

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

OPTIMÁLNÍ METODY VÝMĚNY ŘÍDKÝCH DAT V SENZOROVÉ SÍTI

OPTIMAL METHODS FOR SPARSE DATA EXCHANGE IN SENSOR NETWORKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Alena Valová

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. Pavel Rajmic, Ph.D.

BRNO 2017

Zadání diplomové práce

Ústav: Ústav matematiky
Studentka: **Bc. Alena Valová**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Matematické inženýrství
Vedoucí práce: **doc. Mgr. Pavel Rajmic, Ph.D.**
Akademický rok: 2016/17

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

Optimální metody výměny řídkých dat v senzorové síti

Stručná charakteristika problematiky úkolu:

Práce se bude zabývat analýzou a optimalizací metod pro výměnu dat v senzorové síti. Konkrétněji půjde o decentralizovanou, často "ad hoc" síť, která bude mít za úkol sledovat polohu pohybujícího se objektu. Všechny senzory musejí mít v každém okamžiku dostatečné informace o poloze objektu a k jejímu zpřesnění je nutné vyměňovat relativně velké množství dat. Naštěstí o datech víme, že jsou tzv. řídká, což umožňuje přímou či nepřímou úsporu pomocí komprese dat. V přímém případě to bude realizováno pomocí metody komprimovaného snímání (compressed sensing).

Cíle diplomové práce:

Studentka se seznámí s modelováním senzorových sítí a výměny dat a distribuovanými výpočty v nich. Studentka po diskuzi s vedoucím navrhne metody pro výměnu řídkých dat v síti a způsob, jakým se budou metody porovnávat. Metody budou mj. porovnávány a optimalizovány z hlediska energetické náročnosti výpočtu, energetické náročnosti spojení a výměny dat. Součástí práce budou programy vytvořené v jazyce MATLAB.

Seznam literatury:

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

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

ŠPIŘÍK, J., RAJMIC, P. a VESELÝ, V. Reprezentace signálů: od bází k framům, Elektrevue. 2010. ISSN 1213-1539.

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

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

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

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

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

Abstrakt

Práce je zaměřena na sledování pohybu objektu decentralizovanou sensorovou sítí s využitím distribuovaného částicového filtru s fúzním centrem i s konsenzem. Model zahrnuje šum v měření sensorů i případy, kdy sensor objekt vůbec nezachytí. Uvedený přístup využívá řídkosti globální věrohodnostní funkce, jejíž vhodnou řídkou aproximací a vhodnou volbou slovníku lze významně snížit nároky na komunikaci v decentralizované sensorové síti. Diplomová práce obsahuje návrh metod výměny řídkých dat v sensorové síti a porovnání navržených metod z hlediska přesnosti a energetické náročnosti.

Summary

This thesis is focused on object tracking by a decentralized sensor network using fusion center-based and consensus-based distributed particle filters. The model includes clutter as well as missed detections of the object. The approach uses sparsity of global likelihood function, which, by means of appropriate sparse approximation and the suitable dictionary selection can significantly reduce communication requirements in the decentralized sensor network. The master's thesis contains a design of exchange methods of sparse data in the sensor network and a comparison of the proposed methods in terms of accuracy and energy requirements.

Klíčová slova

Matlab, sensorová síť, částicový filtr, postupné Bayesovské odhady, řídkost, OMP, MNČ

Keywords

Matlab, sensor network, particle filter, sequential Bayes estimation, sparsity, OMP, LS

VALOVÁ, A. *Optimální metody výměny řídkých dat v sensorové síti*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 64 s. Vedoucí doc. Mgr. Pavel Rajmic, Ph.D.

Prohlašuji, že jsem diplomovou práci „Optimální metody výměny řídkých dat v senzorové síti“ vypracovala samostatně a s použitím zdrojů uvedených v seznamu literatury.

Bc. Alena Valová

Děkuji svému vedoucímu panu doc. Mgr. Pavlu Rajmicovi, Ph.D. za trpělivost, věnovaný čas a odborné vedení. Dále děkuji svým spolužákům z Matematického inženýrství za jejich cenné rady a připomínky. V neposlední řadě děkuji i své rodině za vstřícnost, podporu a pochopení.

Bc. Alena Valová

Obsah

1	Úvod	3
2	Teoretický základ	4
2.1	Značení	4
2.2	Norma vektoru, řídlost	4
2.3	Inverze, pseudoinverze	5
2.4	Splajny	6
2.4.1	B-splajny	7
2.4.2	Slovník	8
2.4.3	Normování slovníku	8
2.5	Řídká řešení systému lineárních rovnic	11
2.5.1	Metoda nejmenších čtverců	11
2.5.2	Ortogonální sdružovací metoda	12
2.6	Pravděpodobnost a statistika	13
2.6.1	Pravděpodobnost	13
2.6.2	Bayesův vzorec a nezávislost jevů	14
2.6.3	Náhodná veličina	15
2.6.4	Náhodný vektor	16
2.6.5	Věrohodnostní funkce	18
3	Obecný popis	19
3.1	Senzorová síť	19
3.1.1	Vlastnosti komunikačních linek	19
3.1.2	Komunikační topologie	20
3.2	Postupné Bayesovské odhady	20
3.3	Částicové filtry	22
3.3.1	Algoritmus částicového filtru	22
3.4	Distribuované postupné Bayesovské odhady	23
3.5	Distribuované částicové filtry	24
3.5.1	DPF s fúzním centrem	24
3.5.2	DPF s konsenzem	25
4	Model systému	29
4.1	Lokální věrohodnostní funkce	30
4.2	Globální věrohodnostní funkce	31
4.2.1	Odvození globální LF	31
4.2.2	Oddělení dolní části globální log-LF	31
4.2.3	Distribuovaný výpočet dolní části globální log-LF	33
4.3	Distribuovaný výpočet zbylé části globální log-LF	34
4.3.1	Věrohodnostní konsenzus	34
4.3.2	Výpočet expanzních koeficientů	35
4.3.3	Volba slovníku	37
4.4	Distribuovaný částicový filtr	38

OBSAH

5 Program	40
5.1 Popis simulace	40
5.2 Funkce testbed	41
5.3 Funkce PDAF2, PDAF _{fc} , PDAF _{cb}	43
5.4 Obrázky k popisu programu simulace	45
6 Výsledky simulace a porovnání aproximačních metod	51
6.1 Výsledky	51
6.2 Výpočetní složitost	52
6.3 Obrázky k výsledkům simulace	54
7 Závěr	58
A Obsah přiloženého CD	64

1. Úvod

Signály s tzv. řídkou reprezentací si nacházejí čím dál častěji cestu do matematických aplikací. Název řídká reprezentace znamená, že daný signál je možné vyjádřit přesně či jej lze velmi dobře aproximovat lineární kombinací malého počtu vektorů ze zvoleného reprezentačního systému. Hlavní výhodou je tedy snížení množství informací potřebných k popisu daného signálu.

Použití takto reprezentovaných signálů je hned několik. Nejvíce se využívají v oblasti zpracování obrazu, např. v jejich kompresi. Mezi další významné aplikace patří odšumování signálů, doplňování chybějících úseků signálů či konstrukce nových typů analogově-digitálních převodníků. Příkladem nezvyklého využití tohoto přístupu je např. analýza znečištění vzduchu.

V první části práce je zavedeno značení a teoretický základ nutný k porozumění textu. Je zde zaveden pojem splajnu a splajnového slovníku. Také jsou zde popsána možná řídká řešení lineárních rovnic a jsou zavedeny základní pojmy pravděpodobnosti a statistiky.

V další části práce se věnuji pojmům potřebných pro popis modelu systému. Jsou zde zavedeny pojmy sensorové sítě a postupných Bayesovských odhadů, které jsou následně propojeny s pojmy částicových filtrů. Důležité pro tuto práci je i zavedení distribuovaných Bayesovských odhadů a s nimi spojených distribuovaných částicových filtrů.

V kapitole 4 je popsán model sledování pohybu objektu v decentralizované sensorové síti s využitím distribuovaného částicového filtru. V tomto modelu jsou zahrnuty dvě „nejistoty“ měření. První je šum a druhou je případ, kdy senzor objekt vůbec nezachytí. V modelu jsou k popsání měření využity věrohodnostní funkce. Ty jsou bohužel pro komunikaci mezi senzory složité, proto je přistoupeno k jejich aproximaci. Výhodou těchto věrohodnostních funkcí je jejich řídkost, dají se na ně tedy aplikovat metody řídkých reprezentací signálů a tedy i řídká řešení systému lineárních rovnic.

Část 5 popisuje simulaci modelu zmíněného výše. Model je zpracován v Matlabu a výsledky simulací jsou uvedeny v kapitole 6.

Vlastní přínos spočívá v nalezení metod aproximace zmíněné věrohodnostní funkce, zapracování těchto metod do programu simulace a jejich následné porovnání.

Jelikož práce čerpá převážně z anglických zdrojů, bylo někdy složité najít odpovídající české ekvivalenty k anglickým pojmům. Proto je ve vybraných případech vedle českého názvu uvedený v závorce i název anglický.

Práce vznikla za spolupráce s prof. Franzem Hlawatschem z Technické univerzity ve Vídni, který se uvedené problematice již dlouhou dobu věnuje. Spolu s prof. Hlawatschem se na řešení tohoto tématu podílí či podílelo i několik dalších osob, mezi něž patří i dr. Florian Meyer, do jehož simulace jsem zapracovala aproximační metody decentralizovaného částicového filtru.

2. Teoretický základ

2.1. Značení

Nebude-li řečeno jinak, bude použito následující označení:

Výčet prvků množiny zapisujeme ve složených závorkách, kardinalitu množiny (tedy počet prvků v množině) zapisujeme ve svislicích, např. $|\{1093, 3511\}| = 2$. Množiny přirozených, reálných i komplexních čísel jsou značeny obvyklým způsobem, popořadě \mathbb{N} , \mathbb{R} , \mathbb{C} . Přirozená čísla s nulou značíme \mathbb{N}_0 .

Skaláry jsou označeny malými písmeny v kurzívě, m , n . Komplexní, resp. reálný vektor délky $n \in \mathbb{N}$ je značen malým tučným písmem, $\mathbf{x} \in \mathbb{C}^n$, resp. $\mathbf{x} \in \mathbb{R}^n$. Prvek x_i značí i -tý prvek vektoru \mathbf{x} , $i = 1, \dots, n$. Je-li vektor označen \mathbf{x}^T , jedná se o transponovaný vektor k vektoru \mathbf{x} .

Analogicky komplexní, resp. reálnou matici velikosti $m \times n$, $m, n \in \mathbb{N}$ značíme velkým tučným písmem, $\mathbf{A} \in \mathbb{C}^{m \times n}$, resp. $\mathbf{A} \in \mathbb{R}^{m \times n}$. Pro objasnění velikosti matice používáme označení $\mathbf{A}^{m \times n}$, nebo píšeme, že je matice typu $m \times n$. Prvek $a_{k,l}$, $k = 1, \dots, m$ a $l = 1, \dots, n$, značí prvek matice nacházející se na k -tém řádku a l -tém sloupci. Označení $a_{k,:}$ je rezervováno pro k -tý řádek matice \mathbf{A} , analogicky $a_{:,l}$ značí l -tý sloupec této matice.

Matice označena \mathbf{A}^T je transponovanou maticí k matici \mathbf{A} . Symbolem \mathbf{I}^n značíme jednotkovou matici, tedy matici typu $n \times n$, jejíž všechny prvky jsou rovny 0 s výjimkou prvků na hlavní diagonále, které jsou rovny 1. Jako $\mathbf{0}$ označujeme vektor nul. Analogicky je symbolem $\mathbf{1}$ označen vektor jedniček. Příslušný typ je v textu upřesněn.

Z důvodu implementace jsou někdy vektory i matice uvažovány jako prázdné, tedy velikosti 0×0 , nebo třeba $m \times 0$. Tato informace je však včas uvedena.

2.2. Norma vektoru, řídkost

Ze všeho nejdříve zavedu pojmy normy a řídkosti vektoru. Laicky řečeno, norma vektoru \mathbf{x} označuje vzdálenost tohoto vektoru od počátku, tedy od nulového vektoru $\mathbf{0}$. Řídkost vektoru \mathbf{x} označuje počet nenulových prvků vektoru, či prvků, které jsou větší než předem zadaná hodnota. Korektní definice zní:

Definice 2.1. Pro každé $\mathbf{x} \in \mathbb{C}^n$, $n \in \mathbb{N}$ a $0 \leq p \leq \infty$ definují vztahy

$$\|\mathbf{x}\|_p = \begin{cases} |\{x_i \neq 0, i = 1, \dots, n\}| & \text{pro } p = 0, \\ \sum_{i=1}^n |x_i|^p & \text{pro } 0 < p < 1, \\ \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}} & \text{pro } 1 \leq p < \infty, \\ \max_{i=1, \dots, n} |x_i| & \text{pro } p = \infty, \end{cases}$$

nezápornou funkci na \mathbb{C}^n , tzv. ℓ_p normu vektoru \mathbf{x} .

Poznámka 2.2. Striktně vzato se jedná o normu jen v případě $1 \leq p < \infty$, ostatní jsou označovány jako pseudonormy. V celé této práci je pro zjednodušení použité označení ℓ_p norma pro všechna p .

Poznámka 2.3. Normu ℓ_0 lze také definovat jako počet prvků nosiče vektoru \mathbf{x}

$$\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|. \quad (2.1)$$

Tato norma je důležitá při hledání řídkých řešení, tedy řešení s nejmenším počtem nenulových složek. Neméně důležitá je i ℓ_1 norma představující součet absolutních hodnot prvků vektoru \mathbf{x} :

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|. \quad (2.2)$$

V inženýrské praxi je nejznámější ℓ_2 norma, což je obvyklá eukleidovská norma

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}. \quad (2.3)$$

Definice 2.4. Vektor $\mathbf{x} \in \mathbb{C}^n$ nazveme k -řádkým, jestliže platí

$$\|\mathbf{x}\|_0 \leq k. \quad (2.4)$$

Poznámka 2.5. Tedy k -řádký vektor má nejvýše k nenulových složek. Není časté, aby byly reálné signály řídké ve smyslu uvedené definice. Místo nulových složek obsahují hodnoty velmi blízké nule.

2.3. Inverze, pseudoinverze

Abychom se mohli věnovat řešení lineárních rovnic nutných k výpočtu aproximací funkcí, musíme si uvést několik informací k inverzním a pseudoinverzním maticím. Případné doplnění tohoto textu lze najít v [24] i v [42].

Definice 2.6. Necht \mathbf{A} je čtvercová matice, tedy typu $n \times n$. Čtvercová matice \mathbf{B} se nazývá inverzní k matici \mathbf{A} , jestliže platí

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}^n. \quad (2.5)$$

V takovém případě nazveme matici \mathbf{A} invertibilní. V tomto případě je taková matice určena jednoznačně a značíme ji \mathbf{A}^{-1} .

Věta 2.7. Pro čtvercovou matici \mathbf{A} typu $n \times n$ jsou následující výroky ekvivalentní:

- (a) \mathbf{A} je invertibilní,
- (b) \mathbf{A} je regulární, tj. $\det \mathbf{A} \neq 0$.

V případě platnosti těchto podmínek máme

$$\mathbf{A}^{-1} = \frac{1}{\det \mathbf{A}} \text{adj } \mathbf{A}. \quad (2.6)$$

2.4. SPLAJNY

Poznámka 2.8. Matice adj \mathbf{A} v předchozí větě značí adjungovanou matici, tedy transponovanou matici algebraických doplňků [24].

Matice je tedy invertibilní pouze za předpokladu, je-li čtvercová a regulární. To jsou docela silné předpoklady. Proto přejdeme k obecnějšímu popisu inverze, k tzv. pseudoinverzi. U pseudoinverze již nejsou požadovány tak přísné předpoklady na matici \mathbf{A} . Matice nejen že nemusí být regulární, ale nemusí být ani čtvercová. Pseudoinverze musí splňovat následující podmínky:

Definice 2.9. Matici \mathbf{A}^+ nazveme (Moore-Penroseovou) pseudoinverzí matice \mathbf{A} , platí-li:

- (I1) $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$,
- (I2) $\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+$,
- (I3) $(\mathbf{A}\mathbf{A}^+)^* = \mathbf{A}\mathbf{A}^+$ ($\mathbf{A}\mathbf{A}^+$ je samoadjungovaný),
- (I4) $(\mathbf{A}^+\mathbf{A})^* = \mathbf{A}^+\mathbf{A}$ ($\mathbf{A}^+\mathbf{A}$ je samoadjungovaný).

Poznámka 2.10. Některé další odvození identity pro \mathbf{A}^+

- (I5) $(\mathbf{A}^*)^+ = (\mathbf{A}^+)^*$,
- (I6) $(\mathbf{A}\mathbf{A}^*)^+ = (\mathbf{A}^*)^+\mathbf{A}^+$.

Z těchto vztahů a ze znalosti $(\mathbf{A}\mathbf{A}^+)^* = (\mathbf{A}^+)^*\mathbf{A}^*$ lze jednoduše odvodit identitu nutnou k výpočtu pomocí metody nejmenších čtverců

- (I7) $(\mathbf{A}^*\mathbf{A})^+\mathbf{A}^* = \mathbf{A}^+$.

Důkazy k výše uvedeným identitám jsou k dispozici v [42].

2.4. Splajny

Splajny [39] jsou reálné, spojité, po částech polynomické funkce. S takovými funkcemi se velmi dobře pracuje, například můžeme jednoduše počítat jejich derivace či integrace. Přirozeně se tedy nabízí aproximovat složité funkce právě pomocí splajnů.

Definice 2.11. Řekneme, že funkce $s(x)$ je splajn stupně n (řádu $n + 1$) pro uzly $a = x_0 < x_1 < \dots < x_m = b$, jestliže

- (S1) $s(x)$ je polynom stupně n v každém intervalu $\langle x_i, x_{i+1} \rangle$, $i = 0, \dots, m - 1$,
- (S2) $s(x)$ má spojité derivace až do řádu $(n - 1)$ na celém intervalu $\langle a, b \rangle$.

V této práci budu uvažovat jen splajny, jejichž uzly mají jednotkovou vzdálenost. Takové splajny se nazývají kardinální. Kardinální splajn daného stupně a hladkosti, který je definován na dané části definičního oboru, lze vyjádřit jako lineární kombinaci B-splajnů a to dokonce na stejné části definičního oboru jako původní splajn.

Věta 2.12 (Schoenberg, [35]). Každý kardinální splajn $s(x)$ je jednoznačně charakterizován vztahem

$$s(x) = \sum_{k \in \mathbb{Z}} c(k) \beta^n(x - k), \quad (2.7)$$

kde $x \in \mathbb{R}$, k je celočíselný posun hlavního B-splajnu $\beta^n(x)$ stupně n , a $c(k)$ jsou koeficienty B-splajnů.

Z této věty vychází i původ názvu B-splajnu, který zavedl právě Isaac Jacob Schoenberg jako zkratku pro báze splajny.

2.4.1. B-splajny

B-splajn $\beta^n(x)$ stupně n je symetrická, zvonově vypadající křivka, která vzniká $(n + 1)$ násobnou konvolucí [11] centrovaného normalizovaného obdélníkového pulzu β^0

$$\beta^0(x) = \begin{cases} 1 & \text{pro } -\frac{1}{2} \leq x < \frac{1}{2}, \\ \frac{1}{2} & \text{pro } |x| = \frac{1}{2}, \\ 0 & \text{jinak,} \end{cases} \quad (2.8)$$

$$\beta^n(x) = \underbrace{\beta^0 * \dots * \beta^0(x)}_{(n+1)\text{krát}}. \quad (2.9)$$

Zde uvažujeme jen uniformní bázové splajny, tedy takové, jejichž uzly mají mezi sebou vždy stejnou vzdálenost. Pro lepší představu jsou B-splajny prvních čtyř stupňů vykresleny na obrázku 2.1.

