>	<b>66</b>
<b>9</b>	<b>Přílohy</b>	<b>68</b>



# Seznam obrázků

2.1	Příklad souvislého grafu (vlevo) a grafu, který není souvislý (vpravo) . . .	14
2.2	Jednodimenzionální B-splajny stupně 0 až 3 . . . . .	17
2.3	Ukázka slovníku B-splajnů stupně 3 na intervalu $\langle -2, 2 \rangle$ . . . . .	18
2.4	Ukázka slovníku B-splajnů stupně 3 na ploše $\langle -5, 5 \rangle \times \langle -5, 5 \rangle$ . . . . .	19
2.5	Ukázka 1-D Fourierovy báze pro hodnoty $R = 2$ a $d = 100$ . . . . .	20
2.6	Ukázka atomu $\psi_9(x, y)$ z Fourierova slovníku pro $d = 100$ , $K = 9$ . . . . .	20
2.7	Grafické znázornění systému $\mathbf{y} = \mathbf{Pz}$ . . . . .	24
2.8	Celkové schéma komprimovaného snímání. . . . .	25
3.1	Dva příklady aplikace sensorových sítí. . . . .	28
3.2	Schéma jednoho kroku sekvenčního bayesovského odhadu. . . . .	30
4.1	Model sensorové sítě. . . . .	38
4.2	Ukázka tvaru log-LLF. . . . .	41
4.3	Ukázka tvaru log-LLF s částicemi. . . . .	42
4.4	Ukázka aproximace log-LLF. . . . .	42
4.5	Příklady částicové reprezentace apriorní hustoty pravděpodobnosti. . . . .	44
4.6	Ukázka log-LLF a její částicové reprezentace s „extra“ částicemi . . . . .	45
4.7	Ukázka aproximace log-LLF pomocí B-splajnů . . . . .	45
5.1	Oblast zájmu se sensorovou sítí a zkušební trajektorií sledovaného objektu	49
5.2	Ukázka odhadů polohy pro jeden experiment, oblast zájmu je přiblížena . .	51
5.3	Ukázka průběhu chyb pro jednotlivé senzory. . . . .	52
5.4	Příklad histogramu hodnot chyb RMSE. . . . .	53
5.5	Průměrná řídkost v závislosti na čase pro simulace 1 a 2 . . . . .	57
5.6	Výsledky testu 3 – průměrná chyba v závislosti na čase . . . . .	58

# Seznam tabulek

5.1	Seznam parametrů, které nastavujeme pro každou simulaci. . . . .	50
5.2	Příklad způsobů komunikace . . . . .	55
5.3	Výčet funkcí s největší časovou náročností . . . . .	56
5.4	Výsledky testu 1 . . . . .	56
5.5	Výsledky testu 2 . . . . .	57
5.6	Výsledky testu 3 – Přidání částic . . . . .	58
5.7	Výsledky testu 4 – Komprimované snímání pro Fourierův slovník . . . . .	59
5.8	Výsledky testu 5 – Komprimované snímání pro slovník B-splajnů . . . . .	60
5.9	Výsledky testu 6 . . . . .	61
5.10	Výsledky testu 7 . . . . .	61
9.4	Seznam vybraných proměnných . . . . .	70

# 1. Úvod

Senzorové sítě nacházejí v dnešní době široké uplatnění v mnoha oblastech jako např. vojenství, zdravotnictví, řízení dopravy nebo monitorování životního prostředí. Konkrétním příkladem může být sledování polohy pohyblivých objektů. Speciálním typem sítí je tzv. decentralizovaná síť, ve které není přítomna žádná centrální výpočetní jednotka, takže senzory musí pro správné fungování navzájem spolupracovat.

Princip činnosti senzorů lze matematicky popsat pravděpodobnostním modelem, který k výpočtům používá sekvenční bayesovský odhad. Data, která si senzory mezi sebou posílají, mají tedy pravděpodobnostní charakter.

Vzájemná komunikace je v této problematice klíčovým aspektem a je přirozené hledat způsoby, jak komunikační náklady senzorové sítě šetřit. Abychom dosáhli nějakých úspor, je možné nalézt pro distribuovaná data určitou řídkou reprezentaci, díky které je můžeme dále komprimovat. Pro tuto reprezentaci se typicky používá tzv. Fourierův slovník [11]. V této práci budeme testovat také slovník tzv. B-splajnů, který má tu výhodu, že jsou jeho atomy lokalizované v prostoru stavových veličin.

K další kompresi nám bude sloužit v současnosti stále více využívaná metoda komprimovaného snímání (též komprimační vzorkování) založená na řídkosti dat. Příkladem oblasti, kde se již komprimované snímání úspěšně používá, může být perfuzní zobrazování pomocí magnetické rezonance [4] nebo jedno-pixelová kamera [5].

Práce se zabývá modelem, který předpokládá stálý výskyt právě jednoho pohyblivého objektu. Model pracující s nejistou přítomností sledovaného objektu nalezneme např. v [26]. Dalším příkladem je model, který umožňuje sledování více objektů najednou [17].

Cílem této práce je seznámení se s teorií pravděpodobnostního modelu senzorové sítě a se simulačním programem v prostředí MATLABu. Dalším cílem je implementovat nové metody a jiná vylepšení do tohoto programu a na závěr porovnat přístupy podle různých hledisek (komunikační náročnost, přesnost měření aj.).

Ve druhé kapitole jsou zmíněny potřebné pasáže některých matematických disciplín (teorie pravděpodobnosti, teorie grafů, aproximace funkcí, řídkost a komprimované snímání). Třetí kapitola popisuje obecně senzorovou síť využívající distribuovaný částicový filtr, který je založen na sekvenčním bayesovském odhadu. Ve čtvrté kapitole uvedeme již konkrétní model senzorové sítě pro sledování pohybu objektu. Pátá kapitola obsahuje popis simulace tohoto modelu a výsledky, které jsme z provedených simulací dostali.

Práce vznikla ve spolupráci s prof. Franzem Hlawatschem a Ing. Renem Reppem z Technické univerzity Vídeň, se kterými jsme diskutovali různé možnosti a postupy. Od nich jsme také obdrželi základní simulační program senzorové sítě, do kterého jsme implementovali nové funkce.

## 2. Matematický aparát

V této úvodní kapitole probereme potřebné pasáže některých matematických disciplín, které využíváme v problematice sledování objektu senzorem sítí. Uvedeme některé definice a věty z teorie pravděpodobnosti a základní definice pro teorii grafů. Následně se zaměříme na aproximace funkcí, řídkost a s řídkostí spojené komprimované snímání.

### 2.1. Pravděpodobnost

Z teorie pravděpodobnosti nejdříve uvedeme základní definice a postupně se dostaneme až k náhodné veličině a náhodnému vektoru. Pro náhodný vektor uvedeme Bayesovu větu, která bude důležitým prvkem pro kapitolu 3 o senzorných sítích.

#### 2.1.1. Základní definice

Symbolem  $\omega$  označíme *elementární jev*, což je výsledek nějakého pokusu nebo děje, jenž není jednoznačně určen podmínkami, za nichž se odehrává. Množinu všech elementárních jevů označujeme  $\Omega$  a nazýváme ji *prostor elementárních jevů*. *Náhodným jevem*  $A$  pak rozumíme libovolnou podmnožinu základního prostoru  $\Omega$ , tedy  $A \subseteq \Omega$ . Doplněk k náhodnému jevu  $\bar{A} = \Omega \setminus A$  nazveme *opačný jev* [16]. V následující definici provedeme preciznější vysvětlení.

**Definice 2.1.** Nechť  $\Omega$  je libovolná neprázdná množina. Neprázdný systém  $\Sigma$  podmnožin množiny  $\Omega$  se nazývá  $\sigma$ -*algebra*, jestliže platí následující podmínky:

1. Pro každý náhodný jev  $A \in \Sigma$  je  $\bar{A} \in \Sigma$ .
2. Pro každou posloupnost náhodných jevů  $A_i \in \Sigma, i = 1, 2, \dots$  je

$$\bigcup_{i=1}^{\infty} A_i \in \Sigma.$$

Prvek  $A$  systému  $\Sigma$  nazveme *náhodný jev*, dvojice  $(\Omega, \Sigma)$  se nazývá *jevové pole*.

**Definice 2.2.** Nechť  $\Omega$  je neprázdná množina a nechť  $\Sigma$  je  $\sigma$ -algebra náhodných jevů definovaných na  $\Omega$ . *Pravděpodobnost*  $P(A)$  náhodného jevu  $A \in \Sigma$  je funkce  $P : \Sigma \rightarrow \mathbb{R}$  s vlastnostmi:

1.  $P(A) \geq 0$  pro všechny náhodné jevy  $A \in \Sigma$ .
2.  $P(\Omega) = 1$ .
3. Pro každou posloupnost disjunktních náhodných jevů  $A_i \in \Sigma, i = 1, 2, \dots$  je

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

Uspořádaná trojice  $(\Omega, \Sigma, P)$  se pak nazývá *pravděpodobnostní prostor* (též *Kolmogorovo pravděpodobnostní pole* [24]).

### 2.1.2. Podmíněná pravděpodobnost

**Definice 2.3.** Pravděpodobnost náhodného jevu  $A \in \Sigma$  za podmínky, že nastane náhodný jev  $B \in \Sigma$ ,  $P(B) \neq 0$ , je *podmíněná pravděpodobnost*

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (2.1)$$

Podmíněná pravděpodobnost má tuto základní vlastnost:

$$P(A_1 \cap \dots \cap A_n) = P(A_1)P(A_2|A_1) \dots P(A_n|A_1 \cap \dots \cap A_{n-1}).$$

**Věta 2.4.** (*Věta o úplné pravděpodobnosti*) Nechť  $B_1, B_2, \dots$  je úplná soustava jevů (tvoří rozklad  $\Omega$ ), nechť  $P(B_i) > 0$  pro všechna  $i = 1, 2, \dots$ . Potom pro libovolný jev  $A \in \Sigma$  platí

$$P(A) = \sum_{i=1}^{\infty} P(B_i)P(A|B_i). \quad (2.2)$$

### 2.1.3. Bayesova věta

Ze vztahu pro podmíněnou pravděpodobnost (2.1) a z věty o úplné pravděpodobnosti (2.2) dostáváme přímo následující větu.

**Věta 2.5.** (*Bayesova*) Nechť  $B_1, B_2, \dots$  je úplná soustava jevů a  $P(B_i) > 0$  pro všechna  $i = 1, 2, \dots$ . Pak platí

$$P(B_j|A) = \frac{P(B_j)P(A|B_j)}{\sum_{i=1}^{\infty} P(B_i)P(A|B_i)}, \quad j = 1, 2, \dots \quad (2.3)$$

Pravděpodobnosti  $P(B_j|A)$  se nazývají *aposteriorní* pravděpodobnosti, protože se stanovují až po provedení pokusu. Naproti tomu jednotlivé pravděpodobnosti  $P(B_j)$  nazýváme *apriorní*, neboť se počítají ještě před samotným pokusem, jehož výsledkem je jev  $A$ . Bayesův vzorec nám tedy umožňuje vypočítat aposteriorní pravděpodobnost pomocí apriorních.

### 2.1.4. Nezávislost

**Definice 2.6.** Mějme pravděpodobnostní prostor  $(\Omega, \Sigma, P)$ . Dva náhodné jevy  $A, B \in \Sigma$  nazveme *nezávislé*, jestliže platí

$$P(A \cap B) = P(A)P(B).$$

Náhodné jevy  $A_1, A_2, \dots$  se nazývají *nezávislé*, jestliže pro každé  $k \in \mathbb{N}$  a pro každou  $k$ -tici náhodných jevů  $A_{i_1}, \dots, A_{i_k}$  platí

$$P\left(\bigcap_{j=1}^k A_{i_j}\right) = \prod_{j=1}^k P(A_{i_j}).$$

Pokud jsou dva náhodné jevy nezávislé, zřejmě platí  $P(A|B) = P(A)$  a  $P(B|A) = P(B)$  (přímý důsledek vztahu pro podmíněnou pravděpodobnost (2.1)).

## 2.1. PRAVDĚPODOBNOST

### 2.1.5. Náhodná veličina

Před definicí pojmu náhodné veličiny musíme zavést  $\sigma$ -algebru pro reálnou přímku  $\mathbb{R}$ . Nejmenší  $\sigma$ -algebra (nejmenší vzhledem k množinové inkluzi), která obsahuje všechny typy intervalů na  $\mathbb{R}$  (tj.  $\langle a, b \rangle, \langle a, b], (a, b), (-\infty, b), (-\infty, b], (a, \infty), \langle a, \infty \rangle$ ) se nazývá *Borelovská  $\sigma$ -algebra* (značíme  $\mathcal{B}$ ). K tomu, abychom takovou  $\sigma$ -algebru vytvořili, stačí uvažovat pouze intervaly typu  $(-\infty, x)$  [29]. Uvažujme reálnou funkci  $X$  definovanou na  $\Omega$  a libovolná  $a, b \in \mathbb{R}$ . Zavedeme označení pro podmnožiny  $\Omega$ :

$$(X < b) = \{\omega \in \Omega, X(\omega) < b\}, \quad (2.4)$$

$$(a < X < b) = \{\omega \in \Omega, a < X(\omega) < b\}. \quad (2.5)$$

**Definice 2.7.** Nechť  $(\Omega, \Sigma, P)$  je pravděpodobnostní prostor. Reálnou funkci  $X$  definovanou na  $\Omega$  ( $X : \Omega \rightarrow \mathbb{R}$ ), pro kterou platí

$$x \in \mathbb{R} \Rightarrow (X < x) \in \Sigma,$$

nazveme *náhodná veličina (proměnná)*.

Tato definice nám dává možnost určit pravděpodobnost, že náhodná veličina  $X$  nabývá hodnoty z intervalu  $(-\infty, x)$ ,  $x \in \mathbb{R}$ . Z vlastností Borelovské  $\sigma$ -algebry  $\mathcal{B}$  můžeme dále odvodit následující definici [29].

**Definice 2.8.** Množinová funkce

$$\begin{aligned} P_x(B) &= P(\{\omega \in \Omega, X(\omega) \in B\}) \\ &= P(X^{-1}(B)), \quad B \in \mathcal{B}, \end{aligned}$$

se nazývá *rozdělení pravděpodobnosti* náhodné veličiny  $X$ , kde  $X^{-1}(\cdot)$  představuje vzor množiny  $B$  v  $\Sigma$ :

$$X^{-1}(B) = \{\omega \in \Omega : X(\omega) \in B\} \in \Sigma.$$

**Definice 2.9.** Nechť  $X$  je náhodná veličina definovaná na pravděpodobnostním prostoru  $(\Omega, \Sigma, P)$ . Reálnou funkci

$$F(x) = P(X < x).$$

nazýváme *distribuční funkcí* náhodné veličiny  $X$ .

Distribuční funkcí je náhodná veličina  $X$  jednoznačně určena. To znamená, že distribuční funkce určuje jednoznačně rozdělení pravděpodobnosti  $P$ .

**Definice 2.10.** Náhodná veličina  $X$  je *diskrétní*, jestliže nabývá nejvýše spočetně mnoha hodnot  $x = x_1, x_2, \dots$ . Její *pravděpodobnostní funkce* je posloupnost

$$p(x) = P(X = x) > 0, \quad \text{pro } x = x_1, x_2, \dots,$$

s vlastností  $\sum_x p(x) = 1$ .

**Příklad 2.11.** Řekneme, že náhodná veličina  $X$  má *Poissonovo rozdělení* s parametrem  $\lambda > 0$ , jestliže má pravděpodobnostní funkci

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad \text{pro } k = 0, 1, \dots,$$

což značíme  $X \sim \text{Po}(\lambda)$ . Snadno vidíme, že jde o diskrétní náhodnou veličinu. Poissonovo rozdělení popisuje výskyt jevu v určitém intervalu (časovém, délkovém apod.).

**Definice 2.12.** Náhodná veličina  $X$  je *spojitá*, jestliže má spojitou distribuční funkci  $F(x)$  pro všechna  $x \in (-\infty; +\infty)$ . Její *hustota pravděpodobnosti* (někdy zkráceně jen hustota) je taková nezáporná funkce  $f(x)$ , že pro všechna  $x \in (-\infty; +\infty)$  je

$$F(x) = \int_{-\infty}^x f(t) dt$$

s vlastností  $\int_{-\infty}^{\infty} f(t) dt = 1$ .

**Příklad 2.13.** Řekneme, že náhodná veličina  $X$  má *normální (Gaussovo) rozdělení* s parametry  $\mu$  a  $\sigma^2$ , jestliže má hustotu pravděpodobnosti

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

značíme  $X \sim N(\mu, \sigma^2)$ . Normální rozdělení má v teorii pravděpodobnosti obrovský význam, nejenom protože mnoho náhodných veličin v praktických příkladech má právě toto rozdělení.

### 2.1.6. Náhodný vektor

V praxi se setkáváme s případy, kdy vyšetřujeme několik náhodných veličin současně. Proto rozšíříme pojem náhodná veličina na náhodný vektor. Mějme mnohorozměrnou funkci  $\mathbf{X}(\omega) = (X_1(\omega), \dots, X_n(\omega))^T$  a obdobně jako v (2.4) a (2.5) zavedme označení:

$$\begin{aligned} (\mathbf{X} < \mathbf{x}) &= \{\omega \in \Omega, X_1(\omega) < x_1, \dots, X_n(\omega) < x_n\}, \\ (\mathbf{a} < \mathbf{X} < \mathbf{b}) &= \{\omega \in \Omega, a_1 < X_1(\omega) < x_1, \dots, a_n < X_n(\omega) < x_n\}. \end{aligned}$$

Nerovnostní podmínky chápeme tak, že platí všechny zároveň. Tento zápis budeme používat i dále. Podobně jako u náhodné veličiny označme  $\mathcal{B}_n$  nejmenší  $\sigma$ -algebru nad intervaly tvaru

$$(-\infty, x_1) \times (-\infty, x_2) \times \dots \times (-\infty, x_n) \tag{2.6}$$

pro všechny vektory  $\mathbf{x} \in \mathbb{R}^n$ .

**Definice 2.14.** Mějme pravděpodobnostní prostor  $(\Omega, \Sigma, P)$ . Reálná vektorová funkce  $\mathbf{X}(\omega) = (X_1(\omega), \dots, X_n(\omega))^T$  definovaná na  $\Omega$ , pro kterou platí

$$\mathbf{x} \in \mathbb{R}^n \Rightarrow (\mathbf{X} < \mathbf{x}) \in \Sigma,$$

se nazývá *náhodný vektor*. Podobně jako pro náhodnou veličinu i pro náhodný vektor definujeme *sduženou distribuční funkci*

$$F(\mathbf{x}) = P(\mathbf{X} < \mathbf{x}).$$

## 2.1. PRAVDĚPODOBNOST

**Definice 2.15.** Pro náhodný vektor  $\mathbf{X} = (X_1, \dots, X_n)^\top$ , který nabývá pouze nejvýše spočetně mnoha hodnot, definujeme *sduženou pravděpodobností funkci*

$$p(\mathbf{x}) = P(\mathbf{X} = \mathbf{x}),$$

neboli

$$p(x_1, \dots, x_n) = P(X_1 = x_1, \dots, X_n = x_n).$$

Pomocí sdužené pravděpodobností funkce lze odvodit příslušnou *sduženou distribuční funkci* vztahem:

$$F(\mathbf{x}) = F(x_1, \dots, x_n) = \sum_{t_1 < x_1} \cdots \sum_{t_n < x_n} p(t_1, \dots, t_n). \quad (2.7)$$

**Definice 2.16.** Pro spojité náhodné vektory definujeme *sduženou hustotu pravděpodobnosti* jako funkci  $f: \mathbb{R}^n \rightarrow \langle 0, \infty \rangle$  splňující

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(\mathbf{x}) d\mathbf{x} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x_1, \dots, x_n) dx_1, \dots, dx_n = 1.$$

**Definice 2.17.** Mějme náhodný vektor  $\mathbf{X} = (X_1, \dots, X_n)^\top$ . Nechť  $k_1, \dots, k_r$  jsou různá celá čísla taková, že  $1 \leq k_i \leq n$  pro  $i = 1, \dots, r$ , přičemž  $1 \leq r < n$ . Rozdělení náhodného vektoru  $(X_{k_1}, \dots, X_{k_r})^\top$  se nazývá *marginální*. Ze sdužené distribuční funkce (2.7) snadno odvodíme *marginální distribuční funkci*  $G$  náhodných veličin  $X_1, \dots, X_k$  (pro  $1 \leq k < n$ ) vztahem

$$G(x_1, \dots, x_k) = F(x_1, \dots, x_k, +\infty, \dots, +\infty) = \lim_{x_{k+1}, \dots, x_n \rightarrow \infty} F(x_1, \dots, x_k, x_{k+1}, \dots, x_n).$$

**Věta 2.18.** [1] Nechť náhodný vektor  $\mathbf{X} = (X_1, \dots, X_n)^\top$  má hustotu pravděpodobnosti  $f(\mathbf{x})$ . Pak marginální hustota vektoru  $\mathbf{Y} = (X_1, \dots, X_r)^\top$  je rovna

$$q(y_1, \dots, y_r) = \int_{\mathbb{R}^{n-r}} f(y_1, \dots, y_r, z_1, \dots, z_{n-r}) dv(z_1, \dots, z_{n-r}), \quad (2.8)$$

kde funkce  $v$  je  $\sigma$ -konečná míra na  $\sigma$ -algebře  $\mathcal{B}_{n-r}$ .

**Definice 2.19.** Budeme říkat, že náhodné veličiny  $X$  a  $Y$  jsou *nezávislé*, jestliže platí

$$P(X < x, Y < y) = P(X < x)P(Y < y),$$

tj. jestliže se dvojrozměrná distribuční funkce náhodného vektoru  $(X, Y)$  rovná součinu distribuční funkce náhodné veličiny  $X$  a náhodné veličiny  $Y$ . Obdobně pro  $n$  náhodných veličin  $X_1, \dots, X_n$  řekneme, že jsou *nezávislé*, jestliže pro každou  $n$ -tici  $x_1, \dots, x_n$  reálných čísel platí

$$P(X_1 < x_1, \dots, X_n < x_n) = \prod_{k=1}^n P(X_k < x_k).$$



### 2.1.7. Podmíněná hustota

Na začátku kapitoly jsme definovali pojem podmíněné pravděpodobnosti (2.1) a uvedli jsme Bayesovu větu (2.3) – oboje pro náhodné jevy. V následujících řádcích rozšíříme tyto vztahy pro náhodné vektory.

**Definice 2.20.** Uvažujme opět náhodný vektor  $\mathbf{X} = (X_1, \dots, X_n)^\top$ , jehož složky rozdělíme do dvou nových náhodných vektorů  $\mathbf{Y} = (X_1, \dots, X_r)^\top$ ,  $\mathbf{Z} = (X_{r+1}, \dots, X_n)^\top$ , kde  $1 \leq r < n$ . Podmíněnou hustotou náhodného vektoru  $\mathbf{Z}$  při daném  $\mathbf{Y} = \mathbf{y}$  nazveme takovou nezápornou měřitelnou funkci  $r(\mathbf{z}|\mathbf{y})$ , která pro množiny  $B \in \mathcal{B}_r$  a  $C \in \mathcal{B}_{n-r}$ , viz (2.6), splňuje vztah

$$P(\mathbf{Y} \in B, \mathbf{Z} \in C) = \int_B \left( \int_C r(\mathbf{z}|\mathbf{y}) d\nu(\mathbf{z}) \right) q(\mathbf{y}) d\lambda(\mathbf{y}),$$

kde  $q(\mathbf{y})$  je marginální hustota vektoru  $\mathbf{Y}$ . Funkce  $\nu$  a  $\lambda$  jsou tzv. čítací míry na příslušných  $\sigma$ -algebách [1].

**Věta 2.21.** [1] Označme  $f(\mathbf{y}, \mathbf{z})$  sdruženou hustotu vektorů  $\mathbf{Y}$  a  $\mathbf{Z}$  a

$$q(\mathbf{y}) = \int_{\mathbb{R}^{n-r}} f(\mathbf{y}, \mathbf{z}) d\nu(\mathbf{z}) \quad (2.9)$$

marginální hustota vektoru  $\mathbf{Y}$  podle (2.8). Pak podmíněná hustota vektoru  $\mathbf{Z}$  při daném  $\mathbf{Y} = \mathbf{y}$  je rovna

$$r(\mathbf{z}|\mathbf{y}) = \begin{cases} \frac{f(\mathbf{y}, \mathbf{z})}{q(\mathbf{y})}, & \text{pro } q(\mathbf{y}) \neq 0, \\ 0, & \text{pro } q(\mathbf{y}) = 0. \end{cases} \quad (2.10)$$

Vztah (2.10) lze přepsat do tvaru

$$f(\mathbf{y}, \mathbf{z}) = r(\mathbf{z}|\mathbf{y})q(\mathbf{y}), \quad (2.11)$$

který platí skoro všude vzhledem k příslušným mírám.

**Věta 2.22.** (Bayesova) [1] Buď  $q(\mathbf{y})$  marginální hustota náhodného vektoru  $\mathbf{Y}$  a  $r(\mathbf{z}|\mathbf{y})$  podmíněná hustota vektoru  $\mathbf{Z}$  při daném  $\mathbf{Y}$ . Pak podmíněná hustota  $s(\mathbf{y}|\mathbf{z})$  vektoru  $\mathbf{Y}$  při daném  $\mathbf{Z}$  je rovna

$$s(\mathbf{y}|\mathbf{z}) = \begin{cases} \frac{q(\mathbf{y})r(\mathbf{z}|\mathbf{y})}{\int_{\mathbb{R}^r} q(\mathbf{y})r(\mathbf{z}|\mathbf{y})d\lambda(\mathbf{y})}, & \text{pro } \int q(\mathbf{y})r(\mathbf{z}|\mathbf{y})d\lambda(\mathbf{y}) \neq 0, \\ 0, & \text{jinak.} \end{cases} \quad (2.12)$$

*Důkaz.* Ze vztahu (2.11) je patrné, že  $r(\mathbf{z}|\mathbf{y})q(\mathbf{y})$  je hustota vektoru  $(\mathbf{Y}^\top, \mathbf{Z}^\top)^\top$ . Pak podle věty 2.18 je marginální hustota  $q(\mathbf{z})$  vektoru  $\mathbf{Z}$  rovna

$$q(\mathbf{z}) = \int_{\mathbb{R}^r} r(\mathbf{z}|\mathbf{y})q(\mathbf{y})d\lambda(\mathbf{y}). \quad (2.13)$$

Vztah (2.12) poté dostáváme z věty 2.21 o podmíněné hustotě. □

## 2.1. PRAVDĚPODOBNOST

Všimněme si, že Bayesovu větu můžeme zapsat také ve tvaru

$$s(\mathbf{y}|\mathbf{z}) = \frac{r(\mathbf{z}|\mathbf{y})q(\mathbf{y})}{q(\mathbf{z})}. \quad (2.14)$$

Obdobně jako u Bayesovy věty pro náhodné jevy (2.3) označujeme hustotu  $q(\mathbf{y})$  jako *apriorní hustotu*, protože vyjadřuje naše předběžné znalosti o vektoru  $\mathbf{Y}$ . Podmíněná hustota  $r(\mathbf{z}|\mathbf{y})$ , která vyjadřuje chování vektoru  $\mathbf{Z}$  v závislosti na vektoru  $\mathbf{Y}$ , je nám také známá, resp. ji získáme z provedených experimentů. Získané podmíněné hustotě  $s(\mathbf{y}|\mathbf{z})$  se pak říká *aposteriorní hustota*.

### 2.1.8. Bayesovská statistika

Předchozí dvě věty otevírají nové možnosti, jak přistupovat k řešení statistických úloh jako například testování hypotéz nebo hledání bodových či intervalových odhadů. Uvažujme nyní náhodný vektor  $\mathbf{X} = (X_1, \dots, X_n)^\top$  a vektor parametrů  $\Theta = (\Theta_1, \dots, \Theta_n)^\top$ . Podmíněnou hustotu náhodného vektoru  $\mathbf{X}$  označíme  $f(\mathbf{x}|\theta)$ .

Při klasickém statistickém přístupu k problému odhadu parametrů  $\Theta$  považujeme tento vektor za vektor neznámých konstant. K závěrům o hodnotě parametrů pak použijeme pouze realizaci náhodného vektoru  $\mathbf{X}$  a tvar jeho rozdělení. Příkladem takového postupu může být metoda maximální věrohodnosti [29].

Naproti tomu při bayesovském přístupu použijeme kromě  $\mathbf{X}$  ještě informace o vektoru parametrů  $\Theta$ , který považujeme také za náhodný vektor. Přeznačme v Bayesově větě (2.14) obě podmíněné i obě marginální hustoty na  $f(\cdot|\cdot)$ , resp.  $f(\cdot)$ . Dále tedy budeme rozlišovat jednotlivé hustoty pravděpodobnosti jen podle jejich argumentů. Pro vektory  $\mathbf{X}$  a  $\Theta$  dostáváme

$$f(\theta|\mathbf{x}) = \frac{f(\mathbf{x}|\theta)f(\theta)}{f(\mathbf{x})}. \quad (2.15)$$

Zopakujme, že hustotu  $f(\theta)$  označujeme jako apriorní (anglicky *prior density*) a hustotu  $f(\theta|\mathbf{x})$  jako aposteriorní (*posterior density*). Podmíněná hustota  $f(\mathbf{x}|\theta)$  je tzv. *věrohodnostní funkce (likelihood function)*. Pokud se chceme dozvědět pouze něco o parametrech  $\Theta$ , můžeme v rovnici (2.15) ignorovat jmenovatel  $f(\mathbf{x})$ , protože ten není závislý na  $\Theta$ . Potom píšeme

$$f(\theta|\mathbf{x}) \propto f(\mathbf{x}|\theta)f(\theta), \quad (2.16)$$

kde  $\propto$  značí úměrnost. Aposteriorní hustota je tedy až na nějakou normalizační konstantu rovna součinu věrohodnostní funkce a apriorní hustoty [7]. Věrohodnostní funkce  $f(\mathbf{x}|\theta)$  bude velmi důležitá pro pozdější popis fungování sensorové sítě v kapitole 3.