Protože je výpočet konvolucí výpočetně náročný, lze pomocí Fourierovy transformace a jednostranné mocninné funkce $(x)_+^n$

$$(x)_+^n = \begin{cases} x^n & \text{pro } x \geq 0, \\ 0 & \text{pro } x < 0, \end{cases} \quad (2.10)$$

odvodit explicitní vztah výpočtu B-splajnu

$$\beta^n(x) = \frac{1}{n!} \sum_{k=0}^{n+1} \binom{n+1}{k} (-1)^k \left(x - k + \frac{n+1}{2}\right)_+^n. \quad (2.11)$$

Ovšem ani výše uvedený explicitní výpočet není pro aplikace dostatečně rychlý. Proto se používají již odvozené vzorce pro přímý výpočet B-splajnů daných stupňů. V aplikacích jsou nejvíce používány B-splajny kubické (stupně 3), protože dosahují velmi přesné interpolace. Je pro ně použit předpis:

$$\beta^3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^3}{2} & \text{pro } 0 \leq |x| < 1, \\ \frac{(2-|x|)^3}{6} & \text{pro } 1 \leq |x| < 2, \\ 0 & \text{pro } 2 \leq |x|. \end{cases} \quad (2.12)$$

Obecně mohou být B-splajny vícerozměrné funkce. Z kartézského součinu d jedno-rozměrných B-splajnů $\beta^n(x)$ získáme d -rozměrný bázový splajn. Příklad dvourozměrného B-splajnu lze vidět na obrázku 2.2.

Na závěr této podkapitoly si uvedeme několik vlastností B-splajnů:

Poznámka 2.13 (Vybrané vlastnosti B-splajnů). Uniformní B-splajn křivky jsou:

1. po částech polynomy,
2. spojité,
3. kladné,
4. sudé funkce.

Je-li bázový splajn stupně n , pak má:

5. $n + 2$ uzlů s konstantní vzdáleností,
6. $n + 1$ segmentů,
7. derivace až do řádu $(n - 1)$.

2.4. SPLAJNY

Mezi další ceněné vlastnosti patří:

8. kompaktní nosič,
9. jsou to nejmenší polynomiální splajny ve smyslu nejkratšího nosiče,
10. plocha pod křivkou bázevého splajnu je rovna jedné,
11. B-splajny lze sečíst do konstanty vhodnou volbou posuvu.

2.4.2. Slovník

Jak již bylo řečeno, splajny se používají k aproximaci funkcí. A protože lze každý kardinální splajn vyjádřit jako lineární kombinace bázevých splajnů, můžeme tedy i každou aproximaci funkce kardinálním splajnem vyjádřit pomocí lineární kombinace B-splajnů. Stručně řečeno, při aproximaci řešíme soustavu lineárních rovnic

$$\mathbf{Ax} = \mathbf{b}, \quad (2.13)$$

kde $\mathbf{A} \in \mathbb{C}^{m \times n}$ je matice B-splajnů (viz dále), $\mathbf{x} \in \mathbb{C}^n$ je vektor koeficientů a $\mathbf{b} \in \mathbb{C}^m$ reprezentující pozorování či signál chceme aproximovat.

Matice \mathbf{A} se obvykle nazývá slovník (ang. dictionary), její sloupce, v našem případě bázevé splajny, se nazývají atomy. Slovník je sestaven pomocí posouvání centrálního d -dimenzionálního B-splajnu podél příslušné d -dimenzionální mřížky a jeho následným vyhodnocením na předem daných pozicích.

Zde nastávají problémy s volbou bázevých splajnů pro slovník \mathbf{A} . Jsou zde dva extrémy: Je-li šířka B-splajnu velká vůči rozptylu pozic, na kterých bázevý splajn vyhodnocujeme, nastává situace, kdy jsou všechny pozice vyhodnoceny na jednom jediném B-splajnu. Druhým extrémem je, je-li rozptyl pozic o hodně větší než šířka B-splajnu. V tomto případě je každá pozice příslušná jen jednomu B-splajnu. V obou případech jsou výsledky aproximace nevhodné k dalšímu zpracování.

S tímto extrémem souvisí i další problém s volbou B-splajnů. Jsou-li pozice umístěny pouze na okrajových částech B-splajnu, může při aproximaci docházet k velké aproximační chybě na místech, kde žádná vyhodnocující pozice není.

Uvedeným problémům se lze vyhnout vhodnou volbou šířky B-splajnu nebo rozšířením plochy, na které bázevé splajny vyhodnocujeme.

Slovník sestavený z jednorozměrných kubických B-splajnů je na obrázku 2.3. Vícerozměrné kubické B-splajny lze opět získat kartézským součinem jednorozměrných bázevých splajnů. Slovník sestavený z dvourozměrných kubických splajnů je na obrázku 2.4.

2.4.3. Normování slovníku

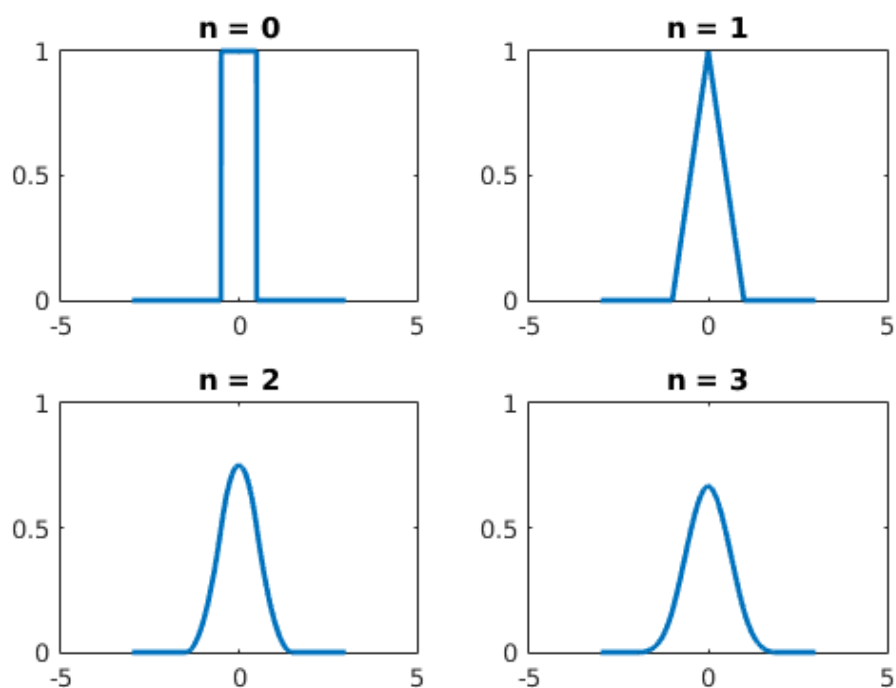
V některých případech je vhodné slovník normovat, například z hlediska urychlení výpočtu řešení. Sloupce takového normovaného slovníku mají jednotkovou energii.

Normovat můžeme pomocí diagonální matice \mathbf{W} , na jejíž diagonále je převrácená hodnota ℓ_2 normy sloupců $\mathbf{a}_{:,l}$ slovníku \mathbf{A} , tedy $1/\|\mathbf{a}_{:,l}\|_2$. Normovaný slovník $\tilde{\mathbf{A}} = \mathbf{AW}$ lze v algoritmech použít obvyklým způsobem místo matice \mathbf{A} (naopak to obecně není možné).

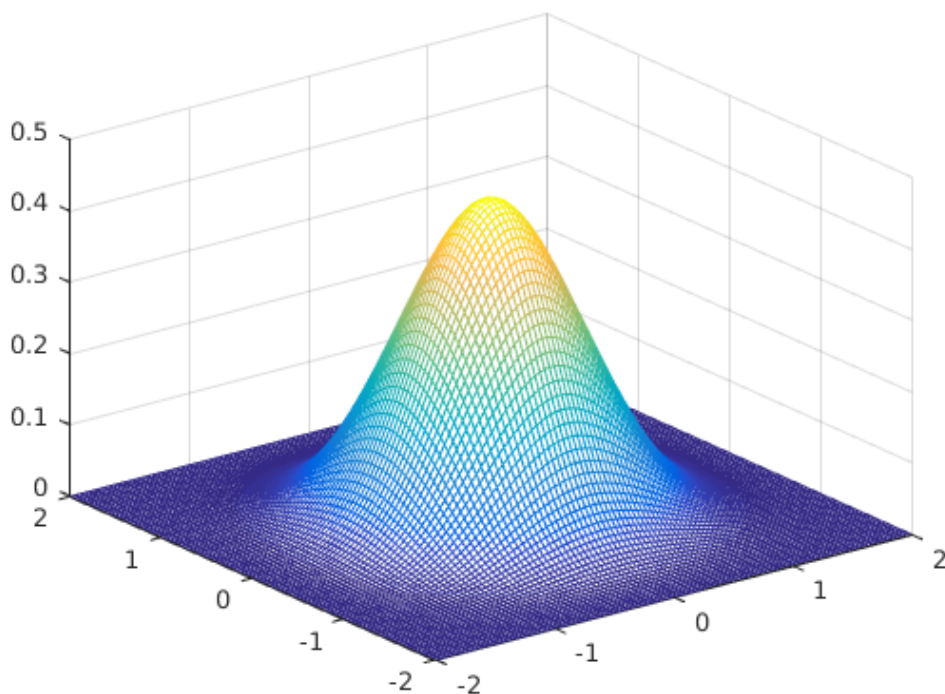
Provedeme-li výpočet řešení pomocí normované matice $\tilde{\mathbf{A}}$, nalezneme jej ve formě vektoru $\tilde{\mathbf{x}}$. Nás ale obvykle zajímá vektor koeficientů příslušných původní matici \mathbf{A} . Ten získáme vynásobením normovaného vektoru zleva maticí \mathbf{W}

$$\mathbf{b} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} = \mathbf{AW}\tilde{\mathbf{x}} = \mathbf{Ax} \Rightarrow \mathbf{x} = \mathbf{W}\tilde{\mathbf{x}}. \quad (2.14)$$

Centrováné bázové splajny stupně 0 až 3



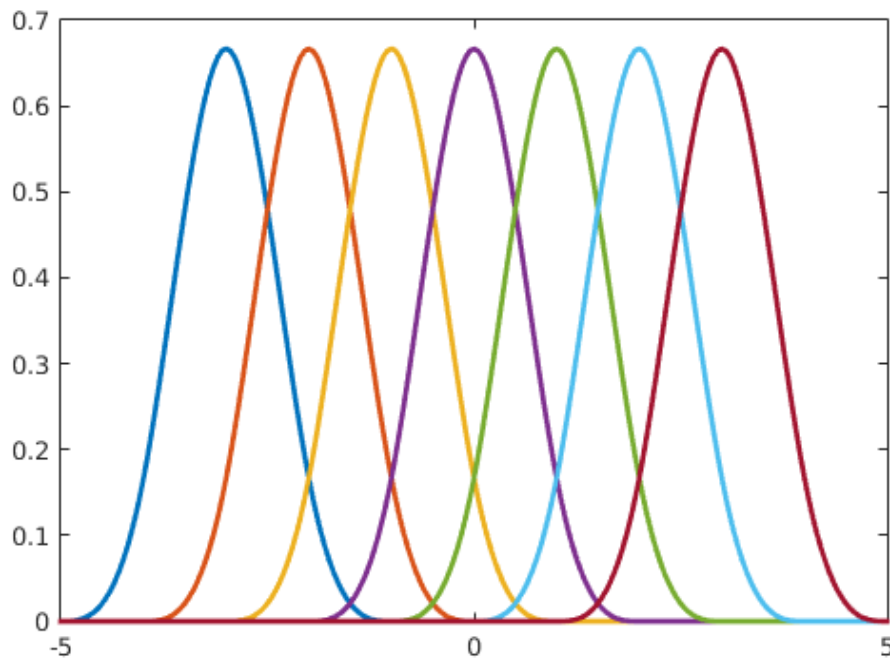
Obrázek 2.1: Centrováné B-splajny vybraných stupňů.

Kubický d-rozměrný bázový splajn, $d = 2$ 

Obrázek 2.2: Dvourozměrný B-splajn stupně 3 vzniklý kartézským součinem dvou jedno-rozměrných B-splajnů.

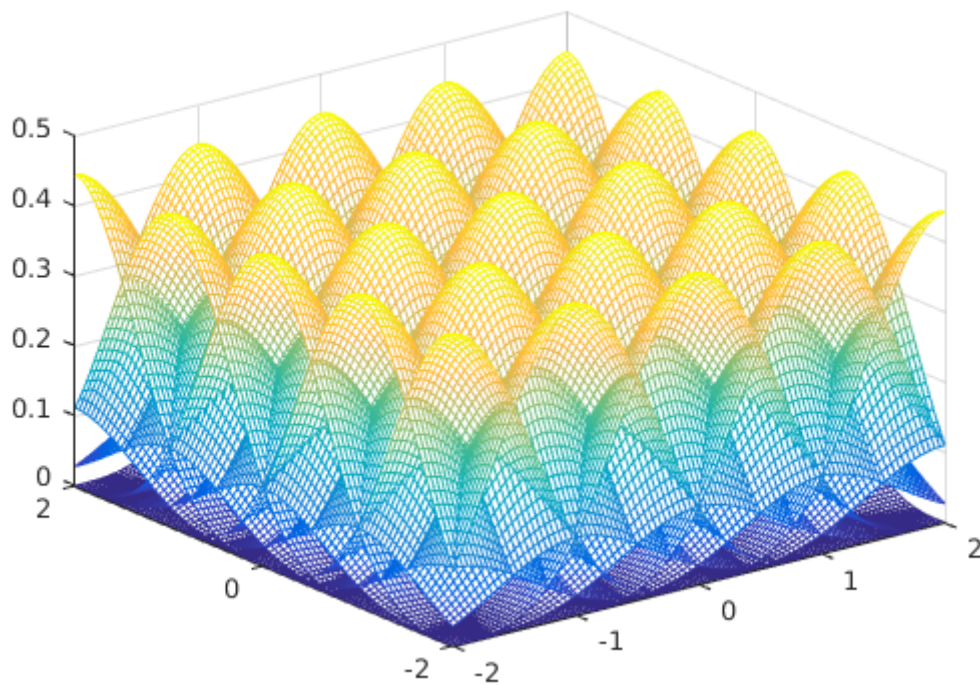
2.4. SPLAJNY

Bázové splajny stupně 3



Obrázek 2.3: Slovník sestavený z jednorozměrných kubických bázových splajnů.

Bázové splajny stupně 3, $d = 2$



Obrázek 2.4: Slovník sestavený s dvourozměrných kubických bázových splajnů.

2.5. Řídká řešení systému lineárních rovnic

Mezi všemi řešeními soustavy lineárních rovnic mají zvláštní postavení ta, ve kterých má vektor řešení co nejvíce složek nulových. Jsou to tzv. řídká řešení [33]. Ta jsou vhodná z důvodů snížení výpočetní náročnosti, numerické stability, usnadnění interpretace dat či snadné a silné komprese.

Úkolem je řešit obvyklou soustavu lineárních rovnic (2.13), ovšem s tím, aby byl hledaný vektor \mathbf{x} co nejřidší. Úlohu lze zapsat jako

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \mathbf{Ax} = \mathbf{b}. \quad (2.15)$$

Vektor \mathbf{x} pak nazveme řídkým řešením systému lineárních rovnic (2.13).

Existuje několik přístupů pro nalezení řídkého řešení [12], [14]. Ačkoli spolu jednotlivé přístupy více či méně souvisí, lze je rozdělit do několika kategorií. Já se budu věnovat pouze dvěma vybraným: algoritmům založených na prahování (ang. thresholding) a hladovým (ang. greedy) algoritmům. Do první skupiny patří algoritmy, které po ukončení celého výpočtu vyberou k nejvýznamějších atomů. Sem bychom mohli zařadit metodu nejmenších čtverců uvedenou v následující podkapitole. Algoritmy druhé skupiny najdou v každé své iteraci jeden nebo více nejvýznamějších atomů. Do této kategorie patří například ortogonální sdružovací metoda, jež bude také popsána dále.

2.5.1. Metoda nejmenších čtverců

Metoda nejmenších čtverců [5] vznikla z potřeby proložit pozorovaná data lineárním matematickým modelem. Aby byl snížen vliv chyby v pozorování, je použito větší množství pozorování než je neznámých parametrů v modelu. To má za následek přeurčení systému lineárních rovnic. Řečeno matematicky: Je dán vektor $\mathbf{b} \in \mathbb{R}^m$ a matice $\mathbf{A} \in \mathbb{R}^{m \times n}$, kde $m > n$, a chceme najít vektor $\mathbf{x} \in \mathbb{R}^n$ tak, aby \mathbf{Ax} byla nejlepší aproximace vektoru \mathbf{b} .

Jednou z možností, jak takovou aproximaci najít, je minimalizovat součet čtverců reziduí $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$

$$\min \sum_{i=1}^m r_i^2 = \min \|\mathbf{r}\|_2^2 = \min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|_2^2. \quad (2.16)$$

Tato minimalizace se nazývá problémem nejmenších čtverců, vektor \mathbf{x} řešením nejmenších čtverců. Takové řešení musí splňovat tzv. systém normálních rovnic

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}. \quad (2.17)$$

Je-li matice \mathbf{A} plně sloupcové hodnosti, je matice $(\mathbf{A}^T \mathbf{A})$ invertibilní [5], a vektor \mathbf{x} je dán jednoznačně vztahem

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}. \quad (2.18)$$

Není-li matice \mathbf{A} plně sloupcové hodnosti, je nutné použít vztah, který vychází z identity (I7) v poznámce 2.10

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^+ \mathbf{A}^T \mathbf{b}. \quad (2.19)$$

2.5. ŘÍDKÁ ŘEŠENÍ SYSTÉMU LINEÁRNÍCH ROVNIC

Toto řešení je dáno také jednoznačně. Je to jediné řešení rovnice $\mathbf{Ax} = \mathbf{b}$, které minimalizuje normu odchylky od pravé strany, a které má samo minimální normu mezi všemi takovými řešeními. Více v [42] nebo v [37].

Chceme-li získat pomocí metody nejmenších čtverců k -řádké řešení, stačí nám vybrat k koeficientů z vektoru řešení \mathbf{x} , které mají na řešení největší podíl. V obecném případě vybíráme k největších koeficientů v absolutní hodnotě, zbylé koeficienty nahrazujeme nulami. V některých případech je vhodné vybírat pouze k největších kladných, popřípadě k největších záporných koeficientů (tj. největších v absolutní hodnotě).

2.5.2. Ortogonální sdružovací metoda

Ortogonální sdružovací metoda (ang. Orthogonal Matching Pursuit) [14] patří mezi nejpoužívanější hladové algoritmy. Je to iterativní metoda, jejíž myšlenka je velmi jednoduchá: V každé iteraci najde jeden atom, který má největší podíl na řešení, tedy ten, který má nejmenší reziduální ℓ_2 chybu při aproximaci.

Vstupem algoritmu je normalizovaný slovník \mathbf{A} typu $m \times n$ a vektor \mathbf{b} typu $m \times 1$, který chceme nahradit jednodušší funkcí. Dále sem patří dvě kritéria ukončující algoritmus, z nichž je nutné použít alespoň jedno, maximální počet iterací i_{\max} a maximální velikost chyby ϵ_{\max} .

Výstupem algoritmu je množina indexů aktivních atomů \mathcal{S} , popřípadě vektor \mathbf{x} typu $n \times 1$, který má nenulová čísla na pozicích odpovídajících aktivním atomům. Jelikož ortogonální sdružovací metoda nachází v každé iteraci jen jeden atom, obsahuje množina \mathcal{S} nejvýše i_{\max} prvků. Zřejmě vektor řešení \mathbf{x} je nejvýše i_{\max} -řádký.

Při inicializaci algoritmu je nastaven vektor počáteční aproximace řešení $\hat{\mathbf{x}} = \mathbf{0}$, počáteční množina indexů $\mathcal{S} = \emptyset$, prázdná matice $\mathbf{S}^{m \times 0}$ a je spočteno počáteční reziduum $\mathbf{r} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$. Také je nastaven počet iterací $i = 1$.

Nyní je nalezen atom $\mathbf{a}_{:,l_0}$ maximalizující absolutní hodnotu součinu sloupců $\mathbf{a}_{:,l}$, $l = 1, \dots, n$, matice \mathbf{A} s reziduem \mathbf{r}

$$l_0 = \arg \max_l |\mathbf{a}_{:,l}^T \mathbf{r}|. \quad (2.20)$$

Do indexové množiny \mathcal{S} je přidán index l_0 atomu $\mathbf{a}_{:,l_0}$. Tento atom je také přiřazen do matice \mathbf{S} jako její poslední sloupec. Je počten nový vektor řešení $\hat{\mathbf{x}}$ a to pomocí metody nejmenších čtverců

$$\hat{\mathbf{x}} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \mathbf{b}. \quad (2.21)$$

S novým vektorem řešení je vypočítáno nové reziduum a jeho ℓ_2 norma

$$\mathbf{r} = \mathbf{b} - \mathbf{S}\hat{\mathbf{x}} \quad (2.22)$$

$$\epsilon = \|\mathbf{r}\|_2. \quad (2.23)$$

Teď zbývá jen zvýšení počtu iterací i o jedna a zkontrolování ukončovacích kritérií. Pokud $\epsilon > \epsilon_{\max}$ a zároveň $i \leq i_{\max}$, pak je celý proces opakován znovu. V opačném případě jsou přiřazeny do vektoru \mathbf{x} na indexy \mathcal{S} popořadě čísla z vektoru $\hat{\mathbf{x}}$ a algoritmus končí.

Analogicky jako u metody nejmenších čtverců je získán aproximovaný vektor $\hat{\mathbf{b}}$ vynásobením slovníku a řídkého řešení: $\hat{\mathbf{b}} = \mathbf{A}\mathbf{x}$.

2.6. Pravděpodobnost a statistika

V této kapitole budou zavedeny pojmy, které jsou nutné pro porozumění matematického popisu problému řešeného v této práci.

V celém textu rozumíme jevovým polem dvojici (Ω, \mathcal{A}) , kde Ω je neprázdná množina elementárních jevů, \mathcal{A} neprázdný systém podmnožin množiny Ω tvořící σ -algebru. Pravděpodobnostním prostorem je pak trojice (Ω, \mathcal{A}, P) , kde P je pravděpodobnost, reálná funkce definovaná na σ -algebře \mathcal{A} . Dvojicí $(\mathbb{R}, \mathcal{B})$ je rozuměn měřitelný prostor, \mathcal{B} je borelovská σ -algebra. Obdobně ve dvojici $(\mathbb{R}^n, \mathcal{B}_n)$ je \mathcal{B}_n systém borelovských podmnožin n -rozměrného euklidovského prostoru \mathbb{R}^n . Podrobnosti jsou k nalezení v [2] nebo v [47].

2.6.1. Pravděpodobnost

Definice 2.14 (Kolmogorov). *Nechť (Ω, \mathcal{A}) je jevové pole. Potom zobrazení $P : \mathcal{A} \rightarrow \mathbb{R}$, které každému jevu $A \in \mathcal{A}$ přiřazuje $P(A)$, nazveme pravděpodobností na jevovém poli (Ω, \mathcal{A}) , jestliže je splněno:*

$$(P1) \quad P(\Omega) = 1 \quad \text{pro všechna } A \in \mathcal{A},$$

$$(P2) \quad P(A) \geq 0,$$

$$(P3) \quad P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i) \quad \text{resp.} \quad P\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i),$$

kde A_1, A_2, \dots je spočetná resp. konečná posloupnost po dvou disjunktních jevů z \mathcal{A} , tj. $A_i \in \mathcal{A}, A_i \cap A_j = \emptyset$, pro všechna $i \neq j$.

Věta 2.15 (Vlastnosti pravděpodobnosti). *Nechť (Ω, \mathcal{A}, P) je pravděpodobnostní prostor. Pak pro libovolné $A_1, A_2, \dots \in \mathcal{A}$ má pravděpodobnost následující vlastnosti:*

1. $P(\emptyset) = 0$,
2. $P(A) \leq 1$,
3. $A_1 \subset A_2 \Rightarrow P(A_1) \leq P(A_2)$,
4. $A_1 \subset A_2 \Rightarrow P(A_2 - A_1) = P(A_2) - P(A_1)$,
5. $P(A^c) = 1 - P(A)$,
6. $P(A_1 \cdot A_2) = P(A_2) - P(A_2 \cap A_1)$,
7. $P(A_1 \cup A_2) = P(A_1) + P(A_2) - P(A_1 \cap A_2)$,
8. $P(A_1 \cup A_2) \leq P(A_1) + P(A_2)$,
9. $P\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} P(A_i)$,
10. $P\left(\bigcap_{i=1}^{\infty} A_i\right) \geq 1 - \sum_{i=1}^{\infty} (1 - P(A_i))$,
11. $A_1 \subset A_2 \subset \dots \Rightarrow P\left(\bigcup_{i=1}^{\infty} A_i\right) = \lim_{i \rightarrow \infty} P(A_i)$,
12. $A_1 \supset A_2 \supset \dots \Rightarrow P\left(\bigcap_{i=1}^{\infty} A_i\right) = \lim_{i \rightarrow \infty} P(A_i)$.

2.6.2. Bayesův vzorec a nezávislost jevů

Definice 2.16. *Nechť (Ω, \mathcal{A}, P) je pravděpodobnostní prostor, nechť náhodné jevy $A, B \in \mathcal{A}$. Podmíněná pravděpodobnost náhodného jevu A za podmínky, že nastal náhodný jev B je zobrazení $P(\cdot|B) : \mathcal{A} \rightarrow \mathbb{R}$ dané vzorcem*

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad (2.24)$$

kde $P(B) > 0$.

Poznámka 2.17. Protože je výraz $P(A \cap B)$ symetrický v obou náhodných jevech, lze za předpokladu $P(A) > 0$ psát i

$$P(B|A) = \frac{P(A \cap B)}{P(A)}. \quad (2.25)$$

Definice 2.18. *Nechť (Ω, \mathcal{A}, P) je pravděpodobnostní prostor. Řekneme, že náhodné jevy $A_1, A_2, \dots \in \mathcal{A}$ tvoří úplný systém jevů, jestliže platí*

$$A_i \cap A_j = \emptyset \text{ pro } i \neq j, \quad (2.26)$$

$$\bigcup_{i=1}^{\infty} A_i = \Omega. \quad (2.27)$$

Věta 2.19 (Úplná pravděpodobnost). *Nechť A_1, A_2, \dots je úplný systém jevů v pravděpodobnostním prostoru (Ω, \mathcal{A}, P) takový, že $P(A_i) > 0$, $i = 1, 2, \dots$. Potom platí*

$$P(B) = \sum_{i=1}^{\infty} P(B|A_i)P(A_i). \quad (2.28)$$

Věta 2.20 (Bayesův vzorec). *Nechť A_1, A_2, \dots je úplný systém jevů v pravděpodobnostním prostoru (Ω, \mathcal{A}, P) takový, že $P(A_i) > 0$, $i = 1, 2, \dots$. Jestliže $P(B) > 0$, pak platí*

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_{i=1}^{\infty} P(B|A_i)P(A_i)}, \text{ pro } j = 1, 2, \dots \quad (2.29)$$

Poznámka 2.21. V aplikacích je $P(A_j)$ nazývána apriorní pravděpodobností a $P(A_j|B)$ je nazývána aposteriorní pravděpodobností.

Definice 2.22. *Nechť (Ω, \mathcal{A}, P) je pravděpodobnostní prostor. Jevy $A, B \in \mathcal{A}$ nazveme nezávislé vzhledem k pravděpodobnosti P , když platí*

$$P(A \cap B) = P(A)P(B). \quad (2.30)$$

Definice 2.23. *Řekneme, že náhodné jevy A_1, \dots, A_n definované na pravděpodobnostním prostoru (Ω, \mathcal{A}, P) jsou skupinově nezávislé vzhledem k pravděpodobnosti P , když platí*

$$P\left(\bigcap_{i=1}^n A_i\right) = \prod_{i=1}^n P(A_i). \quad (2.31)$$

Definice 2.24. *Náhodné jevy A_1, \dots, A_n definované na pravděpodobnostním prostoru (Ω, \mathcal{A}, P) jsou podmíněně nezávislé při daném náhodném jevu B jestliže platí*

$$P\left(\bigcap_{i=1}^n A_i \middle| B\right) = \prod_{i=1}^n P(A_i|B). \quad (2.32)$$

2.6.3. Náhodná veličina

Definice 2.25. *Nechť (Ω, \mathcal{A}) je jevové pole a $(\mathbb{R}, \mathcal{B})$ je měřitelný prostor. Reálnou funkci $X : \Omega \rightarrow \mathbb{R}$ nazveme náhodnou veličinou právě tehdy, když platí*

$$\{\omega \in \Omega : X(\omega) < x\} \in \mathcal{A} \quad \text{pro všechna } x \in \mathbb{R}. \quad (2.33)$$

Poznámka 2.26. Zavedme označení:

$$\begin{aligned} &\{\omega \in \Omega : X(\omega) < x\} \text{ nebo } [X < x] \text{ a } P(X < x), \text{ kde } x \in \mathbb{R}; \\ &\{\omega \in \Omega : X(\omega) = x\} \text{ nebo } [X = x] \text{ a } P(X = x), \text{ kde } x \in \mathbb{R}. \end{aligned}$$

Definice 2.27. *Nechť X je náhodná veličina definovaná na (Ω, \mathcal{A}) a (Ω, \mathcal{A}, P) je pravděpodobnostní prostor. Reálná funkce*

$$F_X(x) = P(X < x) \quad \text{pro všechna } x \in \mathbb{R} \quad (2.34)$$

se nazývá distribuční funkce náhodné veličiny X .

Věta 2.28. *Distribuční funkce F_X náhodné veličiny X je funkce neklesající, zleva spojitá a normovaná. Splňuje tedy*

$$\lim_{x \rightarrow -\infty} F_X(x) = 0 \quad \lim_{x \rightarrow \infty} F_X(x) = 1. \quad (2.35)$$

Poznámka 2.29. Distribuční funkce udává pravděpodobnost, s jakou náhodná veličina X nabývá hodnot menších než x . Ve statistice nacházejí uplatnění dva základní typy distribučních funkcí: Diskrétní (po částech konstantní) a spojitá (absolutně spojitá).

Definice 2.30. *Nechť zobrazení $p : \mathbb{R} \rightarrow \langle 0, 1 \rangle$ je dáno předpisem $p(x) = P(X = x)$. Pak funkci $p(x)$ nazveme pravděpodobnostní funkcí náhodné veličiny X .*

Definice 2.31. *Nechť (Ω, \mathcal{A}, P) je pravděpodobnostní prostor a X je náhodná veličina na (Ω, \mathcal{A}) s distribuční funkcí $F_X(x)$. Řekneme, že náhodná veličina X je diskrétní vzhledem k pravděpodobnosti P právě tehdy, když existuje nezáporná funkce $p(x)$, která je nulová v \mathbb{R} , s výjimkou nejméně jednoho a nejvýše spočetně mnoha bodů (v množině M), kde je kladná a platí pro ni*

$$F_X(x) = \sum_{t \in M, t \leq x} p(t) \quad \text{pro všechna } x \in \mathbb{R}. \quad (2.36)$$

Zde M je oborem hodnot náhodné veličiny X .

Věta 2.32 (Vlastnosti pravděpodobnostní funkce). *Nechť $p(x)$ je pravděpodobnostní funkce diskrétní náhodné veličiny X . Pak platí*

1. $p(x) \geq 0$ pro všechna $x \in \mathbb{R}$,
2. $\sum_{x \in M} p(x) = 1$,
3. $p(x_0) = P(X = x_0)$ pro libovolné, ale pevné $x_0 \in \mathbb{R}$,
4. $P(X \in B) = \sum_{x \in M \cap B} p(x)$ pro všechna $B \in \mathcal{B}$.

2.6. PRAVDĚPODOBNOST A STATISTIKA

Definice 2.33. *Nechť (Ω, \mathcal{A}, P) je pravděpodobnostní prostor a X je náhodná veličina na (Ω, \mathcal{A}) s distribuční funkcí $F_X(x)$. Řekneme, že náhodná veličina X je spojitá vzhledem k pravděpodobnosti P právě tehdy, když existuje nezáporná po částech spojitá funkce $f(x)$ tak, že pro všechna $x \in \mathbb{R}$ platí*

$$F_X(x) = \int_{-\infty}^x f(t)dt. \quad (2.37)$$

Funkce $f(t)$ se nazývá hustota pravděpodobnosti náhodné veličiny X .

Věta 2.34 (Vlastnosti hustoty pravděpodobnosti). *Nechť $f(x)$ je hustota pravděpodobnosti spojitě náhodné veličiny X . Pak platí*

1. $f(x) \geq 0$ pro všechna $x \in \mathbb{R}$,
2. $\int_{-\infty}^{\infty} f(x)dx = 1$,
3. $P(x_0 < X \leq x_0 + h) = \int_{x_0}^{x_0+h} f(x)dx$ pro libovolné, ale pevné $x_0 \in \mathbb{R}$ a $h > 0$,
4. $P(X = x_0) = 0$ pro libovolné, ale pevné $x_0 \in \mathbb{R}$,
5. $f(x) = \frac{d}{dx}F_X(x)$ ve všech bodech spojitosti funkce $f(x)$.

Vybraná rozdělení

Definice 2.35 (Poissonovo rozdělení). *Řekneme, že náhodná veličina X má Poissonovo rozdělení s parametrem λ , a píšeme $X \sim \text{Po}(\lambda)$, jestliže X nabývá pouze hodnot $0, 1, \dots$, a to s pravděpodobnostmi*

$$p(k) = \frac{\lambda^k}{k!}e^{-\lambda}, \quad k = 0, 1, \dots, \quad (2.38)$$

kde $\lambda > 0$ je dané číslo.

Definice 2.36 (Normální rozdělení). *Řekneme, že náhodná veličina X má normální rozdělení s parametry μ a σ , a píšeme $X \sim N(\mu, \sigma^2)$, je-li hustota pravděpodobnosti určena vztahem*

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad (2.39)$$

kde $\mu \in \mathbb{R}$ a $\sigma > 0$ jsou dané konstanty.

2.6.4. Náhodný vektor

Definice 2.37. *Nechť X_1, \dots, X_n jsou náhodné veličiny definované na (Ω, \mathcal{A}) . Pak vektor $\mathbf{X} = (X_1, \dots, X_n)^T$ nazveme náhodným vektorem.*

Poznámka 2.38. Zavedme označení:

$$\begin{aligned} &\{\omega \in \Omega : \mathbf{X}(\omega) < \mathbf{x}\} \text{ nebo } [\mathbf{X} < \mathbf{x}] \text{ a } P(\mathbf{X} < \mathbf{x}), \text{ kde } \mathbf{x} \in \mathbb{R}^n; \\ &\{\omega \in \Omega : \mathbf{X}(\omega) = \mathbf{x}\} \text{ nebo } [\mathbf{X} = \mathbf{x}] \text{ a } P(\mathbf{X} = \mathbf{x}), \text{ kde } \mathbf{x} \in \mathbb{R}^n. \end{aligned}$$

Definice 2.39. *Nechť \mathbf{X} je náhodný vektor na $(\Omega, \mathcal{A}, \mathbb{P})$. Pak funkci danou předpisem*

$$F_{\mathbf{X}}(\mathbf{x}) = \mathbb{P}(\mathbf{X} < \mathbf{x}) \quad \text{pro všechna } \mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n \quad (2.40)$$

nazýváme distribuční funkce náhodného vektoru \mathbf{X} , případně sdružená distribuční funkce náhodných veličin X_1, \dots, X_n .

Věta 2.40. *Distribuční funkce náhodného vektoru \mathbf{X} je funkce neklesající a zleva spojitá vzhledem ke každé jednotlivé náhodné veličině X_i , $i = 1, \dots, n$ a je normovaná, tedy pro všechna $i = 1, \dots, n$ splňuje*

$$\lim_{x_i \rightarrow -\infty} F_{\mathbf{X}}(x_i) = 0, \quad \lim_{x_i \rightarrow \infty} F_{\mathbf{X}}(x_i) = 1. \quad (2.41)$$

Definice 2.41. *Nechť $(\Omega, \mathcal{A}, \mathbb{P})$ je pravděpodobnostní prostor a $\mathbf{X} = (X_1, \dots, X_n)^T$ je náhodný vektor na (Ω, \mathcal{A}) s distribuční funkcí $F_{\mathbf{X}}(\mathbf{x})$. Řekneme, že náhodný vektor je diskrétní vzhledem k \mathbb{P} právě tehdy, když existuje nezáporná funkce $p(\mathbf{x}) = p(x_1, \dots, x_n)$ s neprázdnou spočetnou množinou M takovou, že $p(\mathbf{x}) > 0$, $\mathbf{x} \in M$ a $p(\mathbf{x}) = 0$, $\mathbf{x} \notin M$ a platí pro ni*

$$F_{\mathbf{X}}(x_1, \dots, x_n) = \sum_{t_1 \leq x_1} \dots \sum_{t_n \leq x_n} p(t_1, \dots, t_n) \quad \text{pro všechna } \mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n. \quad (2.42)$$

Funkci $p(\mathbf{x})$ nazveme pravděpodobnostní funkcí náhodného vektoru \mathbf{X} (ang. probability mass function, zkr. PMF), popřípadě simultánní (sdruženou) pravděpodobnostní funkcí náhodných veličin X_1, \dots, X_n . Množina M je obor hodnot náhodného vektoru \mathbf{X} .

Věta 2.42 (Vlastnosti pravděpodobnostní funkce). *Pro pravděpodobnostní funkci $p(\mathbf{x})$ náhodného vektoru $\mathbf{X} = (X_1, \dots, X_n)^T$ platí*

1. $p(\mathbf{x}) \geq 0$ pro všechna $\mathbf{x} \in \mathbb{R}^n$,
2. $\sum_{\mathbf{x} \in M} p(\mathbf{x}) = 1$,
3. $\mathbb{P}(\mathbf{X} \in B) = \sum_{\mathbf{x} \in M \cap B} p(\mathbf{x})$ pro všechna $B \in \mathcal{B}_n$.

Definice 2.43. *Nechť $(\Omega, \mathcal{A}, \mathbb{P})$ je pravděpodobnostní prostor a $\mathbf{X} = (X_1, \dots, X_n)^T$ je náhodný vektor na (Ω, \mathcal{A}) s distribuční funkcí $F_{\mathbf{X}}(\mathbf{x})$. Řekneme, že náhodný vektor je spojitý vzhledem k \mathbb{P} právě tehdy, když existuje nezáporná po částech spojitá funkce $f(\mathbf{x}) = f(x_1, \dots, x_n)$ tak, že pro všechna $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$*

$$F_{\mathbf{X}}(x_1, \dots, x_n) = \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_n} f(t_1, \dots, t_n) dt_1 \dots dt_n. \quad (2.43)$$

Funkce $f(\mathbf{x})$ je hustota pravděpodobnosti náhodného vektoru \mathbf{X} (ang. probability density function, zkr. PDF), popřípadě simultánní (sdružená) hustota náhodných veličin X_1, \dots, X_n . Množina M je obor hodnot náhodného vektoru \mathbf{X} .