#### Volba apriorního rozdělení

Apriorní hustota  $f(\theta)$  by měla nějakým způsobem zahrnovat naše apriorní informace (subjektivní i objektivní) o parametrech  $\theta$ . Možnosti se dají klasifikovat do čtyř skupin [14]:

1. Tvar hustoty  $f(\theta)$  vyplývá z apriorní informace.
2. Jako hustotu použijeme histogram.

3. Volíme hladkou hustotu, která dobře aproximuje histogram.
4. Předpokládáme, že hustota má funkcionální tvar a neznáme pouze parametry.

Někdy můžeme za apriorní hustotu použít i tzv. nevlastní hustotu, která je definovaná jako nezáporná měřitelná funkce (nemusí být integrovatelná).

### Vyhodnocení aposteriorní hustoty

Podobně jako u klasických statistických odhadů, rozlišujeme při bayesovském přístupu různé typy odhadů [7]:

- Bodový odhad pomocí maximální věrohodnosti, kdy vybereme tu hodnotu  $\boldsymbol{\theta}$  (případně jednu z více hodnot), kde aposteriorní hustota  $f(\boldsymbol{\theta}|\mathbf{x})$  nabývá nejvyšší hodnoty:

$$\hat{\boldsymbol{\theta}} = \arg_{\boldsymbol{\theta}} \max\{f(\boldsymbol{\theta}|\mathbf{x})\}.$$

- Bodový odhad jako střední hodnota aposteriorního rozdělení:

$$\hat{\boldsymbol{\theta}} = E(\boldsymbol{\Theta}|\mathbf{X}). \quad (2.17)$$

- Bayesovský  $(1 - \alpha)$  konfidenční interval pro testování hypotéz definujeme jako takovou množinu  $I$ , pro kterou platí

$$P(\boldsymbol{\Theta} \in I|\mathbf{X}) = 1 - \alpha.$$

Podrobnější přehled a popis bayesovských metod lze nalézt např. v [14].

## 2.2. Grafy

Senzorová síť, kterou se v práci zabýváme, tvoří souvislý graf. V následujících definicích vysvětlíme, co tento pojem znamená. Vycházíme při tom z [8, 21].

**Definice 2.23.** Nechť je dána libovolná neprázdná množina  $n(G)$ . Označme dále  $e(G) \subseteq \{(u, v) | u, v \in n(G), u \neq v\}$  množinu neuspořádaných dvojic z  $n(G)$ . *Neorientovaným grafem*  $G$  pak označíme uspořádanou dvojici  $G = (n(G), e(G))$ . Prvky množiny  $n(G)$  označíme *uzly (vrcholy)* a prvky množiny  $e(G)$  *hrany*.

**Definice 2.24.** Řekneme, že hrana  $e = (u, v)$  *inciduje* s uzly  $u$  a  $v$ . Uzly  $u$  a  $v$  nazveme *sousední*, pokud jsou v grafu spojeny hranou. *Okolí* (sousedství) uzlu  $u$  je množina všech vrcholů sousedních s tímto uzlem. Značíme  $N(u) = \{v \in n(G), (u, v) \in e(G)\}$

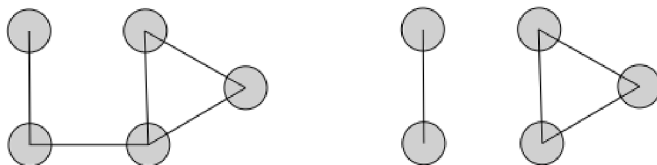
**Definice 2.25.** Předpokládejme, že je dán neorientovaný graf  $G = (n(G), e(G))$ . Jestliže je možno sestrojít konečnou posloupnost uzlů a hran grafu  $G$  tvaru

$$(u_0, e_1, u_1, e_2, \dots, u_{n-1}, e_n, u_n),$$

pak tuto posloupnost nazveme *sled* mezi uzly  $u_0$  a  $u_n$ . Číslo  $n$  je *délka* daného sledu.

**Definice 2.26.** Nechť je dán neorientovaný graf  $G = (n(G), e(G))$ . Jestliže mezi každými dvěma uzly  $x$  a  $y$  existuje alespoň jeden sled, nazveme graf *souvislý*.

## 2.3. APROXIMACE FUNKCÍ



Obrázek 2.1: Příklad souvislého grafu (vlevo) a grafu, který není souvislý (vpravo)

### 2.3. Aproximace funkcí

Aproximovat funkci  $f(x)$  znamená najít k ní funkci  $\varphi(x)$ , která je ve stanovených bodech k  $f(x)$  v jistém smyslu blízká. Píšeme  $\varphi(x) \approx f(x)$ . Existují dva základní typy aproximace [3]:

1. **Interpolace** je aproximace, při níž  $\varphi(x)$  nabývá v zadaných bodech  $x_i$  předepsaných hodnot  $y_i = f(x_i) = \varphi(x_i)$ . Zároveň můžeme požadovat, aby funkce  $\varphi$  a  $f$  měly v bodech  $x_i$  také stejné derivace. Při interpolaci se  $\varphi$  nejčastěji hledá ve tvaru polynomu nebo jako funkce po částech polynomiální. Interpolací se v této práci dále zabývat nebudeme.
2. **Metoda nejmenších čtverců** je taková aproximace, při níž  $\varphi(x)$  „prokládáme“ mezi zadanými body  $(x_i, y_i)$  tak, aby „vzdálenost“ funkcí  $f$  a  $\varphi$  v těchto bodech byla v jistém smyslu minimální. Oproti interpolaci se nepožaduje, aby funkce  $\varphi$  procházela předepsanými body. Tuto metodu popíšeme blíže v následující části.

#### 2.3.1. Metoda nejmenších čtverců

Metoda nejmenších čtverců (zkráceně MNČ, anglicky *least squares*) označuje obecně postup pro přibližné řešení přeuročených soustav rovnic, tj. soustav, které mají větší počet rovnic než hledaných neznámých. Proto je tato metoda vhodná také pro aproximaci funkce.

##### Prokládání dat křivkami

Nechť  $x$  je nezávislá proměnná a  $f(x)$  je neznámá funkce, kterou chceme aproximovat. Předpokládáme, že jsme provedli celkem  $J$  pozorování, tj. máme k dispozici  $J$  dvojic

$$\{(x_j, f(x_j))\}_{j=1}^J \quad (2.18)$$

Naším úkolem je modelovat  $f(x)$  lineární kombinací funkcí  $\psi_1(x), \psi_2(x), \dots, \psi_K(x)$  pro nějaké  $K \leq J$ :

$$f(x) \approx \varphi(x) = \alpha_1 \psi_1(x) + \alpha_2 \psi_2(x) + \dots + \alpha_K \psi_K(x) = \sum_{k=1}^K \alpha_k \psi_k(x). \quad (2.19)$$

Funkce  $\psi_k(x)$  označujeme jako *bázové funkce* nebo *atomy*. Množinu všech atomů  $\{\psi_k(x)\}_{k=1}^K$  nazýváme *slovník*. Volba slovníku je velmi důležitým aspektem pro dobrou aproximaci funkce a je ovlivněna konkrétní aplikací či modelem. Hojně používané slovníky jsou např. polynomy nebo trigonometrické polynomy. V části 2.3.3 uvedeme, jaké slovníky budeme využívat v našem modelu.

## 2. MATEMATICKÝ APARÁT

Neznámé parametry  $\{\alpha_k(x)\}_{k=1}^K$  (v dalších kapitolách označované jako *bázové koeficienty*) získáme tak, že do (2.19) dosadíme pozorování (2.18). Dostaneme tak soustavu  $J$  lineárních rovnic:

$$\begin{aligned}\alpha_1\psi_1(x_1) + \alpha_2\psi_2(x_1) + \cdots + \alpha_K\psi_K(x_1) &= f(x_1), \\ \alpha_1\psi_1(x_2) + \alpha_2\psi_2(x_2) + \cdots + \alpha_K\psi_K(x_2) &= f(x_2), \\ &\vdots \\ \alpha_1\psi_1(x_J) + \alpha_2\psi_2(x_J) + \cdots + \alpha_K\psi_K(x_J) &= f(x_J).\end{aligned}$$

Tento systém můžeme přepsat do maticového tvaru

$$\mathbf{\Psi}\boldsymbol{\alpha} = \mathbf{f}, \quad (2.20)$$

kde  $\mathbf{f} = (f(x_1), f(x_2), \dots, f(x_J))^T$  je vektor pravých stran (typicky naměřená data),  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_K)^T$  je vektor neznámých bázových koeficientů a

$$\mathbf{\Psi}_{J \times K} = \begin{pmatrix} \psi_1(x_1) & \psi_2(x_1) & \dots & \psi_K(x_1) \\ \psi_1(x_2) & \psi_2(x_2) & \dots & \psi_K(x_2) \\ \vdots & \vdots & & \vdots \\ \psi_1(x_J) & \psi_2(x_J) & \dots & \psi_K(x_J) \end{pmatrix}$$

je matice soustavy, která je někdy označována také jako *slovník* a její sloupce jako *atomy* (tyto dva pojmy tedy nejsou zcela jednoznačné). Jak už jsme zmínili v úvodu, soustava (2.20) je přeurčená, a tedy řešení obecně neexistuje. Hledáme nějaké „přibližné řešení“, které je v jistém smyslu nejlepší. K tomu potřebujeme zavést následující pojem.

*Rezidua* jsou rozdíly mezi pozorováním a modelem:

$$r_j = f(x_j) - \varphi(x_j) = f(x_j) - \sum_{k=1}^K \alpha_k \psi_k(x_j), \quad j = 1, 2, \dots, J,$$

což lze opět vyjádřit maticově

$$\mathbf{r} = \mathbf{f} - \mathbf{\Psi}\boldsymbol{\alpha},$$

kde  $\mathbf{r} = (r_1, \dots, r_J)^T$ . Metoda nejmenších čtverců je založena na minimalizaci součtu čtverců reziduí:

$$\|\mathbf{r}\|^2 = \sum_{j=1}^J r_j^2 \rightarrow \min. \quad (2.21)$$

Lze ukázat [3], že vektor  $\boldsymbol{\alpha}$ , jenž minimalizuje (2.21), musí splňovat tzv. *normální soustavu rovnic*:

$$\mathbf{\Psi}^T \mathbf{\Psi}\boldsymbol{\alpha} = \mathbf{\Psi}^T \mathbf{f}. \quad (2.22)$$

Když jsou sloupce matice  $\mathbf{\Psi}$  lineárně nezávislé, řešení normální soustavy rovnic je jediné řešení úlohy (2.21).

## 2.3. APROXIMACE FUNKCÍ

### Využití pseudoinverzní matice

Normální soustava rovnic (2.22) se ovšem nehodí pro rozsáhlé soustavy (2.20), tedy pro velký počet pozorování  $J$ . Dalším limitem je výše zmíněná nutnost lineární nezávislosti sloupců matice  $\Psi$ . Pro tyto případy je vhodné použít tzv. *pseudoinverzi*.

**Definice 2.27.** *Pseudoinverzní matice  $\mathbf{A}^+$  k matici  $\mathbf{A}$  je taková, která splňuje:*

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

kde symbol  $*$  značí tzv. hermitovskou transpozici (složení transpozice matice a komplexního sdružení každého jejího prvku).

Lze ukázat, že platí  $\mathbf{A}^+ = \mathbf{A}^*(\mathbf{A}\mathbf{A}^*)^+$ , což v případě plné řádkové hodnosti matice  $\mathbf{A}$  přechází v [22]

$$\mathbf{A}^+ = \mathbf{A}^*(\mathbf{A}\mathbf{A}^*)^{-1}$$

a pro reálnou matici  $\mathbf{A}$

$$\mathbf{A}^+ = \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1}.$$

Pro regulární matici  $\mathbf{A}$  zřejmě platí  $\mathbf{A}^+ = \mathbf{A}^{-1}$ . Pomocí pseudoinverze získáme snadno řešení úlohy (2.21):

$$\boldsymbol{\alpha} = \Psi^+ \mathbf{f}. \quad (2.23)$$

### 2.3.2. Aproximace funkcí dvou proměnných

Výše popsany postup metody nejmenších čtverců lze zobecnit pro aproximaci funkce více proměnných. My se pro jednoduchost omezíme na funkce dvou proměnných  $f(x, y)$ , také vzhledem k tomu, že později budeme aproximovat právě takové funkce. Obdobně jako v předchozím případě (2.18) uvažujeme, že máme k dispozici pozorování

$$\{(x_j, y_j, f_j)\}_{j=1}^J \quad (2.24)$$

kde  $f_j = f(x_j, y_j)$ . Je zřejmé, že slovník, který k aproximaci použijeme, musí být složen z funkcí dvou proměnných. Mějme tedy slovník  $\{\psi_k(x, y)\}_{k=1}^K$  a dosazením pozorování (2.24) do

$$f(x, y) \approx \varphi(x, y) = \sum_{k=1}^K \alpha_k \psi_k(x, y) \quad (2.25)$$

dostáváme soustavu rovnic

$$\begin{aligned} \alpha_1 \psi_1(x_1, y_1) + \alpha_2 \psi_2(x_1, y_1) + \cdots + \alpha_K \psi_K(x_1, y_1) &= f(x_1, y_1), \\ \alpha_1 \psi_1(x_2, y_2) + \alpha_2 \psi_2(x_2, y_2) + \cdots + \alpha_K \psi_K(x_2, y_2) &= f(x_2, y_2), \\ &\vdots \\ \alpha_1 \psi_1(x_J, y_J) + \alpha_2 \psi_2(x_J, y_J) + \cdots + \alpha_K \psi_K(x_J, y_J) &= f(x_J, y_J), \end{aligned}$$

resp. v maticovém tvaru analogickém k (2.20)

$$\Psi \alpha = \mathbf{f}. \quad (2.26)$$

Bázové koeficienty získáme znovu pomocí pseudoinverzní matice

$$\alpha = \Psi^+ \mathbf{f}. \quad (2.27)$$

### 2.3.3. Slovníky pro aproximaci funkce dvou proměnných

Pro aproximaci věrohodnostní funkce v kapitole 4 budeme používat dva typy slovníků – B-splajny a Fourierův slovník.

#### Slovník B-splajnů

Splajny jsou hladké a po částech polynomiální funkce, které jsou typicky používány spíše pro interpolační úlohy. V našem případě je využijeme pro aproximaci funkce.

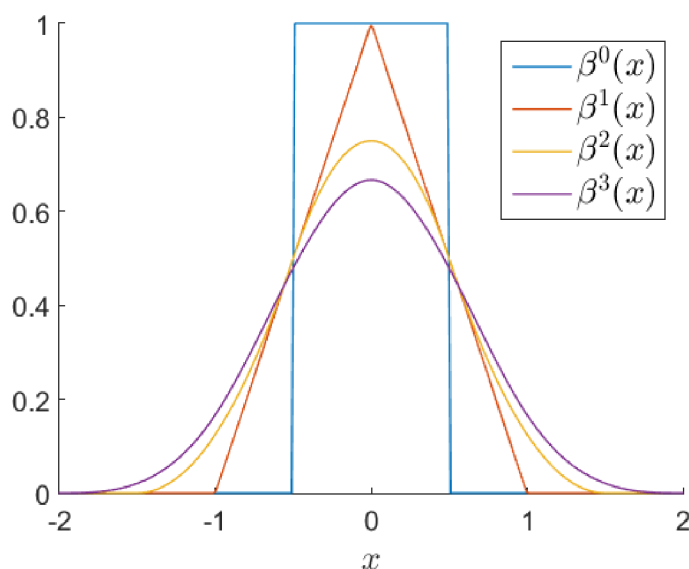
Uvažujme obdélníkový puls  $\beta^0(x)$  [27]:

$$\beta^0(x) = \begin{cases} 1, & \text{pro } -\frac{1}{2} \leq |x| < \frac{1}{2}, \\ \frac{1}{2}, & \text{pro } |x| = \frac{1}{2}, \\ 0, & \text{jinak.} \end{cases}$$

Z tohoto signálu, pak složenou konvolucí [19]

$$\beta^n(x) = \underbrace{\beta^0 * \beta^0 * \dots * \beta^0(x)}_{(n+1) \text{ krát}} \quad (2.28)$$

získáme předpis pro tzv. bázové splajny řádu  $n$ , zkráceně B-splajny. Na obrázku 2.2 vidíme B-splajny stupně 0 až 3.



Obrázek 2.2: Jednodimenzionální B-splajny stupně 0 až 3

### 2.3. APROXIMACE FUNKCÍ

Pomocí Fourierovy transformace můžeme odvodit i explicitní vzorec pro B-splajny [27]. Nejprve zavedme tzv. jednostrannou mocninnou funkci

$$(x)_+^n = \begin{cases} x^n, & \text{pro } x \geq 0 \\ 0, & \text{pro } x < 0 \end{cases}$$

pomocí které jsme již schopni vyjádřit  $\beta^n(x)$ :

$$\beta^n(x) = \frac{1}{n!} \sum_{k=0}^{n+1} \binom{n+1}{k} (-1)^k \left(x - k + \frac{n+1}{2}\right)_+^n. \quad (2.29)$$

Jako příklad uveďme explicitní vzorec pro B-splajn stupně 3 neboli kubický B-splajn:

$$\beta^3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{1}{2}|x|^3, & \text{pro } 0 \leq |x| < 1, \\ \frac{1}{6}(2 - |x|^3), & \text{pro } 1 \leq |x| < 2, \\ 0, & \text{jinak.} \end{cases}$$

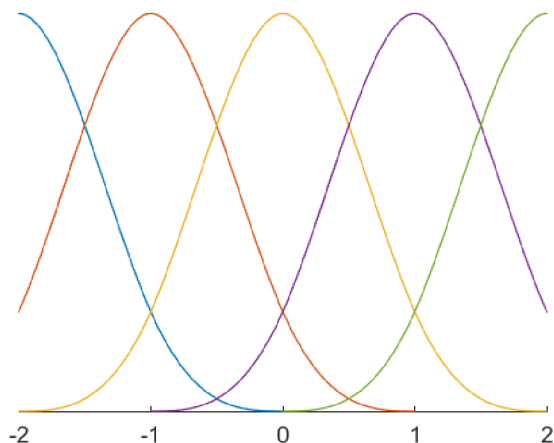
Slovník B-splajnů je pak tvořen jednotlivými splajny rovnoměrně rozloženými přes uvažovaný interval. Mějme tedy interval délky  $d$  symetrický kolem nuly  $\langle -d/2, d/2 \rangle$ , na kterém chceme sestavit slovník  $R$  splajnů  $\{\beta_r^3(x)\}_{r=1}^R$ . Pak lze tento slovník vyjádřit jako

$$\beta_r^3(x) = \begin{cases} \frac{2}{3} - |\bar{x}|^2 + \frac{1}{2}|\bar{x}|^3, & \text{pro } 0 \leq |\bar{x}| < 1, \\ \frac{1}{6}(2 - |\bar{x}|^3), & \text{pro } 1 \leq |\bar{x}| < 2, \\ 0, & \text{jinak,} \end{cases} \quad (2.30)$$

kde  $\bar{x} = (x - S)\delta$ . Posunutí  $S$  a dilataci  $\delta$  získáme ze vztahů

$$S = d \left( \frac{r-1}{R-1} - \frac{1}{2} \right), \quad \delta = \frac{R-1}{d}, \quad (2.31)$$

pro  $r = 1, \dots, R$ . Slovník  $\{\beta_r^3(x)\}_{r=1}^5$  pro interval délky  $d = 4$  a počet splajnů  $R = 5$  je znázorněn na obrázku 2.3.



Obrázek 2.3: Ukázka slovníku B-splajnů stupně 3 na intervalu  $\langle -2, 2 \rangle$



### Vlastnosti B-splajnů

B-splajny mají několik vhodných vlastností [27]:

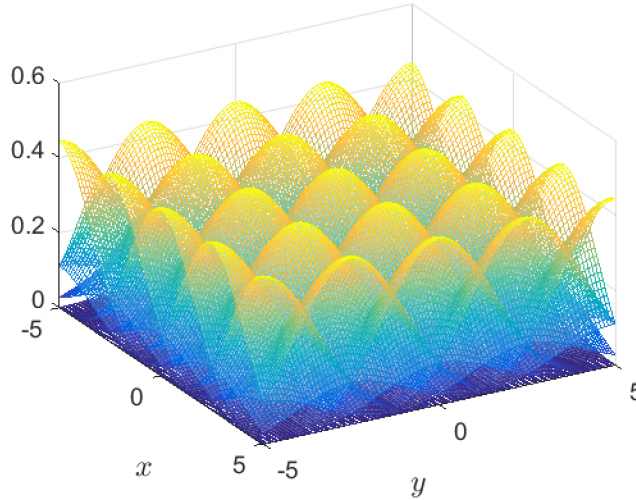
- Jsou spojité a mají spojité derivace až do řádu  $k - 1$ .
- Mají kompaktní nosič.
- $d$ -rozměrné splajny jsou rovny kartézskému součinu  $d$  jednorozměrných splajnů.
- Pro vhodně zvolené  $\Delta x$  je nekonečná suma všech posunutých splajnů  $(\beta^3(x - i\Delta x))$ ,  $i \in \mathbb{Z}$  rovna konstantě.
- Každý splajn může být reprezentován jako lineární kombinace splajnů s (vhodně zvoleným) menším měřítkem.

### Slovník dvojrozměrných B-splajnů

Ze slovníku jednorozměrných B-splajnů popsaného vztahy (2.30) a (2.31) zkonstruuujeme jednoduše slovník pro aproximaci funkcí dvou proměnných  $\{\psi_k(x)\}_{k=1}^K$  pomocí kartézského součinu (viz vlastnosti B-splajnů). Pro  $k$ -tý atom slovníku  $\psi_k(x, y)$  položíme

$$\psi_k(x, y) = \beta_{r_x}^3(x) \beta_{r_y}^3(y). \quad (2.32)$$

Dvojici indexů  $(r_x, r_y) \in \{1, \dots, R\}^2$  jsme přeindexovali na  $k = 1, \dots, K = R^2$ . Na obrázku 2.4 můžeme vidět ukázkou slovníku pro  $d = 10$  a  $R = 5$ , resp.  $K = 25$ .



Obrázek 2.4: Ukázkou slovníku B-splajnů stupně 3 na ploše  $\langle -5, 5 \rangle \times \langle -5, 5 \rangle$

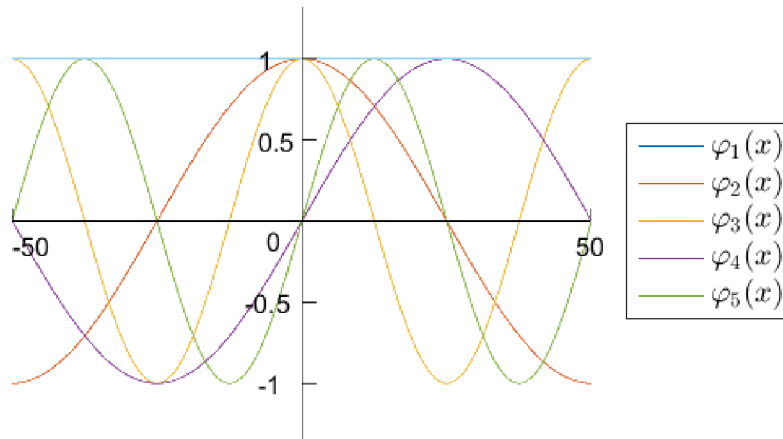
### Fourierův slovník

Druhý slovník, který budeme používat je tzv. *Fourierův*. Uvedme nejprve jeho tvar pro jednorozměrný případ  $\{\varphi_r(x)\}_{r=1}^{2R+1}$  na intervalu délky  $d$  [11]:

$$\varphi_r(x) = \begin{cases} 1, & \text{pro } r = 1, \\ \cos\left(\frac{2\pi}{d}(r-1)x\right), & \text{pro } r = 2, \dots, R+1, \\ \sin\left(\frac{2\pi}{d}(r-1-R)x\right), & \text{pro } r = R+2, \dots, 2R+1. \end{cases} \quad (2.33)$$

### 2.3. APROXIMACE FUNKCÍ

Vidíme, že oproti slovníku B-splajnů nelze libovolně zvolit velikost Fourierova slovníku. Důvodem je, že tento slovník vždy obsahuje konstantní atom a stejný počet atomů s funkcemi  $\sin(\cdot)$  a  $\cos(\cdot)$ . Ač je slovník složen z kosinových a sinových funkcí, lze ho vyjádřit pouze pomocí  $\sin(\cdot)$  a jeho posunutí (nebo  $\cos(\cdot)$  a jeho posunutí), což je patrné také z obrázku 2.5, kde uvádíme příklad Fourierova slovníku pro  $d = 100$  a  $R = 2$ .

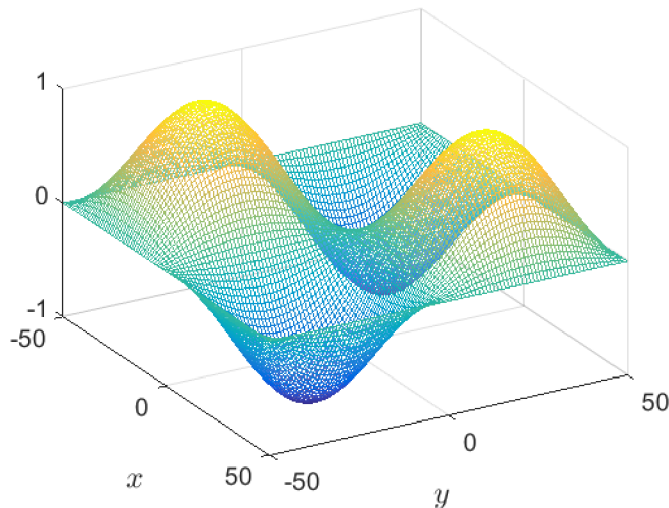


Obrázek 2.5: Ukázka 1-D Fourierovy báze pro hodnoty  $R = 2$  a  $d = 100$

Dvojměrný Fourierův slovník vytvoříme stejným způsobem jako u B-splajnů:

$$\psi_k(x, y) = \varphi_{r_x}(x)\varphi_{r_y}(y). \quad (2.34)$$

Dvojici indexů  $(r_x, r_y) \in \{1, \dots, 2R + 1\}^2$  opět pro přehlednost přeindexujeme na  $k = 1, \dots, K = (2R + 1)^2$  a máme tak slovník  $\{\psi_k(x, y)\}_{k=1}^K$ . Na obrázku 2.6 vidíme jeden atom ze slovníku  $\{\psi_k(x, y)\}_{k=1}^9$  na ploše  $100 \times 100$ .



Obrázek 2.6: Ukázka atomu  $\psi_9(x, y)$  z Fourierova slovníku pro  $d = 100$ ,  $K = 9$

### 2.3.4. Řídké řešení soustav rovnic

Vraťme se nyní k soustavě lineárních rovnic z části 2.3.1:

$$\Psi \boldsymbol{\alpha} = \mathbf{f}. \quad (2.35)$$

Ukázali jsme, jak najít nejlepší přibližné řešení  $\boldsymbol{\alpha}$  pomocí metody nejmenších čtverců. Dále přidáme další požadavek pro hledané řešení. Nejenom že budeme chtít malý (nikoliv nejmenší) součet reziduí (2.21), ale také se pokusíme hledat tzv. *řídké* řešení. Tento pojem vysvětlíme v následujících definicích.

#### Normy, řídkost vektoru

**Definice 2.28.** Pro vektorový prostor  $V$  nad tělesem  $\mathbb{F}$  můžeme definovat reálný funkcionál  $\|\cdot\| : X \rightarrow \mathbb{R}^+$  splňující pro všechna  $\mathbf{x}, \mathbf{y} \in V$  a všechna  $c \in \mathbb{F}$  podmínky

1.  $\|c\mathbf{x}\| = |c|\|\mathbf{x}\|$  (absolutně homogenní),
2.  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$  (splňuje trojúhelníkovou nerovnost),
3.  $\|\mathbf{x}\| \geq 0$  a  $\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$  (rozlišuje body).

Toto zobrazení nazveme *norma* a prostor  $V$  *normovaný vektorový prostor*.

**Definice 2.29.** Nechť  $p \in \mathbb{R}, p \geq 1$ , pak  $l_p$ -norma vektoru  $\mathbf{x} \in \mathbb{C}^N$  je dána

$$\|\mathbf{x}\|_p = \left( \sum_{i=1}^N |x_i|^p \right)^{\frac{1}{p}}$$

a pro  $p = \infty$

$$\|\mathbf{x}\|_\infty = \max_i |x_i|.$$

Je zřejmé, že  $l_2$  norma je klasická Euklidovská norma, kterou budeme pro jednoduchost v některých případech zapisovat bez dolního indexu  $\|\cdot\|$ .