Věta 2.44 (Vlastnosti hustoty pravděpodobnosti). *Pro hustotu pravděpodobnosti $f(\mathbf{x})$ náhodného vektoru $\mathbf{X} = (X_1, \dots, X_n)^T$ platí*

1. $f(\mathbf{x}) \geq 0$ pro všechna $\mathbf{x} \in \mathbb{R}^n$,
2. $\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(x_1, \dots, x_n) dx_1 \dots dx_n = 1$,
3. $\mathbb{P}(\mathbf{X} \in B) = \int \dots \int_{(x_1, \dots, x_n) \in B} f(x_1, \dots, x_n) dx_1 \dots dx_n$ pro všechna $B \in \mathcal{B}_n$,
4. ve všech bodech spojitosti funkce $f(x_1, \dots, x_n)$ je

$$f(x_1, \dots, x_n) = \frac{\partial^n}{\partial x_1 \dots \partial x_n} F_{\mathbf{X}}(x_1, \dots, x_n).$$

2.6. PRAVDĚPODOBNOST A STATISTIKA

2.6.5. Věrohodnostní funkce

Definice 2.45. Necht' $\mathbf{X} = (X_1, \dots, X_n)^T$ je náhodný vektor, kde X_i jsou nezávislé stejně rozdělené náhodné veličiny s hustotou $f(x, \boldsymbol{\theta})$, přičemž $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)^T$ je neznámý vektor parametrů. Při pevné hodnotě \mathbf{x} nazveme sdruženou hustotu $f(\mathbf{x}, \boldsymbol{\theta})$ jakožto funkci parametru $\boldsymbol{\theta}$ věrohodnostní funkcí (ang. likelihood function, zkr. LF).

Poznámka 2.46. Sdružená hustota $f(\mathbf{x}, \boldsymbol{\theta})$ náhodného vektoru $\mathbf{X} = (X_1, \dots, X_n)^T$ z předchozí definice je dána předpisem

$$f(\mathbf{x}; \boldsymbol{\theta}) = \prod_{i=1}^n f(x_i, \boldsymbol{\theta}). \quad (2.44)$$

Poznámka 2.47. Zpravidla je technicky výhodnější pracovat s přirozeným logaritmem věrohodnostní funkce

$$\ln f(\mathbf{x}; \boldsymbol{\theta}) = \sum_{i=1}^n \ln f(x_i, \boldsymbol{\theta}), \quad (2.45)$$

tedy s tzv. logaritmickou věrohodnostní funkcí. Zde \ln je přirozený logaritmus. Podobně v případě diskrétního rozdělení charakterizovaného pravděpodobnostmi $p(x_i, \boldsymbol{\theta})$, $i = 1, 2, \dots$, definujeme logaritmickou věrohodnostní funkci jako

$$\ln p(\mathbf{x}; \boldsymbol{\theta}) = \sum_{i=1}^n \ln p(x_i, \boldsymbol{\theta}). \quad (2.46)$$

Poznámka 2.48. Logaritmus je monotónně rostoucí funkce. Je tedy zřejmé, že logaritmus jakékoli funkce dosahuje svého maxima ve stejném bodě jako funkce původní. To je důležité např. pro maximálně věrohodný odhad, kde je nutné najít maximum věrohodnostní funkce. Tento přístup je výhodný i z hlediska numerických metod. Je totiž mnohem výhodnější pracovat se součtem funkcí než s jejich součinem.

3. Obecný popis

V následujících kapitolách se budu věnovat popisu a řešení problému, který představil profesor Franz Hlawatsch a jeho tým. Jedná se o sledování pohybu objektu v senzorové síti [18]. Objektem může být téměř cokoli: auto s nepřitelem na zabraném území i radioaktivní oblak po výbuchu jaderné elektrárny.

Nejdříve představím obecné pojmy používané v aplikacích. Jedná se o senzorovou síť, postupné bayesovské odhady a částicový filtr. Následně se budu věnovat distribuovaným postupným (sekvenčním) bayesovským odhadům a souvisejícím distribuovaným částicovým filtrům. Nakonec uvedu popis konkrétních modelů, jež jsou použity v simulaci, která je součástí této práce (podrobnosti lze nalézt v části 5).

3.1. Senzorová síť

Senzorová síť zaznamenává stav sledovaného objektu. V obecném případě se jedná o tzv. síť zprostředkovatelů (ang. agent networks, zkr. AN). Zprostředkovatelé jsou senzory [44], kamery [1], roboti [9], nebo dokonce i autonomní vozidla [34]. V aplikacích se AN používají nejen ke sledování pohybujícího se objektu [44], ale i k lokalizaci zdroje znečištění [45] nebo k monitorování zdravotní péče [25].

Velikost zprostředkovatelů je omezena jen a pouze fyzikálními zákony. Mohou to být malé, levné senzory fungující na baterie s omezenou výpočetní i komunikační kapacitou, ale i již zmiňovaná autonomní vozidla, která zvládají i několik tisíc operací zároveň.

Každý zprostředkovatel by měl obsahovat: jeden nebo více senzorů, komunikační rozhraní, jednotku na zpracování informací a pohonnou jednotku (např. baterie, motor). Senzory měří fyzikální veličiny např. polohu, rychlost, teplotu, tlak, intenzitu světla, amplitudu. Pohonné jednotky umožňují senzoru vypnutí, v případě lepších senzorů i pohyb.

V kontextu této práce každý zprostředkovatel odhaduje parametry sledovaného objektu na základě jeho vlastního měření. Měření jednotlivých senzorů nepostačují k odhadnutí hledaných parametrů, proto spolu musí zprostředkovatelé komunikovat. Takové síť zprostředkovatelů se nazývají distribuované. Odhad parametrů objektu získaný z měření distribuovaných sítí nazveme distribuovaným odhadem.

Pro algoritmy distribuovaných odhadů jsou důležitá dvě omezení: Prvním jsou vlastnosti komunikačních linek, např. spolehlivost nebo zpoždění signálu. Ve zde uvedeném modelu uvažujeme ideální síť bez zpoždění a chyb. Druhým omezením je komunikační topologie, tedy které senzory spolu mohou komunikovat.

3.1.1. Vlastnosti komunikačních linek

Mnoho algoritmů na výpočty distribuovaných odhadů předpokládá časovou synchronizaci senzorů. Obvykle se také poloha senzorů uvažuje jako známá.

Algoritmy odhadů uvažují i různá fyzikální omezení: množství dostupné energie (omezená kapacita baterií zprostředkovatelů), výpočetní náročnost (limitovaná velikost paměti či omezení procesoru), komunikace (signál může odeslán jen do určité vzdálenosti a jen na určitých frekvencích, chyby v signálu).

V závislosti na aplikacích mohou algoritmy obsahovat dobu životnosti zprostředkovatele, jeho pohyb, reakční čas či větší robustnost komunikační sítě.

3.2. POSTUPNÉ BAYESOVSKÉ ODHADY

3.1.2. Komunikační topologie

Komunikační topologie sítě zprostředkovatelů může být popsána prostým grafem [38].

Definice 3.1. *Prostý graf G je dvojice $G = (V_G, E_G)$, kde V_G je konečná množina uzlů a $E_G = \{\{u, v\} | u, v \in V_G, u \neq v\} \subseteq V_G \times V_G$ je konečná množina hran. Píšeme $e = \{u, v\}$, $e \in E_G$, jestliže hrana e spojuje vrcholy $u, v \in V_G$.*

Každý uzel u_1, \dots, u_n , $n \in \mathbb{N}$, představuje jednoho zprostředkovatele, zatímco každá hrana $e = \{u_i, u_j\}$, $i, j \in \mathbb{N}$, $1 \leq i, j \leq n$, představuje neorientovanou komunikační linku mezi zprostředkovateli u_i a u_j .

Obecně nemůže každý zprostředkovatel komunikovat s každým, proto zavádíme pojem sousedství zprostředkovatele k jako množinu všech zprostředkovatelů $\mathcal{N}_k \subseteq V_G$ definovanou: $\mathcal{N}_k = \{k' \in V_G | \{k, k'\} \in E_G\}$. Prvky této množiny jsou označovány jako sousedé zprostředkovatele k . Sousedé vybraného zprostředkovatele tvoří podgraf komunikačního grafu.

Definice 3.2. *Nechť G je prostý graf. Graf G' se nazývá podgraf grafu G , jestliže $V_{G'} \subseteq V_G$, $E_{G'} \subseteq E_G$.*

Abychom mohli informaci o sousednosti implementovat, zavedeme pojem matice sousednosti. Ta vyjadřuje, kteří zprostředkovatelé spolu mohou komunikovat:

Definice 3.3. *Nechť G je prostý graf. Matice $\mathbf{A}^{n \times n}$ se nazývá matice sousednosti, jestliže*

$$a_{k,l} = \begin{cases} 1 & \text{existuje-li hrana } e = \{u_k, u_l\}, \\ 0 & \text{jinak.} \end{cases} \quad (3.1)$$

Je zřejmé, že komunikace probíhá jen mezi zprostředkovateli spojenými hranou. Graf, v němž neexistuje zprostředkovatel, který by nebyl spojený s jiným zprostředkovatelem se nazývá souvislý:

Definice 3.4. *V grafu G definujeme cestu mezi vrcholy u, v jako alternující posloupnost vrcholů a hran ($u = u_0, e_1, u_1, e_2, \dots, u_{n-1}, e_n, u_n = v$), $n > 0$, kde $i \neq j \Rightarrow u_i \neq u_j$, $1 \leq i, j \leq n$. Číslo n pak nazveme délkou cesty.*

Definice 3.5. *Řekneme, že graf je souvislý, jestliže zde existuje cesta mezi libovolnými dvěma uzly $u, v \in V_G$.*

Vzdálenost mezi dvěma zprostředkovateli je určena počtem hran v nejkratší cestě, která dané zprostředkovatele spojuje. Největší vzdálenost mezi jakýmkoli dvěma zprostředkovateli nazveme průměrem komunikační sítě.

3.2. Postupné Bayesovské odhady

Komunikaci mezi jednotlivými senzory popisují algoritmy distribuovaného částicový filtru (ang. distributed particle filter, zkr. DPF). Ty jsou vhodné pro nelineární a negaussovské systémy, jako je systém popsáný v této práci. Teoretický základ DPF tvoří postupné Bayesovské odhady (ang. sequential Bayesian estimation) [3], které jsou nezbytné pro následný obecný popis algoritmů částicových filtrů.

Uvažujme stavový vektor \mathbf{x}_n , kde $n \in \mathbb{N}$ je diskrétní časový index. Tento vektor se mění s časem dle tzv. systémového modelu

$$\mathbf{x}_n = \mathbf{g}_n(\mathbf{x}_{n-1}, \mathbf{u}_n). \quad (3.2)$$

Zde $\mathbf{g}_n(\cdot, \cdot)$ je známá, obecně nelineární (stavově-přechodová) funkce a \mathbf{u}_n je bílý šum¹, nezávislý na minulém či přítomném stavu, se známou hustotou pravděpodobnosti $f(\mathbf{u}_n)$.

V čase n je obdržén vektor měření (pozorování) \mathbf{z}_n , který je se stavem \mathbf{x}_n spojen tzv. měřicím modelem

$$\mathbf{z}_n = \mathbf{h}_n(\mathbf{x}_n, \mathbf{v}_n). \quad (3.3)$$

Opět $\mathbf{h}_n(\cdot, \cdot)$ je známá, obecně nelineární (měřicí) funkce, \mathbf{v}_n je bílý šum, nezávislý na minulém či přítomném stavu, se známou hustotou pravděpodobnosti $f(\mathbf{v}_n)$. Dále definujeme $\mathbf{z}_{1:n} = (\mathbf{z}_1^T, \dots, \mathbf{z}_n^T)^T$, což je vektor všech měření až do času n .

Vztahy (3.2) a (3.3) spolu se statistickými předpoklady určují pravděpodobnostní formulaci systémového modelu a měřicího modelu. První je určeno pomocí stavově přechodové PDF² $f(\mathbf{x}|\mathbf{x}_{n-1})$, druhé pomocí lokální LF³ $f(\mathbf{z}_n|\mathbf{x}_n)$. Obě funkce jsou závislé na čase. Ze statistických předpokladů také plyne podmíněná nezávislost stavu \mathbf{x}_n na všech minulých měřeních $\mathbf{z}_{1:n-1}$ při daném předchozím stavu \mathbf{x}_{n-1} , tedy

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

Navíc z předpokladů plyne podmíněná nezávislost \mathbf{z}_n na všech minulých měřeních $\mathbf{z}_{1:n-1}$ při daném aktuálním stavu \mathbf{x}_n , tedy

$$f(\mathbf{z}_n|\mathbf{x}_n, \mathbf{z}_{1:n-1}) = f(\mathbf{z}_n|\mathbf{x}_n). \quad (3.5)$$

Cílem je odhadnout stav \mathbf{x}_n na základě všech minulých měření $\mathbf{z}_{1:n}$ až do času n , a to postupně (rekurzivně) s použitím výsledků obdržných v předchozích krocích. K tomu je vhodné spočítat aposteriorní hustotu pravděpodobnosti $f(\mathbf{x}_n|\mathbf{z}_{1:n})$. Z té lze následně spočíst mnoho různých odhadů stavu \mathbf{x}_n , např. odhad, který minimalizuje střední kvadratickou chybu [3], [?Sedmnactka?]:

$$\hat{\mathbf{x}}_n^{\text{MMSE}} = \mathbb{E}(\mathbf{x}_n|\mathbf{z}_{1:n}) = \int \mathbf{x}_n f(\mathbf{x}_n|\mathbf{z}_{1:n}) d\mathbf{x}_n. \quad (3.6)$$

Aposteriorní hustota pravděpodobnosti $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ může být vypočtena sekvenčně v následujících dvou krocích:

Predikce Spočteme predikci aposteriorní funkce $f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$ pomocí předchozí aposteriorní hustoty pravděpodobnosti $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ a stavově přechodové hustoty pravděpodobnosti $f(\mathbf{x}_n|\mathbf{x}_{n-1})$ následovně:

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

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

¹Bílý šum je posloupnost nekorelovaných náhodných proměnných $\{X_t\}$ s nulovým průměrem μ a s rozptylem σ^2 . Obvykle je tato skutečnost vyjadřována jako $\{X_t\} \sim \text{WN}(0, \sigma^2)$.

²Hustota pravděpodobnosti (ang. probability density function).

³Věrohodnostní funkce (ang. likelihood function).

3.3. ČÁSTICOVÉ FILTRY

Aktualizace S předchozími výsledky a s lokální LF $f(\mathbf{z}_n|\mathbf{x}_n)$ převedeme predikci $f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$ na aposteriorní funkci $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ dle Bayesovy věty:

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

Jelikož na počátku algoritmu neexistuje předchozí aposteriorní hustota, předpokládáme, že $f(\mathbf{x}_n|\mathbf{z}_{1:n})|_{n=0} = f(\mathbf{x}_0)$.

Bohužel přímý výpočet vztahů (3.7) – (3.9) nebývá jednoduchý. Analytické řešení v mnoha případech neexistuje a numerické řešení vyžaduje implementaci mnoharozměrných integrálů. Ze všech dostupných algoritmů vynikají částicové filtry, které se umí dobře vypořádat s nelinearitami, rozmanitými rozděleními i se šumem. V případech, kdy jiné algoritmy selhávají, dávají částicové filtry stále dobré výsledky. Proto o nich pojednává následující kapitola.

3.3. Částicové filtry

Částicové filtry (ang. particle filters, zkr. PF) [16], [3], [13], [10] jsou aproximací postupných Bayesovských odhadů. Spojitá rozdělení pravděpodobnosti jsou nahrazena diskrétními náhodnými měřeními, která jsou reprezentována váženými částicemi. Částice jsou vzorky neznámých stavů ze stavového prostoru a jejich váhy, pravděpodobnostní funkce, jsou počítány použitím Bayesovy věty. Korektněji:

Negaussovská aposteriorní hustota pravděpodobnosti $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ je reprezentována množinou $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$ náhodně rozmístěných částic s pozicí $\mathbf{x}_n^{(j)}$ a jim korespondujících vah s hodnotou $w_n^{(j)}$. Prvky množiny určují diskrétní aproximaci aposteriorní PDF

$$f(\mathbf{x}_n|\mathbf{z}_{1:n}) \approx \sum_{j=1}^J w_n^{(j)} \delta(\mathbf{x}_n - \mathbf{x}_n^{(j)}), \quad (3.10)$$

kde $\delta(\cdot)$ označuje multidimenzionální Diracovu delta funkci. Díky této reprezentaci lze spočítat aproximaci odhadu z (3.6), který minimalizuje střední kvadratickou chybu

$$\hat{\mathbf{x}}_n^{\text{MMSE}} \approx \hat{\mathbf{x}}_n = \sum_{j=1}^J w_n^{(j)} \mathbf{x}_n^{(j)}. \quad (3.11)$$

PF v čase n rekurzivně aktualizuje předchozí částice $\mathbf{x}_{n-1}^{(j)}$ i váhy $w_{n-1}^{(j)}$ na základě měření \mathbf{z}_n . Částice $\mathbf{x}_{n-1}^{(j)}$ jsou náhodně rozmístěny s danou hustotou pravděpodobnosti s tzv. návrhem hustoty pravděpodobnosti (ang. proposal probability density function), která je různá od aposteriorní PDF $f(\mathbf{x}_n|\mathbf{z}_{1:n})$. Predikce v (3.7), (3.8) je provedena vzorkováním z návrhu PDF $q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_n)$, čímž se získají částice, které jsou použity k aproximaci predikce aposteriorní PDF $f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$. Aktualizace v (3.9) je provedena spočtením vah $w_n^{(j)}$ částic, jež jsou získány pomocí LF $f(\mathbf{z}_n|\mathbf{x}_n)$ a výše spočítané $f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$.

3.3.1. Algoritmus částicového filtru

V čase $n = 0$ je algoritmus inicializován náhodným navzorkováním J částic $\mathbf{x}_0^{(j)}$, $j = 1, \dots, J$. Vzorkování je dáno apriorní PDF $f(\mathbf{x}_0)$. Váhy jsou zpočátku stejné $w_0^{(j)} = 1/J$. V čase $n \geq 1$ se rekurzivně provádí následující kroky:

Predikce Z návrhu PDF $q(\mathbf{x}_n|\mathbf{x}_{n-1}^{(j)}, \mathbf{z}_n) = q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_n)|_{\mathbf{x}_{n-1}=\mathbf{x}_{n-1}^{(j)}}$ je pro každou částici $\mathbf{x}_{n-1}^{(j)}$ spočtena (predikována) nová částice $\mathbf{x}_n^{(j)}$.

Aktualizace K vypočteným částicím $\mathbf{x}_n^{(j)}$ z predikčního kroku jsou spočteny váhy

$$\tilde{w}_n^{(j)} = w_{n-1}^{(j)} \frac{f(\mathbf{z}_n|\mathbf{x}_n^{(j)})f(\mathbf{x}_n^{(j)}|\mathbf{x}_{n-1}^{(j)})}{q(\mathbf{x}_n^{(j)}|\mathbf{x}_{n-1}^{(j)}, \mathbf{z}_n)}, \quad (3.12)$$

kteř jsou následně normalizovány

$$w_n^{(j)} = \frac{\tilde{w}_n^{(j)}}{\sum_{j=1}^J \tilde{w}_n^{(j)}}. \quad (3.13)$$

Získaná množina $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$ reprezentuje aposteriorní hustotu pravděpodobnosti $f(\mathbf{x}_n|\mathbf{z}_{1:n})$.

Spočtení odhadu Z množiny $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$ je spočtena minimální střední kvadratická chyba

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

Převzorkování Převzorkování množiny $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$ obdržíme z množiny $\{\mathbf{x}_n^{(j)}\}_{j=1}^J$ jejím navzorkováním a přeskládáním, kde $\mathbf{x}_n^{(j)}$ je vzorkováno s pravděpodobností $w_n^{(j)}$. Výsledkem je J převzorkovaných částic $\mathbf{x}_n^{(j)}$, jejichž váhy musí být opět totožné $w_n^{(j)} = 1/J$.

3.4. Distribuované postupné Bayesovské odhady

Distribuované postupné Bayesovské odhady tvoří teoretický základ distribuovaným částicovým filtrům, které se využívají k odhadu budoucího stavu objektu, jenž je sledován distribuovanou sítí zprostředkovatelů. V těchto sítích je měření aktuálního stavu rozptýleno mezi zprostředkovatele sítě.