**Definice 2.30.** *Nosičem vektoru*  $\mathbf{x} \in \mathbb{C}^N$  rozumíme množinu jeho indexů, v nichž má vektor nenulové hodnoty. Značíme  $\text{supp}(\mathbf{x})$ :

$$\text{supp}(\mathbf{x}) = \{i | x_i \neq 0\}.$$

**Definice 2.31.** *Řídkostí vektoru*  $\mathbf{x} \in \mathbb{C}^N$  rozumíme nezáporné celé číslo

$$\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|, \quad (2.36)$$

kde  $|\cdot|$  značí mohutnost množiny. Vektor  $\mathbf{x}$  nazveme *k-řídký*, jestliže platí

$$\|\mathbf{x}\|_0 \leq k. \quad (2.37)$$

Relativní řídkostí vektoru  $\mathbf{x}$  délky  $N$  pak budeme rozumět poměr  $\frac{k}{N}$ .

Vidíme, že řídkost označujeme stejně jako  $l_p$  normu pro  $p = 0$ . Ve skutečnosti se ovšem o normu nejedná, protože nesplňuje podmínku homogenosti.

## 2.3. APROXIMACE FUNKCÍ

### Řídké řešení nedourčené soustavy rovnic

Uvažujme nyní soustavu lineárních rovnic

$$\mathbf{Ax} = \mathbf{y} \quad (2.38)$$

takovou, že pro matici soustavy  $\mathbf{A}_{J \times K}$  platí  $J < K$  a zároveň je matice  $\mathbf{A}$  plně (řádkové) hodnosti. V tomto případě označíme soustavu (2.38) jako *nedourčenou* a platí pro ní, že má nekonečně mnoho přípustných řešení [15]. Proto mezi všemi řešeními můžeme hledat to nejřidší<sup>1</sup>, tzn. s nejmenší hodnotou  $\|\mathbf{x}\|_0$ . Řešíme tedy úlohu

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_0, \quad \text{vzhledem k } \mathbf{Ax} = \mathbf{y}. \quad (2.39)$$

Metody pro řešení této úlohy se dají (hrubě) rozdělit do dvou základních typů [12].

První skupina metod vychází z tzv.  $l_1$  relaxace, a proto se nazývají *relaxační*. Místo úlohy (2.39) řešíme úlohu

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1, \quad \text{vzhledem k } \mathbf{Ax} = \mathbf{y}, \quad (2.40)$$

kteřou lze řešit algoritmy konvexní optimalizace. Podmínky ekvivalentnosti úloh (2.39) a (2.40) lze najít v [12], příklady relaxačních metod lze najít v [6].

Druhou skupinu tvoří tzv. *hladové (greedy) algoritmy*, jejichž hlavní princip spočívá v tom, že v každé iteraci najdou jeden (nebo více) „nejvýznamnějších“ atomů. Výhodou těchto algoritmů je obvykle nízká složitost, nevýhodou pak, že není zaručeno dosažení globálního optima [12]. Nejvíce používaný algoritmus tohoto typu je *Orthogonal Matching Pursuit* (OMP), který budeme využívat v našem modelu a který si dále podrobně popíšeme. Vycházíme při tom z [6].

### Algoritmus OMP

- **Vstupy:** slovník  $\mathbf{A}$ , vektor  $\mathbf{y}$ , maximální počet iterací  $K_{\max}$ , maximální chyba aproximace  $\varepsilon_0$ .
- **Inicializace:**
  - čítač iterací  $k = 0$ ,
  - řešení  $\mathbf{x}_0 = \mathbf{0}$ ,
  - residuum  $\mathbf{r}_0 = \mathbf{y}$ ,
  - nosič řešení  $S_0 = \emptyset$ .
- **Hlavní iterace:**
  1. Zvedni čítač  $k$  o 1.
  2. Vypočítej  $\boldsymbol{\varepsilon}_k = \mathbf{A}^\top \mathbf{r}_{k-1}$ .
  3. Aktualizuj nosič řešení  $S_k = S_{k-1} \cup i$ , kde  $i = \arg \max_i \boldsymbol{\varepsilon}_k(i)$ .
  4. Aktualizuj řešení  $\mathbf{x}_k = \arg \min_{\mathbf{u}} \|\mathbf{y} - \mathbf{A}_{S_k} \mathbf{u}\|$ .
  5. Aktualizuj residuum  $\mathbf{r}_k = \mathbf{y} - \mathbf{A}_{S_k} \mathbf{x}_k$ .
  6. Pokud  $\|\mathbf{r}^k\|_2 < \varepsilon_0$  nebo  $k = K_{\max}$  ukonči cyklus, jinak pokračuj další iterací.
- **Výstup:** Řídké řešení  $\mathbf{x}_k$ , nosič řešení  $S_k$ .

---

<sup>1</sup>V případě, kdy mluvíme o „větší“ řídkosti vektoru  $\mathbf{x}$ , budeme tím myslet větší hodnotu  $\|\mathbf{x}\|_0$  (podobně pro „menší“ řídkost menší hodnotu  $\|\mathbf{x}\|_0$ ). Ovšem pokud zmiňujeme „řidší“ nebo „nejřidší“ vektor, myslíme tím takový, který má co nejmenší hodnotu  $\|\mathbf{x}\|_0$ .

Ve třetím kroku hlavního cyklu výrazem  $\varepsilon_k(i)$  myslíme  $i$ -tou složku vektoru  $\varepsilon_k$ . Ve čtvrtém kroku výraz  $\mathbf{A}_{S_k}$  označuje matici, která vznikne z matice  $\mathbf{A}$  tak, že vybereme ty sloupce, jejichž indexy jsou v aktuální nosiči  $S_k$ . V tomto kroku vlastně řešíme minimalizaci metodou nejmenších čtverců, takže hledané aktuální řešení dostaneme pomocí pseudoinverze  $\mathbf{x}_k = \mathbf{A}_{S_k}^+ \mathbf{y}$ .

Algoritmus končí, jestliže je norma rezidua menší než požadovaná maximální chyba aproximace  $\varepsilon_0$  nebo jestliže čítač  $k$  dosáhl maximálního počtu iterací  $K_{\max}$ . Můžeme ovšem zvolit pouze jednu z těchto podmínek. Výstupem algoritmu je řešení  $\mathbf{x}_k$ , které je  $k$ -řídké.

Velká výhoda tohoto algoritmu tkví v tom, že můžeme předepsat, jak řídké nebo jak přesné řešení chceme získat.

### Řídké řešení přeuročené soustavy rovnic

Předchozí algoritmus můžeme použít i pro soustavu lineárních rovnic, která je přeuročená. Připomeňme tvar naší soustavy rovnic

$$\Psi_{J \times K} \boldsymbol{\alpha}_{K \times 1} = \mathbf{f}_{J \times 1}, \quad (2.41)$$

ve které platí  $K \leq J$ , v našem případě bude platit dokonce  $K \ll J$ . Volba parametrů  $K_{\max}$  a  $\varepsilon_0$  pak ovlivňuje, jaké řešení  $\boldsymbol{\alpha}$  dostaneme. Čím větší řídkost připustíme, tzn. čím bude hodnota  $\|\boldsymbol{\alpha}\|_0$  větší, tím bude naše aproximace přesnější. Největší možná řídkost odpovídá zřejmě dimenzi vektoru  $\boldsymbol{\alpha}$ , platí tedy  $K_{\max} \leq K$ . V případě rovnosti  $K_{\max} = K$  pak algoritmus OMP dává stejný výsledek jako metoda nejmenších čtverců, neboť v poslední iteraci obsahuje nosič  $S_k$  všechny indexy.

Druhým „extrémem“ by byl případ, kdy nastavíme  $K_{\max} = 1$ . Získali bychom tak nejřidší možné řešení (s výjimkou nulového), ovšem naše aproximace funkce by byla nejspíše velmi nepřesná.

Při řešení úlohy aproximace funkce dvou proměnných (2.25) se tedy budeme snažit balancovat mezi těmito dvěma extrémy. Pro pozdější úsporu komunikační náročnosti budeme chtít najít řídké řešení, ovšem se zachováním určité přesnosti aproximace.

## 2.4. Komprimované snímání

Komprimované snímání (též komprimované vzorkování) je jedna z důležitých aplikací v oblasti řídké reprezentace signálů. Hlavní myšlenka spočívá v neadaptivním snímání signálu jen takovým počtem „vzorků“, který je skutečně potřeba [13].

V předchozí části jsme se zabývali hledáním řídkého řešení soustavy lineárních rovnic (v případě nedourčených systémů), resp. hledáním nějaké řídké aproximace (v případě přeuročených systémů), tedy úlohou

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_0, \quad \text{vzhledem k } \mathbf{A}\mathbf{x} = \mathbf{y}. \quad (2.42)$$

V úloze komprimovaného snímání budeme řešit stejnou úlohu, ovšem matice soustavy  $\mathbf{A}$  bude mít speciální tvar.

## 2.4. KOMPRIMOVANÉ SNÍMÁNÍ

### 2.4.1. Způsoby komprimování dat

Běžný způsob komprimování dat je *adaptivní* vzhledem k signálu. Tento přístup se zakládá na tom, že pro různé typy signálů (např. fotografie) můžeme nalézt takovou reprezentaci (resp. slovník), ve které je tento signál řídký nebo přibližně řídký.

Postupujeme tak, že nasbíráme všechna data, provedeme vhodnou transformaci pomocí našeho slovníku a vyhodnotíme získané koeficienty. Většinu koeficientů pak nevyužijeme („vyhodíme je“), protože nesou málo informací. Příkladem může být kompresní formát obrázků JPEG. Nasbíraná data odpovídají hodnotám všech pixelů fotografie, následně provedeme dvojrozměrnou diskretní kosinovou transformaci (DCT) a kvantování (diskretizaci) výsledných koeficientů. Nakonec si ponecháme pouze  $k$  (většinou největších) koeficientů, u kterých zakódujeme jejich pozici/index a velikost [13].

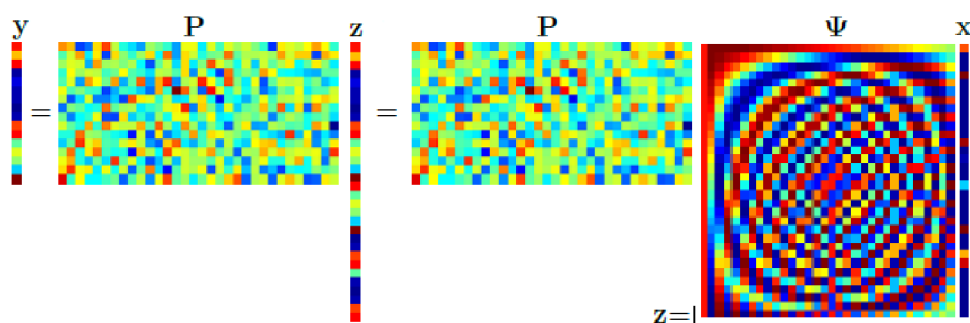
Jinou strategii nabízí *komprimované snímání* (anglicky *compressed sensing*, *compressive sampling*, CS). Pokud je signál řídký (nebo alespoň přibližně řídký) v nějaké vhodné reprezentaci, můžeme ho snímat *lineárně* a *neadaptivně*. Navíc nám stačí pouze určitý potřebný počet vzorků.

### 2.4.2. Princip komprimovaného snímání

Signál  $\mathbf{z}$  tedy předpokládáme řídký ve slovníku  $\Psi$ . Můžeme psát  $\mathbf{z} = \Psi\mathbf{x}$ , kde  $\mathbf{x}$  je  $k$ -řídká reprezentace signálu  $\mathbf{z}$ . Provedeme „malý počet“ neadaptivních měření, které odpovídají skalárním součinům se signálem  $\mathbf{z}$ , což můžeme zapsat jako

$$\mathbf{y} = \mathbf{P}\mathbf{z} = \mathbf{P}\Psi\mathbf{x}. \quad (2.43)$$

Matice  $\mathbf{P}$  je tzv. *kompresní* (též *měřicí*, *snímací*) *matice* rozměru  $m \times N$ . Složky vektoru  $\mathbf{y}$  odpovídají výsledkům měření, které vznikají jako lineární kombinace vzorků signálu. Počet měření je  $m \ll N$ , tedy dimenze vektoru  $\mathbf{y}$  je výrazně menší než dimenze původního signálu  $\mathbf{z}$ . Systém (2.43) je znázorněn na obrázku 2.7.



Obrázek 2.7: [13, 22] Grafické znázornění systému (2.43). Je zde použito pseudobarevné schéma, kdy modrá barva reprezentuje nulu a čím teplejší barva, tím vyšší hodnota. Vidíme, že vektor  $\mathbf{z}$  není sám řídký, ale je řídký ve slovníku  $\Psi$ , resp. je řídká jeho reprezentace  $\mathbf{x}$ .

Zpětná rekonstrukce signálu  $\mathbf{z}$  je nelineární a časově náročná. Nejprve řešíme úlohu (2.42), kde matice  $\mathbf{A}$  má tvar  $\mathbf{A} = \mathbf{P}\Psi$ , tj.

$$\mathbf{x}_{\text{opt}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_0, \quad \text{vzhledem k } \mathbf{P}\Psi\mathbf{x} = \mathbf{y}. \quad (2.44)$$

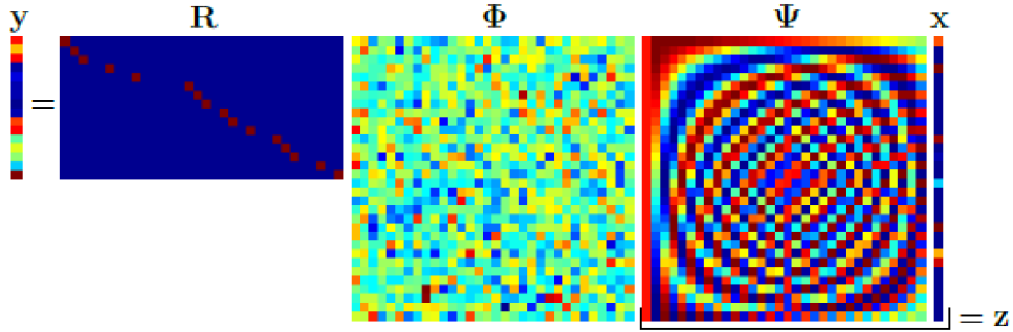
Tuto úlohu můžeme řešit opět algoritmem OMP. Alternativou k nalezení řešení  $\mathbf{x}_{\text{opt}}$  může být algoritmus CoSaMP (Compressive Sampling Matching Pursuit), který je založen na OMP a který je uzpůsobený pro úlohy s kompresním snímáním [18]. V našem modelu budeme používat pouze základní OMP.

Samotný signál pak získáme jednoduše jako  $\mathbf{z} = \Psi \mathbf{x}_{\text{opt}}$ . Dále uvedeme požadavky na kompresní matici  $\mathbf{P}$ , abychom byli schopni nalézt  $\mathbf{x}_{\text{opt}}$  z (2.44).

### 2.4.3. Volba kompresní matice

Kompresní matici  $\mathbf{P}$  je vhodné zvolit ve tvaru  $\mathbf{P} = \mathbf{R}\Phi$ . Zde  $\Phi$  je (zatím nspecifikovaná) čtvercová matice  $N \times N$  a  $\mathbf{R}$  je matice, která vznikne z jednotkové matice  $N \times N$  ponecháním pouze  $m$  náhodně vybraných řádků. Součin  $\mathbf{R}\Phi$  tedy představuje náhodný výběr řádků z matice  $\Phi$ , který se řídí rovnoměrným rozdělením pravděpodobnosti.

Matice  $\mathbf{A}$  je tedy nyní ve tvaru  $\mathbf{A} = \mathbf{R}\Phi\Psi$ . Konkrétní příklad realizace matice  $\mathbf{R}$  a ilustrace celkové situace vidíme na obrázku 2.8.



Obrázek 2.8: [22] Celkové schéma komprimovaného snímání: naměřený vektor  $\mathbf{y}$  je roven součinu kompresní matice  $\mathbf{P} = \mathbf{R}\Phi$ , slovníku  $\Phi$  a řídkého vektoru  $\mathbf{x}$ . Matice  $\mathbf{R}$  vznikla z jednotkové matice  $32 \times 32$  náhodným vybráním  $m = 16$  řádků.

Zbývá vyřešit otázku volby matice  $\Phi$  a počtu řádků  $m$  tak, abychom byli schopni rekonstruovat signál pomocí  $l_1$  technik<sup>2</sup>. Nejdříve si zavedeme pojem vzájemné koherence matice.

**Definice 2.32.** Vzájemná koherence matice  $\mathbf{A}$  je definována jako největší absolutní normalizovaný skalární součin dvou různých sloupců matice, tj.

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

kde  $\mathbf{a}_j$  označuje  $j$ -tý sloupec matice  $\mathbf{A}$  [22].

Ve speciálním případě, kdy je matice  $\mathbf{A}$  složená ze dvou ortonormálních bází<sup>3</sup>  $\Phi$  a  $\Psi$ , tj.  $\mathbf{A} = [\Phi, \Psi]$ , dostaneme vzájemnou koherenci jako

$$\mu([\Phi, \Psi]) = \max_{1 \leq i, j \leq N} |\psi_i^\top \phi_j|.$$

<sup>2</sup>V části 2.3.4 jsme uvedli, že za určitých podmínek můžeme  $l_1$  relaxací řešit úlohu (2.42), resp. (2.44).

<sup>3</sup>Báze  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]$  je ortonormální, jestliže pro skalární součin atomů platí  $\mathbf{b}_i^\top \mathbf{b}_j = 0$ , pro  $i \neq j$ , a norma všech atomů je rovna jedné, tj.  $\|\mathbf{b}_i\| = 1$  [22].

## 2.4. KOMPRIMOVANÉ SNÍMÁNÍ

Hodnota  $\mu([\Phi, \Psi])$  se pohybuje v intervalu  $\langle 1/\sqrt{N}, 1 \rangle$ . Pomocí vzájemné koherence jsme schopni formulovat podmínku, za jaké je zaručena přesná rekonstrukce z  $m$  měření.

**Věta 2.33.** [13] *Nechť je dán signál  $\mathbf{z}$ , který má ve slovníku  $\Psi$   $k$ -řádkou reprezentaci  $\mathbf{x}$ . Pak řešení  $l_1$  minimalizace*

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1, \quad \text{vzhledem k } \mathbf{R}\Phi\Psi\mathbf{x} = \mathbf{y}, \quad (2.45)$$

*kde  $\mathbf{y}$  jsou měření, je současně s vysokou pravděpodobností nejřidší možné, pokud pro počet řádků  $m$  matice  $\mathbf{P}$  platí*

$$m \geq C \cdot \mu^2([\Phi, \Psi]) \cdot k \cdot N \cdot \ln N, \quad (2.46)$$

*pro nějakou kladnou konstantu  $C$ .*

Z uvedené nerovnice (2.46) je patrné, že počet měření  $m$ , potřebný pro úspěšnou rekonstrukci, závisí na řidkosti  $k$  pouze lineárně. Naopak u koherence  $\mu^2([\Phi, \Psi])$  jde o závislost kvadratickou. Proto se v praxi snažíme hledat takové dvojice bází, které mají co nejmenší koherenci. Pokud např. nalezneme ideální dvojici s koherencí  $\frac{1}{\sqrt{N}}$ , počet měření musí být řádově pouze  $k \cdot \ln N$ . S rostoucí koherencí přestává být měření výhodné, až v určitém momentu přestává mít zcela smysl, neboť  $m$  přeroste dimenzi signálu  $N$  [13].



## 3. Distribuovaný částicový filtr v senzorových sítích

V této kapitole přiblížíme problematiku distribuovaných částicových filtrů v senzorových sítích. Sensorovou síť si představme jako skupinu přístrojů, které snímají určitý (většinou fyzikální) jev distribuovaným způsobem, tzn. jednotlivé senzory spolu navzájem spolupracují. V mnoha případech, kdy zkoumaný jev obsahuje nelineární či negaussovské prvky, je vhodné použít částicové filtry. Tato kapitola čerpá především z [10].

### 3.1. Sensorové sítě

Senzory mohou být různého typu – od malých, levných, bateriových jednotek s omezenými schopnostmi výpočtu a komunikace až po multifunkční mobilní roboty schopné řešit komplexní úkoly. Každý senzor obsahuje některé z těchto prvků: čidla, komunikační rozhraní, výpočetní jednotku a pohonnou jednotku. Čidla snímají fyzikální vlastnosti jako např. teplotu, tlak, vlhkost, koncentraci, vzdálenost, rychlost, intenzitu světla apod.

My budeme dále uvažovat sensorové sítě, které se snaží distribuovaně odhadnout určité parametry nebo stavy sledovaného jevu na základě lokálních měření jednotlivých senzorů. Vzájemná spolupráce je důležitá, protože lokální měření jsou většinou nedostatečná pro získání dobrého odhadu. Je proto zřejmé, že vzájemná komunikace mezi senzory je klíčovým aspektem pro fungování takových sensorových sítí.

Navíc se budeme zabývat pouze decentralizovanou sensorovou sítí, ve které nefiguruje žádná centrální výpočetní jednotka. Každý senzor v síti musí mít tedy aktuální informace o sledovaném jevu. Decentralizované sensorové sítě mají řadu výhod jako např. odolnost vůči výpadku senzorů nebo přerušení spojení mezi senzory.

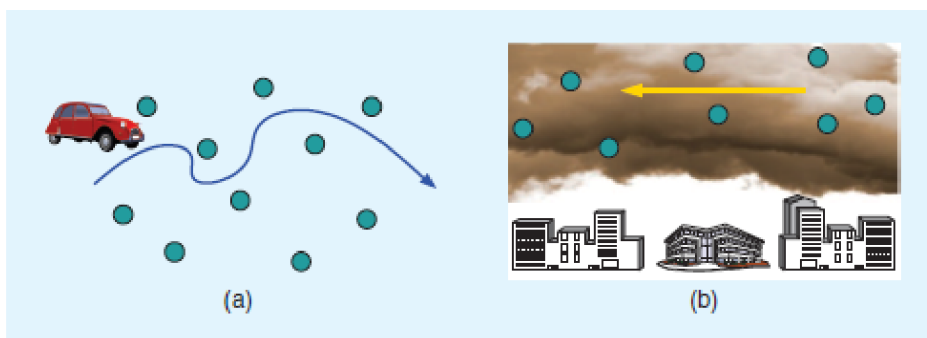
V našem textu budeme vždy předpokládat, že sensorová síť tvoří souvislý graf, viz část 2.2. Jednotlivé senzory tvoří uzly grafu a hrany znázorňují komunikační spojení mezi dvěma senzory. Souvislost sensorové sítě tedy znamená, že všechny senzory jsou propojeny (přímo nebo nepřímo přes řadu jiných senzorů). Dále předpokládáme, že senzory nejsou mobilní, tzn. jejich pozice se v čase nemění. V práci se nebudeme zabývat problémem časové synchronizace použitých algoritmů.

Typickým příkladem sensorových sítí využívajícím distribuovaný výpočet je sledování pohyblivého objektu (anglicky target tracking). Sledovaný objekt (např. vozidlo, letadlo, člověk nebo zvíře) se pohybuje prostředím, ve kterém je rozmístěna sensorová síť. Objekt generuje nějaký signál (aktivně vyzařuje nebo pasivně odráží), který je následně zpracováván senzory. Ty pak odhadují např. pozici nebo rychlost objektu. Tomuto problému se budeme věnovat i v naší práci. Druhým příkladem může být sledování chemického oblaku, u kterého senzory zaznamenávají pozici, rychlost, velikost či tvar. Oba příklady jsou znázorněny na obrázku 3.1.

Pro tyto typy úloh je vhodné použít metodu *sekvenčního<sup>1</sup> bayesovského odhadu*, která je založena na Bayesově větě (2.14). Hledaný odhad i měření senzorů budeme chápat jako náhodné veličiny (náhodné vektory), které budou dané jejich hustotami pravděpodobnosti.

<sup>1</sup>Některé zdroje uvádí „rekurzivní“ bayesovský odhad. Dále budeme tyto dva termíny pokládat významově za shodné.

### 3.2. SEKVENČNÍ BAYESOVSKÝ ODHAD



Obrázek 3.1: Dva příklady aplikace senzorových sítí. (a) Sledování pohyblivého objektu. (b) Sledování chemického oblaku. Sensory jsou znázorněny kruhy [10].

### 3.2. Sekvenční bayesovský odhad

Uvažujme stavový vektor  $\mathbf{x}_n$  dimenze  $d$ , tedy  $\mathbf{x}_n \in \mathbb{R}^d$ , kde  $n$  značí diskretní časový okamžik. Tento vektor se řídí *rovnici stavového prostoru*

$$\mathbf{x}_n = \mathbf{g}_n(\mathbf{x}_{n-1}, \mathbf{u}_n), \quad n = 1, 2, \dots, \quad (3.1)$$

kde  $\mathbf{g}_n(\cdot, \cdot)$  je známá, obecně nelineární funkce. Dále  $\mathbf{u}_n$  značí (budící) bílý šum<sup>2</sup>, který nezávisí na minulých ani současných stavech a jehož hustota pravděpodobnosti je známa. Rovnice (3.1) představuje vlastně diskretní náhodný Markovův proces řádu jedna [9]. V čase  $n$  získáme vektor měření  $\mathbf{z}_n$ , který dostaneme z vektoru  $\mathbf{x}_n$  pomocí *rovnice měření*

$$\mathbf{z}_n = \mathbf{h}_n(\mathbf{x}_n, \mathbf{v}_n), \quad n = 1, 2, \dots, \quad (3.2)$$

kde  $\mathbf{h}_n(\cdot, \cdot)$  je známá, obecně nelineární funkce. Dále  $\mathbf{v}_n$  značí (měřicí) bílý šum, který opět nezávisí na minulých ani současných stavech a jehož hustota pravděpodobnosti je známa. Vektor všech měření do času  $n$  označíme  $\mathbf{z}_{1:n} = (\mathbf{z}_1^\top, \dots, \mathbf{z}_n^\top)^\top$ .

Ve většině problémů uvažujeme, že jsou funkce  $\mathbf{g}_n$  a  $\mathbf{h}_n$  nezávislé na čase, případně že se mění v čase velmi pomalu. Můžeme tedy psát  $\mathbf{g}_n = \mathbf{g}$  a  $\mathbf{h}_n = \mathbf{h}$  pro všechny časové kroky  $n$ . Rovnice (3.1) a (3.2) společně s našimi statistickými předpoklady určují pravděpodobnostní formulaci systému podle přechodové hustoty pravděpodobnosti  $f(\mathbf{x}_n|\mathbf{x}_{n-1})$  a měřicího modelu podle věrohodnostní funkce  $f(\mathbf{z}_n|\mathbf{x}_n)$ <sup>3</sup>. Předpokládáme dále, že platí

$$f(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_{1:n-1}) = f(\mathbf{x}_n|\mathbf{x}_{n-1}), \quad (3.3)$$

což znamená, že stavový vektor  $\mathbf{x}_n$  je podmíněně nezávislý na všech předešlých měřeních  $\mathbf{z}_{1:n-1}$ . Podobně platí

$$f(\mathbf{z}_n|\mathbf{x}_n, \mathbf{z}_{1:n-1}) = f(\mathbf{z}_n|\mathbf{x}_n), \quad (3.4)$$

tedy také vektor měření je nezávislý na všech předchozích měřeních  $\mathbf{z}_{1:n-1}$ .

Úkolem je odhadnout stav  $\mathbf{x}_n$  ze všech předešlých měření  $\mathbf{z}_{1:n}$  rekurzivním způsobem. Bayesovský přístup je založen na výpočtu *aposteriorní hustoty pravděpodobnosti*,

<sup>2</sup>Bílý šum (anglicky white noise) je posloupnost nekorelovaných náhodných veličin  $\{Y_t, t \in \mathbb{Z}\}$  s nulovou střední hodnotou a konečným kladným rozptylem  $\sigma^2$  [20]. Zapisujeme  $\{Y_t\} \sim \text{WN}(0, \sigma^2)$ .

<sup>3</sup>V souladu s literaturou budeme značit všechny hustoty pravděpodobnosti písmenem  $f(\cdot)$  a různé hustoty budeme rozlišovat podle jejich argumentů.

### 3. DISTRIBUOVANÝ ČÁSTICOVÝ FILTR V SENZOROVÝCH SÍTÍCH

viz část 2.1.8. Pomocí aposteriorní pravděpodobnosti  $f(\mathbf{x}_n|\mathbf{z}_{1:n})$  již můžeme vypočítat odhad stavu  $\mathbf{x}_n$ . Například odhad pomocí minimalizace střední kvadratické chyby (MMSE) získáme jako střední hodnotu (2.17) pravděpodobnosti  $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ :

$$\begin{aligned}\hat{\mathbf{x}}_n^{\text{MMSE}} &= \mathbb{E}(f(\mathbf{x}_n|\mathbf{z}_{1:n})) \\ &= \int \mathbf{x}_n f(\mathbf{x}_n|\mathbf{z}_{1:n}) d\mathbf{x}_n, \quad n = 1, 2, \dots\end{aligned}\quad (3.5)$$

Pro další postup tedy potřebujeme znát hustotu pravděpodobnosti  $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ . Ukážeme, jak tuto pravděpodobnost získáme rekurzivně pomocí aposteriorní hustoty v předchozím kroku  $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$  a aktuálního vektoru měření  $\mathbf{z}_n$ .