Uvažujme síť K zprostředkovatelů, $K \in \mathbb{N}$. V čase n naměří k -tý zprostředkovatel vektor pozorování $\mathbf{z}_{n,k}$ příslušící ke stavu objektu \mathbf{x}_n následovně

$$\mathbf{z}_{n,k} = \mathbf{h}_{n,k}(\mathbf{x}_n, \mathbf{v}_{n,k}). \quad (3.15)$$

Zde $\mathbf{h}_{n,k}(\cdot, \cdot)$ je známá, obecně nelineární (měřicí) funkce závisující na zprostředkovateli, $\mathbf{v}_{n,k}$ je lokální šum měření, bílý šum⁴, nezávislý na minulém či přítomném stavu, nezávislý na \mathbf{u}_n z (3.2), se známou hustotou pravděpodobnosti $f(\mathbf{v}_{n,k})$.

⁴Bílý šum je posloupnost nekorelovaných náhodných proměnných $\{X_t\}$ s nulovým průměrem μ a s rozptylem σ^2 . Obvykle je tato skutečnost vyjadřována jako $\{X_t\} \sim \text{WN}(0, \sigma^2)$.

3.5. DISTRIBUOVANÉ ČÁSTICOVÉ FILTRY

Globální vektor pozorování \mathbf{z}_n ve (3.3) je určen na základě lokálních měření $\mathbf{z}_{n,k}$ od všech zprostředkovatelů. Rovnice (3.15) spolu se statistickými předpoklady určuje lokální LF $f(\mathbf{z}_n|\mathbf{x}_n)$ k -tého zprostředkovatele. Navíc, uvažujeme-li $\mathbf{v}_{n,k}$ nezávislou na lokálním šumu měření $\mathbf{v}_{n,k'}$ pro $k \neq k'$, je globální LF rovna součinu lokálních věrohodnostních funkcí

$$f(\mathbf{z}_n|\mathbf{x}_n) = \prod_{k=1}^K f(\mathbf{z}_{n,k}|\mathbf{x}_n). \quad (3.16)$$

Cílem distribuovaných postupných odhadů je zjistit \mathbf{x}_n , a to sekvenčně a na základě měření $\mathbf{z}_{n',k}$ všech zprostředkovatelů k (nebo jejich podmnožiny) pro všechny časy n' až do aktuálního času n .

3.5. Distribuované částicové filtry

Vzhledem k tomu, že jsou částicové filtry vhodné pro nelineární a negaussovské modely, jsou distribuované částicové filtry vhodnou volbou pro aproximace distribuovaných postupných Bayesovských odhadů.

V důsledku nutných aproximací nejsou algoritmy distribuovaných částicových filtrů tak přesné, jako centralizované PF. Aproximace jsou důležité z důvodů rozumné výpočetní náročnosti, stability či potřeby synchronizace náhodných generátorů.

Různé algoritmy DPF se od sebe liší výpočetní náročností, zpracováním dat, přesností odhadu, požadavky kladenými na paměť, množstvím komunikovaných dat, komunikačním dosahem, robustností stabilitou i rychlostí výpočtu. Z těchto důvodů se DPF dělí do dvou základních kategorií: První kategorie obsahuje centrální procesní jednotku, do níž jsou data odesílána, a kde se také zpracovávají, tj. je zde počítána globální LF. Jedná se o distribuované částicové filtry s fúzním centrem (ang. fusion center-based DPFs). Druhou kategorií jsou tzv. decentralizované DPF, které nemají centrum, kam data odesílaly a kde by byla také zpracovávána. Zprostředkovatelé komunikují jen mezi sebou a data jsou zpracovávána současně každým zprostředkovatelem zvlášť. Algoritmy distribuovaných částicových filtrů založené na konsenzu (ang. consensus-based DPFs), o nichž bude pojednáno dále, patří do této kategorie.

3.5.1. DPF s fúzním centrem

V algoritmech DPF s fúzním centrem [46], [23], [41] převádí každý zprostředkovatel pomocí PF své měření na lokální aposteriorní hustotu pravděpodobnosti. Ta je následně odeslána do fúzního centra, kde je spočítána globální aposteriorní hustota pravděpodobnosti, a kde je odhadnut globální stav. Výpočetní náročnost může být snížena aproximací lokální aposteriorní hustoty.

Není-li možná přímá komunikace s fúzním centrem, je možné použít tzv. multihop komunikaci, v níž jsou lokální aposteriorní hustoty přeposílány přes další zprostředkovatele. Nevýhodou tohoto typu komunikace je nutnost tvorby směrovacích tabulek (ang. routing tables) při každé změně komunikační topologie (např. v důsledku pohybu zprostředkovatelů), ale také nerovnoměrné energetické zatížení zprostředkovatelů v závislosti na jejich poloze (čím blíže k fúznímu centru jsou, tím větší množství dat musí přijmout i odeslat).

K dalšímu snížení komunikační náročnosti se používají hierarchické komunikační topologie. V nich jsou zprostředkovatelé rozděleni do skupin, kde každý člen skupiny může komunikovat jen s vedoucím zprostředkovatelem skupiny. Ti spočítají pomocí PF lokální hustotu pravděpodobnosti, kterou následně odešlou do fúzního centra. Zde je ze získaných dat odhadnut globální stav.

Simulace obsažená v příloze obsahuje DPF s fúzním centrem, do kterého jsou data odesílána každým zprostředkovatelem přímo.

3.5.2. DPF s konsenzem

Algoritmy konsenzu

V kontextu sítí zprostředkovatelů „konsenzus“ znamená shodu všech zprostředkovatelů na nějaké hodnotě, jenž závisí na datech těchto zprostředkovatelů. Algoritmy konsenzu (ang. consensus algorithms, zkr. CA) [43], [29], [30] specifikují výměnu dat mezi sousedními zprostředkovateli a dále specifikují, které výpočty jsou zprostředkovateli prováděny.

CA jsou iterativní algoritmy, které obvykle dosahují globální shody pouze přibližně. Přibližná shoda je způsobena počtem iterací (který je konečný), velikostí i topologií komunikační sítě.

Velkou výhodou CA je neexistence slabých míst, nehrozí, že zničením jediného zprostředkovatele přestane síť fungovat. Tyto algoritmy jsou robustní vzhledem ke změně topologie komunikační sítě i k jejím chybám.

Mezi nejpoužívanější typy algoritmů konsenzu patří konsenzus průměru (ang. average consensus), nebo konsenzus maxima (ang. max-consensus).

Uvažujme souvislou AN⁵, kde každý zprostředkovatel $k \in 1, \dots, K$ obsahuje skalární informaci s_k . Cílem je distribuovaný výpočet buď konsenzu průměru $\frac{1}{K} \sum_{k=1}^K s_k$, nebo konsenzu maxima $\max_{k \in 1, \dots, K} \{s_k\}$. Výsledek konsenzu je pak znám všem zprostředkovatelům.

V každé iteraci aktualizuje každý zprostředkovatel svůj vlastní stav ζ_k , na základě hodnot stavů sousedních zprostředkovatelů. Operace prováděné k -tým zprostředkovatelem jsou:

Inicializace Nastavení stavu zprostředkovatele $\zeta_k^{(0)} = s_k$.

Aktualizace Pro $i = 1, 2, \dots$ se aktualizuje stav zprostředkovatele vzhledem k $\zeta_k^{(0)} = u(\zeta_k^{(i-1)}, \{\zeta_{k'}^{(i-1)}\}_{k' \in \mathcal{N}_k})$. Zde \mathcal{N}_k je množina sousedů zprostředkovatele k , $u(\cdot)$ je aktualizující stavová funkce, která kombinuje předchozí stav $\zeta_k^{(i-1)}$ k -tého zprostředkovatele s předchozími stavy jeho sousedů $\{\zeta_{k'}^{(i-1)}\}_{k' \in \mathcal{N}_k}$. Následuje přenos stavu $\zeta_k^{(i-1)}$ všem sousedním zprostředkovatelům $k' \in \mathcal{N}_k$.

Aktualizující stavová funkce $u(\cdot)$ závisí na typu CA:

Konsenzus průměru $u(\zeta_k^{(i)}, \{\zeta_{k'}^{(i)}\}_{k' \in \mathcal{N}_k}) = \omega_{k,k}^{(i)} \zeta_k^{(i)} + \sum_{k' \in \mathcal{N}_k} \omega_{k,k'}^{(i)} \zeta_{k'}^{(i)}$, kde určení vhodných vah $\omega_{k,k}^{(i)}$ je v [43] a [29], a součet všech vah je roven jedné. Nový stav je tedy lineární kombinací předchozího stavu k -tého zprostředkovatele a předchozích stavů jeho sousedů $\{\zeta_{k'}^{(i)}\}_{k' \in \mathcal{N}_k}$. Pro $i \rightarrow \infty$ každý stav $\zeta_k^{(i)}$ konverguje k průměru s_k všech zprostředkovatelů k .

⁵Sít zprostředkovatelů (ang. agent networks), která zaznamenává stav daného objektu.

3.5. DISTRIBUOVANÉ ČÁSTICOVÉ FILTRY

Konsenzus maxima $u(\zeta_k^{(i)}, \{\zeta_{k'}^{(i)}\}_{k' \in \mathcal{N}_k}) = \max\{\{\zeta_{k'}^{(i)}\}_{k' \in \mathcal{N}_k}\}$. Každý stav $\zeta_k^{(i)}$ se stane maximem všech s_k , a to po konečném počtu iterací. Počet iterací je nejvýše roven průměru komunikačního grafu.

DPF s konsenzem

V částicových filtrech založených na konsenzu zná každý zprostředkovatel částicovou reprezentaci aposteriorní funkce hustoty pravděpodobnosti a filtrování částic je prováděno všemi zprostředkovateli současně. V ideálním případě je tato aposteriorní hustota globální aposteriorní PDF $f(\mathbf{x}_n | \mathbf{z}_{1:n})$, která zahrnuje současná i minulá měření všech zprostředkovatelů $\mathbf{z}_{1:n}$. Toho je dosaženo pomocí algoritmu decentralizovaného konsenzu, jehož cílem je shoda všech zprostředkovatelů ohledně hledaných globálních veličin. Po získání konsenzu každý zprostředkovatel vypočítá lokální aproximaci částicové reprezentace globální aposteriorní PDF $f(\mathbf{x}_n | \mathbf{z}_{1:n})$. Je zřejmé, že k použití algoritmu je nutná komunikace mezi sousedními zprostředkovateli.

Tento druh algoritmů má několik následujících výhod. Zprostředkovatelé komunikují jen se svými sousedy. Na rozdíl od některých typů DPF s fúzním centrem nepožadují existenci směrovacích tabulek ani žádnou jinou znalost komunikační sítě. Každý zprostředkovatel zná odhad globálního stavu, což je důležité obzvlášť u robotických sítí zprostředkovatelů, jejichž úkony jsou na globální stav vázány. Algoritmy konsenzu jsou robustní vzhledem ke komunikační topologii, k chybám v datech i vzhledem k poškození zprostředkovatelů.

Nevýhodou jsou vyšší požadavky na komunikaci. Čím větší síť zprostředkovatelů je, tím více iterací musí proběhnout než síť dojde ke konsenzu.

Existuje několik podkategorií algoritmů založených na konsenzu, např. výpočet částicových vah, výpočet aposteriorních parametrů, nebo výpočet parametrů LF. Popsány budou pouze dva z těchto algoritmů: výpočet částicových vah a výpočet parametrů LF, a to z důvodu použití v simulaci, která je k dispozici v příloze, a o níž je pojednáno v kapitole 5.

DPF s konsenzem a výpočtem částicových vah

V tomto algoritmu [40], [15], [7] jsou z lokálních vah $w_{n,k}^{(j)}$, které závisí na měření k -tého zprostředkovatele, distribuovaně vypočítány globální částicové váhy $w_n^{(j)}$, které odráží měření všech zprostředkovatelů. Ke každé váze $w_n^{(j)}$ je proveden jeden výpočet konsenzu průměru. Tento přístup předpokládá, že v každém ze zprostředkovatelů jsou vzorkovány identické množiny částic $\{\mathbf{x}_n^{(j)}\}_{j=1}^J$. To má za následek požadavek na synchronizaci lokálních generátorů „náhodných“ čísel všech zprostředkovatelů v síti. Tedy všichni zprostředkovatelé sítě obdrží stejné pseudonáhodné číslo. Na základě toho jsou částicové reprezentace $\{(\mathbf{x}_n^{(j)}, w_n^{(j)})\}_{j=1}^J$ předchozí globální aposteriorní PDF $f(\mathbf{x}_{n-1} | \mathbf{z}_{1:n-1})$ identické a známé všem zprostředkovatelům.

Výpočet konsenzu globálních vah je založen na faktorizaci (3.16) globálních věrohodnostních funkcí $f(\mathbf{z}_n | \mathbf{x}_n)$. Použitím (3.16) a (3.12) lze zjistit, že j -tá globální váha $w_n^{(j)}$ je úměrná součinu lokálních věrohodnostních funkcí pro $\mathbf{x}_n^{(j)}$:

$$w_n^{(j)} \propto \tilde{w}_n^{(j)} = f(\mathbf{z}_n | \mathbf{x}_n^{(j)}) = \prod_{k=1}^K f(\mathbf{z}_{n,k} | \mathbf{x}_n^{(j)}). \quad (3.17)$$

V tomto případě si můžeme dovolit ignorovat zbylé členy vyskytující se ve výrazu (3.12): $w_{n-1}^{(j)}$, $f(\mathbf{x}_n^{(j)}|\mathbf{x}_{n-1}^{(j)})$ a $q(\mathbf{x}_n^{(j)}|\mathbf{x}_{n-1}^{(j)}, \mathbf{z}_n)$, protože jsou lokálně dostupné každému zprostředkovateli [15]. Zlogaritmování výrazu (3.17) získáme výraz

$$\log(\tilde{w}_n^{(j)}) = \sum_{k=1}^K \log f(\mathbf{z}_{n,k}|\mathbf{x}_n^{(j)}), \quad (3.18)$$

v němž je suma přes všechny zprostředkovatele spočtena distribuovaně, konsenzem. Každý zprostředkovatel lokálně vypočítá $\log f(\mathbf{z}_{n,k}|\mathbf{x}_n^{(j)})$ na základě předpokladu, že $\mathbf{x}_n^{(j)}$ jsou pro všechny zprostředkovatele identické. Následně je spočtena aproximace $\tilde{A}(\mathbf{z}_n, \mathbf{x}_n^{(j)})$ průměru

$$A(\mathbf{z}_n, \mathbf{x}_n^{(j)}) = \frac{1}{K} \sum_{k=1}^K \log f(\mathbf{z}_{n,k}|\mathbf{x}_n^{(j)}) \quad (3.19)$$

a to tak, že je pro každou částici $\mathbf{x}_n^{(j)}$ proveden CA. Jelikož máme k dispozici konečný počet iterací, jedná se opravdu o aproximaci. S pomocí vztahů (3.18) (3.17) získáme aproximaci $\tilde{w}_n^{(j)}$ následovně:

$$KA(\mathbf{z}_n, \mathbf{x}_n^{(j)}) = \log(\tilde{w}_n^{(j)}), \quad (3.20)$$

$$K\tilde{A}(\mathbf{z}_n, \mathbf{x}_n^{(j)}) \approx \log(\tilde{w}_n^{(j)}), \quad (3.21)$$

$$\exp(K\tilde{A}(\mathbf{z}_n, \mathbf{x}_n^{(j)})) \approx \tilde{w}_n^{(j)}. \quad (3.22)$$

Jelikož je K všem zprostředkovatelům známé (je buď dané, nebo ho lze zjistit distribuovaným algoritmem [28]), může každý vypočítat aproximaci $\tilde{w}_n^{(j)}$. Následný odhad globálních vah $w_n^{(j)}$ získáme z $\tilde{w}_n^{(j)}$ jednoduše normalizací

$$w_n^{(j)} = \frac{\tilde{w}_n^{(j)}}{\sum_{j'=1}^J \tilde{w}_n^{(j')}}. \quad (3.23)$$

Vypočtené globální váhy $w_n^{(j)}$ nejsou u každého zprostředkovatele stejné následkem konečného počtu iterací konsenzu. Abychom dosáhli identických vah, je nutné např. použít konsenzus maxima, konsenzus minima a spočítat jejich průměr. Tedy pro každé $w_n^{(j)}$ je spuštěn jeden konsenzus pro najetí maxima a jeden konsenzus pro najetí minima. Počet iterací pro oba konsenzusy je roven dvojnásobku průměru komunikačního grafu. Průměr maxima a minima nám zaručí stejné váhy pro všechny zprostředkovatele. Zajištění identických množin vah ve všech zprostředkovatelích umožní, že budou vzorkovány stejné částice.

DPF s konsenzem a výpočtem parametrů věrohodnostní funkce

DPF s konsenzem a výpočtem částicových vah využívá algoritmus konsenzu k fúzi aproximací lokálních aposteriorních hustot pravděpodobnosti do aproximace globální aposteriorní PDF $f(\mathbf{x}_n|\mathbf{z}_{1:n})$. DPF s konsenzem a výpočtem parametrů LF [20], [19], [36] je využíván k určení aproximace globální LF $f(\mathbf{z}_n|\mathbf{x}_n)$, která je získána v každém zprostředkovateli jako funkce \mathbf{x}_n . To každému zprostředkovateli umožňuje vypočítat globální LF

3.5. DISTRIBUOVANÉ ČÁSTICOVÉ FILTRY

na pozicích částic $\mathbf{x}_n^{(j)}$, což je vyžadováno pro spočtení vah $w_n^{(j)}$ v aktualizacním kroku PF.

Každý zprostředkovatel spouští lokálně algoritmus PF, který je ekvivalentní globálnímu PF, protože využívá globální LF a vypočítává globální odhad zahrnující vektor měření \mathbf{z}_n od všech sensorů. Lokální PF provádí výpočty nezávisle na ostatních částicových filtrech, pouze globální LF je spočítána kooperativně. V tomto případě není nutná synchronizace generátorů náhodných čísel. Globální odhad, který je získán každým zprostředkovatelem, se může mírně lišit, a to právě kvůli nesynchronizaci náhodných generátorů čísel a také kvůli nedostatečnému zkonvergování algoritmu konsenzu.

Globální LF lze získat na základě konsenzu jen v případě, existuje-li postačující statistika s danou strukturou [20]. Pokud globální LF tuto strukturu nemá, je možné spočítat její aproximaci pomocí věrohodnostního konsenzu (ang. likelihood consensus, zkr. LC). Tento konenzus je založen na faktorizaci globální LF (3.16), který předpokládá statistickou nezávislost lokálního šumu měření $\mathbf{v}_{n,k}$. Použitím logaritmu na výraz (3.16) získáváme

$$\log f(\mathbf{z}_n|\mathbf{x}_n) = \sum_{k=1}^K \log f(\mathbf{z}_{n,k}|\mathbf{x}_n). \quad (3.24)$$

Výpočet této sumy není možný pomocí konsenzu, protože obecně závisí lokální log-LF $\log f(\mathbf{z}_{n,k}|\mathbf{x}_n)$ na neznámém stavu \mathbf{x}_n . Kvůli tomu je využívána následující aproximace bázovými funkcemi

$$\log f(\mathbf{z}_{n,k}|\mathbf{x}_n) \approx \sum_{r=1}^R \alpha_{n,k,r}(\mathbf{z}_{n,k}) \varphi_{n,r}(\mathbf{x}_n). \quad (3.25)$$

Zde $\alpha_{n,k,r}(\mathbf{z}_{n,k})$ jsou expanzní koeficienty obsahující všechny lokální informace zprostředkovatele včetně lokálního měření $\mathbf{z}_{n,k}$, funkce $\varphi_{n,r}(\mathbf{x}_n)$ jsou pevně dané bázové funkce nezávislé na zprostředkovateli, které jsou všem zprostředkovatelům známé. Vložení vztahu (3.25) do vztahu (3.24) a následné exponování vede k aproximaci globální LF ve tvaru:

$$f(\mathbf{z}_n|\mathbf{x}_n) \approx \exp\left(\sum_{r=1}^R \alpha_{n,r}(\mathbf{z}_n) \varphi_{n,r}(\mathbf{x}_n)\right), \quad (3.26)$$

kde

$$\alpha_{n,r}(\mathbf{z}_n) = \sum_{k=1}^K \alpha_{n,k,r}(\mathbf{z}_{n,k}). \quad (3.27)$$

Zprostředkovatel může díky znalosti $\alpha_{n,r}(\mathbf{z}_{n,k})$ vypočítat aproximaci globální LF $f(\mathbf{z}_n|\mathbf{x}_n)$ pro všechny hodnoty \mathbf{x}_n . Navíc, každý koeficient $\alpha_{n,r}(\mathbf{z}_{n,k})$ je dán sumou přes všechny zprostředkovatele, takže člen $\alpha_{n,k,r}(\mathbf{z}_{n,k})$ odpovídající k -tému zprostředkovateli závisí pouze na lokálním měření $\mathbf{z}_{n,k}$. Pak $\alpha_{n,r}(\mathbf{z}_n)$ může být vypočítána distribuovaně algoritmem konsenzu.