Podle Bayesovy věty (2.14) můžeme hustotu pravděpodobnosti  $f(\mathbf{x}_n|\mathbf{z}_{1:n})$  rozepsat jako

$$f(\mathbf{x}_n|\mathbf{z}_{1:n}) = \frac{f(\mathbf{z}_{1:n}|\mathbf{x}_n) f(\mathbf{x}_n)}{f(\mathbf{z}_{1:n})}. \quad (3.6)$$

Aposteriorní hustota  $f(\mathbf{x}_n|\mathbf{z}_{1:n})$  je podmíněná hustota stavového vektoru  $\mathbf{x}_n$  při daných vektorech měření až po aktuální časový krok. Vektor  $\mathbf{z}_{1:n}$  můžeme uměle rozepsat a přeskádat na  $(\mathbf{z}_n^\top, \mathbf{z}_{1:n-1}^\top)^\top$  a rovnici (3.6) pak přepíšeme do tvaru

$$f(\mathbf{x}_n|\mathbf{z}_{1:n}) = \frac{f(\mathbf{z}_n, \mathbf{z}_{1:n-1}|\mathbf{x}_n) f(\mathbf{x}_n)}{f(\mathbf{z}_n, \mathbf{z}_{1:n-1})}. \quad (3.7)$$

Použitím vztahu pro podmíněnou hustotu (2.11) na členy  $f(\mathbf{z}_n, \mathbf{z}_{1:n-1}|\mathbf{x}_n)$  a  $f(\mathbf{z}_n, \mathbf{z}_{1:n-1})$  dostáváme

$$f(\mathbf{x}_n|\mathbf{z}_{1:n}) = \frac{f(\mathbf{z}_n|\mathbf{z}_{1:n-1}, \mathbf{x}_n) f(\mathbf{z}_{1:n-1}|\mathbf{x}_n) f(\mathbf{x}_n)}{f(\mathbf{z}_n|\mathbf{z}_{1:n-1}) f(\mathbf{z}_{1:n-1})}. \quad (3.8)$$

Na hustotu  $f(\mathbf{z}_{1:n-1}|\mathbf{x}_n)$  opět aplikujeme Bayesovu větu (2.14)

$$f(\mathbf{z}_{1:n-1}|\mathbf{x}_n) = \frac{f(\mathbf{x}_n|\mathbf{z}_{1:n-1}) f(\mathbf{z}_{1:n-1})}{f(\mathbf{x}_n)}$$

a po dosazení do (3.8) získáme

$$\begin{aligned}f(\mathbf{x}_n|\mathbf{z}_{1:n}) &= \frac{f(\mathbf{z}_n|\mathbf{z}_{1:n-1}, \mathbf{x}_n) f(\mathbf{x}_n|\mathbf{z}_{1:n-1}) f(\mathbf{z}_{1:n-1}) f(\mathbf{x}_n)}{f(\mathbf{z}_n|\mathbf{z}_{1:n-1}) f(\mathbf{z}_{1:n-1}) f(\mathbf{x}_n)} \\ &= \frac{f(\mathbf{z}_n|\mathbf{z}_{1:n-1}, \mathbf{x}_n) f(\mathbf{x}_n|\mathbf{z}_{1:n-1})}{f(\mathbf{z}_n|\mathbf{z}_{1:n-1})}.\end{aligned}\quad (3.9)$$

Využitím předpokladu (3.4) dostáváme konečný vztah pro aposteriorní hustotu pravděpodobnosti

$$f(\mathbf{x}_n|\mathbf{z}_{1:n}) = \frac{f(\mathbf{z}_n|\mathbf{x}_n) f(\mathbf{x}_n|\mathbf{z}_{1:n-1})}{f(\mathbf{z}_n|\mathbf{z}_{1:n-1})}. \quad (3.10)$$

Pro to, abychom měli kompletní rekurzivní schéma, potřebujeme vyjádřit apriorní hustotu  $f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$  pomocí předchozí aposteriorní hustoty. K tomu použijeme Chapman–Kolmogorovovu rovnici [2]

$$f(\mathbf{x}_n|\mathbf{z}_{1:n-1}) = \int f(\mathbf{x}_n|\mathbf{x}_{n-1}) f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1}) d\mathbf{x}_{n-1}, \quad (3.11)$$

### 3.3. ČÁSTICOVÉ FILTRY

která vyplývá ze vztahu (2.13) z důkazu Bayesovy věty a předpokladu (3.3).

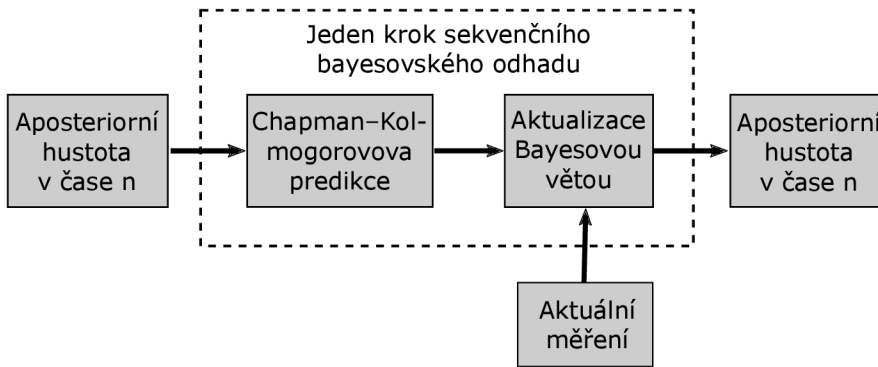
Výpočtu  $f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$  ve (3.11) se říká *predikce*, po které pak následuje *aktualizace* pomocí (3.10). Hustotu  $f(\mathbf{z}_n|\mathbf{z}_{1:n-1})$  ve (3.10), označujeme jako normalizační konstantu, protože není závislá na hledaném stavu  $\mathbf{x}_n$ . Můžeme ji vyjádřit jako

$$f(\mathbf{z}_n|\mathbf{z}_{1:n-1}) = \int f(\mathbf{z}_n|\mathbf{x}_n) f(\mathbf{x}_n|\mathbf{z}_{1:n-1}) d\mathbf{x}_n.$$

Je jasné, že pro výpočet je nutné znát apriorní hustotu pravděpodobnosti v čase  $n = 0$ :

$$f(\mathbf{x}_n|\mathbf{z}_{1:n}) = f(\mathbf{x}_0). \quad (3.12)$$

Pokud nemáme k dispozici přesnou apriorní hustotu  $f(\mathbf{x}_0)$ , můžeme použít například rovnoměrné rozdělení. Schéma jedné iterace sekvenčního bayesovského odhadu je znázorněno na obrázku 3.2.



Obrázek 3.2: Schéma jednoho kroku sekvenčního bayesovského odhadu [9]

Přímý výpočet (3.11) a (3.10) je ovšem obvykle nemožný, protože analytické řešení není ve většině případů k dispozici a klasický numerický postup by vyžadoval výpočet vícerozměrných integrálů. Důležitou výjimkou je případ, kdy jsou rovnice stavového prostoru (3.1) a rovnice měření (3.2) lineární, bílé šумы  $\mathbf{u}_n$  a  $\mathbf{v}_n$  mají Gaussovo rozdělení a také apriorní hustota  $f(\mathbf{x}_0)$  má Gaussovo rozdělení. Lze dokázat, že  $f(\mathbf{x}_n|\mathbf{z}_{1:n})$  má potom také Gaussovo rozdělení. Pro tento případ můžeme použít *Kalmanův filtr* [2].

Pro nelineární a negaussovské problémy existují rozšířený Kalmanův filtr, Gaussův sumační filtr [10], nebo metody založené na diskretizaci stavového prostoru (*grid-based methods*) [2]. V našem modelu použijeme tzv. *částicové filtry* (*particle filters*), které jsou často výhodnější než předchozí zmíněné metody.

### 3.3. Částicové filtry

Částicové filtry jsou jedním z hojně používaných metod k řešení sekvenčního bayesovského odhadu skrze (3.10), (3.11) a (3.5). Jde přitom vlastně o známou Monte Carlo metodu [9]. Hlavní myšlenka těchto metod spočívá v reprezentování námi hledané aposteriorní hustoty pravděpodobnosti  $f(\mathbf{x}_n|\mathbf{z}_{1:n})$  množinou náhodných vzorků s přidruženými vahami  $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$ . Váhy jsou normalizované tak, aby platilo  $\sum_{j=1}^J w_n^{(j)} = 1$ . Připomeňme, že hledaný stavový vektor  $\mathbf{x}_n$  má dimenzi  $d$ , a tedy i pro jednotlivé vzorky platí  $\mathbf{x}_n^{(j)} \in \mathbb{R}^d$ .

### 3. DISTRIBUOVANÝ ČÁSTICOVÝ FILTR V SENZOROVÝCH SÍTÍCH

Máme tedy množinu uspořádaných dvojic vzorků (částic) a jejich vah, pomocí kterých aproximujeme aposteriorní hustotu pravděpodobnosti jako

$$f(\mathbf{x}_n | \mathbf{z}_{1:n}) \approx \sum_{j=1}^J w_n^{(j)} \delta(\mathbf{x}_n - \mathbf{x}_n^{(j)}), \quad (3.13)$$

kde  $\delta(\cdot)$  značí  $d$ -rozměrnou Diracovu delta funkci [10]. Užitím vážených částic můžeme získat různé odhady stavu  $\mathbf{x}_n$ . Odhad minimalizující střední kvadratickou chybu (3.5) pak aproximujeme podle

$$\hat{\mathbf{x}}_n^{\text{MMSE}} \approx \hat{\mathbf{x}}_n = \sum_{j=1}^J w_n^{(j)} \mathbf{x}_n^{(j)}. \quad (3.14)$$

Tato aproximace je dostatečně přesná, pokud je počet částic  $\mathbf{x}_n^{(j)}$  v místech s významnou hustotou pravděpodobnosti dostatečně velký a pokud jsou váhy  $w_n^{(j)}$  počítány vhodným způsobem [10].

Úlohu rekurzivního výpočtu aposteriorní hustoty pravděpodobnosti

$$f(\mathbf{x}_{n-1} | \mathbf{z}_{1:n-1}) \rightarrow f(\mathbf{x}_n | \mathbf{z}_{1:n}), \quad \text{s použitím vektoru měření } \mathbf{z}_n, \quad (3.15)$$

tak přeformujeme na úlohu rekurzivního výpočtu částic a jejich vah

$$\{(\mathbf{x}_{n-1}^{(j)}, w_{n-1}^{(j)})\}_{j=1}^J \rightarrow \{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J, \quad \text{s použitím vektoru měření } \mathbf{z}_n. \quad (3.16)$$

Částice  $\mathbf{x}_n^{(j)}$  jsou náhodně generovány podle tzv. *návrhové hustoty pravděpodobnosti*  $q(\cdot | \cdot)$  (anglicky proposal/importance density), která je odlišná od aposteriorní hustoty  $f(\mathbf{x}_n | \mathbf{z}_{1:n})$ .

Specifika pro volbu návrhové hustoty pravděpodobnosti lze najít v [2]. Dále blíže popíšeme *obecný částicový filtr*.

#### Algoritmus obecného částicového filtru

- **Inicializace** ( $n = 0$ ):

Generujeme částice  $\mathbf{x}_0^{(j)}, j = 1, \dots, J$  z apriorní hustoty pravděpodobnosti  $f(\mathbf{x}_0)$ .

Váhy částic nastavíme pro všechny částice stejně, tj.  $w_0^{(j)} = 1/J, j = 1, \dots, J$ .

- **Hlavní iterace** ( $n \geq 1$ ):

1. *Predikce*: Pro každou částici z předchozího časového kroku  $\mathbf{x}_{n-1}^{(j)}$  je vypočítána nová částice podle návrhové hustoty pravděpodobnosti

$$q(\mathbf{x}_n | \mathbf{x}_{n-1}^{(j)}, \mathbf{z}_n) \equiv q(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{z}_n) |_{\mathbf{x}_{n-1} = \mathbf{x}_{n-1}^{(j)}}. \quad (3.17)$$

2. *Aktualizace*: Aktuální váhy jsou přepočítány podle

$$\tilde{w}_n^{(j)} = w_{n-1}^{(j)} \frac{f(\mathbf{z}_n | \mathbf{x}_n^{(j)}) f(\mathbf{x}_n^{(j)} | \mathbf{x}_{n-1}^{(j)})}{q(\mathbf{x}_n^{(j)} | \mathbf{x}_{n-1}^{(j)}, \mathbf{z}_n)}, \quad j = 1, \dots, J, \quad (3.18)$$

a následně normalizovány podle  $w_n^{(j)} = \tilde{w}_n^{(j)} / \sum_{j'=1}^J \tilde{w}_n^{(j')}$ . Množina  $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$  reprezentuje aposteriorní hustotu  $f(\mathbf{x}_n | \mathbf{z}_{1:n})$ .

### 3.4. DISTRIBUOVANÝ SEKVENČNÍ BAYESOVSKÝ ODHAD

3. *Výpočet odhadu*: Z  $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$  vypočítáme odhad stavu pomocí (3.14), tj.  $\hat{\mathbf{x}}_n = \sum_{j=1}^J w_n^{(j)} \mathbf{x}_n^{(j)}$ .
  4. *Převzorkování (resampling)*: Množinu  $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$  můžeme převzorkovat, pokud je to žádoucí. Podmínky pro převzorkování a samotný algoritmus převzorkování nalezneme v [2].
- **Výstup**: Odhad stavového vektoru  $\hat{\mathbf{x}}_n$  pro  $n = 1, 2, \dots$

Existuje několik dalších algoritmů, které jsou většinou odvozeny z tohoto obecného filtru. Některé z nich můžeme najít v [2] a [9].

## 3.4. Distribuovaný sekvenční bayesovský odhad

Nyní se už zaměříme na případ, kdy měření  $\mathbf{z}_n$  nezískáváme pouze z jednoho zdroje, ale ze skupiny senzorů. Uvažujme sensorovou síť skládající se z  $S$  senzorů. V čase  $n$  sensor  $s \in \{1, \dots, S\}$  zaznamená lokální měřicí vektor  $\mathbf{z}_n^{(s)}$ , který dostaneme analogicky s (3.2) ze stavového vektoru  $\mathbf{x}_n$  pomocí *lokální rovnice měření*

$$\mathbf{z}_n^{(s)} = \mathbf{h}_n^{(s)}(\mathbf{x}_n, \mathbf{v}_n^{(s)}) \quad (3.19)$$

Funkce  $\mathbf{h}_n^{(s)}(\cdot, \cdot)$  je známá, závislá na senzoru  $s$  a obecně nelineární. Lokální bílý šum  $\mathbf{v}_n^{(s)}$  není závislý na stavovém vektoru  $\mathbf{x}_n$  ani na budícím šumu  $\mathbf{u}_n$  z (3.1).

Z lokálních vektorů měření sestavíme globální vektor měření zahrnující pozorování všech senzorů

$$\mathbf{z}_n = \left( \mathbf{z}_n^{(1)\top}, \dots, \mathbf{z}_n^{(S)\top} \right)^\top.$$

Z rovnice (3.19) a z našich statistických předpokladů jsme schopni odvodit *lokální věrohodnostní funkci* (anglicky local likelihood function – LLF)  $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$  pro každý sensor  $s$ . Většina metod pro distribuované bayesovské sekvenční odhady navíc předpokládá, že lokální měřicí šumy  $\mathbf{v}_n^{(s)}$  jsou nezávislé pro různá  $s$ . Díky tomu můžeme získat *globální věrohodnostní funkci* (anglicky global likelihood function – GLF) jako součin lokálních věrohodnostních funkcí, tj.

$$f(\mathbf{z}_n|\mathbf{x}_n) = \prod_{s=1}^S f(\mathbf{z}_n^{(s)}|\mathbf{x}_n). \quad (3.20)$$

Tuto GLF pak dosadíme do výpočtu aposteriorní hustoty pravděpodobnosti (3.10).

Cílem je tedy odhadnout stav  $\mathbf{x}_n$  sekvenčním způsobem pomocí měření  $\mathbf{z}_{n'}^{(s)}$  všech senzorů v časech  $n'$  až do aktuálního času  $n$ . Opět jsou k dispozici metody založené na Kalmanově filtru či jeho modifikacích. Druhou velkou skupinou algoritmů jsou distribuované částicové filtry, kterými se budeme zabývat v následující části.

## 3.5. Distribuované částicové filtry

Vzhledem k hojnému využití částicových filtrů pro úlohu sekvenčního bayesovského odhadu lze předpokládat, že bude tato metoda účinná i pro distribuované odhady. V tomto případě mluvíme o *distribuovaných částicových filtrech* (anglicky distributed particle filtering – DPF).

### 3.5.1. Přehled DPF algoritmů

Existuje mnoho různých algoritmů tohoto typu, které se liší v různých aspektech jako např. typ a množství distribuovaných dat, dosah komunikace, výpočetní složitost, přesnost odhadu a podobné [10]. Algoritmy můžeme rozdělit do dvou hlavních skupin. Ty, které používají určitou centrální výpočetní jednotku označujeme jako *distribuované částicové filtry s fúzním centrem* (anglicky fusion center (FC)-based DPF). Druhou skupinu tvoří tzv. *decentralizované distribuované částicové filtry*, které centrální výpočetní jednotku nepoužívají. Mluvíme pak o *decentralizované sensorové síti*. U těchto algoritmů tedy požadujeme, aby v každém časovém okamžiku měly všechny senzory informaci (v ideálním případě shodnou) o aktuálním odhadu.

Podle typu dat, které si senzory navzájem posílají, rozlišujeme dvě třídy decentralizovaných sensorových sítí. Buď si senzory vyměňují přímo svoje zaznamenaná měření, nebo si vyměňují nějakou statistickou reprezentaci měření (např. reprezentaci aposteriorní funkce nebo reprezentaci věrohodnostní funkce). My se budeme dále zabývat algoritmem, který spadá do druhé třídy – tzv. *věrohodnostní konsenzus*. Podrobný přehled rozdělení algoritmů i jejich popis nalezneme v [10].

### 3.5.2. Věrohodnostní konsenzus

Algoritmus věrohodnostního konsenzu spadá do skupiny decentralizovaných distribuovaných částicových filtrů, ve kterých všechny senzory vykonávají částicový filtr současně. Každý senzor má svoji reprezentaci aposteriorní funkce, která je v ideálním případě rovna globální aposteriorní funkci (3.20) zohledňující měření všech sensorů. Pomocí decentralizovaného algoritmu konsenzu se senzory „domlouvají“ na určité globální informaci, díky které pak mohou stanovit lokální aproximaci globální aposteriorní funkce  $f(\mathbf{z}_n|\mathbf{x}_n)$ , resp. lokální odhad stavu  $\mathbf{x}_n^{(s)}$ . Uvedme si nejprve algoritmus obecného konsenzu.

#### Algoritmus obecného konsenzu

Uvažujme decentralizovanou sensorovou síť, ve které má každý senzor  $s \in \{1, \dots, S\}$  určitou skalární veličinu  $a^{(s)}$ . Cílem je získat distribuovaným výpočtem průměr  $\frac{1}{S} \sum_{s=1}^S a^{(s)}$  nebo maximum  $\max_s \{a^{(s)}\}$  ze všech hodnot  $a^{(s)}$  tak, že výsledek je dostupný v každém senzoru. Pak mluvíme o *konsenzu průměru*, nebo o *max-konsenzu*. V každé iteraci  $i$  algoritmu konsenzu jednotlivé senzory aktualizují svou interní hodnotu  $\xi_i^{(s)}$  pomocí interních hodnot jejich sousedních sensorů. Množinu sousedních sensorů každého senzoru  $s$  označíme  $\mathcal{N}^{(s)} \subseteq \{1, \dots, S\} \setminus s$ .

- **Inicializace** ( $i = 0$ ):  $\xi_0^{(s)} = a^{(s)}$ .
- **Hlavní iterace** ( $i = 1, \dots, I$ ):

1. Všechny senzory  $s$  aktualizují interní stav, pro konsenzus průměru podle

$$\xi_i^{(s)} = \gamma_{i-1}^{(s,s)} \xi_{i-1}^{(s)} + \sum_{s' \in \mathcal{N}^{(s)}} \gamma_{i-1}^{(s,s')} \xi_{i-1}^{(s')}$$

a pro max-konsenzus podle

$$\xi_i^{(s)} = \max \left\{ \xi_{i-1}^{(s)}, \left\{ \xi_{i-1}^{(s')} \right\}_{s' \in \mathcal{N}^{(s)}} \right\},$$

kde  $\gamma^{(s,s')}$  jsou vhodně zvolené váhy [28].

### 3.5. DISTRIBUOVANÉ ČÁSTICOVÉ FILTRY

2. Každý senzor  $s$  pošle svůj aktuální interní stav  $\xi_i^{(s)}$  všem sousedním sensorům  $s' \in \mathcal{N}^{(s)}$ .

Při konsenzu průměru konvergují interní stavy  $\xi_i^{(s)}$  k požadovanému průměru pro  $I \rightarrow \infty$ . Je zřejmé, že počet iterací  $I$  významně ovlivňuje komunikační (datové) náklady algoritmu.

Konsenzové algoritmy mají několik výhodných vlastností. Jsou odolné vůči změnám v sensorové síti, vůči selhání jednotlivých spojení mezi senzory i selhání samotných sensorů.

#### Věrohodnostní konsenzus

Globální věrohodnostní funkce (3.20) je potřebná pro výpočet vah v obecném částicovém filtru (3.18). Pomocí věrohodnostního konsenzu tedy chceme z lokálních věrohodnostních funkcí  $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$  získat globální věrohodnostní funkci  $f(\mathbf{z}_n|\mathbf{x}_n)$  nebo alespoň její aproximaci. To umožňuje, aby každý senzor prováděl samostatně lokální částicový filtr, který je ekvivalentní ke globálnímu částicovému filtru, protože senzory využívají globální věrohodnostní funkci. Získané odhady stavu  $\mathbf{x}_n$  se můžou lehce lišit kvůli nesynchronizovaným náhodným generátorům a neúplně zkonvergovanému algoritmu konsenzu [10].

Pokud rovnici (3.20) zlogaritmujeme, dostaneme

$$\ln f(\mathbf{z}_n|\mathbf{x}_n) = \ln \left( \prod_{s=1}^S f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) \right) = \sum_{s=1}^S \ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n). \quad (3.21)$$

Algoritmus konsenzu nelze aplikovat přímo na funkci (resp. hustotu pravděpodobnosti) – v tomto případě na logaritmus lokální věrohodnostní funkce  $\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$  (zkráceně log-LLF). Proto tuto funkci aproximujeme pomocí konečného slovníku  $\{\psi_k\}_{k=1}^K$ , viz část 2.3:

$$\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) \approx \sum_{k=1}^K \alpha_k^{(s)} \psi_k(\mathbf{x}_n) \quad (3.22)$$

pro senzory  $s = 1, \dots, S$ . Všechny senzory používají stejný slovník. Vektor lokálních bázových koeficientů  $\boldsymbol{\alpha}^{(s)} = (\alpha_1^{(s)}, \dots, \alpha_K^{(s)})^\top$  je vypočítán v jednotlivých senzorech např. pomocí metody nejmenších čtverců nebo algoritmu OMP. Pokud k výpočtu použijeme OMP, dostaneme  $K_{\text{OMP}}$ -řídový vektor  $\boldsymbol{\alpha}^{(s)}$ .

Vektor globálních bázových koeficientů  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_K)^\top$  získáme konsenzem průměru s celkovým počtem iterací  $I$  pro jednotlivé lokální bázové koeficienty:

$$\beta_k = \frac{1}{S} \sum_{s=1}^S \alpha_k^{(s)}, \quad k = 1, \dots, K. \quad (3.23)$$

V každé iteraci konsenzu  $i = 1, \dots, I$  se aktualizuje lokální odhad globálních bázových koeficientů  $\hat{\beta}_{k,i}^{(s)}$  podle

$$\hat{\beta}_{k,i}^{(s)} = \gamma_{i-1}^{(s,s)} \hat{\beta}_{k,i-1}^{(s)} + \sum_{s' \in \mathcal{N}^{(s)}} \gamma_{i-1}^{(s,s')} \hat{\beta}_{k,i-1}^{(s')} \quad (3.24)$$

pro  $k = 1, \dots, K$ . Konsenzus je inicializován hodnotami lokálních bázových koeficientů, tj.  $\hat{\beta}_{k,0}^{(s)} = \alpha_k^{(s)}$ .



### 3. DISTRIBUOVANÝ ČÁSTICOVÝ FILTR V SENZOROVÝCH SÍTÍCH

Dále, označíme-li

$$\lambda(\mathbf{x}_n) = \frac{1}{S} \ln f(\mathbf{z}_n | \mathbf{x}_n), \quad (3.25)$$

pak ze (3.21) dostaneme

$$\lambda(\mathbf{x}_n) = \frac{1}{S} \sum_{s=1}^S \ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n) \quad (3.26)$$

a globální věrohodnostní funkci můžeme vyjádřit ve tvaru

$$f(\mathbf{z}_n | \mathbf{x}_n) = \exp(\ln f(\mathbf{z}_n | \mathbf{x}_n)) = \exp(S\lambda(\mathbf{x}_n)). \quad (3.27)$$

Dosazením (3.22) do (3.25) dostáváme

$$\lambda(\mathbf{x}_n) \approx \frac{1}{S} \sum_{s=1}^S \sum_{k=1}^K \alpha_k^{(s)} \psi_k(\mathbf{x}_n), \quad (3.28)$$

což lze pomocí globálních bazových koeficientů zjednodušit na

$$\lambda(\mathbf{x}_n) \approx \sum_{k=1}^K \beta_k \psi_k(\mathbf{x}_n). \quad (3.29)$$

Dosazením aproximace  $\lambda(\mathbf{x}_n)$  do (3.27) získáváme konečný vztah pro výpočet globální věrohodnostní funkce:

$$f(\mathbf{z}_n | \mathbf{x}_n) = \exp \left( S \sum_{k=1}^K \beta_k \psi_k(\mathbf{x}_n) \right). \quad (3.30)$$

Každý sensor bude mít po provedení věrohodnostního konsenzu svůj (lokální) odhad globálních bazových koeficientů  $\hat{\beta}^{(s)} = (\hat{\beta}_1^{(s)}, \dots, \hat{\beta}_K^{(s)})^\top$ . Z těchto lokálních odhadů již můžeme podle (3.30) vypočítat lokální aproximaci globální věrohodnostní funkce:

$$\hat{f}^{(s)}(\mathbf{z}_n | \mathbf{x}_n) = \exp \left( S \sum_{k=1}^K \hat{\beta}_k^{(s)} \psi_k(\mathbf{x}_n) \right). \quad (3.31)$$

Dosazením do (3.10) dostáváme lokální aposteriorní hustotu pravděpodobnosti

$$f^{(s)}(\mathbf{x}_n | \mathbf{z}_{1:n}) = \frac{f^{(s)}(\mathbf{z}_n | \mathbf{x}_n) f(\mathbf{x}_n | \mathbf{z}_{1:n-1})}{f(\mathbf{z}_n | \mathbf{z}_{1:n-1})}, \quad (3.32)$$

pomocí které jednotlivé senzory určí svůj lokální odhad stavu  $\hat{\mathbf{x}}_n^{(s)}$ .

#### 3.5.3. Částicová reprezentace

Stejně jako v části 3.3, i pro distribuované částicové filtry zavedeme reprezentaci pomocí částic a jejich vah. Analogicky označíme množiny těchto dvojic pro každý sensor  $s = 1, \dots, S$ :

$$\{(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)})\}_{j=1}^J,$$

### 3.5. DISTRIBUOVANÉ ČÁSTICOVÉ FILTRY

kde  $J$  značí počet částic. Pro jednoduchost nyní uvažujeme, že je počet částic pro všechny senzory stejný. Avšak není problém uvažovat případ různého počtu částic pro různé senzory. Jednotlivé hustoty pravděpodobnosti, pomocí kterých počítáme odhad stavu  $\hat{\mathbf{x}}_n^{(s)}$ , jsou tedy reprezentovány konečným počtem vzorků. Aproximaci log-LLF v (3.22)

$$\ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n) \approx \sum_{k=1}^K \alpha_k^{(s)} \psi_k(\mathbf{x}_n)$$

pak provádíme pouze vzhledem k aktuálním částicím. Tyto částice odpovídají „pozorování“ z části 2.3.1.

Chceme tedy vypočítat lokální bázové koeficienty  $\alpha_k^{(s)}$ . Pro jednotlivé senzory označme vektor těchto zatím neznámých bázových koeficientů

$$\boldsymbol{\alpha}^{(s)} = (\alpha_1^{(s)}, \dots, \alpha_K^{(s)})^\top$$

a vektor funkce log-LLF vyhodnocené v částicích

$$\boldsymbol{\eta}^{(s)} = (\ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n^{(s,1)}), \dots, \ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n^{(s,J)}))^\top.$$

K aproximaci používáme slovník  $\{\psi_k\}_{k=1}^K$ . Matici soustavy pro výpočet koeficientů  $\boldsymbol{\alpha}^{(s)}$  získáme vyhodnocením atomů slovníku v částicích. Označme

$$\boldsymbol{\psi}_k^{(s)} = (\psi_k(\mathbf{x}_n^{(s,1)}), \dots, \psi_k(\mathbf{x}_n^{(s,J)}))^\top$$

a dále

$$\boldsymbol{\Psi}^{(s)} = (\boldsymbol{\psi}_1^{(s)}, \dots, \boldsymbol{\psi}_K^{(s)}),$$

což můžeme pro přehlednost rozepsat na

$$\boldsymbol{\Psi}^{(s)} = \begin{bmatrix} \psi_1(\mathbf{x}_n^{(s,1)}) & \cdots & \psi_k(\mathbf{x}_n^{(s,1)}) \\ \vdots & \ddots & \vdots \\ \psi_1(\mathbf{x}_n^{(s,J)}) & \cdots & \psi_k(\mathbf{x}_n^{(s,J)}) \end{bmatrix}. \quad (3.33)$$

S použitým označením dostáváme systém rovnic

$$\boldsymbol{\Psi}^{(s)} \boldsymbol{\alpha}^{(s)} = \boldsymbol{\eta}^{(s)}, \quad (3.34)$$

ze kterého již získáme lokální bázové koeficienty  $\boldsymbol{\alpha}^{(s)}$  pomocí OMP nebo MNČ.

## 4. Model sensorové sítě

V této kapitole popíšeme již konkrétní model sensorové sítě, který byl sestaven prof. Hlawatschem a jeho týmem a kterým se budeme nadále v této práci zabývat. Uvažujme pohyblivý objekt (anglicky též target – cíl), který má časově proměnný stav  $\mathbf{x}_n = (x_n, y_n, \dot{x}_n, \dot{y}_n)^\top \in \mathbb{R}^4$ , kde  $x, y$  značí polohu objektu (2-D souřadnice),  $\dot{x}, \dot{y}$  značí složky rychlosti objektu v příslušných směrech a  $n \in \mathbb{N}$  značí diskrétní časový index. Stavový vektor objektu se řídí rovnicí stavového prostoru (3.1). V našem modelu se objekt pohybuje podle rovnice:

$$\mathbf{x}_n = \mathbf{G}\mathbf{x}_{n-1} + \mathbf{E}\mathbf{u}_n, \quad (4.1)$$

kde matice  $\mathbf{G}$  je tvaru

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

a  $\mathbf{E}$  je jednotková matice  $4 \times 4$ . Dále  $\mathbf{u}_n \sim \mathbf{N}_4(\mathbf{0}, \Sigma_{\mathbf{u}})$  značí gaussovský budící šum s nulovou střední hodnotou a varianční maticí

$$\Sigma_{\mathbf{u}} = \begin{bmatrix} \sigma_u^2 & 0 & 0 & 0 \\ 0 & \sigma_u^2 & 0 & 0 \\ 0 & 0 & \sigma_u^2 & 0 \\ 0 & 0 & 0 & \sigma_u^2 \end{bmatrix}.$$

Z tvaru matice  $\mathbf{G}$  vidíme, že se poloha objektu vyvíjí podle  $x_n = x_{n-1} + \dot{x}_{n-1}$ , resp.  $y_n = y_{n-1} + \dot{y}_{n-1}$ , což odpovídá pohybu s jednotkovým časovým krokem. K oběma souřadnicím je následně přičtena realizace  $\mathbf{u}_n$ . Složky rychlosti objektu  $\dot{x}, \dot{y}$  se mění jen podle tohoto budícího šumu. Objekt budeme sledovat pouze na předem dané oblasti zájmu (anglicky region of interest).

Dále uvažujme sensorovou síť, která se skládá z  $S$  sensorů tvořících souvislý graf. Pozici jednotlivých sensorů označíme<sup>1</sup>  $\bar{\mathbf{x}}^{(s)} = (x^{(s)}, y^{(s)})^\top$ . Každý sensor  $s \in \{1, \dots, S\}$  může komunikovat se svými sousedními senzory. Množinu sousedních sensorů označíme stejně jako v části 3.5.2, tj.  $\mathcal{N}^{(s)} \subseteq \{1, \dots, S\} \setminus s$ . Dalším parametrem sensorů je jejich maximální dosah měření, který označíme  $R^{(s)}$ . Touto sensorovou sítí sledujeme pohyb zmíněného objektu. Model je znázorněn na obrázku 4.1.

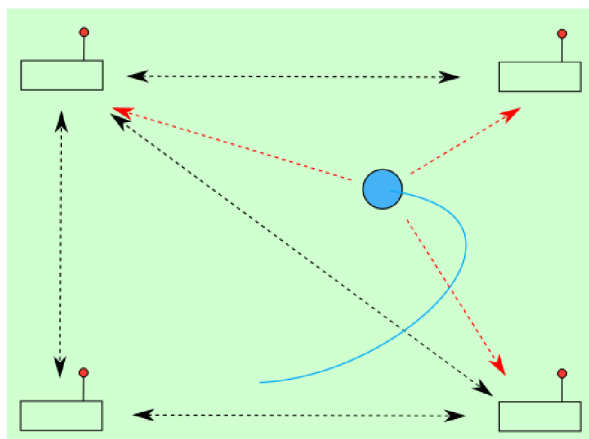
### 4.1. Měřící proces

V čase  $n$  každý sensor  $s$  detekuje náhodný počet  $M_n^{(s)}$  měření  $\mathbf{z}_{n,m}^{(s)}$ , kde  $m \in \mathcal{M}_n^{(s)} = \{1, \dots, M_n^{(s)}\}$ . Měříme<sup>2</sup> vzdálenost objektu od senzoru  $r \in \mathbb{R}_+$  (anglicky range) a úhel  $\rho \in \langle -180^\circ, 180^\circ \rangle$  (anglicky bearing) od senzoru k danému bodu  $\bar{\mathbf{x}}_{n,m} = (x_{n,m}, y_{n,m})^\top$

<sup>1</sup>Označením  $\bar{\mathbf{x}}$  budeme myslet vektor, který obsahuje pouze první dvě složky vektoru  $\mathbf{x}$ , tj. polohu objektu.

<sup>2</sup>V práci se nezabýváme, jakým způsobem senzory měření získají. Můžeme uvažovat případ, kdy sledovaný objekt vysílá akustický nebo rádiový signál [10].

#### 4.1. MĚŘÍCÍ PROCES



Obrázek 4.1: Model sensorové sítě: Zelené pozadí odpovídá oblasti zájmu, v rozích jednotlivé senzory, černé šipky symbolizují komunikační spojení, červené šipky symbolizují zachycení sledovaného objektu (modrý bod). Vidíme, že senzor v levém dolním rohu objekt nezachytil, protože vzdálenost je větší než jeho maximální dosah.

(poloha objektu, nebo falešného měření). Kladný směr osy  $x$  odpovídá nulovému úhlu  $\rho$ . Vektor měření dostáváme z rovnice měření (3.19), která je pro náš model ve tvaru

$$\mathbf{z}_{n,m}^{(s)} = (r_{n,m}^{(s)}, \rho_{n,m}^{(s)})^\top + \mathbf{v}_n^{(s)}, \quad (4.2)$$

kde

$$r_{n,m}^{(s)} = \|\bar{\mathbf{x}}_{n,m} - \bar{\mathbf{x}}^{(s)}\| \quad \text{a} \quad \rho_{n,m}^{(s)} = \text{atan2}(y_{n,m} - y^{(s)}, x_{n,m} - x^{(s)}).$$

Funkce  $\text{atan2}(y, x)$  vrací úhel, mezi kladným směrem osy  $x$  a průvodičem bodu  $(x, y)$  ve 2-D rovině v požadovaném intervalu  $(-180^\circ, 180^\circ)$ . Explicitně ji můžeme zapsat ve tvaru:

$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right), & \text{pokud } x > 0, \\ \arctan\left(\frac{y}{x}\right) + 180^\circ, & \text{pokud } x < 0 \text{ a } y \geq 0, \\ \arctan\left(\frac{y}{x}\right) - 180^\circ, & \text{pokud } x < 0 \text{ a } y < 0, \\ 90^\circ, & \text{pokud } x = 0 \text{ a } y > 0, \\ -90^\circ, & \text{pokud } x = 0 \text{ a } y < 0, \\ \text{nedefinováno}, & \text{pokud } x = 0 \text{ a } y = 0. \end{cases}$$

Dále  $\mathbf{v}_n^{(s)} \sim \mathbf{N}_2(\mathbf{0}, \Sigma_{\mathbf{v}}^{(s)})$  značí gaussovský šum měření s varianční maticí

$$\Sigma_{\mathbf{v}}^{(s)} = \begin{bmatrix} \sigma_r^{2(s)} & 0 \\ 0 & \sigma_\rho^{2(s)} \end{bmatrix}.$$

Rozptyl  $\sigma_r^{2(s)}$  odpovídá měřené vzdálenosti a rozptyl  $\sigma_\rho^{2(s)}$  odpovídá měřenému úhlu. Všimněme si, že je měření závislé pouze na souřadnicích  $x, y$  a nikoliv na složkách rychlosti  $\dot{x}, \dot{y}$ . Rychlost tedy ovlivňuje pouze vývoj polohy objektu, ovšem na měřící proces senzorů nemá vliv.

Nejvýše jedno z těchto měření odpovídá sledovanému objektu, ostatní označíme jako falešná měření (anglicky clutter measurements). Jednotlivá měření  $\mathbf{z}_{n,m}^{(s)}$  uvažujeme jako statisticky nezávislá přes všechny senzory při daném stavu  $\mathbf{x}_n$ . Dále označme vektor všech měření senzoru  $s$  v čase  $n$ :

$$\mathbf{z}_n^{(s)} = \left( \mathbf{z}_{n,1}^{(s)\top}, \dots, \mathbf{z}_{n,M_n^{(s)}}^{(s)\top} \right)^\top$$

a vektor měření všech senzorů v čase  $n$ :

$$\mathbf{z}_n = (\mathbf{z}_n^{(1)\top}, \dots, \mathbf{z}_n^{(s)\top})^\top.$$

Abychom zohlednili, které měření odpovídá sledovanému objektu, zavedeme indikátor  $a_n^{(s)} \in \{0, 1, \dots, M_n^{(s)}\} = \{0\} \cup \mathcal{M}_n^{(s)}$ , pro který platí

$$a_n^{(s)} = \begin{cases} m \in \mathcal{M}_n^{(s)}, & \text{pokud v čase } n \text{ objekt generuje měření } \mathbf{z}_{n,m}^{(s)} \text{ v senzoru } s, \\ 0, & \text{pokud v čase } n \text{ objekt negeneruje měření } \mathbf{z}_{n,m}^{(s)} \text{ v senzoru } s. \end{cases} \quad (4.3)$$

To znamená, že  $a_n^{(s)}$  se rovná indexu  $m$ , pokud sensor v čase  $n$  objekt zachytí. Pravděpodobnost detekce sledovaného objektu označíme  $P_d^{(s)}$ . Tato pravděpodobnost je nám známa. Pokud sensor objekt nezachytí, klademe  $a_n^{(s)} = 0$ .

Dále uvažujeme, že počet falešných měření senzoru  $s$  odpovídá Poissonovu rozdělení (viz příklad 2.11) se střední hodnotou  $\mu^{(s)}$  nezávislou na čase. Veličiny  $a_n^{(s)}$  a  $M_n^{(s)}$  jsou tedy diskrétní náhodné veličiny, pro které můžeme zavést sdruženou pravděpodobnostní funkci [23]:

$$p(a_n^{(s)}, M_n^{(s)}) = \begin{cases} C_1(M_n^{(s)})P_d^{(s)}, & \text{pro } a_n^{(s)} \in \mathcal{M}_n^{(s)}, \\ C_1(M_n^{(s)})(1 - P_d^{(s)})\mu^{(s)}, & \text{pro } a_n^{(s)} = 0, \end{cases} \quad (4.4)$$

kde  $C_1(M_n^{(s)})$  je normalizační člen závislý pouze na počtu měření  $M_n^{(s)}$ .

Podmíněná hustota pravděpodobnosti náhodného vektoru měření  $\mathbf{z}_{n,m}^{(s)}$  při daném stavu  $\mathbf{x}_n$  je dána

$$f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)}) = \begin{cases} f_t(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m), & \text{pro } a_n^{(s)} = m \in \mathcal{M}_n^{(s)}, \\ f_c(\mathbf{z}_{n,m}^{(s)}), & \text{pro } a_n^{(s)} = 0, \end{cases} \quad (4.5)$$

kde  $f_t(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m)$  odpovídá zachycení sledovaného objektu (index „t“ – target) a  $f_c(\mathbf{z}_{n,m}^{(s)})$  odpovídá falešnému měření (index „c“ – clutter). Obě podmíněné hustoty pravděpodobnosti jsou známé. Měřicí proces našeho modelu (4.2) je pak popsán právě hustotou pravděpodobnosti  $f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)})$ .

Nyní uvedeme konkrétní tvar funkcí  $f_t(\cdot)$  a  $f_c(\cdot)$ , který budeme používat v simulacích. Označme  $r_{\mathbf{x}}^{(s)}$  vzdálenost libovolného bodu  $\bar{\mathbf{x}} = (x, y)^\top$  od senzoru  $s$ , tj.  $r_{\mathbf{x}}^{(s)} = \|\bar{\mathbf{x}} - \bar{\mathbf{x}}^{(s)}\|$  a  $\rho_{\mathbf{x}}^{(s)}$  úhel k tomuto bodu od senzoru  $\rho_{\mathbf{x}}^{(s)} = \arctan\left(\frac{y - y^{(s)}}{x - x^{(s)}}\right)$ . Pak pomocí rovnice měření (4.2) dostáváme:

$$\begin{aligned} f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)}) &\equiv f_t(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m) \equiv f_c(\mathbf{z}_{n,m}^{(s)}) = \\ &= \frac{360(R^{(s)})^2}{2r_{n,m}^{(s)}} \frac{1}{\sqrt{2\pi\sigma_r^{2(s)}}} \frac{1}{\sqrt{2\pi\sigma_\rho^{2(s)}}} \exp\left(\frac{-(r_{\mathbf{x}}^{(s)} - r_{n,m}^{(s)})^2}{2\sigma_r^{2(s)}} + \frac{-(\rho_{\mathbf{x}}^{(s)} - \rho_{n,m}^{(s)})^2}{2\sigma_\rho^{2(s)}}\right). \end{aligned} \quad (4.6)$$

Vidíme, že se v našem případě hustoty  $f_t(\cdot)$  a  $f_c(\cdot)$  rovnají. Funkce (4.6) vlastně odpovídá dvojrozměrnému normálnímu rozdělení pro polární souřadnice  $r$  a  $\rho$  (až na multiplikativní konstantu).

## 4.2. ODVOZENÍ LOKÁLNÍ VĚROHODNOSTNÍ FUNKCE

### 4.2. Odvození lokální věrohodnostní funkce

V této části odvodíme tvar lokální věrohodnostní funkce  $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$  pro náš model, viz části 3.4 a 3.5. Pro jednotlivá měření  $m$  jsou podmíněné hustoty pravděpodobnosti  $f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)})$  ze (4.5) statisticky nezávislé, takže pro podmíněnou hustotu náhodného vektoru měření  $\mathbf{z}_n^{(s)}$  při daném  $\mathbf{x}_n, a_n^{(s)}$  a  $M_n^{(s)}$  můžeme psát [23]:

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)}, M_n^{(s)}) = \prod_{m \in \mathcal{M}_n^{(s)}} f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)}). \quad (4.7)$$

Je zřejmé, že pro bližší odvození tvaru podmíněné hustoty pravděpodobnosti z předcházejícího vztahu musíme rozlišit dva případy ze (4.5):

1. Pokud  $a_n^{(s)} = 0$ , všechny měření jsou falešná. Z (4.5) a (4.7) pak dostáváme:

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)}, M_n^{(s)}) = \prod_{m \in \mathcal{M}_n^{(s)}} f_c(\mathbf{z}_{n,m}^{(s)}). \quad (4.8)$$

2. V případě, kdy  $a_n^{(s)} = m \in \mathcal{M}_n^{(s)}$ , odpovídá měření  $\mathbf{z}_{n,m}^{(s)}$  sledovanému objektu a zbylých  $M_n^{(s)} - 1$  měření  $\mathbf{z}_{n,m'}^{(s)}, m' \in \mathcal{M}_n^{(s)} \setminus m$ , jsou měření falešná. Pak opět pomocí (4.5) a (4.7) dostáváme:

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)}, M_n^{(s)}) = f_t(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)} = m) \prod_{m' \in \mathcal{M}_n^{(s)} \setminus \{m\}} f_c(\mathbf{z}_{n,m'}^{(s)}) \quad (4.9)$$

Využitím věty o úplné pravděpodobnosti (2.2) dostaneme:

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) = \sum_{a_n^{(s)}=0}^{M_n^{(s)}} f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)}, M_n^{(s)}) p(a_n^{(s)}, M_n^{(s)}). \quad (4.10)$$

Pokud sumu ve (4.10) „rozdělíme“ pro zmíněné dva případy a dosadíme (4.4), (4.8) a (4.9), získáme konečný tvar pro výpočet lokální věrohodnostní funkce [25]:

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) = C_2(\mathbf{z}_n^{(s)}) \left[ (1 - P_d^{(s)}) \mu^{(s)} + P_d^{(s)} \sum_{m=1}^{M_n^{(s)}} \frac{f_t(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)} = m)}{f_c^{(s)}(\mathbf{z}_{n,m}^{(s)})} \right], \quad (4.11)$$

kde normalizační konstanta je dána

$$C_2(\mathbf{z}_n^{(s)}) = C_1(M_n^{(s)}) \prod_{m \in \mathcal{M}_n^{(s)}} f_c(\mathbf{z}_{n,m}^{(s)}).$$

Poznamenejme, že kdybychom neuvažovali falešná měření, tj.  $M_n^{(s)} = 1, \mu^{(s)} = 0$ , a každý senzor  $s$  by jistě zachytil sledovaný objekt, tj.  $P_d^{(s)} = 1$ , zjednodušil by se výpočet lokální věrohodnostní funkce z (4.11) na  $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) = f_t(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ .

V simulačním programu provedeme zjednodušení pro výpočet lokální věrohodnostní funkce. Jednak uvažujeme, že parametry senzorů  $P_d^{(s)}$  a  $\mu^{(s)}$  jsou pro všechny senzory shodné, tj.  $P_d^{(s)} = P_d$  a  $\mu^{(s)} = \mu$ . Dále, aby pro každý senzor  $s$  ležely hustoty pravděpodobnosti  $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$  na stejné „nulové hladině“  $(1 - P_d)\mu$ , položíme  $C_2(\mathbf{z}_n^{(s)}) = 1$ .

### 4.3. Aproximace lokální věrohodnostní funkce

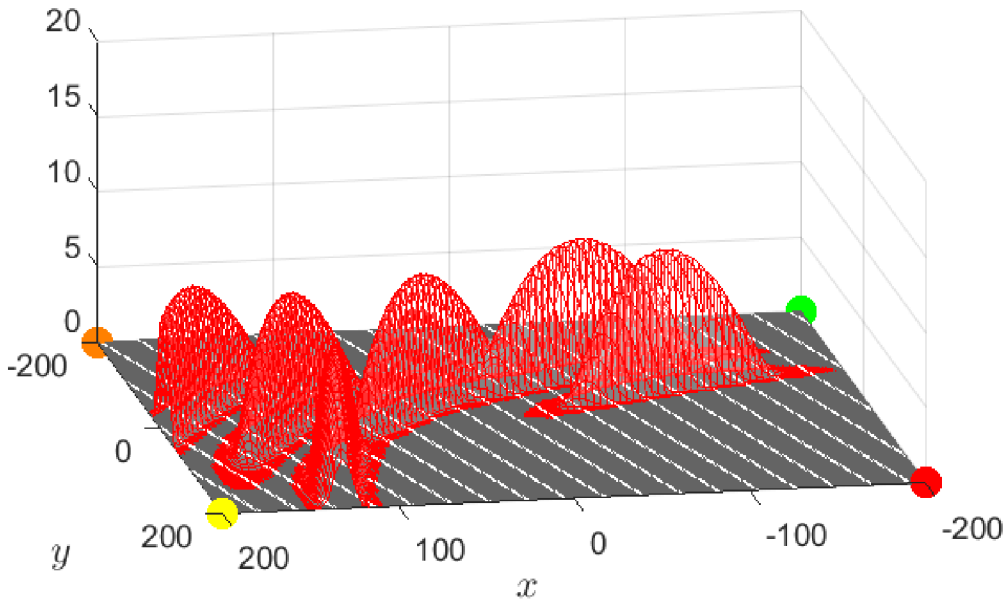
Připomeňme, že pro odhad stavu  $\mathbf{x}_n$  potřebujeme globální věrohodnostní funkci (3.20)

$$f(\mathbf{z}_n|\mathbf{x}_n) = \prod_{s=1}^S f(\mathbf{z}_n^{(s)}|\mathbf{x}_n). \quad (4.12)$$

Abychom mohli použít věrohodnostní konsenzus (viz část 3.5.2) musíme pracovat s logaritmovanými funkcemi:

$$\ln f(\mathbf{z}_n|\mathbf{x}_n) = \sum_{s=1}^S \ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n). \quad (4.13)$$

Na obrázku 4.2 můžeme vidět typický tvar logaritmu lokální věrohodnostní funkce (log-LLF), který vychází ze (4.6). Jednotlivé „kopečky“ odpovídají hustotám pravděpodobnosti  $f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)})$  ze (4.6), resp. jejich logaritmům.



Obrázek 4.2: Ukázka tvaru log-LLF pro velký rozptyl měřeného úhlu  $\sigma_\rho^2$ . Barevné puntíky odpovídají čtyřem sensorům v síti. Na obrázku je znázorněna log-LLF pouze pro „červený“ sensor na pozici  $(-200, 200)$ .

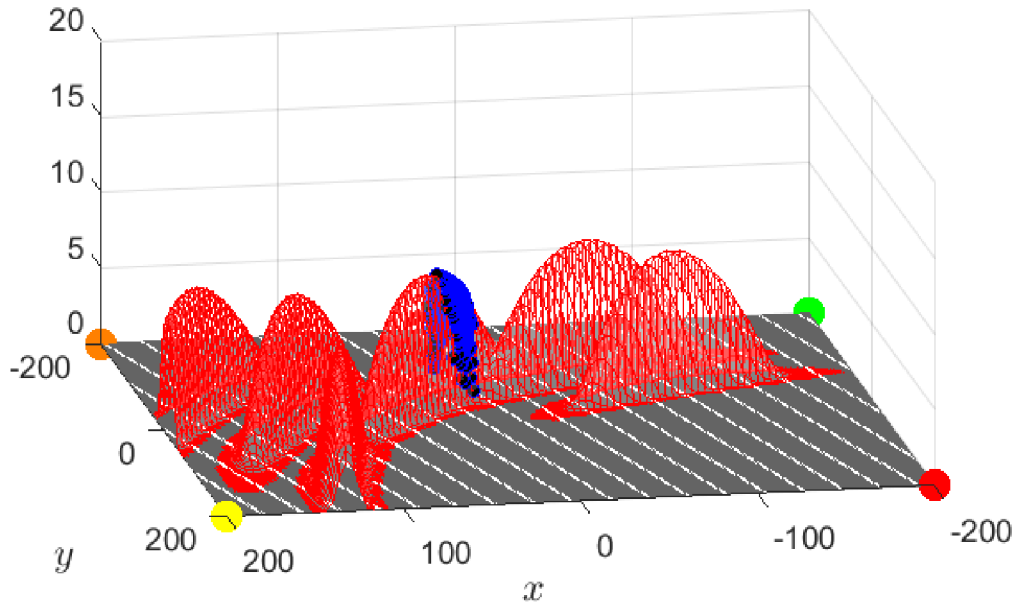
Všimněme si, že pro jednodušší použití slovníků pro aproximaci posouváme nulovou hladinu funkcí log-LLF do nuly<sup>3</sup> odečtením členu  $\ln((1 - P_d)\mu)$  od logaritmu věrohodnostní funkce  $\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$  ze (4.13). Posunutí můžeme provést díky tomu, že se log-LLF funkce sčítají při výpočtu globální věrohodnostní funkce. To znamená, že se tvar GLF nezmění, ale pouze se posune její obor hodnot, což odhad stavu neovlivní.

Podle kopcovitěho tvaru log-LLF můžeme vybrat vhodné typy slovníků pro aproximaci těchto funkcí, viz (3.22). V naší práci budeme používat a následně i porovnávat dva typy slovníků, které jsme zmínili v části 2.3.3. Jde o tzv. B-splajny a Fourierův slovník.

<sup>3</sup>Díky tomu nemusíme ke slovníku B-splajnů přidávat konstantní atom.

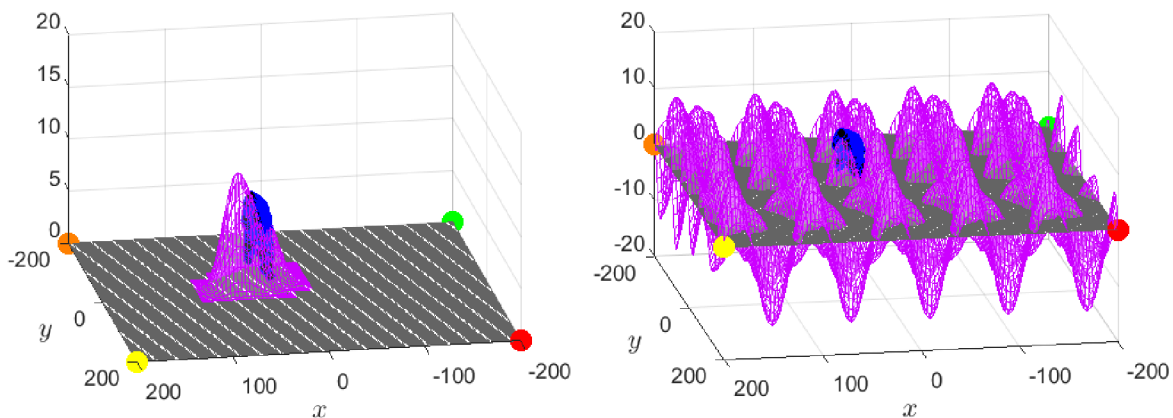
### 4.3. APROXIMACE LOKÁLNÍ VĚROHODNOSTNÍ FUNKCE

Jak je vysvětleno v části 3.5.3, nepotřebujeme aproximovat funkci na celé oblasti zájmu, ale pouze v místech výskytu částic. Na obrázku 4.3 můžeme vidět stejnou log-LLF jako na obrázku 4.2 s tím rozdílem, že jsou na něm zobrazené i částice.



Obrázek 4.3: Ukázka tvaru log-LLF pro velký rozptyl měřeného úhlu  $\sigma_\rho^2$ . Barevné puntíky odpovídají čtyřem sensorům v síti. Na obrázku je znázorněna log-LLF pouze pro „červený“ sensor na pozici  $(-200, 200)$ . Modré body znázorňují částice.

Na dalším obrázku 4.4 již vidíme aproximaci log-LLF s použitím obou slovníků. Je zjevné, že v případě Fourierova slovníku dochází k chybné aproximaci na oblastech bez částic. To ovšem není důležité, neboť jak jsme zmínili výše, hledáme aproximaci log-LLF pouze vzhledem k částicím.



Obrázek 4.4: Ukázka aproximace log-LLF z obrázku 4.3. Vlevo aproximace pomocí slovníku B-splajnů, vpravo pomocí Fourierova slovníku.



## 4.4. Částicový filtr

Nyní si uvedeme, jak senzory generují částice a jakým způsobem probíhá výpočet jejich vah. Každý sensor provádí svůj lokální částicový filtr nezávisle na lokálních částicových filtrech ostatních sensorů, ovšem za pomoci měření všech sensorů. Máme tedy reprezentaci

$$\{(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)})\}_{j=1}^J, \quad \text{pro senzory } s = 1, \dots, S,$$

kde  $J$  značí počet částic.

V našem modelu použijeme nejjednodušší algoritmus částicového filtru, tzv. *Sampling Importance Resampling Filter*. Je to varianta zmíněného algoritmu obecného částicového filtru, při které za návrhovou hustotu pravděpodobnosti  $q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_n)$  volíme přechodovou hustotu pravděpodobnosti, tj.

$$q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_n) \equiv f(\mathbf{x}_n|\mathbf{x}_{n-1}) \quad (4.14)$$

a zároveň provádíme převzorkování při každém časovém kroku [2]. Dále uvedeme podrobnější popis tohoto algoritmu.

### Algoritmus Sampling Importance Resampling Filter

- **Inicializace** ( $n = 0$ ):  
Generujeme částice  $\mathbf{x}_0^{(s,j)}$ ,  $j = 1, \dots, J$  z apriorní hustoty pravděpodobnosti  $f^{(s)}(\mathbf{x}_0)$ .
- **Hlavní iterace** ( $n \geq 1$ ):
  1. *Predikce*: Pro každou částici z předchozího časového kroku  $\mathbf{x}_{n-1}^{(j)}$  je vypočítána nová částice podle přechodové hustoty pravděpodobnosti

$$f(\mathbf{x}_n|\mathbf{x}_{n-1}^{(s,j)}) \equiv f(\mathbf{x}_n|\mathbf{x}_{n-1})|_{\mathbf{x}_{n-1}=\mathbf{x}_{n-1}^{(s,j)}}. \quad (4.15)$$

2. *Aktualizace*: Aktuální váhy jsou přepočítány podle

$$w_n^{(s,j)} = \frac{\hat{f}(\mathbf{z}_n|\mathbf{x}_n^{(s,j)})}{\sum_{j=1}^J \hat{f}(\mathbf{z}_n|\mathbf{x}_n^{(s,j)})}, \quad j = 1, \dots, J, \quad (4.16)$$

kde  $\hat{f}(\mathbf{z}_n|\mathbf{x}_n^{(s,j)})$  je lokální aproximace globální věrohodnostní funkce z (3.31) vyhodnocená v částici  $\mathbf{x}_n^{(s,j)}$ .