4. Model systému

V modelu uvažujeme síť senzorů sledující daný objekt. Tento objekt má neznámý a časově proměnný stav $\mathbf{x}_n \in \mathbb{R}^d$, kde $n \in \mathbb{N}_0$ je diskrétní časový index. Ve stavovém vektoru \mathbf{x}_n jsou obsaženy parametry jako je poloha či rychlost objektu. Předpokládáme, že známe stavově přechodovou hustotu pravděpodobnosti $f(\mathbf{x}_n|\mathbf{x}_{n-1})$.

Objekt je sledovaný decentralizovanou distribuovanou sítí S senzorů, $s = 1, \dots, S$. Každý senzor s může komunikovat s podmnožinou ostatních senzorů $\mathcal{N}_s \subseteq \{1, \dots, S\} \setminus \{s\}$, se sousedy senzoru s . Počet sousedů \mathcal{N}_s závisí na pokrytí oblasti komunikačním systémem. Předpokládáme, že graf odpovídající komunikační síti je souvislý (viz definice 3.5).

V čase n senzor s naměří $M_n^{(s)}$ hodnot $\mathbf{z}_{n,m}^{(s)}$, $m = 1, \dots, M_n^{(s)}$, může se jednat např. o vzdálenost od senzoru a úhel k danému bodu. Tato měření jsou uvažována jako podmíněně statisticky nezávislá přes senzory s při daném stavu objektu \mathbf{x}_n . Dále definujeme $\mathbf{z}_n^{(s)} = [\mathbf{z}_{n,1}^{(s)T}, \dots, \mathbf{z}_{n,M_n^{(s)}}^{(s)T}]^T$, $\mathbf{z}_n = [\mathbf{z}_n^{(1)T}, \dots, \mathbf{z}_n^{(S)T}]^T$ a $\mathcal{M}_n^{(s)} = \{1, \dots, M_n^{(s)}\}$.

U měření $\mathbf{z}_{n,m}^{(s)}$ uvažujeme dva typy nejistot: Prvním je falešné měření neboli šum (ang. clutter), které nastává, když měření není generováno sledovaným objektem. Druhým typem nejistoty je případ, kdy senzor objekt vůbec nezachytí. Protože je objekt ve sledované oblasti pouze jeden, předpokládáme, že senzor s naměří nejvýše jednu hodnotu odpovídající objektu [4], [27]. Ostatní hodnoty jsou šum.

Souvislost mezi objektem a měřením senzoru s v čase n popisuje množina $a_n^{(s)} \in \{0, 1, \dots, M_n^{(s)}\}$, kterou definujeme

$$a_n^{(s)} = \begin{cases} m \in \mathcal{M}_n^{(s)}, & \text{jestliže v čase } n \text{ objekt generuje měření } \mathbf{z}_{n,m}^{(s)} \text{ v senzoru } s, \\ 0, & \text{jestliže v čase } n \text{ objekt negeneruje měření v senzoru } s. \end{cases} \quad (4.1)$$

Tedy $a_n^{(s)}$ je rovno indexu m z měření $\mathbf{z}_{n,m}^{(s)}$, pokud je toto měření generováno objektem. V tomto případě říkáme, že objekt byl detekován senzorem s a pravděpodobnost detekce $P_d^{(s)}$ objektu senzorem s známe. Jsou-li všechna měření $\mathbf{z}_{n,m}^{(s)}$ šum, pak klademe $a_n^{(s)} = 0$. Předpokládáme, že počet falešných měření v senzoru s je dán Poissonovým rozdělením se střední hodnotou $\mu^{(s)}$, která se v čase nemění. Z uvedených předpokladů plyne, že sdružená pravděpodobnostní funkce náhodných veličin $a_n^{(s)}$ a $M_n^{(s)}$ má tvar

$$p(a_n^{(s)}, M_n^{(s)}) = \begin{cases} C_1(M_n^{(s)})P_d^{(s)}, & a_n^{(s)} \in \mathcal{M}_n^{(s)}, \\ C_1(M_n^{(s)})(1 - P_d^{(s)})\mu^{(s)}, & a_n^{(s)} = 0, \end{cases} \quad (4.2)$$

kde $C_1(M_n^{(s)})$ je normalizační konstanta, která závisí pouze na $M_n^{(s)}$. Dále definujeme $\mathbf{a}_n = [a_n^{(1)}, \dots, a_n^{(S)}]^T$.

Je-li objekt detekován senzorem s , pak je rozdělení $\mathbf{z}_{n,m}^{(s)}$ při daném stavu \mathbf{x}_n popsáno podmíněnou PDF $f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)} = m)$, o níž předpokládáme, že je známá. Na druhé straně, každé falešné měření senzoru s má rozdělení dle PDF $f_c(\mathbf{z}_{n,m}^{(s)})$, o kterém také předpokládáme, že je známé. Z toho plyne

$$f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)}) = \begin{cases} f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)} = m), & a_n^{(s)} = m \in \mathcal{M}_n^{(s)}, \\ f_c(\mathbf{z}_{n,m}^{(s)}), & a_n^{(s)} = 0. \end{cases} \quad (4.3)$$

Měřící proces je pak popsán podmíněnou PDF $f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)})$.

4.1. Lokální věrohodnostní funkce

V této kapitole bude odvozena lokální LF, která vyjadřuje statistickou závislost mezi lokálním vektorem měření $\mathbf{z}_n^{(s)}$ a stavem objektu \mathbf{x}_n . Nejdříve bude pro pevně daný senzor s vypočtena podmíněná PDF $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)}, M_n^{(s)})$. Jelikož jsou měření $\mathbf{z}_{n,m}^{(s)}$ podmíněně nezávislá přes všechna m vzhledem k danému stavu \mathbf{x}_n a daným náhodným veličinám $a_n^{(s)}$ a $M_n^{(s)}$, lze psát

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)}, M_n^{(s)}) = \prod_{m \in \mathcal{M}_n^{(s)}} f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)}), \quad (4.4)$$

kde funkce $f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)})$ je určena vztahem (4.3).

K dalšímu postupu je nutné rozlišit případy $a_n^{(s)} = m \in \mathcal{M}_n^{(s)}$ a $a_n^{(s)} = 0$:

- Je-li $a_n^{(s)} = m \in \mathcal{M}_n^{(s)}$, pak v čase n objekt generuje měření $\mathbf{z}_{n,m}^{(s)}$ v senzoru s a zbývajících $M_n^{(s)} - 1$ měření $\mathbf{z}_{n,m'}^{(s)}$, $m' \in \mathcal{M}_n^{(s)} \setminus \{m\}$ tvoří šum. Pro $a_n^{(s)} = m \in \mathcal{M}_n^{(s)}$ pomocí (4.3) a (4.4) obdržíme:

$$\begin{aligned} f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)} = m, M_n^{(s)}) &= f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)} = m) \prod_{m' \in \mathcal{M}_n^{(s)} \setminus \{m\}} f_c(\mathbf{z}_{n,m'}^{(s)}) \\ &= f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)} = m) \frac{f_c(\mathbf{z}_n^{(s)}|M_n^{(s)})}{f_c(\mathbf{z}_{n,m}^{(s)})}, \end{aligned} \quad (4.5)$$

kde

$$f_c(\mathbf{z}_n^{(s)}|M_n^{(s)}) = \prod_{m \in \mathcal{M}_n^{(s)}} f_c(\mathbf{z}_{n,m}^{(s)}). \quad (4.6)$$

- Je-li $a_n^{(s)} = 0$, pak jsou všechna měření šum a ze vztahů (4.3) a (4.4) obdržíme:

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)} = 0, M_n^{(s)}) = f_c(\mathbf{z}_n^{(s)}|M_n^{(s)}). \quad (4.7)$$

S využitím věty o úplné pravděpodobnosti je odvozena lokální LF senzoru s

$$f(\mathbf{z}_n^{(s)}, M_n^{(s)}|\mathbf{x}_n) = \psi(\mathbf{z}_n^{(s)}, M_n^{(s)}) \sum_{a_n^{(s)}=0}^{M_n^{(s)}} f(\mathbf{z}_n^{(s)}|\mathbf{x}_n, a_n^{(s)}, M_n^{(s)}) p(a_n^{(s)}, M_n^{(s)}), \quad (4.8)$$

kde

$$\psi(\mathbf{z}_n^{(s)}, M_n^{(s)}) = \begin{cases} 1, & \text{je-li } M_n^{(s)} \text{ rovno počtu subvektorů } \mathbf{z}_{n,m}^{(s)} \text{ v } \mathbf{z}_n^{(s)}, \\ 0, & \text{jinak.} \end{cases} \quad (4.9)$$

V dalším bude uvažován pouze případ, kdy $M_n^{(s)}$ odpovídá počtu subvektorů $\mathbf{z}_{n,m}^{(s)}$ v $\mathbf{z}_n^{(s)}$, tedy $\psi(\mathbf{z}_n^{(s)}, M_n^{(s)}) = 1$. To může nastat v případě, je-li $M_n^{(s)}$ deterministicky určená $\mathbf{z}_n^{(s)}$, tedy když $M_n^{(s)}$ není náhodná proměnná. Navíc, závisí-li $M_n^{(s)}$ deterministicky na $\mathbf{z}_n^{(s)}$, můžeme místo $f(\mathbf{z}_n^{(s)}, M_n^{(s)}|\mathbf{x}_n)$ psát jen $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$.

Zbývá rozdělit sumu (4.8) na část platící pro $a_n^{(s)} = 0$ a část platící pro $a_n^{(s)} \in \mathcal{M}_n^{(s)}$. Následným použitím výrazů (4.2), (4.5) a (4.6) obdržíme lokální LF ve tvaru:

$$f(\mathbf{z}_n^{(s)} | \mathbf{x}_n) = C_2(\mathbf{z}_n^{(s)}) \left[(1 - P_d^{(s)})\mu^{(s)} + P_d^{(s)} \sum_{m \in \mathcal{M}_n^{(s)}} \frac{f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m)}{f_c(\mathbf{z}_{n,m}^{(s)})} \right], \quad (4.10)$$

kde $C_2(\mathbf{z}_n^{(s)}) = C_1(M_n^{(s)})f_c(\mathbf{z}_n^{(s)} | M_n^{(s)})$.

4.2. Globální věrohodnostní funkce

V předchozí podkapitole byla odvozena lokální LF. Každému senzoru je známa jeho vlastní lokální LF, která zahrnuje lokální měření $\mathbf{z}_n^{(s)}$ daného senzoru. Jak ale bylo ukázáno v kapitole 3, k odhadnutí následujícího stavu systému je nutné znát globální LF $f(\mathbf{z}_n | \mathbf{x}_n)$.

4.2.1. Odvození globální LF

Jelikož jsou vektory měření $\mathbf{z}_n^{(s)}$ různých senzorů s podmíněně nezávislé vzhledem k danému \mathbf{x}_n je globální LF dána součinem:

$$f(\mathbf{z}_n | \mathbf{x}_n) = \prod_{s=1}^S f(\mathbf{z}_n^{(s)} | \mathbf{x}_n). \quad (4.11)$$

Aby bylo možné vypočítat $f(\mathbf{z}_n | \mathbf{x}_n)$ distribuovaně pomocí konsenzu průměru, je nutné převést součin (4.11) na sumu. Toho lze dosáhnout zlogaritmováním součinu:

$$f(\mathbf{z}_n | \mathbf{x}_n) = \exp(\ln f(\mathbf{z}_n | \mathbf{x}_n)) = \exp(Sl(\mathbf{x}_n)), \quad (4.12)$$

kde $l(\mathbf{x}_n) = \ln f(\mathbf{z}_n | \mathbf{x}_n)/S$. Pomocí (4.11) získáme průměr S lokálních logaritmovaných věrohodnostních funkcí (zkr. log-LF) $\ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n)$, $s = 1, \dots, S$, které jsou funkcemi $\mathbf{x}_n \in \mathbb{R}^d$

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

Kvůli závislosti na \mathbf{x}_n nelze tento průměr spočítat přímo pomocí algoritmu konsenzu průměru. Algoritmus je určen k průměrování čísel a ne funkcí [29]. Proto je v LC¹ [17] každá lokální log-LF aproximována pomocí funkčního rozvoje konečného řádu. Konsenzus průměru je následně použit na koeficienty tohoto rozvoje. Nicméně, ve zde uvedeném případě má lokální LF vlastnosti umožňující dále popsané modifikace LC.

4.2.2. Oddělení dolní části globální log-LF

Lokální LF $f(\mathbf{z}_n^{(s)} | \mathbf{x}_n)$ ze vztahu (4.10) se skládá s následujících částí:

- konstantní člen odpovídající situaci $a_n^{(s)} = 0$

$$C_2(\mathbf{z}_n^{(s)})(1 - P_d^{(s)})\mu^{(s)}; \quad (4.14)$$

¹Věrohodnostní konsenzus (ang. likelihood consensus).

4.2. GLOBÁLNÍ VĚROHODNOSTNÍ FUNKCE

- $M_n^{(s)}$ kladných členů

$$C_2(\mathbf{z}_n^{(s)})P_d^{(s)} \frac{f(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_n, a_n^{(s)} = m)}{f_c(\mathbf{z}_{n,m}^{(s)})} \quad m = 1, \dots, M_n^{(s)}, \quad (4.15)$$

kde každý odpovídá jednomu měření $\mathbf{z}_{n,m}^{(s)}$.

Konstantní člen budeme nazývat dolní částí lokální LF $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ a členy související s měřením budeme nazývat mody lokální LF $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$. Každý modus odpovídá měření $\mathbf{z}_{n,m}^{(s)}$. V tomto modelu jsou pro pevně daný senzor s buď všechna měření šum, a nebo je jedno měření generováno objektem a zbylá jsou šum. Modus odpovídající měření, která byla generována objektem, budeme nazývat modem objektu (ang. object mode), modus odpovídající měření, která byla generována šumem, budeme nazývat mody šumu (ang. clutter modes). Ačkoli nevíme, který modus je modem objektu a který modem šumu, je vidět, že podobnou strukturu „dolní část plus násobky modů“ vykazuje log-LF $\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$. Lze ukázat, že počet modů je dán součtem $\mu^{(s)} + P_d^{(s)}$.

Jelikož jsou mody kladné, tvoří dolní část minimum lokální LF:

$$\min_{\mathbf{x}_n \in \mathbb{R}^d} \{f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)\} = C_2(\mathbf{z}_n^{(s)})(1 - P_d^{(s)})\mu^{(s)}. \quad (4.16)$$

A protože je \ln rostoucí funkcí, nemůže být lokální log-LF $C_2(\mathbf{z}_n^{(s)})(1 - P_d^{(s)})\mu^{(s)}$ menší než

$$\lambda^{(s)} = \min_{\mathbf{x}_n \in \mathbb{R}^d} \{\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)\} = \ln(C_2(\mathbf{z}_n^{(s)})(1 - P_d^{(s)})\mu^{(s)}). \quad (4.17)$$

Pak může být $\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ zapsána jako

$$\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) = \lambda^{(s)} + \Lambda^{(s)}(\mathbf{x}_n), \quad (4.18)$$

kde $\Lambda^{(s)}(\mathbf{x}_n) \geq 0$. Průměr v (4.13) tedy přechází na:

$$l(\mathbf{x}_n) = \frac{1}{S} \sum_{s=1}^S (\lambda^{(s)} + \Lambda^{(s)}(\mathbf{x}_n)) = \bar{\lambda} + \bar{\Lambda}(\mathbf{x}_n). \quad (4.19)$$

Dolní část globální log-LF má tedy tvar

$$\bar{\lambda} = \frac{1}{S} \sum_{s=1}^S \lambda^{(s)} \quad (4.20)$$

a druhý člen v součtu (4.19), který již není dolní částí, je globální log-LF tvaru

$$\bar{\Lambda}(\mathbf{x}_n) = \frac{1}{S} \sum_{s=1}^S \Lambda^{(s)}(\mathbf{x}_n). \quad (4.21)$$

Z rovnic (4.12) a (4.19) získáme vztah pro globální LF ve tvaru:

$$f(\mathbf{z}_n|\mathbf{x}_n) = \exp(S(\bar{\lambda} + \bar{\Lambda}(\mathbf{x}_n))). \quad (4.22)$$

Nyní přistoupíme k distribuovanému výpočtu globální LF $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$: konstantní dolní část globální log-LF $\bar{\lambda}$ je vypočtena pomocí konsenzu průměru, jelikož se jedná o skalár, zatímco $\bar{\Lambda}(\mathbf{x}_n)$ je určena LC.

Navíc může být část lokální LF $\Lambda^{(s)}(\mathbf{x}_n)$ vypočtena lokálně v senzoru s ze vztahu (4.18):

$$\Lambda^{(s)}(\mathbf{x}_n) = \ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) - \lambda^{(s)}. \quad (4.23)$$

Tato funkce se skládá z $M_n^{(s)}$ modů, z nichž každý odpovídá měření $\mathbf{z}_{n,m}^{(s)}$. Tyto mody jsou buď všechny mody šumu, a nebo je jeden mod objektu a zbývající jsou mody šumu.

4.2.3. Distribuovaný výpočet dolní části globální log-LF

Každý senzor s vypočítá $\lambda^{(s)}$ a to dle vztahu (4.17). Cílem této kapitoly je distribuovaně určit průměr $\bar{\lambda}$ všech $\lambda^{(s)}$ tak, aby byl tento průměr známý všem senzorům. Toho lze dosáhnout dvěma způsoby:

Fúzním centrem Dolní části lokálních log-LF $\lambda^{(s)}$ jsou odeslány přímo do výpočetního centra, kde jsou zprůměrovány a výsledek je odeslán zpět jednotlivým senzorům.

Konsenzem Konkrétněji konsenzem průměru. V něm se využívá jen komunikace se sousedními senzory $s' \in \mathcal{N}_s$. Každému senzoru s je znám jeho vlastní stav, který aktualizuje po každé iteraci konsenzu. O této metodě bude pojednáno dále.

Vlastní stav senzoru s inicializujeme jako $u_0^{(s)} = \lambda^{(s)}$. V iteraci i odešle senzor s svůj předchozí vlastní stav $u_{i-1}^{(s)}$ sousedním senzorům $s' \in \mathcal{N}_s$. Zároveň od nich obdrží jejich předchozí vlastní stavy $u_{i-1}^{(s')}$. Senzor s následně aktualizuje svůj vlastní stav dle

$$u_i^{(s)} = \sum_{s' \in \{s\} \cup \mathcal{N}_s} \gamma_{s,s'} u_{i-1}^{(s')}, \quad i = 1, 2, \dots, \quad (4.24)$$

kde $\gamma_{s,s'}$ označuje váhy splňující

$$\sum_{s' \in \{s\} \cup \mathcal{N}_s} \gamma_{s,s'} = 1. \quad (4.25)$$

Jelikož předpokládáme, že je komunikační síť souvislá, pak je zaručeno [43], že pro vhodné zvolené váhy $\gamma_{s,s'}$ a pro $i \rightarrow \infty$ konsenzus (4.24) konverguje k $\bar{\lambda}$.