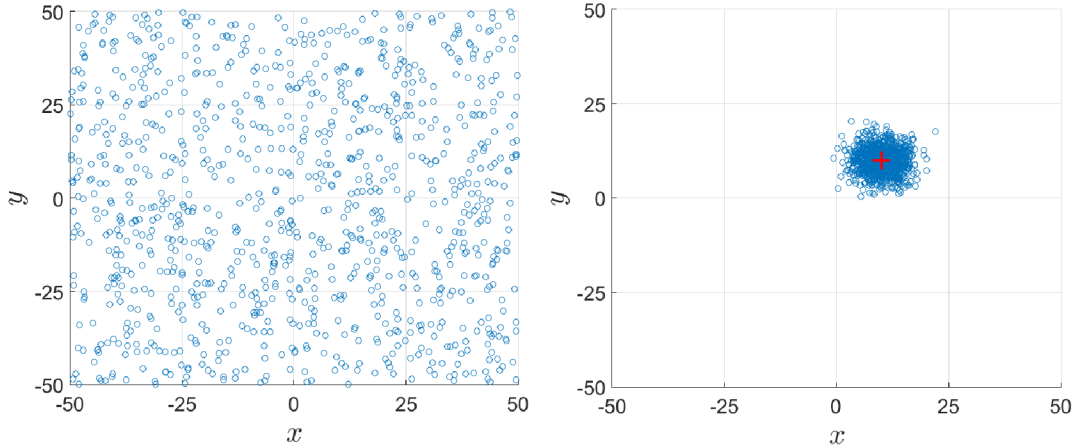
3. *Výpočet odhadu*: Z  $\{(\mathbf{x}_n^{(s,j)}, w_n^{(j)})\}_{j=1}^J$  vypočítáme odhad stavu pomocí (3.14), tj.  $\hat{\mathbf{x}}_n^{(s)} = \sum_{j=1}^J w_n^{(s,j)} \mathbf{x}_n^{(s,j)}$ .
  4. *Převzorkování (resampling)*: Množinu  $\{(\mathbf{x}_n^{(s,j)}, w_n^{(j)})\}_{j=1}^J$  převzorkujeme [2]. Váhy nastavíme rovnoměrně na  $w_n^{(s,j)} = 1/J$ .
- **Výstup**: Lokální odhady stavu  $\hat{\mathbf{x}}_n^{(s)}$  pro senzory  $s = 1, \dots, S$  a časy  $n = 1, 2, \dots$

### Volba apriorní hustoty pravděpodobnosti

Za apriorní hustotu pravděpodobnosti  $f^{(s)}(\mathbf{x}_0)$  volíme jednu z těchto možností (viz také obrázek 4.5):

#### 4.5. PROMAZÁVÁNÍ SLOVNÍKU

- A) **Rovnoměrné rozdělení** - částice jsou rozděleny rovnoměrně po celé oblasti zájmu. Tuto možnost zvolíme, pokud nemají senzory apriorní informaci o počátečním stavu  $\mathbf{x}_0$  sledovaného objektu.
- B) **Normální rozdělení** - částice jsou rozděleny podle dvojrozměrného normálního rozdělení se střední hodnotou v počátečním stavu  $\mathbf{x}_0$ . Tuto možnost tedy zvolíme, pokud mají senzory k dispozici apriorní informaci o počátečním stavu  $\mathbf{x}_0$  sledovaného objektu.



Obrázek 4.5: Příklady částicové reprezentace apriorní hustoty pravděpodobnosti  $f^{(s)}(\mathbf{x}_0)$  pro 1000 částic: vlevo rovnoměrné rozdělení, vpravo normální rozdělení se střední hodnotou v bodě  $\bar{\mathbf{x}}_0 = (-10, 10)^\top$ , který je znázorněn červeným křížkem.

#### Přidání částic

V simulacích budeme zkoušet také variantu, kdy v prvním kroku predikce přidáme k aktuálním částicím také  $J_z$  částic okolo každého získaného měření  $\mathbf{z}_{n,m}^{(s)}$ , viz (4.2). Tyto „extra“ částice budou podobně jako v případě známého počátečního stavu generovány z normálního rozdělení se střední hodnotou v bodě odpovídajícímu měření a rozptylem  $\sigma_z^2$ . Na obrázku 4.6 vidíme příklad log-LLF a její částicové reprezentace s přidávanými částicemi kolem všech měření.

### 4.5. Promazávání slovníku

Zopakujme, že při aproximaci funkcí log-LLF hledáme nějaké přibližné řešení  $\boldsymbol{\alpha}^{(s)}$  soustavy

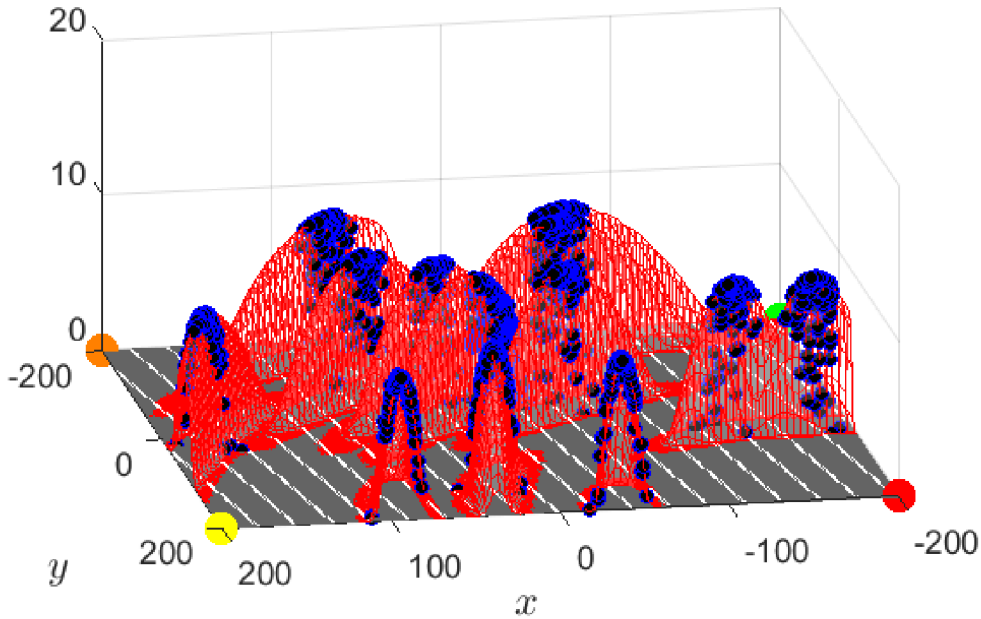
$$\boldsymbol{\Psi}^{(s)} \boldsymbol{\alpha}^{(s)} = \boldsymbol{\eta}^{(s)}. \quad (4.17)$$

Slovník  $\boldsymbol{\Psi}^{(s)}$  je složen z jednotlivých atomů

$$\boldsymbol{\Psi}^{(s)} = \left( \boldsymbol{\psi}_1^{(s)}, \dots, \boldsymbol{\psi}_K^{(s)} \right),$$

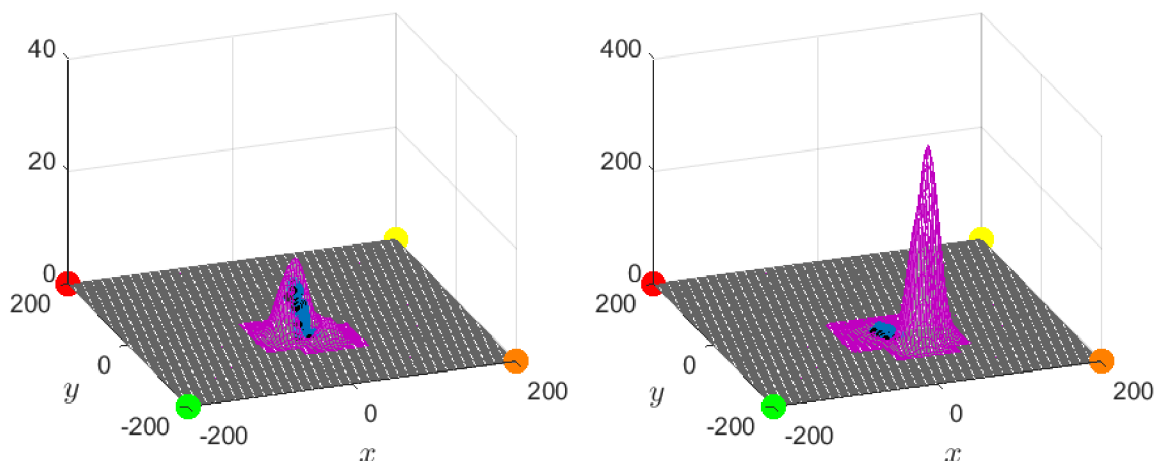
které získáme vyhodnocením v aktuální částicích

$$\boldsymbol{\psi}_k^{(s)} = \left( \psi_k(\mathbf{x}_n^{(s,1)}), \dots, \psi_k(\mathbf{x}_n^{(s,J)}) \right)^\top,$$



Obrázek 4.6: Ukázka log-LLF a její částicové reprezentace s „extra“ částicemi

viz část 3.5.3. Dříve než hledáme řešení soustavy (4.17) pomocí OMP nebo MNČ, můžeme slovník „promazat“. Promazáním slovníku myslíme nahrazení některých prvků slovníku, které splňují určitou podmínku, nulovou hodnotou. Při nízkých hodnotách prvků slovníku může totiž dojít k nežádoucím jevům. Když například atom slovníku B-splajnů obsahuje jen malý počet nízkých hodnot (tzn. jen pár částic na „okraji“ splajnu), může při aproximaci vyjít velmi vysoká hodnota odpovídajícího bazového koeficientu, viz obrázek 4.7. To může mít negativní vliv v následujícím věrohodnostním konsenzu.



Obrázek 4.7: Ukázka aproximace log-LLF pomocí B-splajnů: vlevo s použitím promazání, vpravo bez promazání, částice jsou v obou případech shodné. Bez promazání má aproximace log-LLF výrazně vyšší hodnoty, a tedy i hodnota jednoho z koeficientů  $\alpha_k^{(s)}$  je vysoká.

#### 4.6. KOMPRESNÍ VĚROHODNOSTNÍ KONSENZUS

V našem modelu jsme vyzkoušeli dva typy promazávání:

1. **Promazávání jednotlivých hodnot** (clearing values)

Pokud je absolutní hodnota prvku slovníku menší než určitá prahová hodnota  $\varepsilon_v$  (index „v“ – values), tj.  $|\psi_k(\mathbf{x}_n^{(s,j)})| < \varepsilon_v$ , tento prvek vynulujeme, tj.  $\psi_k(\mathbf{x}_n^{(s,j)}) = 0$ .

2. **Promazávání celých atomů** (clearing atoms)

Pokud je velikost atomu slovníku menší než určitá prahová hodnota  $\varepsilon_a$  (index „a“ – atoms), tj.  $\|\boldsymbol{\psi}_k^{(s)}\| < \varepsilon_a$ , celý tento atom vynulujeme, tj.  $\psi_k(\mathbf{x}_n^{(s,j)}) = 0$ , pro  $j = 1, \dots, J$ .

Případně můžeme tyto dva typy i kombinovat. Vhodnou prahovou hodnotu pro promazávání celých atomů  $\varepsilon_a$  je nutno odhadnout experimentálně, přičemž je zřejmé, že závisí především na počtu částic  $J$ . Jakmile je slovník promazán, pokračujeme dále výpočtem lokálních bázových koeficientů  $\boldsymbol{\alpha}^{(s)}$ .

### 4.6. Kompresní věrohodnostní konsenzus

Při distribuovaném výpočtu globálních bázových koeficientů  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_K)^\top$ , pro které platí

$$\beta_k = \frac{1}{S} \sum_{s=1}^S \alpha_k^{(s)}, \quad k = 1, \dots, K, \quad (4.18)$$

provádí sensorová síť celkem  $K$  věrohodnostních konsenzů, viz část 3.5.2. To samozřejmě zapříčiní vysoké komunikační náklady. Abychom nemuseli provádět tak velký počet konsenzů a ušetřili tak komunikační náklady sensorové sítě, aplikujeme metodu komprimovaného snímání, kterou jsme uvedli v části 2.4.

Připomeňme, že funkce log-LLF  $\ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n)$  jsou reprezentovány právě lokálními bázovými koeficienty  $\boldsymbol{\alpha}^{(s)} = (\alpha_1^{(s)}, \dots, \alpha_K^{(s)})^\top$  ze (4.18). Pokud k výpočtu  $\boldsymbol{\alpha}^{(s)}$  použijeme metodu OMP s  $K_{\text{OMP}} \leq K$  iteracemi, vektor  $\boldsymbol{\alpha}^{(s)}$  je  $K_{\text{OMP}}$ -řádký, tj. nejvíce  $K_{\text{OMP}}$  složek  $\alpha_k^{(s)}$  je nenulových, viz část 2.3.4.

Uvažujme kompresní matici  $\Phi$  o rozměrech  $P \times K$ , přičemž  $P \ll K$ , která je známa všem sensorům. Prvky matice  $\Phi$  volíme jako nezávislé realizace náhodné proměnné s normálním rozdělením s nulovou střední hodnotou. Pro jednotlivé senzory je vektor lokálních bázových koeficientů  $\boldsymbol{\alpha}^{(s)}$  vynásoben kompresní maticí  $\Phi$ , čímž dostáváme komprimovaný vektor

$$\mathbf{u}^{(s)} = \Phi \boldsymbol{\alpha}^{(s)} \quad (4.19)$$

dimenze  $P$ , tj.  $\mathbf{u}^{(s)} = (u_1^{(s)}, \dots, u_P^{(s)})^\top$ . Průměr komprimovaných vektorů  $\mathbf{u}^{(s)}$  ze všech sensorů získáme podle

$$\mathbf{v} = \frac{1}{S} \sum_{s=1}^S \mathbf{u}^{(s)}. \quad (4.20)$$

Vektor  $\mathbf{v} = (v_1, \dots, v_P)^\top$  je pak vypočítán distribuovaně pomocí konsenzu průměru. Stejně jako ve věrohodnostním konsenzu (viz část 3.5.2) v každé iteraci  $i = 1, \dots, I$  aktualizujeme lokální odhad kompresního vektoru  $\hat{v}_{p,i}^{(s)}$  podle

$$\hat{v}_{p,i}^{(s)} = \gamma_{i-1}^{(s,s)} \hat{v}_{p,i-1}^{(s)} + \sum_{s' \in \mathcal{N}^{(s)}} \gamma_{i-1}^{(s,s')} \hat{v}_{p,i-1}^{(s')} \quad (4.21)$$

pro  $p = 1, \dots, P$ . Konsenzus je inicializován hodnotami lokálních bázových koeficientů, tj.  $\hat{v}_{p,0}^{(s)} = u_p^{(s)}$ .

Senzorová síť v tomto případě provádí pouze  $P$  konsenzů oproti původnímu počtu  $K$  konsenzů. Vzhledem k tomu, že platí  $P \ll K$ , šetříme výrazně komunikační náklady sensorové sítě, čehož jsme chtěli dosáhnout.

Zbývá popsat, jakým způsobem získáme z vektoru  $\mathbf{v}$  hledané globální koeficienty  $\boldsymbol{\beta}$ . Díky (4.18) a (4.19) platí

$$\frac{1}{S} \sum_{s=1}^S \mathbf{u}^{(s)} = \frac{1}{S} \sum_{s=1}^S \Phi \boldsymbol{\alpha}^{(s)} = \Phi \left( \frac{1}{S} \sum_{s=1}^S \boldsymbol{\alpha}^{(s)} \right) = \Phi \boldsymbol{\beta}. \quad (4.22)$$

Z předchozích vztahů (4.20) a (4.22) dostaneme soustavu rovnic  $\Phi \boldsymbol{\beta} = \mathbf{v}$ . Vzhledem k tomu, že platí  $P < K$ , jedná se o soustavu nedourčenou, a tedy můžeme hledat nejjednodušší řešení. Řešíme úlohu

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_0 \quad \text{vzhledem k} \quad \Phi \boldsymbol{\beta} = \mathbf{v}. \quad (4.23)$$

Tuto úlohu jsme opět schopni řešit algoritmem OMP.

Analogicky s věrohodnostním konsenzem v části 3.5.2 bude mít každý senzor  $s$  po provedení konsenzu svůj lokální odhad vektoru  $\hat{\mathbf{v}}^{(s)}$ . Jednotlivé senzory tedy řeší místo úlohy (4.23) úlohu

$$\hat{\boldsymbol{\beta}}^{(s)} = \arg \min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_0 \quad \text{vzhledem k} \quad \Phi \boldsymbol{\beta} = \hat{\mathbf{v}}^{(s)}. \quad (4.24)$$

Pomocí lokálního odhadu globálních bázových koeficientů  $\hat{\boldsymbol{\beta}}^{(s)}$  jsme pak schopni určit odhad stavu  $\hat{\mathbf{x}}_n^{(s)}$  stejně jako v části 3.5.2.

## 4.7. Algoritmus modelu sensorové sítě

Pro větší přehlednost shrneme předcházející části do uceleného algoritmu, který popisuje proces fungování našeho modelu sensorové sítě. Hvězdičkou \* označujeme kroky, které jsou pro algoritmus nepovinné.

### Algoritmus pro sledování objektu sensorovou sítí

- **Inicializace** ( $n = 0$ ):

Z apriorní hustoty pravděpodobnosti  $f^{(s)}(\mathbf{x}_0)$  generují senzory  $J$  částic  $\mathbf{x}_0^{(s,j)}$ , kde  $j = 1, \dots, J$ , viz část 4.4.

- **Hlavní iterace** ( $n \geq 1$ ):

1. *Predikce*: Pro každou částici z předchozího časového kroku  $\mathbf{x}_{n-1}^{(s,j)}$  senzory vypočítají novou částici  $\mathbf{x}_n^{(s,j)}$  podle  $f(\mathbf{x}_n | \mathbf{x}_{n-1})$ , viz část 4.4.
2. *Měřicí proces*: Senzory zaznamenávají svá měření  $\mathbf{z}_{n,m}^{(s)}$  podle (4.2) a následně vypočítají svoji lokální věrohodnostní funkci  $f(\mathbf{z}_n^{(s)} | \mathbf{x}_n)$  podle (4.11).

\*3. *Přidání částic*: Kolem všech měření přidávají senzory nových  $J_z$  částic, viz část 4.4.

4. *Aproximace log-LLF*: Senzory určí aproximaci log-LLF v následujících krocích:
  - (i) Sestavení slovníku  $\Psi^{(s)}$  podle (3.33).

#### 4.7. ALGORITMUS MODELU SENZOROVÉ SÍTĚ

- \*(ii) Promazání slovníku  $\Psi^{(s)}$ , viz část 4.5.
- (iii) Výpočet lokálních bázových koeficientů  $\alpha^{(s)}$  z rovnice (3.34) pomocí OMP.
- \*5. *Komprimované snímání*: Sensory vypočítají své komprimované vektory  $\mathbf{u}^{(s)}$  podle (4.19).
- 6. Celá sensorová síť provádí věrohodnostní konsenzus pro výpočet globálních bázových koeficientů  $\beta$ , resp. jejich lokálních odhadů  $\hat{\beta}^{(s)}$ , viz část 3.5.2. Pokud sensory použily komprimované snímání (krok \*5.), probíhá výpočet  $\hat{\beta}^{(s)}$  podle (4.24).
- 7. *Aktualizace*: Aktuální váhy jsou přepočítány podle

$$w_n^{(s,j)} = \frac{\hat{f}(\mathbf{z}_n | \mathbf{x}_n^{(s,j)})}{\sum_{j=1}^J \hat{f}(\mathbf{z}_n | \mathbf{x}_n^{(s,j)})}, \quad j = 1, \dots, J,$$

kde  $\hat{f}(\mathbf{z}_n | \mathbf{x}_n^{(s,j)})$  je lokální aproximace globální věrohodnostní funkce z (3.31) vyhodnocená v částici  $\mathbf{x}_n^{(s,j)}$ .

- 8. *Výpočet odhadu*: Z vážených částic  $\{(\mathbf{x}_n^{(s,j)}, w_n^{(j)})\}_{j=1}^J$  vypočítají sensory odhad stavu pomocí (3.14), tj.  $\hat{\mathbf{x}}_n^{(s)} = \sum_{j=1}^J w_n^{(s,j)} \mathbf{x}_n^{(s,j)}$ .
- 9. *Převzorkování (resampling)*: Sensory provedou převzorkování množiny částic  $\{(\mathbf{x}_n^{(s,j)}, w_n^{(j)})\}_{j=1}^J$ .
- **Výstup**: Lokální odhady stavu  $\hat{\mathbf{x}}_n^{(s)}$  pro sensory  $s = 1, \dots, S$  a časy  $n = 1, 2, \dots$

## 5. Simulace a výsledky

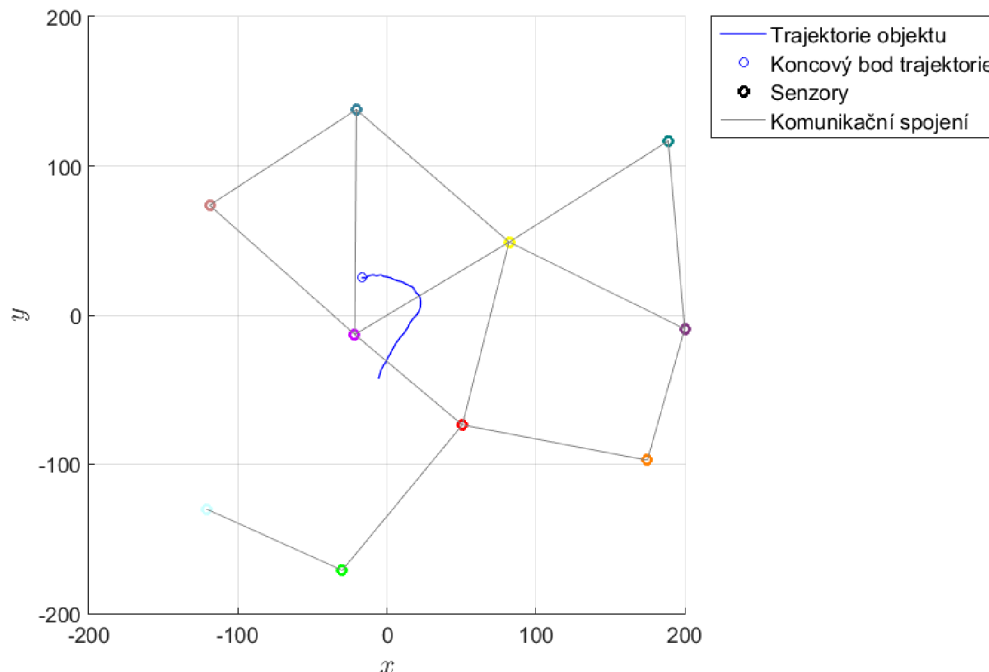
V této kapitole uvedeme výsledky simulací, které jsme provedli pro model sensorové sítě představený v předchozí kapitole 4. Nejprve popíšeme nastavení modelu, parametry, které budou proměnné pro různé simulace, a hodnotící kritéria. Dále se zaměříme na samotné výsledky, které jsme ze simulací získali a které následně interpretujeme a vyhodnotíme.

### 5.1. Nastavení modelu

Sledovaný objekt mění svůj stav  $\mathbf{x}_n$  podle (4.1) s rozptylem budícího šumu  $\sigma_u^2 = \frac{1}{9}$  na oblasti zájmu  $\langle -200, 200 \rangle \times \langle -200, 200 \rangle$  délkových jednotek<sup>1</sup>. Sensorová síť se skládá z celkem  $S = 10$  senzorů, které mají – až na pozici – stejné parametry (proto nyní vynecháme index ( $s$ ) u těchto parametrů):

- dosah měření  $R = 500$ ,
- rozptyl měřené vzdálenosti  $\sigma_r^2 = \left(\frac{5}{3}\right)^2$ ,
- rozptyl měřeného úhlu  $\sigma_\rho^2 = \left(\frac{10}{3}\right)^2$ ,
- pravděpodobnost detekce  $P_d = 0,95$ ,
- střední hodnota počtu falešných měření  $\mu = 5$ .

Pozici jednotlivých senzorů a jejich komunikační spojení můžeme vidět na obrázku 5.1. Uvažujeme případ, kdy senzory nemají apriorní informaci o počátečním stavu, takže za apriorní hustotu pravděpodobnosti  $f^{(s)}(\mathbf{x}_0)$  volíme rovnoměrné rozdělení. Senzory sledují objekt přes padesát časových kroků, tj.  $n = 1, \dots, 50$ . Zkušební trajektorii objektu vidíme také na obrázku 5.1.



Obrázek 5.1: Oblast zájmu se sensorovou sítí a zkušební trajektorií sledovaného objektu

<sup>1</sup>Můžeme uvažovat např. metry nebo kilometry. Pro simulace je to nepodstatné, a proto nebudeme dále jednotky délky uvádět.

## 5.1. NASTAVENÍ MODELU

Doposud zmíněná nastavení tedy zůstávají stejná pro všechny provedené simulace. Nyní si vypíšeme seznam parametrů, které můžeme v různých simulacích měnit<sup>2</sup>. Pro přehlednost je uvádíme v tabulce 5.1.

Tabulka 5.1: Seznam parametrů (jejich označení, popis a přípustné hodnoty), které nastavujeme pro každou simulaci.

	označení	popis	přípustné hodnoty
1) nastavení částicového filtru	$J$	základní počet částic	$\mathbb{N}$
	–	přidání částic kolem všech měření	ano/ne
	$J_z$	počet přidávaných částic kolem každého měření	$\mathbb{N}$
	$\sigma_z^2$	rozptyl přidávaných částic	$\mathbb{R}_+$
2) nastavení slovníku	–	typ slovníku	B-splajny / Fourierův slovník
	$K$	velikost slovníku	$\mathbb{N}$
3) nastavení věrohodnostního konsenzu	–	promazávání jednotlivých hodnot slovníku	ano/ne
	$\varepsilon_v$	prahová hodnota pro promazávání hodnot	$\mathbb{R}_+$
	–	promazávání celých atomů slovníku	ano/ne
	$\varepsilon_a$	prahová hodnota pro promazávání celých atomů	$\mathbb{R}_+$
	$K_{\text{OMP}}$	maximální řídkost	$\mathbb{N}, K_{\text{OMP}} < K$
	$I$	počet iterací konsenzu	$\mathbb{N}$
4) nastavení komprimovaného snímání	–	použití komp. snímání	ano/ne
	$P$	počet řádků kompresní matice	$\mathbb{N}, P < K$

Vidíme, že počet parametrů, které můžeme nastavovat pro jednotlivé simulace je poměrně velký. Většinou budeme mezi sebou porovnávat simulace, které se liší pouze jedním parametrem (či několika málo parametry), zatímco zbylé parametry budou nastaveny stejně.

<sup>2</sup>Později bude z výčtu simulací vidět, že některé parametry byly pro všechny simulace stejné. Nicméně přiložené soubory jsou navrženy tak, že parametry z tabulky 5.1 lze jednoduše nastavit v základním simulačním skriptu.

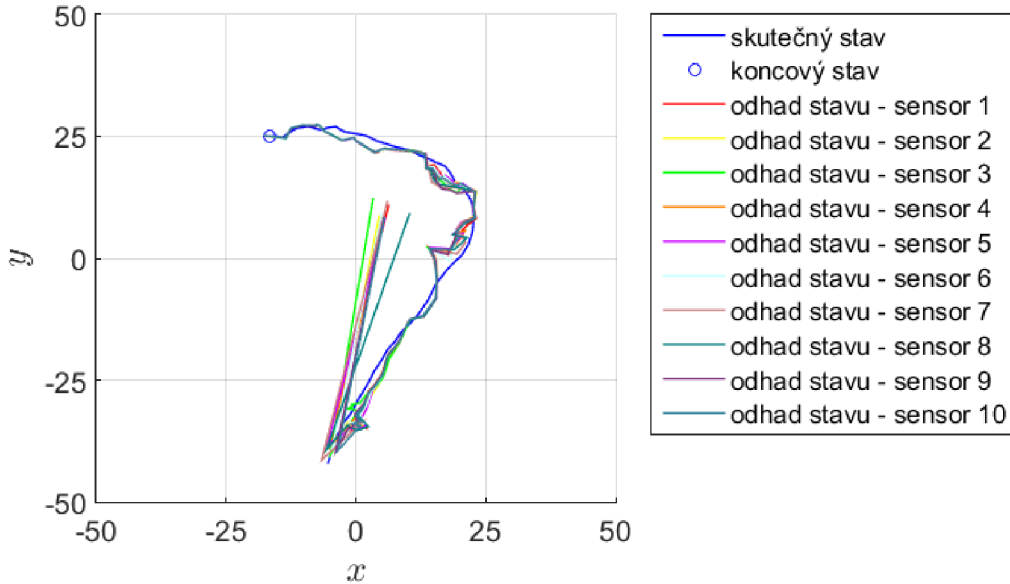


## 5.2. Hodnotící kritéria

Jednotlivé simulace, které se skládají ze 100 experimentů, budeme porovnávat podle několika hodnotících kritérií, které dále popíšeme. Patří mezi ně úspěšnost a přesnost sledování objektu, komunikační (datové) náklady a výpočetní (časová) náročnost.

### 5.2.1. Chyba sledování

Základním hodnotícím parametrem při úloze sledování objektu je evidentně přesnost tohoto sledování. V souladu s kapitolou 4 označme  $\bar{\mathbf{x}}_n = (x_n, y_n)^\top$  skutečnou polohu objektu v čase  $n$ . Podobně  $\hat{\mathbf{x}}_n^{(s)}$  značí lokální odhad (senzoru  $s$ ) polohy objektu v čase  $n$ . Na obrázku 5.2 můžeme vidět příklad odhadů polohy pro naši zkušební trajektorii pro jeden experiment.



Obrázek 5.2: Ukázka odhadů polohy pro jeden experiment, oblast zájmu je přibližena  
Chybu sledování senzoru  $s$  v čase  $n$  potom dostaneme jako

$$E_n^{(s)} = \|\bar{\mathbf{x}}_n - \hat{\mathbf{x}}_n^{(s)}\|, \quad (5.1)$$

což odpovídá vzdálenosti skutečné polohy a odhadu. Střední kvadratickou chybu  $\text{RMSE}^{(s)}$  (z angl. root mean square error) senzoru  $s$  dostaneme jako průměr ze všech časových kroků  $n = 1, \dots, N$ :

$$\text{RMSE}^{(s)} = \frac{1}{N} \sum_{n=1}^N E_n^{(s)}. \quad (5.2)$$

Celkovou střední kvadratickou chybu senzorové sítě  $\text{RMSE}$  pak vypočítáme jako průměr středních kvadratických chyb všech senzorů  $s = 1, \dots, S$ , tj.:

$$\text{RMSE} = \frac{1}{S} \sum_{s=1}^S \text{RMSE}^{(s)}. \quad (5.3)$$

## 5.2. HODNOTÍČÍ KRITÉRIA

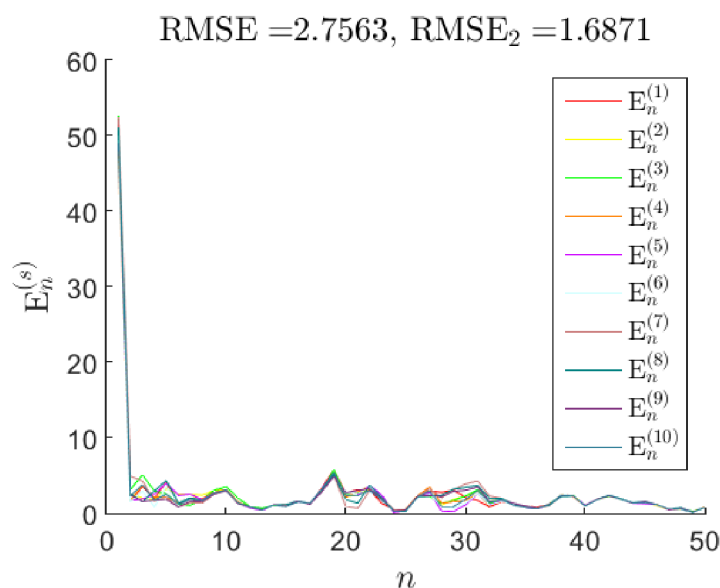
Vzhledem k tomu, že při simulacích používáme rovnoměrné rozdělení apriorní hustoty pravděpodobnosti  $f^{(s)}(\mathbf{x}_0)$ , může docházet na začátku sledování k větším chybám, resp. několik časových kroků trvá než odhady „zkonvergují“ ke skutečné poloze, viz obrázky 5.2 a 5.3. Proto je vhodné měřit také průměrnou chybu sledování až od určitého pozdějšího časového kroku. V našem případě jsme zvolili  $n = 10$ , a tedy dostáváme střední kvadratickou chybu  $\text{RMSE}_2^{(s)}$  senzoru  $s$  pro  $n = 10, \dots, N$  jako

$$\text{RMSE}_2^{(s)} = \frac{1}{N-9} \sum_{n=10}^N E_n^{(s)}. \quad (5.4)$$

Analogicky s (5.3) dostaneme celkovou střední kvadratickou chybu sensorové sítě pro  $n = 10, \dots, N$  jako

$$\text{RMSE}_2 = \frac{1}{S} \sum_{s=1}^S \text{RMSE}_2^{(s)}. \quad (5.5)$$

Při vyhodnocování simulací tedy budeme porovnávat jak chybu RMSE, která zahrnuje všechny časové kroky, tak i chybu  $\text{RMSE}_2$ .



Obrázek 5.3: Ukázka průběhu chyb pro jednotlivé senzory, chyby odpovídají experimentu znázorněnému na obrázku 5.2. Vidíme, že se chyby RMSE a  $\text{RMSE}_2$  výrazně liší.

### 5.2.2. Úspěšnost sledování

S chybou sledování úzce souvisí i pojem úspěšnosti sledování. To, kdy je jeden experiment úspěšný, lze zadefinovat několika způsoby. Ve všech způsobech se porovnává určitá hodnota spjatá s chybou sledování s předem stanovenou prahovou hodnotou  $\varepsilon_{\text{RMSE}}$ .

V [10] navrhují označit experiment za úspěšný, pokud je průměrná chyba v posledním časovém kroku  $n = N$  menší než prahová hodnota tj.

$$\frac{1}{S} \sum_{s=1}^S E_N^{(s)} < \varepsilon_{\text{RMSE}}.$$

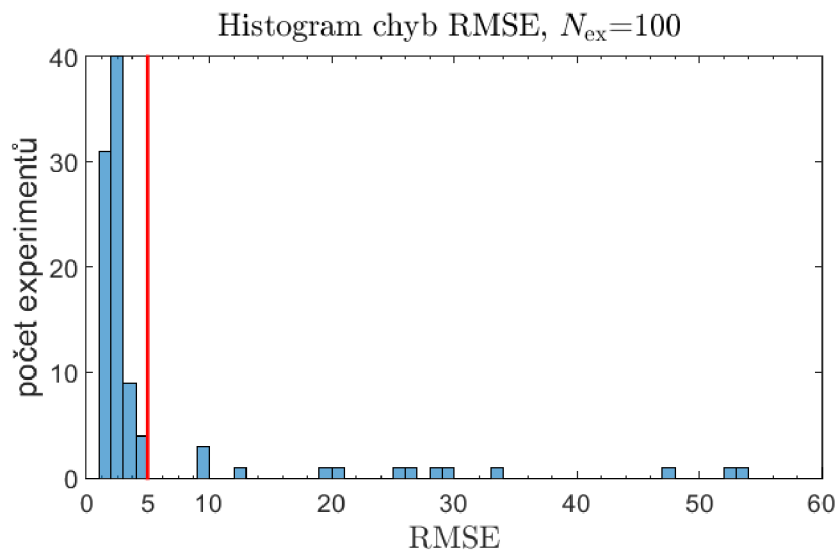
Tento způsob ovšem nezaručuje, že sledování probíhalo dobře během celé doby. Také se může stát, že hodnota průměrné chyby uskočí pouze na konci sledování, a experiment tak bude chybně označen jako neúspěšný. Další způsob, používaný např. v [25], zase porovná prahovou hodnotu s maximální průměrnou chybou během všech časových kroků, tj. experiment je úspěšný pokud

$$\max_n \frac{1}{S} \sum_{s=1}^S E_n^{(s)} < \varepsilon_{\text{RMSE}}.$$

V této práci přidáme ještě třetí možnost, podle které budeme rozlišovat úspěšný experiment. V našich simulacích pokládáme experiment za úspěšný, jestliže je průměr středních kvadratických chyb přes všechny senzory RMSE menší než prahová hodnota, tj.

$$\text{RMSE} < \varepsilon_{\text{RMSE}}. \quad (5.6)$$

Prahou hodnotu  $\varepsilon_{\text{RMSE}}$  určíme experimentálně. Vidíme, že v podmínce (5.6) figuruje průměrná chyba za celou dobu sledování. Kvůli předpokládané vyšší chybě na začátku musíme zvolit prahovou hodnotu s určitou rezervou. Ke správnému určení této hodnoty jsme použili histogramy hodnot RMSE přes experimenty  $n_{\text{ex}} = 1, \dots, N_{\text{ex}}$ , kde  $N_{\text{ex}} = 100$ . Na obrázku 5.4 vidíme příklad takového histogramu pro jednu ze simulací. Z většího množství histogramů pro různé simulace jsme určili naši prahovou hodnotu jako  $\varepsilon_{\text{RMSE}} = 5$ .



Obrázek 5.4: Příklad histogramu hodnot chyb RMSE, prahová hodnota  $\varepsilon_{\text{RMSE}} = 5$  je znázorněna červenou čarou. Experimenty, jejichž chyba se vyskytuje napravo od prahové hodnoty, označíme jako neúspěšné.

Poměr úspěšných experimentů, tj. splňujících podmínku (5.6), k celkovému počtu experimentů v simulaci budeme značit %UE. Nastavení modelu pokládáme za „funkční“, pokud pro dostatečný počet experimentů  $N_{\text{ex}}$  (tj.  $N_{\text{ex}} \geq 100$ ) platí %UE  $\geq 90\%$ .

Při vyhodnocování průměrné chyby RMSE, resp.  $\text{RMSE}_2$ , ze všech experimentů poté vyloučíme ty experimenty, které nebyly úspěšné. Označme  $\mathcal{N}_{\text{UE}} \subseteq \{1, \dots, N_{\text{ex}}\}$  množinu

## 5.2. HODNOTÍCÍ KRITÉRIA

indexů úspěšných experimentů a  $N_{\text{UE}}$  jejich počet, tj.  $N_{\text{UE}} = |\mathcal{N}_{\text{UE}}|$ . Průměrné chyby ze všech úspěšných experimentů  $\overline{\text{RMSE}}$ , resp.  $\overline{\text{RMSE}}_2$ , tedy dostaneme jako

$$\overline{\text{RMSE}} = \frac{1}{N_{\text{UE}}} \sum_{n_{\text{ex}} \in \mathcal{N}_{\text{UE}}} \text{RMSE}_{n_{\text{ex}}}, \quad (5.7)$$

resp.

$$\overline{\text{RMSE}}_2 = \frac{1}{N_{\text{UE}}} \sum_{n_{\text{ex}} \in \mathcal{N}_{\text{UE}}} \text{RMSE}_{2,n_{\text{ex}}}, \quad (5.8)$$

kde  $\text{RMSE}_{n_{\text{ex}}}$ , resp.  $\text{RMSE}_{2,n_{\text{ex}}}$ , značí příslušné střední kvadratické chyby všech senzorů pro experiment  $n_{\text{ex}}$ .

### 5.2.3. Komunikační náklady

Dalším důležitým hodnotícím kritériem jsou komunikační náklady sensorové sítě. Připomeňme, že v každém časovém kroku  $n$  provádí sensorová síť věrohodnostní konsenzus pro výpočet globálních bazových koeficientů  $\beta$ , viz část 3.5.2 (v případě kompresního věrohodnostního konsenzu pro výpočet vektoru  $\mathbf{v}$ , viz část 4.6).

Označme nyní jednotně distribuovaný vektor  $\mathbf{w} = (w_1, \dots, w_T)^\top$ , kde  $T = K$  (pro věrohodnostní konsenzus), nebo  $T = P$  (pro kompresní věrohodnostní konsenzus). V jednotlivých iteracích konsenzu  $i = 1, \dots, I$  musí každý senzor  $s$  poslat svůj aktuální odhad  $\mathbf{w}_i^{(s)} = (w_{1,i}^{(s)}, \dots, w_{T,i}^{(s)})^\top$  všem svým sousedním senzorům  $\mathcal{N}^{(s)}$ , viz komunikační spojení znázorněné na obrázku 5.1.

Zaměříme se tedy nyní na to, jakým způsobem si můžou senzory posílat tento vektor  $\mathbf{w}_i$  (v následujících řádcích vynecháme index  $^{(s)}$ ). Připomeňme, že při věrohodnostním konsenzu jsou posílané vektory řídké, a proto nemusí být výhodné posílat je „jednoduše“ celé. Komunikační náklady budeme vyjadřovat v bitech, přičemž uvažujeme, že jednotlivé prvky (čísla) distribuovaného vektoru lze popsat pomocí  $N_b = 32$  bitů (číselný datový typ „single“). V simulacích uvažujeme tři způsoby komunikace:

#### A) Výměna všech prvků

Nejjednodušší způsob, jak distribuovat vektory, je posílat celý vektor. Potom dostaneme komunikační náklady  $\text{KN}_A$  na transport jednoho vektoru  $\mathbf{w}_i$  jako

$$\text{KN}_A = TN_b. \quad (5.9)$$

#### B) Výměna nenulových prvků a indexového pole

V tomto způsobu posíláme pouze nenulové prvky vektoru společně s indexovým polem nul a jedniček o velikosti  $T$ . Pro každý nulový prvek tímto způsobem ušetříme 31 bitů (místo hodnoty posíláme pouze jeden nulový bit), zatímco nenulové prvky zabírají o jeden bit více (posíláme hodnotu a jeden jedničkový bit). Komunikační náklady  $\text{KN}_B$  vypočítáme podle

$$\text{KN}_B = T + \|\mathbf{w}_i\|_0 N_b, \quad (5.10)$$

kde aktuální řídkost vektoru  $\|\mathbf{w}_i\|_0$  označuje právě počet nenulových prvků, resp. počet jedniček indexového pole.

C) **Výměna dvojic – index prvku a jeho hodnota**

Distribuuovat budeme zřejmě tolik dvojic, jaká je aktuální řídkost  $\|\mathbf{w}_i\|_0$ . Pro popis indexu budeme potřebovat  $\lceil \log_2 T \rceil$  bitů<sup>3</sup>. Komunikační náklady  $\text{KN}_C$  v tomto případě dostaneme jako

$$\text{KN}_C = (\lceil \log_2 T \rceil + N_b) \|\mathbf{w}_i\|_0. \quad (5.11)$$

Pro názornost uveďme jednoduchý příklad. Uvažujme vektor tvaru  $\mathbf{w}_i = (3, 0, 0, -1, 0)^\top$ . Platí tedy  $T = 5$ ,  $\|\mathbf{w}_i\|_0 = 2$  a  $\lceil \log_2 T \rceil = 3$ . Výsledné komunikační náklady pro všechny tři způsoby obsahuje následující tabulka 5.2.

Tabulka 5.2: Příklad způsobů komunikace

způsob I)	posílaná data	$\text{KN}_I$ [bity]
A)	$\{(3, 0, 0, -1, 0)\}$	160
B)	$\{(3, -1); (10010)\}$	69
C)	$\{(1, 3); (4, -1)\}$	70

Vidíme, že v případě námi zvoleného vektoru  $\mathbf{w}_i$  je nejvýhodnější způsob B) – o jeden bit výhodnější než způsob C). Pro neřídké vektory, které jsou distribuovány v kompresním věrohodnostním konsenzu, je zřejmě vždy nejvýhodnější způsob A), tedy výměna všech prvků. Pro řídké vektory ve věrohodnostním konsenzu je výhodný způsob B) nebo C), a to podle relativní řídkosti  $\frac{\|\mathbf{w}_i\|_0}{T}$ . Pro menší relativní řídkost je výhodnější způsob C). Poznamenejme, že řídkost vektoru se v průběhu věrohodnostního konsenzu může zvětšovat kvůli tomu, že různé senzory mohou mít různé nenulové bázové koeficienty (tj. vektory bázových koeficientů mají různé nosiče).

V simulacích budeme počítat komunikaci přes všechny senzory  $s = 1, \dots, S$  a všechny iterace konsenzu  $i = 1, \dots, I$  a následně vypočítáme průměr přes všechny časové kroky  $n = 1, \dots, N$  pro všechny tři způsoby. Nejmenší z těchto hodnot označíme  $\min \Sigma \text{KN}$ . Stejně jako u chyby RMSE budeme zaznamenávat také průměr komunikačních nákladů od časového kroku  $n = 10$ . Nejmenší z těchto hodnot označíme  $\min \Sigma \text{KN}_2$ .

Při vyhodnocování průměrných komunikačních nákladů  $\min \Sigma \text{KN}$ , resp.  $\min \Sigma \text{KN}_2$ , ze všech experimentů opět vyloučíme ty neúspěšné. Tyto průměry pak označíme  $\overline{\min \Sigma \text{KN}}$ , resp.  $\overline{\min \Sigma \text{KN}_2}$ .

### 5.2.4. Výpočetní náročnost

Posledním hodnotícím kritériem je výpočetní (časová) náročnost modelu. U tohoto kritéria se zaměříme čistě na dobu trvání jednotlivých simulací. Průměrnou dobu jednoho časového kroku  $n$  ze všech úspěšných experimentů simulace označíme  $\bar{\tau}_n$ .

Pomocí funkce „Run and Time“ lze v MATLABu vypsát časovou náročnost použitých funkcí v simulaci. V tabulce 5.3 uvádíme seznam časově nejnáročnějších funkcí pro jeden vybraný experiment. Celková doba trvání experimentu byla 130 sekund.

Z tabulky vidíme, že časově nejnáročnější je sestavení slovníku  $\Psi$  s rozměry  $J \times K$ . Doba trvání experimentu tedy závisí nejvíce na počtu částic  $J$  a počtu atomů slovníku  $K$ . Dalším aspektem, který výrazně ovlivňuje rychlost výpočtu je použití funkce promazávání atomů.

<sup>3</sup>Symbol  $\lceil x \rceil$  značí horní celou část čísla  $x$ .

### 5.3. VÝSLEDKY

Tabulka 5.3: Výčet funkcí s největší časovou náročností

funkce	doba trvání [s]	poměr k celkové době [%]
sestavení slovníku	50,3	38,7
promazávání atomů	29,4	22,6
výpočet báz. koeficientů	20,8	16,0
výpočet LLF v částicích	8,6	6,6
aktualizace částic	8,0	6,2
ostatní	12,9	9,9

Poznamenejme, že v reálném případě by výpočty probíhaly paralelně na jednotlivých senzorech, na rozdíl od simulace na počítači. Zároveň doba trvání závisí samozřejmě na výkonu výpočetní jednotky počítače (senzorů). Získané časy  $\bar{\tau}_n$  tedy slouží pouze pro vzájemné porovnání jednotlivých simulací.

## 5.3. Výsledky

Z provedených simulací jsme vyvodili tři základní vyhodnocení: porovnání slovníku B-splajnů a Fourierova slovníku z hlediska komunikace a chyby sledování, efektivita použití komprimovaného snímání z hlediska komunikace a efektivita použití promazávání slovníku B-splajnů z hlediska chyby sledování.

### 5.3.1. Porovnání slovníků

Fourierův slovník, popsáný v části 2.3.3, je v současnosti hojně používaným slovníkem pro aproximaci funkcí ve věrohodnostním konsenzu. Slovník B-splajnů je atraktivní alternativou zvláště kvůli tomu, že jeho atomy jsou tzv. lokalizované (na rozdíl od atomů Fourierova slovníku). To znamená, že atomy jsou nenulové pouze na určité části oblasti zájmu, viz opět část 2.3.3. Jelikož jsou částice během sledování také značně lokalizované, viz např. obrázek 4.3, předpokládáme, že k aproximaci může stačit jen malý počet atomů, a tím se ušetří komunikační náklady sensorové sítě. Další možnou výhodou B-splajnů může být lepší přesnost sledování, tj. menší chyby RMSE nebo RMSE<sub>2</sub>.

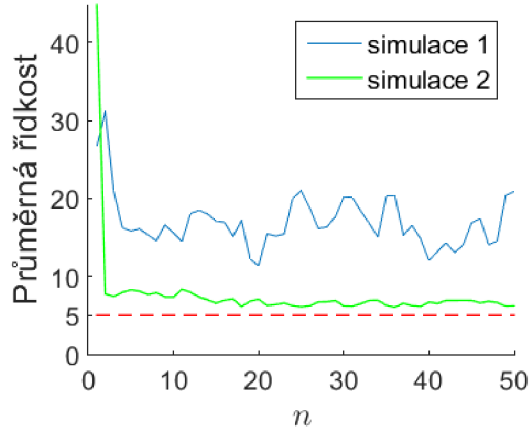
#### Test 1

Pokusíme se nalézt taková nastavení parametrů modelu, při kterých slovník B-splajnů bude efektivnější v komunikaci než Fourierův slovník se zachováním podobné chyby RMSE a RMSE<sub>2</sub>. Zvolili jsme shodně parametry:  $J = 10000$ , žádné přidané částice,  $K = 961$ ,  $K_{\text{OMP}} = 5$ ,  $I = 50$ , bez komprimovaného snímání. Výsledky jsou shrnuty v tabulce 5.4.

Tabulka 5.4: Výsledky testu 1

	slovník	promaz. atomů	$\varepsilon_a$	%UE	$\overline{\text{RMSE}}$	$\overline{\text{RMSE}}_2$
simulace 1	Fourierův	ne	–	92%	1,88	1,15
simulace 2	B-splajny	ano	1	92%	1,20	0,86
	$\min \overline{\Sigma \text{KN}}$ [kB]	$\min \overline{\Sigma \text{KN}}_2$ [kB]	$\bar{\tau}_n$ [s]			
simulace 1	122,5	119,0	13,3			
simulace 2	54,4	47,7	17,1			

Zaměříme se nyní na komunikační náklady. Vidíme, že při shodné velikosti slovníku i maximální řídkosti  $K_{OMP} = 5$  se výsledky překvapivě výrazně liší. Je to dáno tím, že v případě Fourierova slovníku jednotlivé senzory vybírají při OMP častěji odlišné koeficienty (odlišné od ostatních sensorů) a při věrohodnostním konsenzu pak narůstá řídkost distribuovaných vektorů a tím také komunikační náklady. Zato u B-splajnů vybírají senzory většinou stejné koeficienty. Popsaný jev je dobře patrný z následujícího obrázku 5.5.



Obrázek 5.5: Průměrná řídkost v závislosti na čase pro simulace 1 a 2

Ovšem zatímco pro B-splajny je tato velikost slovníku nutnou podmínkou pro zachování úspěšnosti sledování %UE, Fourierův slovník funguje dobře i s výrazně menším počtem atomů. Navíc chyba sledování, která je v tomto testu pro Fourierův slovník větší, se pro menší počet atomů zmenší, což ukazuje následující test.

## Test 2

Opět porovnáváme Fourierův slovník a B-splajny z hlediska komunikačních nákladů. Zvolili jsme shodně parametry:  $J = 10000$ , žádné přidané částice,  $I = 50$ , bez použití komprimovaného snímání. Výsledky jsou shrnuty v tabulce 5.5.

Tabulka 5.5: Výsledky testu 2

	slovník	$K$	prom. atomů	$\varepsilon_a$	$K_{OMP}$	%UE	$\overline{RMSE}$	$\overline{RMSE}_2$
simulace 3	Fourierův	121	ne	–	4	93%	1,77	0,84
simulace 2	B-splajny	961	ano	1	5	92%	1,20	0,86

	$\min \overline{\Sigma KN}$ [kB]	$\min \overline{\Sigma KN}_2$ [kB]	$\bar{\tau}_n$ [s]
simulace 3	55,7	51,3	1,5
simulace 2	54,4	47,7	17,1

Vidíme, že v těchto simulacích je komunikace srovnatelná, přičemž časová náročnost je vzhledem k velikosti slovníků výrazně menší u simulace 3. Ukázali jsme tedy, že komunikační náklady nelze se slovníkem B-splajnů výrazně ušetřit (ve srovnání s Fourierovým slovníkem). Při menší velikosti slovníku se totiž eliminuje efekt vybírání odlišných bázevých koeficientů pro různé senzory.

Všimněme si také, že oproti předcházejícímu testu jsou obě průměrné chyby pro Fourierův slovník menší, ačkoliv byl výrazně redukován počet atomů. Zatímco  $\overline{RMSE}_2$  je pro

### 5.3. VÝSLEDKY

obě simulace srovnatelná, průměrná chyba za všechny časové kroky  $\overline{\text{RMSE}}$  je poměrně výrazně menší u B-splajnů. V následujícím testu tuto chybu pro B-splajny ještě zmenšíme pomocí přidání částic.

#### Test 3 – Přidání částic

V tomto testu se zaměříme na srovnání průměrné chyby  $\overline{\text{RMSE}}$ . Komunikační a výpočetní náročnost nyní nebudeme brát za důležité, takže pro B-splajny můžeme navýšit parametry  $K$  a  $K_{\text{OMP}}$ . Zároveň pro slovník B-splajnů použijeme v prvních devíti časových krocích přidání částic kolem všech měření (i kvůli tomu je nutné zvýšit parametr  $K_{\text{OMP}}$ ). Pro obě simulace jsme zvolili shodně parametry:  $J = 10000$ ,  $I = 50$ , bez použití komprimovaného snímání. Výsledky shrnuje následující tabulka 5.6.

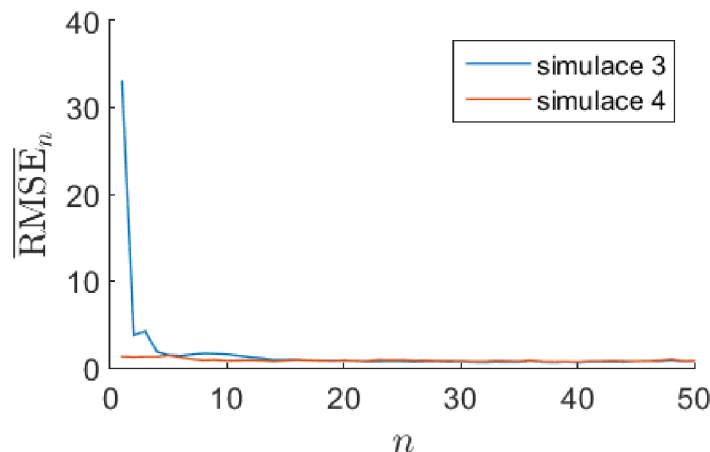
Tabulka 5.6: Výsledky testu 3 – Přidání částic

	slovník	$J_z$	$\sigma_z^2$	$K$	$K_{\text{OMP}}$	%UE	$\overline{\text{RMSE}}$	$\overline{\text{RMSE}}_2$
simulace 3	Fourierův	0	–	121	4	93%	1,77	0,84
simulace 4	B-splajny	200	$\frac{100}{9}$	1600	60	97%	0,87	0,80

	$\min \overline{\Sigma \text{KN}}$ [kB]	$\min \overline{\Sigma \text{KN}}_2$ [kB]	$\bar{\tau}_n$ [s]
simulace 3	55,7	51,3	1,5
simulace 4	494,5	59,3	35,5

Z tabulky je patrné, že průměrná chyba  $\overline{\text{RMSE}}$  je pro B-splajny podstatně menší než pro Fourierův slovník. To znamená, že na začátku sledování je model s B-splajny a s použitím přidávaných částic mnohem přesnější než Fourierův slovník. Ještě názorněji to ukazuje graf průměrné  $\text{RMSE}_n$  za všechny úspěšné experimenty  $\overline{\text{RMSE}}_n$  v závislosti na čase  $n$ , viz obrázek 5.6.



Obrázek 5.6: Výsledky testu 3 – průměrná chyba v závislosti na čase

Pokud tedy požadujeme přesné sledování objektu od samotného začátku, je vhodné použít k aproximaci slovník B-splajnů a zároveň přidat (během několika prvních časových kroků) částice kolem každého měření sensorů. Cenou za vysokou přesnost jsou však obrovské komunikační náklady.

Důležité je poznamenat, že pro Fourierův slovník nelze použít tyto přidávané částice. Připomeňme, že ze všech měření sensorů nejvýše jedno odpovídá sledovanému objektu,



ostatní jsou tzv. falešná měření. Tato měření jsou na náhodných místech, a tak různé senzory mají falešná měření na různých pozicích. Díky tomu, že jsou atomy B-splajnů lokalizované, ve věrohodnostním konsenzu senzory „poznají“ měření odpovídající objektu. Přesněji, po provedení konsenzu budou největší hodnoty bazových koeficientů právě ty, které odpovídají pozici sledovaného objektu, protože v místech, kde má jeden senzor falešná měření, nemají ostatní senzory pravděpodobně žádné měření.

To ovšem neplatí pro Fourierův slovník, u kterého falešná měření ovlivní tvar aproximace log-LLF na celé oblasti zájmu, nejenom na pozici měření. Pro potvrzení této úvahy jsme provedli jednu simulaci s Fourierovým slovníkem a s přidanými částicemi. Simulace měla podle předpokladů nulovou úspěšnost sledování %UE.

### 5.3.2. Použití komprimovaného snímání

Dalším způsobem, jak ušetřit komunikační náklady senzorové sítě, může být použití komprimovaného snímání, viz části 2.4 a 4.6. Všechna ostatní hodnotící kritéria mohou v nejlepším případě zůstat stejná jako při nastavení bez komprimovaného snímání, neboť principem této metody je pouze komprese původních bazových koeficientů a následná rekonstrukce.

Nejdůležitějším parametrem komprimovaného snímání je počet řádků kompresní matice  $P$ , který odpovídá dimenzi distribuovaného neřídkého vektoru. Sensorová síť totiž provádí právě  $P$  konsenzů, a komunikační náklady jsou tedy lineárně závislé na hodnotě  $P$ . Ovšem naopak pro úspěšnou rekonstrukci, která je klíčová pro fungování metody, je potřebný určitý minimální počet vzorků  $P$ .