Je zřejmé, že v praxi nikdy nebude dosaženo nekonečného množství iterací pro dokonalý konsenzus. Množství provedených iterací n_{it} (kterých je konečně mnoho) je určeno buď předem, a nebo za běhu algoritmu poté, co je dosažena rozumná konvergence u všech senzorů. Výsledný konečný vlastní stav $u_{n_{it}}^{(s)}$ senzoru s pak aproximuje dolní část globální log-LF $\bar{\lambda}$

$$\bar{\lambda} \approx u_{n_{it}}^{(s)}. \quad (4.26)$$

Možnou volbou vah jsou Metropolis váhy, které jsou dané vztahem [43]

$$\gamma_{s,s'} = \begin{cases} \frac{1}{1 + \max\{|\mathcal{N}_s|, |\mathcal{N}_{s'}|\}}, & s' \neq s, \\ 1 - \sum_{\nu \in \mathcal{N}_s} \gamma_{s,\nu}, & s' = s. \end{cases} \quad (4.27)$$

4.3. DISTRIBUOVANÝ VÝPOČET ZBYLÉ ČÁSTI GLOBÁLNÍ LOG-LF

Tato volba vyžaduje, aby každý senzor znal nejen počet svých sousedů, ale i počet sousedů všech svých sousedů: $|\mathcal{N}_{s'}|$ pro $s' \in \{s\} \cup \mathcal{N}_s$. Navíc je nutné, aby iterace konsenzu probíhaly synchronizovaně ve všech senzorech.

4.3. Distribuovaný výpočet zbylé části globální log-LF

Každý senzor s umí lokálně spočítat $\Lambda^{(s)}(\mathbf{x}_n)$ dle vztahu (4.23). Nyní je potřeba vypočítat průměr všech $\Lambda^{(s)}(\mathbf{x}_n)$, tedy zbylou část globální log-LF $\bar{\Lambda}(\mathbf{x}_n)$ určenou vztahem (4.21). Tento průměr nemůže být kvůli závislosti na \mathbf{x}_n spočítán přímo, může být ale vypočítán pomocí LC².

4.3.1. Věrohodnostní konsenzus

V algoritmu LC je každá lokální funkce $\Lambda^{(s)}(\mathbf{x}_n)$ aproximována rozvojem funkcí konečného řádu, v našem případě:

$$\Lambda^{(s)}(\mathbf{x}_n) \approx \sum_{b=1}^{n_\psi} \alpha_b^{(s)} \psi_b(\mathbf{x}_n), \quad (4.28)$$

kde funkce $\{\psi_b(\cdot)\}_{b=1}^{n_\psi}$ jsou pevně dané funkce, které jsou identické pro všechny senzory, a které jsou všem sensorům známé. Množina těchto funkcí se nazývá slovník a její členy se nazývají atomy [8], [14]. Zde je jistá nejednoznačnost v názvosloví, kdy se pojem slovníku používá i pro již navzorkované funkce naskládáné jako sloupce (atomy) matice slovníku.

Protože každý senzor s aproximuje funkci závislejší na lokálním měření $\mathbf{z}_n^{(s)}$, je zřejmé, že expanzní koeficienty $\{\alpha_b(\cdot)\}_{b=1}^{n_\psi}$ jsou pro každý senzor s různé.

Vložení aproximace (4.28) do vztahu pro průměr (4.21) a záměnou pořadí sumací získáme vztah pro výpočet zbylé části globální log-LF ve tvaru

$$\bar{\Lambda}^{(s)}(\mathbf{x}_n) \approx \frac{1}{S} \sum_{s=1}^S \sum_{b=1}^{n_\psi} \alpha_b^{(s)} \psi_b(\mathbf{x}_n) = \sum_{b=1}^{n_\psi} \alpha_b \psi_b(\mathbf{x}_n) \quad (4.29)$$

s globálními expanzními koeficienty

$$\alpha_b = \frac{1}{S} \sum_{s=1}^S \alpha_b^{(s)}, \quad b = 1, \dots, n_\psi. \quad (4.30)$$

Ty mohou být vypočítány opět dvěma způsoby:

Fúzním centrem Expanzní koeficienty $\{\alpha_b(\cdot)\}_{b=1}^{n_\psi}$ jsou odeslány přímo do výpočetního centra, kde jsou zprůměrovány. Výsledné α_b jsou poté odeslány zpět všem S sensorům.

Konsenzem Expanzní koeficienty α_b mohou být vypočítány paralelně, pro každý koeficient je spuštěn jeden konsenzus průměru. Dohromady je provedeno n_ψ konsenzů. Tato metoda je rozepsána dále.

²Věrohodnostní konsenzus (ang. likelihood consensus).

Na rozdíl od výše uvedeného algoritmu konsenzu průměru zde každý senzor s aktualizuje n_ψ vlastních stavů $y_i^{(s,b)}$. Stavů jsou inicializovány jako

$$y_0^{(s,b)} = \alpha_b^{(s)}. \quad (4.31)$$

Po konečném počtu iterací m_{it} konsenzu aproximují vlastní stavy $y_{m_{\text{it}}}^{(s,b)}$ globální expanzní koeficienty a_b dané vztahem (4.30)

$$a_b \approx y_{m_{\text{it}}}^{(s,b)}. \quad (4.32)$$

Pomocí vztahu (4.29) získáváme výpočet zbylé části globální log-LF ve tvaru

$$\bar{\Lambda}^{(s)}(\mathbf{x}_n) \approx \sum_{b=1}^{n_\psi} y_{m_{\text{it}}}^{(s,b)} \psi_b(\mathbf{x}_n). \quad (4.33)$$

Nyní již můžeme zapsat vztah pro výpočet aproximace globální LF. Vložíme-li výrazy (4.26) a (4.33) do výrazu (4.22) získáváme aproximaci globální LF

$$f(\mathbf{z}_n | \mathbf{x}_n) \approx \hat{f}_s(\mathbf{z}_n | \mathbf{x}_n) = \exp \left(S \left(u_{n_{\text{it}}}^{(s)} + \sum_{b=1}^{n_\psi} y_{m_{\text{it}}}^{(s,b)} \psi_b(\mathbf{x}_n) \right) \right). \quad (4.34)$$

Aby mohl výraz vypočítán, je nutné, aby každý senzor s znal počet všech senzorů S . Toho lze dosáhnout pomocí distribuovaného výpočtu [28]. Jelikož jsou k výpočtům využívány částicové filtry, není známa přesná globální LF ($\mathbf{z}_n | \mathbf{x}_n$), ale pouze její aproximace vyčíslená na J částicích $\mathbf{x}_n^{(s,j)}$. Taková aproximace globální LF bude nazývána vzorkovaná globální LF a bude značena $f(\mathbf{z}_n | \mathbf{x}_n^{(s,j)})$. S ohledem na výraz (4.34) získáme vzorkovanou globální LF vztahem

$$\hat{f}_s(\mathbf{z}_n | \mathbf{x}_n^{(s,j)}) = \exp \left(S \left(u_{n_{\text{it}}}^{(s)} + \sum_{b=1}^{n_\psi} y_{m_{\text{it}}}^{(s,b)} \psi_b(\mathbf{x}_n^{(s,j)}) \right) \right). \quad (4.35)$$

4.3.2. Výpočet expanzních koeficientů

Lokální vektor expanzních koeficientů $\boldsymbol{\alpha}^{(s)} = [\alpha_1^{(s)}, \dots, \alpha_{n_\psi}^{(s)}]^T$ je spočítán každým senzorem s zvlášť a to způsoby uvedenými dále.

Nejdříve je nutné zavést značení pro funkce vypočtené na částicích $\{\mathbf{x}_n^{(s,j)}\}_{j=1}^J$. Zbylá část lokální log-LF $\Lambda^{(s)}(\mathbf{x}_n)$ vypočtená na výše uvedených J částicích $\mathbf{x}_n^{(s,j)}$ bude značena $\Lambda^{(s)}(\mathbf{x}_n^{(s,j)})$, dále funkce $\psi_b(\mathbf{x}_n)$, $b = 1, \dots, n_\psi$, vypočtené na daných částicích budou značeny $\psi_b(\mathbf{x}_n^{(s,j)})$. Také zavedeme vektor $\boldsymbol{\beta}^{(s)}$ a matici $\boldsymbol{\Psi}^{(s)}$ jako

$$\boldsymbol{\beta}^{(s)} = \left[\Lambda^{(s)}(\mathbf{x}_n^{(s,1)}), \dots, \Lambda^{(s)}(\mathbf{x}_n^{(s,J)}) \right]^T, \quad (4.36)$$

$$\boldsymbol{\Psi}^{(s)} = \begin{bmatrix} \psi_1(\mathbf{x}_n^{(s,1)}) & \cdots & \psi_{n_\psi}(\mathbf{x}_n^{(s,1)}) \\ \vdots & \ddots & \vdots \\ \psi_1(\mathbf{x}_n^{(s,J)}) & \cdots & \psi_{n_\psi}(\mathbf{x}_n^{(s,J)}) \end{bmatrix}. \quad (4.37)$$

Vypočtený vektor expanzních koeficientů označíme $\hat{\boldsymbol{\alpha}}^{(s)} = [\hat{\alpha}_1^{(s)}, \dots, \hat{\alpha}_{n_\psi}^{(s)}]^T$. Prvky vektoru jsou následně substituovány za $\alpha_b^{(s)}$ ze vztahu (4.31). Inicializace LC tedy přejde v rovnost

$$y_0^{(s,b)} = \hat{\alpha}_b^{(s)}. \quad (4.38)$$

4.3. DISTRIBUOVANÝ VÝPOČET ZBYLÉ ČÁSTI GLOBÁLNÍ LOG-LF

Metoda nejmenších čtverců

K získání aproximace $\Lambda^{(s)}(\mathbf{x}_n)$ metodou nejmenších čtverců [5] je nutné minimalizovat součet čtverců reziduí $\mathbf{r}^{(s)} = \boldsymbol{\beta}^{(s)} - \boldsymbol{\Psi}^{(s)}\boldsymbol{\alpha}^{(s)}$ (které jsou tedy vypočtené na částicích $\{\mathbf{x}_n^{(s,j)}\}_{j=1}^J$):

$$\min \sum_{j=1}^J (r_j^{(s)})^2 = \min \sum_{j=1}^J \left(\Lambda^{(s)}(\mathbf{x}_n^{(s,j)}) - \sum_{b=1}^{n_\psi} \alpha_b^{(s)} \psi_b(\mathbf{x}_n^{(s,j)}) \right)^2. \quad (4.39)$$

Předpokládáme, že $J > n_\psi$, je zde alespoň tolik částic kolik je expanzních koeficientů. Řešení metodou nejmenších čtverců je pak dáno vztahem

$$\hat{\boldsymbol{\alpha}}^{(s)} = (\boldsymbol{\Psi}^{(s)T} \boldsymbol{\Psi}^{(s)})^{-1} \boldsymbol{\Psi}^{(s)T} \boldsymbol{\beta}^{(s)}, \quad (4.40)$$

za předpokladu, že matice $\boldsymbol{\Psi}^{(s)}$ je plně sloupcové hodnosti. Není-li matice $\boldsymbol{\Psi}^{(s)}$ plně sloupcové hodnosti, použijeme vztah

$$\hat{\boldsymbol{\alpha}}^{(s)} = (\boldsymbol{\Psi}^{(s)T} \boldsymbol{\Psi}^{(s)})^+ \boldsymbol{\Psi}^{(s)T} \boldsymbol{\beta}^{(s)}. \quad (4.41)$$

Jak již bylo řečeno, aproximovaná funkce $\Lambda^{(s)}(\mathbf{x}_n)$ je řídká. Řešení, které je k -řídké obdržíme vybráním k nejvýznamnějších expanzních koeficientů. V případě aproximace kladné funkce, kterou $\Lambda^{(s)}(\mathbf{x}_n)$ je, je možné vybrat k největších kladných koeficientů.

Pro výpočet vektoru expanzních koeficientů jsou využity i modifikace metody nejmenších čtverců. První modifikací je použití normalizovaného slovníku jako vstupu do algoritmu. Tato metoda je nazývána normalizovanou metodou nejmenších čtverců. Opět lze vybrat k největších koeficientů pro získání k -řídkého řešení. Druhou modifikací je využití tzv. přefitování. K této metodě je využito k -řídké řešení normalizované metody nejmenších čtverců. Z atomů odpovídajících k vybraným expanzním koeficientům je sestaven slovník, který je vstupem do metody nejmenších čtverců. Získaný vektor expanzních koeficientů je již k -řídký.

Ortogonální sdružovací metoda

V tomto případě je vstupem algoritmu ortogonální sdružovací metody [14] normalizovaný slovník $\boldsymbol{\Psi}^{(s)}$, vektor $\boldsymbol{\beta}^{(s)}$ a maximální počet iterací i_{\max} nebo maximální velikost chyby ϵ_{\max} . Výstup algoritmu tvoří množina indexů aktivních atomů \mathcal{S} a vektor $\{\hat{\alpha}_b^{(s)}\}_{b \in \mathcal{S}}$ obsahující koeficienty řídké aproximace (4.28).

Při inicializaci je nastaven vektor počáteční aproximace řešení $\hat{\mathbf{c}} = \mathbf{0}$, počáteční množina indexů $\mathcal{S} = \emptyset$, prázdná matice $\mathbf{S}^{J \times 0}$, počáteční reziduuum $\mathbf{r} = \boldsymbol{\beta}^{(s)}$. Dále je nastaven počet iterací $i = 1$.

Prvním krokem algoritmu je nalezení atomu, který maximalizuje absolutní hodnotu součinu sloupců $[\psi_b(\mathbf{x}_n^{(s,1)}), \dots, \psi_b(\mathbf{x}_n^{(s,J)})]^T$ matice $\boldsymbol{\Psi}^{(s)}$ s reziduem \mathbf{r}

$$b_0 = \arg \max_b \left| [\psi_b(\mathbf{x}_n^{(s,1)}), \dots, \psi_b(\mathbf{x}_n^{(s,J)})]^T \mathbf{r} \right|. \quad (4.42)$$

Indexová množina \mathcal{S} je aktualizována přidáním indexu b_0 do této množiny. Atom matice $\boldsymbol{\Psi}^{(s)}$ odpovídající b_0 -tému sloupci je přiřazen do matice \mathbf{S} jako její poslední sloupec. Pomocí metody nejmenších čtverců je vypočítán nový vektor řešení $\hat{\mathbf{c}}$

$$\hat{\mathbf{c}} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \boldsymbol{\beta}^{(s)}. \quad (4.43)$$

S vektorem řešení $\hat{\mathbf{c}}$ je spočítáno reziduum a následně je spočítána i jeho ℓ_2 norma

$$\mathbf{r} = \boldsymbol{\beta}^{(s)} - \mathbf{S}\hat{\mathbf{c}}, \quad (4.44)$$

$$\epsilon = \|\mathbf{r}\|_2. \quad (4.45)$$

Počet iterací i je navýšen o jedna a jsou zkontrolována ukončovací kritéria. Pokud $\epsilon > \epsilon_{\max}$ a zároveň $i \leq i_{\max}$, pak proběhne další iterace. V opačném případě jsou do vektoru $\{\hat{\alpha}_b^{(s)}\}_{b \in \mathcal{S}}$ na indexy \mathcal{S} přiřazeny popořadě čísla z vektoru $\hat{\mathbf{c}}$ a na indexy vektoru $\{\hat{\alpha}_b^{(s)}\}_{b \notin \mathcal{S}}$ neodpovídající indexům z vektoru $\hat{\mathbf{c}}$ jsou přiřazeny nuly. Algoritmus končí.

Řešení získané ortogonální sdružovací metodou je nejméně k -řádké, kde k je počet provedených iterací i .

4.3.3. Volba slovníku

Nyní bude uvedena modifikace LC, o kterém bylo pojednáno v kapitole 4.3.1. Tato modifikace je motivována faktem, že zbylé části lokální log-LF $\Lambda^{(s)}(\mathbf{x}_n)$ jsou řídké. Obsahují velmi mnoho malých nebo dokonce nulových hodnot. Zbylá část globální log-LF v (4.21), $\bar{\Lambda}(\mathbf{x}_n) = \frac{1}{S} \sum_{s=1}^S \Lambda^{(s)}(\mathbf{x}_n)$, je dokonce ještě řidší, jelikož je mnoho hodnot „odprůměrováno“ pryč. Průměrování má největší efekt na mody šumu, které jsou obsažené v lokálních funkcích $\Lambda^{(s)}(\mathbf{x}_n)$. Mody šumu se stávají nepatrnými na rozdíl od modů měření, což je vhodné pro získání dobrých výsledků pomocí distribuovaného PF.

Cílem této kapitoly je využití řídkosti zbylých částí lokálních log-LF $\Lambda^{(s)}(\mathbf{x}_n)$ k vytvoření ještě řidší reprezentace zbylé části globální log-LF $\bar{\Lambda}(\mathbf{x}_n)$. Výhodou řídké reprezentace jsou nižší nároky na komunikaci, a tedy i nižší nároky na cenu komunikace v LC. Řídkosti lze dosáhnout nejen volbou slovníku, ale i volbou nejmenší možné ℓ_2 normy atomů slovníku.

Jak už bylo řečeno dříve, každé jednotlivé měření $\mathbf{z}_{n,m}^{(s)}$ vede k odpovídajícímu modu v $\ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n)$ a tedy i k odpovídajícímu modu v $\Lambda^{(s)}(\mathbf{x}_n) = \ln f(\mathbf{z}_n^{(s)} | \mathbf{x}_n) - \lambda^{(s)}$. Tyto mody jsou ve stavovém prostoru \mathbb{R}^d rozloženy řídkce, což je důležitý fakt pro vhodnou volbu slovníku $\{\psi_b(\cdot)\}_{b=1}^{n_\psi}$ ze vztahu (4.28). Slovník by měl (i) dobře reprezentovat jednotlivé mody funkcí $\Lambda^{(s)}(\mathbf{x}_n)$, a (ii) by měl využívat řídkosti těchto funkcí pro řídkou reprezentaci globální funkce $\bar{\Lambda}(\mathbf{x}_n)$. První vlastnost je vyžadována pro přesnou aproximaci globální LF v (4.34) a z toho vyplývající dobrý odhad v distribuovaných PF. Druhá vlastnost vede ke snížení výpočetních nároků a nároků na komunikaci.

Představeny byly např. Fourierovy či polynomiální slovníky [20], [17], ale bohužel nesplňovaly obě výše uvedené vlastnosti. Také je potřeba, aby atomy $\psi_b(\cdot)$ byly dostatečně lokalizovány a dostatečně řídkce rozmístěny ve stavovém prostoru \mathbb{R}^d . V článku [32] je navrženo použití rovnoměrně rozmístěných Gaussovských atomů [31]. Já v této práci navrhuji využití splajnů, kdy je slovník sestaven posouváním centrálního d -dimenzionálního B-splajnu $\psi(\mathbf{x})$ podél příslušné d -dimenzionální mřížky

$$\psi_b(\mathbf{x}) = \psi(\mathbf{x} - \mathbf{x}_b), \quad \mathbf{x} \in \mathbb{R}^d, b = 1, \dots, n_\psi, \quad (4.46)$$

kde \mathbf{x}_b značí b -tý bod mřížky. Následně je slovník vyhodnocen na pozicích částic \mathbf{x}_n . Důležité jsou následující dvě vlastnosti: každý B-splajn $\psi_b(\mathbf{x})$ odpovídá jednomu bodu \mathbf{x}_b mřížky, a všechny atomy $\psi_b(\mathbf{x})$ jsou rovny základnímu centrálnímu B-splajnu $\psi(\mathbf{x})$.