#### Test 4 – Komprimované snímání pro Fourierův slovník

Nejdříve uvedeme test pro Fourierův slovník. Jako výchozí simulaci pro porovnání jsme zvolili simulaci 3 z předchozích testů. Parametry shodné pro všechny simulace jsou:  $J = 10000$ , žádné přidané částice, Fourierův slovník,  $K = 121$ , bez použití promazávání,  $I = 50$ , a  $K_{OMP} = 4$ . Výsledky jsou uvedeny v následující tabulce 5.7.

Tabulka 5.7: Výsledky testu 4 – Komprimované snímání pro Fourierův slovník

	kompr. snímání	$P$	%UE	$\overline{\text{RMSE}}$	$\overline{\text{RMSE}}_2$
simulace 3	ne	–	93%	1,77	0,84
simulace 5	ano	30	8%	3,93	1,79
simulace 6	ano	40	55%	3,26	1,28
simulace 7	ano	50	84%	2,41	1,04
simulace 8	ano	60	88%	2,05	0,96

	$\min \overline{\Sigma \text{KN}}$ [kB]	$\min \overline{\Sigma \text{KN}}_2$ [kB]	$\bar{\tau}_n$ [s]
simulace 3	55,7	51,3	1,5
simulace 5	164,1	164,1	1,5
simulace 6	218,8	218,8	1,5
simulace 7	273,4	273,4	1,5
simulace 8	328,1	328,1	1,6

### 5.3. VÝSLEDKY

Vidíme, že rekonstrukce začíná být úspěšná již pro  $P = 30$ . Ovšem abychom získali úspěšnost sledování alespoň 80%, je nutné nastavit  $P$  minimálně na hodnotu 50. V simulaci 8, kde  $P = 60$ , již dostáváme srovnatelné hodnoty průměrných chyb  $\overline{\text{RMSE}}$  a  $\overline{\text{RMSE}}_2$  jako u simulace 3 bez komprimovaného snímání.

Avšak když se zaměříme na komunikační náklady, zjišťujeme, že metoda komprimovaného snímání nenaplnila naše očekávání. Ačkoliv jsme snížili dimenzi distribuovaného vektoru z  $K = 121$  na  $P = 60$ , kvůli principiální neřídkosti komprimovaného vektoru není možné komunikační náklady snížit. Je to dáno tím, že způsoby posílání řídkých dat, které jsme uvedli v části 5.2.3, jsou velmi úsporné. Pokud bychom i v simulaci 3 použili základní způsob posílání dat, tj. distribuce celých vektorů, vyšly by celkové komunikační náklady 661,7 kB. Při použití úsporného způsobu tyto náklady klesly na 55,7 kB, resp. 51,3 kB.

#### Test 5 – Komprimované snímání pro slovník B-splajnů

Uveďme ještě pro úplnost test s komprimovaným snímáním pro B-splajny. Jako výchozí simulaci pro porovnání jsme zvolili simulaci 2 z předchozích testů. Parametry shodné pro všechny simulace jsou:  $J = 10000$ , žádné přidané částice, slovník B-splajnů,  $K = 961$ , s použitím promazávání atomů,  $\varepsilon_a = 1$ ,  $I = 50$ , a  $K_{\text{OMP}} = 5$ . Výsledky jsou uvedeny v následující tabulce 5.8.

Tabulka 5.8: Výsledky testu 5 – Komprimované snímání pro slovník B-splajnů

	kompr. snímání	$P$	%UE	$\overline{\text{RMSE}}$	$\overline{\text{RMSE}}_2$
simulace 2	ne	–	92%	1,20	0,86
simulace 9	ano	80	56%	1,69	0,92

	min $\Sigma\text{KN}$ [kB]	min $\Sigma\text{KN}_2$ [kB]	$\bar{\tau}_n$ [s]
simulace 2	54,4	47,7	17,1
simulace 9	437,5	437,5	17,7

Z výsledků je patrné, že ani v případě B-splajnů nešetří komprimovaného snímání komunikační náklady senzorové sítě. Už pro  $P = 80$  jsou tyto náklady několikanásobně vyšší než u simulace 2, přitom tato hodnota není postačující pro dobrou úspěšnost sledování.

Můžeme tedy říci, že využití komprimovaného snímání nepřináší žádné výhody, pokud senzory dokáží posílat řídká data úsporně.

#### 5.3.3. Efektivita promazávání slovníku B-splajnů

V části 4.5 jsme krátce uvedli, jaký je důvod pro promazávání slovníku při použití B-splajnů. Pro níže uvedené testy jsme se uchýlili pouze k testování promazávání celých atomů, neboť promazávání jednotlivých prvků slovníku oproti tomu nepřinášelo žádné vylepšení.

#### Test 6

Nejdříve otestujeme promazávání v případě, kdy nepoužijeme přidané částice. Jako výchozí simulaci zvolíme opět simulaci 2. Parametry shodné pro obě simulace jsou:  $J = 10000$ , žádné přidané částice, slovník B-splajnů,  $K = 961$ ,  $I = 50$ , a  $K_{\text{OMP}} = 5$ . Výsledky jsou uvedeny v následující tabulce 5.9.

Tabulka 5.9: Výsledky testu 6

	promaz. atomů	$\varepsilon_a$	%UE	$\overline{\text{RMSE}}$	$\overline{\text{RMSE}}_2$
simulace 2	ano	1	92%	1,20	0,86
simulace 10	ne	–	91%	1,37	0,91

	min $\Sigma\text{KN}$ [kB]	min $\Sigma\text{KN}_2$ [kB]	$\bar{\tau}_n$ [s]
simulace 2	54,4	47,7	17,1
simulace 10	54,7	48,0	17,1

Vidíme, že v případě použití promazávání atomů dostáváme o něco menší průměrné chyby sledování  $\overline{\text{RMSE}}$  a  $\overline{\text{RMSE}}_2$ , nicméně rozdíly nejsou výrazné.

### Test 7

V tomto testu provedeme simulace s použitím přidanych částic v prvních devíti časových krocích. Parametry shodné pro obě simulace jsou:  $J = 10000$ ,  $J_z = 200$ ,  $\sigma_z^2 = 11,11$ , slovník B-splajnů,  $K = 1600$ ,  $I = 50$ ,  $K_{\text{OMP}} = 60$ . Výsledky shrnuje tabulka 5.10.

Tabulka 5.10: Výsledky testu 7

	promaz. atomů	$\varepsilon_a$	%UE	$\overline{\text{RMSE}}$	$\overline{\text{RMSE}}_2$
simulace 4	ano	1	97%	0,87	0,80
simulace 11	ne	–	58%	1,41	0,90

	min $\Sigma\text{KN}$ [kB]	min $\Sigma\text{KN}_2$ [kB]	$\bar{\tau}_n$ [s]
simulace 4	494,5	59,3	35,5
simulace 11	545,1	68,0	34,6

Z výsledků testu vidíme, že promazávání atomů má velmi pozitivní vliv na úspěšnost a přesnost měření. V tomto nastavení je dokonce nutné použít promazávání, abychom dosáhli potřebné úspěšnosti sledování (nad 90%).

Lze tedy říci, že pokud model využívá aproximaci pomocí B-splajnů, je prospěšné využít promazávání atomů slovníku. Jak již bylo zmíněno v části 4.5, vhodnou prahovou hodnotu  $\varepsilon_a$  je nutné určit experimentálně.

## 6. Závěr

Tato diplomová práce se zabývala modelem decentralizované sensorové sítě sledující pohyb jednoho objektu. Prvním cílem bylo seznámení se s problematikou a s dodaným simulačním programem. Hlavním cílem bylo do tohoto programu implementovat nové metody a postupy a porovnat je zejména z hlediska přesnosti sledování a komunikačních nákladů.

Obecná teorie distribuovaných částicových filtrů, které jsou základem pro matematický model sensorové sítě, byla uvedena v kapitole 3. V následující kapitole 4 byl již popsán konkrétní model, na kterém byla provedena řada simulací. Byl navržen nový typ slovníku pro aproximaci logaritmu lokální věrohodnostní funkce – tzv. bázové splajny, neboli B-splajny. Vzhledem k řídké reprezentaci této aproximace bylo dále navrženo použití komprimovaného snímání v tzv. kompresním věrohodnostním konsenzu pro úsporu komunikace mezi senzory. Další nové metody spočívaly v přidávání částic kolem všech měření sensorů a také v promazávání slovníku.

Výsledky simulací jsou shrnuty v kapitole 5. Všechny simulace byly provedeny pro uvedenou sensorovou síť, skládající se z deseti sensorů, a pro jednu testovací trajektorii sledovaného objektu. Nejdříve byly porovnány dva typy slovníků: B-splajny a Fourierův slovník. Bylo zjištěno, že s pomocí Fourierova slovníku získáváme dobrou přesnost sledování i pro velmi malý počet atomů, což pro B-splajny není možné. Pokud ovšem senzory nelimituje výpočetní náročnost, použití B-splajnů dává menší chybu na začátku sledování. Jestliže navíc připustíme vysoké komunikační náklady, má model s B-splajny společně s přidávanými částicemi v několika prvních časových krocích ještě lepší přesnost sledování.

Simulace s kompresním věrohodnostním konsenzem ukázaly, že metoda komprimovaného snímání není vhodná pro úsporu komunikace v sensorových sítích. Důvodem jsou vysoké úspory při distribuci řídkých vektorů během věrohodnostního konsenzu bez komprese.

Nakonec bylo ukázáno, že promazávání atomů pro slovník B-splajnů má pozitivní vliv na přesnost měření. V případě použití přidávaných částic dokonce i na celkovou úspěšnost sledování.

Na práci je možné navázat několika způsoby. Lze provést simulace na jiných zkušebních trajektoriích. Dále se mohou hledat jiné scénáře (tj. počet sensorů a jejich parametry, případně parametry pohybu objektu), při kterých mohou vycházet výsledky simulací odlišně, např. ve větší prospěch splajnů. Dalším rozšířením může být aplikování komprimovaného snímání přímo na vektor funkce log-LLF vyhodnocené v částicích  $\eta^{(s)}$ , viz část 3.5.3.

Součástí práce je přiložené CD, na kterém jsou k dispozici skripty a funkce v prostředí MATLABu, které byly použity pro simulace modelu. Výčet těchto souborů, jejich krátký popis a seznam vybraných proměnných použitých v kódech lze nalézt v přílohách.

## 7. Seznam použitých zdrojů

### Literatura

- [1] ANDĚL, J. *Matematická statistika*. 2. vyd. Praha: SNTL, 1978, 346 s.
- [2] ARULAMPALAM, M.S., S. MASKELL, N. GORDON a T. CLAPP. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*. 2002, **50**(2), 174-188. DOI: 10.1109/78.978374. ISSN 1053587X. Dostupné také z: <http://ieeexplore.ieee.org/document/978374/>
- [3] ČERMÁK, L. a R. HLAVIČKA. *Numerické metody*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2008. ISBN 978-80-214-3752-4.
- [4] DAŇKOVÁ, M. *Komprimované snímání v perfuzním zobrazování pomocí magnetické rezonance*. Diplomová práce, Vysoké učení technické v Brně, 2014.
- [5] DUARTE, M. F., M. A. DAVENPORT, D. TAKHAR, J. N. LASKA, T. SUN, K. F. KELLY a R. G. BARANIUK. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*. 2008, **25**(2), 83-91. DOI: 10.1109/MSP.2007.914730. ISSN 1053-5888. Dostupné také z: <http://ieeexplore.ieee.org/document/4472247/>
- [6] ELAD, M. *Sparse and redundant representations: from theory to applications in signal and image processing*. New York: Springer, 2010. ISBN 978-1-4419-7010-7.
- [7] FRIESL, M. Bayesovské odhady v některých modelech. In: *Analýza dat 2004 II*. Pardubice, 2004, s. 21-33. ISBN 80-239-4847-4.
- [8] GROSS, J.L. a J. YELLEN. *Handbook of graph theory*. Boca Raton: CRC Press, c2004. ISBN 15-848-8090-2.
- [9] HAUG, A. J. *Bayesian estimation and tracking: a practical guide*. Hoboken, N.J.: Wiley, c2012. ISBN 978-0-470-62170-7.
- [10] HLINKA, O., F. HLAWATSCH a P. M. DJURIC. Distributed particle filtering in agent networks: A survey, classification, and comparison. *IEEE Signal Processing Magazine*. 2013, **30**(1), 61-81. DOI: 10.1109/MSP.2012.2219652. ISSN 1053-5888. Dostupné také z: <http://ieeexplore.ieee.org/document/6375933/>
- [11] HLINKA, O., F. HLAWATSCH a P. M. DJURIC. Consensus-based Distributed Particle Filtering With Distributed Proposal Adaptation. *IEEE Transactions on Signal Processing*. 2014, **62**(12), 3029-3041. DOI: 10.1109/TSP.2014.2319777. ISSN 1053-587X. Dostupné také z: <http://ieeexplore.ieee.org/document/6804018/>
- [12] HRBÁČEK, R., P. RAJMIC, V. VESELÝ a J. ŠPIŘÍK. Řídké reprezentace signálů: úvod do problematiky. *Elektrorevue*. Brno, 2011, **13**(5), 1-10. ISSN 1213-1539. Dostupné také z: <http://elektrorevue.cz/cz/download/ridke-reprezentace-signalu-uvod-do-problematiky/>

## LITERATURA

- [13] HRBÁČEK, R., P. RAJMIC, V. VESELÝ a J. ŠPIŘÍK. Řídké reprezentace signálů: komprimované snímání. *Elektrorevue*. 2011, **13**(6), 1-8. ISSN 1213-1539. Dostupné také z: <http://elektrorevue.cz/cz/download/ridke-reprezentace-signalu-komprimovane-snimani/>
- [14] HUŠKOVÁ, M. *Bayesovské metody*. Praha: SPN, 1985.
- [15] KARÁSEK, J a L. SKULA. *Lineární algebra: cvičení*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3000-1.
- [16] KARPÍŠEK, Z. *Matematika IV: statistika a pravděpodobnost*. 3., dopl. vyd. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-802-1433-809.
- [17] MEYER, F., T. KROPFREITER, J. L. WILLIAMS, R. LAU a F. HLAWATSCH. *Proceedings of the IEEE*. 2018, **106**(2). DOI: 10.1109/JPROC.2018.2789427. ISSN 0018-9219. Dostupné také z: <http://ieeexplore.ieee.org/document/8290605/>
- [18] NEEDELL, D. a J.A. TROPP. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*. 2009, **26**(3), 301-321. DOI: 10.1016/j.acha.2008.07.002. ISSN 10635203. Dostupné také z: <https://linkinghub.elsevier.com/retrieve/pii/S1063520308000638>
- [19] OPPENHEIM, A.V., A.S. WILLSKY a S.H. NAWAB. *Signals and Systems*. 2nd ed. Upper Saddle River, N.J.: Prentice Hall, c1997. ISBN 978-0138147570.
- [20] PRÁŠKOVÁ, Z. *Základy náhodných procesů II*. 2., upravené vydání. Praha: Univerzita Karlova, nakladatelství Karolinum, 2016. ISBN 978-802-4635-163.
- [21] SEDLÁČEK, J. *Úvod do teorie grafů*. 2. vyd. Praha: Academia, 1977, 227 s.
- [22] RAJMIC, P. *Řídké a nízkohodnostní reprezentace signálů s aplikacemi*. Brno, 2014, 154 s. Habilitační práce. Vysoké učení technické v Brně.
- [23] RAJMIC, P., F. MEYER, G. PAPA a F. HLAWATSCH. *Distributed Object Tracking with Sparsity-Exploiting Clutter Rejection*. Nепublikovaná zpráva.
- [24] RÉNYI, A. *Teorie pravděpodobnosti*. Praha: Academia, 1972, 511 s.
- [25] REPP, R., P. RAJMIC, F. MEYER a F. HLAWATSCH. Target Tracking Using a Distributed Particle-Pda Filter With Sparsity-Promoting Likelihood Consensus. *2018 IEEE Statistical Signal Processing Workshop (SSP)*. Freiburg, Germany: IEEE, 2018, (June), 653-657. DOI: 10.1109/SSP.2018.8450815. ISBN 978-1-5386-1571-3. Dostupné také z: <https://ieeexplore.ieee.org/document/8450815/>
- [26] REPP, R., G. PAPA, F. MEYER, P. BRACA a F. HLAWATSCH. A Distributed Bernoulli Filter Based on Likelihood Consensus with Adaptive Pruning. *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018, 2018, (July), 1-5. DOI: 10.23919/ICIF.2018.8455302. ISBN 978-0-9964527-6-2. Dostupné také z: <https://ieeexplore.ieee.org/document/8455302/>

## LITERATURA

- [27] UNSER, M. Splines: a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*. 1999, **16**(6), 22-38. DOI: 10.1109/79.799930. ISSN 10535888. Dostupné také z: <http://ieeexplore.ieee.org/document/799930/>
- [28] L. XIAO, S. BOYD a S. LALL. A scheme for robust distributed sensor fusion based on average consensus. *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*. IEEE, 2005, 63-70. DOI: 10.1109/IPSN.2005.1440896. ISBN 0-7803-9201-9. Dostupné také z: <http://ieeexplore.ieee.org/document/1440896/>
- [29] ZVÁRA, K. a J. ŠTĚPÁN. *Pravděpodobnost a matematická statistika*. Praha: Matfyzpress, 1997. ISBN 80-858-6324-3.

## 8. Seznam použitých zkratek a symbolů

MNČ	metoda nejmenších čtverců
OMP	Orthogonal Matching Pursuit
LLF	local likelihood function – lokální věrohodnostní funkce $f(\mathbf{z}_n^{(s)} \mathbf{x}_n)$
GLF	global likelihood function – globální věrohodnostní funkce $f(\mathbf{z}_n \mathbf{x}_n)$
log-LLF	logaritmus lokální věrohodnostní funkce $\ln f(\mathbf{z}_n^{(s)} \mathbf{x}_n)$
$\mathbb{N}$	množina přirozených čísel (v této práci včetně 0)
$\mathbb{R}$	množina reálných čísel
$\mathbb{R}_+$	množina kladných reálných čísel
$\mathbf{A}^+$	pseudoinverzní matice k matici $\mathbf{A}$
$\mathbf{A}^*$	hermitovská transpozice matice $\mathbf{A}$
$\mathbf{v}^\top, \mathbf{A}^\top$	transpozice vektoru $\mathbf{v}$ , resp. matice $\mathbf{A}$
$\ \mathbf{x}\ _p$	$l_p$ -norma vektoru $\mathbf{x}$
$\ \mathbf{x}\ _0$	řídkost vektoru $\mathbf{x}$
$\mathbf{x}_n$	stavový vektor v čase $n$
$\bar{\mathbf{x}}_n$	poloha v čase $n$
$\hat{\mathbf{x}}_n^{(s)}$	lokální odhad stavového vektoru $\mathbf{x}_n$ senzoru $s$ v čase $n$
$\mathbf{z}_n$	vektor měření všech senzorů v čase $n$
$\boldsymbol{\alpha}^{(s)}$	vektor lokálních bazových koeficientů senzoru $s$
$\hat{\boldsymbol{\beta}}^{(s)}$	lokální odhad vektoru globálních bazových koeficientů senzoru $s$
$\mathbf{u}^{(s)}$	lokální komprimovaný vektor senzoru $s$
$s$	index senzoru
$S$	počet senzorů
$\bar{\mathbf{x}}^{(s)}$	pozice senzoru
$\sigma_r^2$	rozptyl měřené vzdálenosti
$\sigma_\rho^2$	rozptyl měřeného úhlu
$P_d$	pravděpodobnost detekce
$\mu$	stř. hodnota počtu falešných měření
$R$	dosah měření senzoru
$\mathcal{N}^{(s)}$	množina sousedních senzoru



## 8. SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

$J$	počet částic
$J_z$	počet přidaných částic
$\sigma_z^2$	rozptyl přidaných částic
$K$	velikost slovníku
$K_{\text{OMP}}$	maximální řídkost
$\varepsilon_a$	prahová hodnota pro promaz. atomů
$\varepsilon_v$	prahová hodnota pro promaz. prvků
$I$	počet iterací konsenzů
$P$	počet řádků kompresní matice
$E_n^{(s)}$	chyba sledování senzoru $s$ v čase $n$
$\text{RMSE}^{(s)}$	střední kvadratická chyba senzoru $s$
$\text{RMSE}$	celková střední chyba sensorové sítě
$\text{RMSE}_2^{(s)}$	střední kvadratická chyba senzoru $s$ od času $n = 10$
$\text{RMSE}_2$	celková střední chyba sensorové sítě od času $n = 10$
$\overline{\text{RMSE}}$	průměr z chyb $\text{RMSE}$ ze všech úspěšných experimentů
$\overline{\text{RMSE}}_2$	průměr z chyb $\text{RMSE}_2$ ze všech úspěšných experimentů
$\% \text{UE}$	poměr úspěšných experimentů
$\min \overline{\Sigma \text{KN}}$	nejmenší průměr komunikačních nákladů z úspěšných experimentů
$\min \overline{\Sigma \text{KN}}_2$	nejmenší průměr komunikačních nákladů od času $n = 10$
$\bar{\tau}_n$	průměrná doba jednoho časového kroku $n$ z úspěšných experimentů

# 9. Přílohy

## Obsah příloženého CD

Příložené CD obsahuje všechny potřebné skripty a funkce v jazyce MATLAB, dále soubor `Results.xlsx` s výsledky simulací popsaných v kapitole 5 i s výsledky jiných simulací, které nebyly v této práci zmíněny. Složka „Simulations“, obsahující výsledky simulací ve formátu `.mat`, je vzhledem k velikosti dostupná pouze v příloženém CD. V elektronické verzi práce je v souboru `readme.txt` odkaz ke stažení této složky z internetového disku. Níže uvádíme výčet všech souborů.

### hlavní složka CD

<code>Database1 – 10sensors.mat</code>	databáze pro simulace s deseti senzory, viz kapitola 5
<code>Database2 – 4sensors.mat</code>	jiná databáze pro simulace se čtyřmi senzory
<code>Demo_lcFilter.m</code>	skript pro spuštění jednoho experimentu
<code>readme.txt</code>	soubor s odkazem na stažení složky „Simulations“
<code>Results.xlsx</code>	excel soubor s výsledky simulací
<code>Test.m</code>	skript pro spuštění simulace/simulací

### podložka `Main_Functions`

Obsahuje potřebné funkce pro simulační skripty `Demo_lcFilter.m` a `Test.m`.

<code>clearAtoms.m</code>	funkce pro promazání atomů slovníku, viz část 4.5
<code>clearValues.m</code>	funkce pro promazání prvků slovníku, viz část 4.5
<code>computeAproxLogLLF.m</code>	funkce pro výpočet aproximace log-LLF pro pozdější vykreslení
<code>computeBasisCoefficients.m</code>	funkce pro výpočet lokálních bázevých koeficientů
<code>computeConsensusMatrix.m</code>	funkce pro realizaci výměny dat při konsenzu
<code>computeLogLikelihood.m</code>	funkce pro výpočet log-LLF (4.11) v částicích
<code>computeLogLikelihoodMesh.m</code>	funkce pro výpočet log-LLF (4.11) pro pozdější vykreslení
<code>fourierBasis.m</code>	funkce pro sestavení (Fourierova) slovníku (3.33)
<code>generateMeasurement.m</code>	funkce pro generování jednoho měření (4.2)
<code>generateMeasurementSequence.m</code>	funkce pro generování všech měření
<code>generateStateSequence.m</code>	funkce pro generování stavového vektoru sledovaného objektu (4.1)
<code>getErrorSuccess.m</code>	funkce pro vyhodnocení výsledků simulace: $\overline{RMSE}$ , $\overline{RMSE}_2$ , $\mathcal{N}_{UE}$ , %UE
<code>getHistogram.m</code>	funkce pro vykreslení nebo uložení histogramu chyb, viz obrázek 5.4
<code>getinfo.m</code>	funkce pro textové uložení parametrů simulace
<code>lcFilter.m</code>	funkce částicového filtru pro skript <code>Demo_lcFilter.m</code> – jádro výpočtu, viz část 4.7

lcFilter_Test.m	funkce částicového filtru pro skript Test.m – jádro výpočtu, viz část 4.7
lcUpdate.m	funkce pro aktualizaci částic a vah, viz část 4.7
measurementFunction.m	funkce pro výpočet (4.6)
omp.m	funkce algoritmu OMP, viz část 2.3.4
plotLLF_All.m	funkce pro zobrazení GUI vykreslující log-LLF a jejich aproximace během experimentu
PlotTrackingInTime.m	funkce pro vykreslování sledování objektu během experimentu
predict.m	funkce pro predikci částic, viz část 4.4
resample.m	funkce pro převzorkování částic, viz část 4.4
samplePrior.m	funkce pro generování částic apriorní hustoty pravděpodobnosti, viz část 4.4
splineBasis.m	funkce pro sestavení slovníku (B-splajnů) (3.33)
testing_demo_lcFilter.m	funkce pro provedení jedné simulace
wrapTo180.m	funkce pro vyjádření úhlu v intervalu $(-180^\circ, 180^\circ)$

#### podslůžka Other Matlab Files

Obsahuje pomocné skripty a funkce, které byly použity pro analyzování výsledků.

AnalyseBasisCoefficients.m	funkce k analyzování lokálních bázevých koeficientů pro jeden experiment
AnalyseErrorParticles.m	funkce k analyzování reziduí aproximace
ConvolutionOfSplines.m	skript pro ukázkou generování B-splajnů pomocí konvoluce
GenerateDatabase.m	skript pro generování databáze potřebné pro simulace
PlotAtom.m	funkce pro vykreslení atomů báze
PlotAverageTrackingError.m	funkce pro vykreslení průměrné chyby, viz obrázek 5.6
PlotAxisAtOrigin.m	pomocná funkce pro vykreslení Fourierova 1-D slovníku
PlotDictionary.m	skript pro vykreslení celého slovníku
PlotFourierBasis1D.m	skript pro vykreslení Fourierova 1-D slovníku
PlotOneAtom.m	funkce pro vykreslení jednoho atomu báze
PlotTrackAndSensorLinks.m	skript pro vykreslení senzorové sítě a zkušební trajektorie, viz obrázek 5.1
plotTracking.m	funkce pro vykreslení výsledku sledování pro jeden experiment

#### podslůžka Simulations

Obsahuje uložené výsledky vybraných simulací ve formátu .mat.

## Seznam vybraných proměnných

Tabulka 9.4: Seznam vybraných proměnných

proměnná	parametr	popis
surveillanceWidth	–	parametr velikosti oblasti zájmu
Prior	–	typ apriorní hustoty pravděp.
drivingNoiseCovariance	$\Sigma_{\mathbf{u}}$	var. matice budícího šumu
nAgents	$S$	počet senzorů
position	$\bar{\mathbf{x}}^{(s)}$	pozice senzorů
rangeVariance	$\sigma_r^2$	rozptyl měřené vzdálenosti
bearingVariance	$\sigma_\rho^2$	rozptyl měřeného úhlu
detectionProbability	$P_d$	pravděpodobnost detekce
clutterMean	$\mu$	stř. hodnota počtu falešných měření
maxRange	$R$	dosah měření senzorů
neighbors	$\mathcal{N}^{(s)}$	množina sousedních senzorů
nTimeSteps	$N$	počet časových kroků
nParticles	$J$	počet částic
basisType	–	typ slovníku
nBasisParameter	–	parametr pro výpočet $K$
nBasisFunctions	$K$	velikost slovníku
nConsensusIterations	$I$	počet iterací konsenzů
basisSparsity	$K_{\text{OMP}}$	maximální řídkost
stateSequence	$\mathbf{x}_n$	stavový vektor
measurementSequence	$\mathbf{z}_n$	vektor měření
estimatedStateSequence	$\hat{\mathbf{x}}_n^{(s)}$	lokální odhad stavu
basisCoefficients	$\boldsymbol{\alpha}^s$	lokální bázové koeficienty
basisCoefficients_global	$\hat{\boldsymbol{\beta}}^{(s)}$	odhad globálních bázových koeficientů
compressed_vector	$\mathbf{u}^{(s)}$	komprimovaný vektor
ClearAtom	–	promazávání atomů slovníku
thresAtom	$\varepsilon_a$	prahová hodnota pro promaz. atomů
ClearValue	–	promazávání prvků slovníku
thresValue	$\varepsilon_v$	prahová hodnota pro promaz. prvků
MeasurementParticles	–	přidávání částic
nParticlesMeasurement	$J_z$	počet přidávaných částic
varParticles	$\sigma_z^2$	rozptyl přidávaných částic
Compression	–	použití komprimovaného snímání
P	$P$	počet řádků kompresní matice
successThres	$\varepsilon_{\text{RMSE}}$	prahová hodnota úspěšných experimentů
average_error_successful	$\overline{\text{RMSE}}$	prům. chyba ze všech úsp. experimentů
average_error_successful2	$\overline{\text{RMSE}}_2$	prům. chyba ze všech úsp. experimentů 2
percentage_successful	$\% \text{UE}$	poměr úsp. experimentů
index_successful	$\mathcal{N}_{\text{UE}}$	množina indexů úsp. experimentů
tElapsed	$\bar{\tau}_n$	průměrná doba jednoho čas. kroku
CommCostsBasic	–	průměrná komunikace způsobem A)
CommCostsSparse	–	průměrná komunikace způsobem B)
CommCostsPair	–	průměrná komunikace způsobem C)