Pro dobrou aproximaci je vhodné, aby d -dimenzionální mřížka měla dostatečnou hustotu, což vyžaduje i dostatečné množství n_ψ atomů slovníku. Navíc musí být každý atom

4.4. DISTRIBUOVANÝ ČÁSTICOVÝ FILTR

koncentrován okolo daného bodu mřížky \mathbf{x}_b a současně se musí překrývat s ostatními atomy. Vhodná volba koncentrace atomů je nezbytná pro využití řídkosti funkcí $\Lambda^{(s)}(\mathbf{x}_n)$.

Po aproximaci jsou mezi senzory vyměňovány jen krátké vektory expanzních koeficientů (4.28), což vede na velké snížení požadavků na komunikační síť.

4.4. Distribuovaný částicový filtr

V této kapitole je propojen výše uvedený model sledování objektu s DPF³.

Každý senzor s spouští lokálně PF⁴ nezávisle na lokálních částicových filtrech ostatních senzorů. Nicméně lokální PF využívá lokální aproximaci globální LF $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$, nikoli pouze svou lokální LF. Z důvodu využití globální LF každým lokálním PF závisí lokální odhad stavu $\hat{\mathbf{x}}_n^{(s)}$ na měření \mathbf{z}_n všech senzorů a ne pouze na lokálním měření $\mathbf{z}_n^{(s)}$. Lokální aproximace globální LF $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ jsou počítány distribuovaně, jak již bylo zmíněno dříve v textu.

Hlavním cílem lokálního PF prováděným senzorem s je vypočtení lokální aproximace globální aposteriorní PDF $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ v každém čase n a s měřeními $\mathbf{z}_{1:n} = [\mathbf{z}_1^T, \dots, \mathbf{z}_n^T]$. Toho je docíleno rekurzivně: V čase n je předchozí globální aposteriorní PDF $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ převedena na aktuální globální aposteriorní PDF $f(\mathbf{x}_n|\mathbf{z}_{1:n})$. Převedení je provedeno přes globální LF a tedy přes LC. Také zahrnuje aktuální měření \mathbf{z}_n od všech senzorů. Z aktuální globální aposteriorní PDF $f(\mathbf{x}_n|\mathbf{z}_{1:n})$, přesněji řečeno z její aproximace, je spočítán lokální odhad $\hat{\mathbf{x}}_n^{(s)}$ stavu sledovaného cíle

$$\hat{\mathbf{x}}_n^{\text{MMSE}} = E(\mathbf{x}_n|\mathbf{z}_{1:n}) = \int \mathbf{x}_n f(\mathbf{x}_n|\mathbf{z}_{1:n}) d\mathbf{x}_n. \quad (4.47)$$

Odhad je tedy spočítán jako vhodná aproximace minima odhadu střední kvadratické chyby.

V lokálním PF, který je prováděný senzorem s , je globální aposteriorní PDF $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ reprezentována množinou lokálních částic a s nimi souvisejících vah $\{(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)})\}_{j=1}^J$, kde $\sum_{j=1}^J w_n^{(s,j)} = 1$. Rekurzivní aktualizace globální aposteriorní PDF $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ na $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ tudíž odpovídá rekurzivní aktualizaci množiny částic a vah $\{(\mathbf{x}_{n-1}^{(s,j)}, w_{n-1}^{(s,j)})\}_{j=1}^J$ na $\{(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)})\}_{j=1}^J$.

V čase $n = 0$ je rekurze inicializována náhodným rozmístěním (navzorkováním) J částic $\mathbf{x}_0^{(s,j)}$, kde rozmístění částic odpovídá apriorní PDF $f(\mathbf{x}_0)$, a váhy jsou nastaveny na $w_0^{(s,j)} = 1/J$. Pro časy $n \geq 0$ jsou provedeny následující kroky:

Převzorkování Náhodným rozmístěním a přeskládáním prvků z množiny předchozích částic $\{\mathbf{x}_{n-1}^{(s,j')}\}_{j'=1}^J$ je získáno J převzorkovaných částic $\bar{\mathbf{x}}_{n-1}^{(s,j)}$. Pravděpodobnost umístění částice $\mathbf{x}_{n-1}^{(s,j')}$ určuje váha $w_{n-1}^{(s,j')}$.

Vzorkování Pro každou převzorkovanou částici $\bar{\mathbf{x}}_{n-1}^{(s,j)}$ je predikována nová částice $\mathbf{x}_n^{(s,j)}$ a to vzorkováním z $f(\mathbf{x}_n|\bar{\mathbf{x}}_{n-1}^{(s,j)})$. V tomto modelu se jedná o stavově-přechodovou PDF $f(\mathbf{x}_n|\mathbf{x}_{n-1})$ vypočtenou na pozicích $\mathbf{x}_{n-1} = \bar{\mathbf{x}}_{n-1}^{(s,j)}$.

³Distribuované částicové filtry (ang. distributed particle filters).

⁴Částicový filtr (ang. particle filter).

Následuje distribuovaný výpočet lokální aproximace globální LF $\hat{f}_s(\mathbf{z}_n | \mathbf{x}_n)$. Tento krok tedy vyžaduje komunikaci se sousedními senzory. Lokální aproximace v senzoru s může být vypočtena např. metodou nejmenších čtverců, nebo ortogonální sdružovací metodou.

Aktualizace Vypočteným částicím $\mathbf{x}_n^{(s,j)}$ z předchozího kroku jsou přiřazeny váhy, které lze spočítat jako

$$w_n^{(s,j)} = \frac{\hat{f}_s(\mathbf{z}_n | \mathbf{x}_n^{(s,j)})}{\sum_{j=1}^J \hat{f}_s(\mathbf{z}_n | \mathbf{x}_n^{(s,j)})}. \quad (4.48)$$

Ze získané množiny $\{(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)})\}_{j=1}^J$ je možno vypočíst aproximaci minima střední kvadratické chyby odhadu globálního stavu

$$\hat{\mathbf{x}}_n^{(s)} = \sum_{j=1}^J w_n^{(s,j)} \mathbf{x}_n^{(s,j)}. \quad (4.49)$$

5. Program

5.1. Popis simulace

Program simulace je psaný v programovacím prostředí MATLAB ve verzích 2015a a 2015b. Nachází se v příloze této práce a je spustitelný ze souboru `testbed`. V tomto souboru je možné nastavit všechny parametry simulace.

V simulaci uvažujeme pohybující se objekt. Stav \mathbf{x}_n objektu je popsán jeho pozicí a rychlostí. Lze tedy psát $\mathbf{x}_n = [x_{1,n}, x_{2,n}, \dot{x}_{1,n}, \dot{x}_{2,n}]^T$. Pohyb objektu je realizován v rovině xy na tzv. oblasti zájmu (ang. region of interest, zkr. ROI), jejíž velikost je $[-3000, 3000] \times [-3000, 3000]$, v simulaci jí odpovídá parametr `surveillanceRegion`. Hodnoty vzdáleností v simulaci mohou mít libovolné jednotky, zde jsou použity metry. Pro \mathbf{x}_n je použit model s téměř konstantní rychlostí pohybu (ang. near constant-velocity motion model)

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{W}\mathbf{u}_n, \quad (5.1)$$

kde dle [26] matice $\mathbf{A} \in \mathbb{R}^{4 \times 4}$ a $\mathbf{W} \in \mathbb{R}^{4 \times 2}$ jsou tvaru

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & st & 0 \\ 0 & 1 & 0 & st \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} vt & 0 \\ 0 & vt \\ st & 0 \\ 0 & st \end{bmatrix}. \quad (5.2)$$

Zde st je doba trvání jednoho časového kroku `scanTime`, v simulaci $st = 1$, pro vt je použit vztah $vt = 0,5st^2$. V modelu (5.1) je $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \sigma_u^2 \mathbf{I}_2)$ Gaussovský šum pohybu (ang. Gaussian driving noise), pro který byl vybrán rozptyl $\sigma_u^2 = 0,05$ (v simulaci označován jako `drivingNoiseVariance`), a v němž je nezávislá a identicky rozdělená posloupnost 2D Gaussovských náhodných vektorů. Apriorní PDF pro stav sledovaného cíle $f(\mathbf{x}_0) = \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{C}_0)$, kde byla vybrána kovariance $\mathbf{C}_0 = \text{diag}([50^2, 50^2, 5^2, 5^2])$ (v simulaci `priorCovariance`) a průměr `priorMean` odpovídající příkazu `mvnrnd`, tedy více-rozměrná náhodná čísla s parametry `targetStartState` a `priorCovariance`. Parametr `targetStartState` označuje počáteční pozici sledovaného objektu a má hodnotu $[0; 0; 5; 5]$.

V simulaci je uvažováno $S = 5$ senzorů (`numSensors`), které jsou rozloženy rovnoměrně na ROI, na kružnici o poloměru 3000 m , což je číslo ve druhém sloupci druhého řádku parametru `surveillanceRegion`. Pozice senzorů jsou vypočteny funkcí `getSensorPosition` a jsou uchovány v parametru `sensorPositions`. Každý senzor měří tzv. range (vzdálenost) a tzv. bearing (úhel) až do vzdálenosti 6000 od pozice senzoru. Je-li vzdálenost menší než 6000 m , je měření, které je generováno objektem, dáno jako

$$\mathbf{z}_{n,m}^{(s)} = \left[\|\tilde{\mathbf{x}}_n - \mathbf{p}^{(s)}\| \varphi(\tilde{\mathbf{x}}_n, \mathbf{p}^{(s)}) \right]^T + \mathbf{v}_{n,m}^{(s)} = \mathbf{h}_{n,m}^{(s)}(\tilde{\mathbf{x}}_n) + \mathbf{v}_{n,m}^{(s)}, \quad (5.3)$$

kde $\tilde{\mathbf{x}}_n = [x_{1,n}, x_{2,n}]^T$, $\mathbf{p}^{(s)}$ je pozice senzoru s , $\varphi(\tilde{\mathbf{x}}_n, \mathbf{p}^{(s)})$ je úhel (ve stupních) vektoru $\tilde{\mathbf{x}}_n$ vzhledem k vektoru $\mathbf{p}^{(s)}$ a $\mathbf{v}_{n,m}^{(s)} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_v)$ označuje šum měření, pro nějž byla vybrána kovariance $\mathbf{C}_v = \text{diag}([5^2, 0,75])$, kde hodnoty kovariance pochází z parametrů `measurementVarianceRange` a `measurementVarianceBearing`. Analogicky

jako \mathbf{u}_n je $\mathbf{v}_{n,m}^{(s)}$ nezávislá a identicky rozdělená posloupnost 2D Gaussovských náhodných vektorů.

Dalšími hodnotami pro nastavení simulace je pravděpodobnost detekce $P_d^{(s)} = 0,9$ objektu senzorem s a střední hodnota $\mu^{(s)} = 7$ určující počet falešných měření senzoru s . Dále 200 provedených kroků `timeSteps`, 2000 částic `numPredParticles`, použití nejvýše 40 `numberOfAtoms` atomů k aproximaci, nastavení parametru šířky splajnů `dilation` na hodnotu 0,08 a nastavení tolerance na 0,4 pro „vyhození“ atomů slovníku s normou menší než je tato tolerance. Pro metody konsenzu je nutné nastavit i parametr `spline.consensusWeightModel`, kde byla použita možnost 4, tedy váhy Metropolis. V simulaci je nastaveno, že každý z pěti sensorů může komunikovat jen se dvěma svými bližšími sousedy. Zbývá nastavení výběru metod pro aproximaci, kde pro potřeby této práce byly použity všechny metody a to s volbou částic 1.

Lze si všimnout, že v souladu s modelem (5.3) je věrohodnostní funkce $f(\mathbf{z}_{n,m}^{(s)} | \mathbf{x}_n, a_n^{(s)} = m)$ ve (4.3) pouze funkcí $\tilde{\mathbf{x}}_n = [x_{1,n}, x_{2,n}]^T$ a to pro pozorované a pevně dané $\mathbf{z}_{n,m}^{(s)}$. Po vykreslení logaritmu této funkce v rovině xy , tedy v $[x_{1,n}, x_{2,n}]$, lze vidět, že se skládá z tzv. banánových funkcí, obrázky 5.8, 5.10, 5.11 v kapitole 5.4.

5.2. Funkce testbed

Kromě základních parametrů simulace, zmíněných v předchozí kapitole, se ve funkci `testbed` nachází i parametry vizualizace, aproximace a splajnového systému. Tato kapitola obsahuje bližší popis parametrů obsažených v této funkci. Parametry jsou řazeny dle jejich pořadí ve funkci `testbed`. Obrázky obsažené v popisu lze nalézt v kapitole 5.4.

visualizationMode Nastavení metody vizualizace simulace. V úvahu připadají tři možnosti: 0 pro vykreslení jednoho grafu obsahujícího celou trasu sledovaného objektu i odhad této trasy různými metodami; 1 pro vykreslení „videa“ simulace, které je spustitelné mezerníkem; 2 pro vykreslení krok po kroku, kde na další krok simulace lze opět přejít stisknutím mezerníku; 3 nechceme-li vykreslovat žádný graf. Ilustrativní příklad vizualizace je na obrázku 5.1. Vykreslení této vizualizace zajišťuje funkce `plotAll`.

timeSteps Počet kroků simulace. Pro výše uvedený ROI a $st = 1$ doporučuji maximálně 400 kroků, protože pak se již zvyšuje možnost, že konec trasy sledovaného objektu nebude vidět na vizualizaci (objekt „odjede“ ze záběru).

approximationMethod Dvouřádková matice, kde první řádek obsahuje možnosti aproximace lokální LF a druhý řádek obsahuje možnosti rozložení částic pro vybranou metodu aproximace. Metod aproximace je hned devět: 1 tzv. originální metoda, kde probíhá jen součin vah (součin lokálních LF), tím je získána globální LF [15]; 2–9 jsou metody popsané v kapitole 4.3.2, kde pro 2–5 se jedná o decentralizovaný PF s fúzním centrem 3.5.1 a pro 6–9 se jedná o decentralizovaný PF s konsenzem 3.5.2. Metody volby částic jsou čtyři: 1 obvyklé částice ve smyslu DPF; 2 rovnoměrně rozložené částice okolo posledních `numLast` odhadnutých pozic objektu; 3 rovnoměrně rozložené částice na celém ROI; 4 částice tvoří pravidelnou mřížku přes celou plochu ROI.

5.2. FUNKCE TESTBED

numberOfAtoms Parametr řídkosti slovníku a vektoru expanzních koeficientů. Tedy maximální počet atomů matice $\Psi^{(s)}$ při aproximaci $\Lambda^{(s)}(\mathbf{x}_n^{(s,j)})$. V případě OMP se jedná o maximální počet iterací i_{\max} této metody.

dilation Parametr šířky (dilatace) splajně. Pro nižší hodnoty jsou splajny širší, pro vyšší hodnoty zase užší. Aby byly pokryty požadavky z kapitoly 4.3.3, je pro jakoukoli volbu dilatace vypočten vhodný posuv atomu.

tolerance Ve slovníku splajně $\Psi^{(s)}$ jsou ponechány pouze atomy, jejichž ℓ_2 norma je vyšší než tato tolerance. Tím jsou vyřazeny atomy slovníku obsahující čísla nulová nebo blízka nule.

regularMesh Jedná se o parametr „hustoty“ bodů na mřížce. Pouze pro některé druhy vizualizace. Tento parametr nemá žádný vliv na průběh simulace.

splineSys Zde jsou obsaženy další možnosti vizualizace: 1 pro vykreslení slajnového systému na regulární mřížce a jeho vyčíslení na pozicích částic, obrázky 5.2 a 5.3; 2 pro funkci $\Lambda^{(s)}(\mathbf{x}_n^{(s,j)})$ vypočtenou na regulární mřížce, vykreslení vah a jejich aproximace, obrázky 5.4 a 5.5; 3 pro vykreslení expanzních koeficientů (všech senzorů) seřazených dle velikosti 5.6; 4 zobrazení funkce $\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ na pozicích částic, obrázky 5.7 a 5.8.

consensusWeightModel Program simulace obsahuje několik různých typů konsenzů. Použitý a zmíněný v textu je pouze ten s číslem 4, který využívá váhy Metropolis. Ostatní metody jsou zde ponechány pro možné další simulace.

numPredParticles Počet částic použitých pro distribuovaný PF.

numUniform Počet rovnoměrně rozložených částic.

numRegular Počet částic v rovnoměrné mřížce.

surveillanceRegion Velikost ROI, jedná se o plochu zobrazenou ve vizualizacích funkcí `plotAll` a `approxPlot`.

minmax Velikost oblasti pro parametr `approximationMethod`.

numLast Pro kolik posledních měření objektu se mají přidávat rovnoměrné částice.

drivingNoiseVariance Parametr toho, jak se mění pohyb objektu. Jedná se o rozptyl σ_u^2 v Gaussovském šumu pohybu $\mathbf{u}_n \sim N(\mathbf{0}, \sigma_u^2 \mathbf{I}_2)$.

measurementVarianceRange, measurementVarianceBearing Zvýšením těchto hodnot se zvýší nepřesnost měření. Zvětšíme-li první parametr, budou banánové funkce širší. Zvětšíme-li druhý parametr, banánové funkce budou delší. Parametry pochází z kovariance \mathbf{C}_v Gaussovského šumu měření $\mathbf{v}_{n,m}^{(s)} \sim N(\mathbf{0}, \mathbf{C}_v)$. Zvýšení hodnot těchto parametrů je patrné na obrázcích 5.10, 5.11.

meanClutterParameter Střední hodnota $\mu^{(s)}$ Poissonova rozdělení, které upravuje počet falešných měření senzoru s během časového kroku sT . Zvýšením této hodnoty bude více banánových funkcí v $\ln f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$. Zvýšení toho parametru je vidět na obrázcích 5.7-5.10

detectionProbability Pravděpodobnost detekce $P_d^{(s)}$ objektu senzorem s .

numSensors Počet senzorů S . Doporučuji nejvýše šest.

sensorPositions Pozice senzorů, které jsou vypočítány funkcí `getSensorPosition`. Sensory jsou rovnoměrně rozloženy na kružnici o poloměru pocházejícím z parametru `surveillanceRegion`.

measurementRange Vzdálenost, do které může senzor měřit parametry sledovaného objektu. V nastavení simulace je to průměr kružnice, na které senzory leží.

scanTime Délka časového kroku.

targetStartState Počáteční bod pohybu objektu.

trajectory Trajektorie sledovaného objektu generována modelem s téměř konstantní rychlostí pohybu (5.1). V programu simulace je generování této trajektorie zajištěno funkcí `generateTrajectory`.

priorCovariance, priorMean Hodnoty \mathbf{C}_0 a $\boldsymbol{\mu}_0$ apriorní PDF pro stav sledovaného cíle $f(\mathbf{x}_0) = \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{C}_0)$.

measurements Hodnoty vzdálenosti a úhlu, které naměří senzor s . Tyto hodnoty jsou generovány pomocí funkce `generateMeasurements`.

clutteredMeasurements Výše uvedené měření `measurements` je rozšířeno o falešná měření pomocí funkce `generateClutteredMeasurements`.

approxMenu Funkce, která má na starosti zpracování hodnot z výše uvedeného parametru `approximationMethod`. Z této funkce je možný přístup k funkcím `PDAF2`, `PDAFfc`, `PDAFcb` a tedy i k funkcím zajišťujícím metody aproximace.

plotAll Jedna z množností vizualizace dat. Příkladem je obrázek 5.1.

5.3. Funkce `PDAF2`, `PDAFfc`, `PDAFcb`

Tyto funkce představují jádro celé simulace. Ačkoli jsou z velké části stejné, byly kvůli lepší orientaci rozděleny do tří funkcí. Souhrnně jsou nazývány funkcemi `PDAF`. V této kapitole budou popsány jejich důležité části.

predictedParticles Částice, které jsou generovány funkcí `particles`. Dle nastavení v parametru `approximationMethod` mohou být generovány čtyři typy částic, z nichž pouze první typ jsou částice ve smyslu DPF.

constantFactor Konstantní faktor cf ve tvaru

$$cf = \frac{P_d^{(s)}}{\mu^{(s)}} \frac{1}{\sqrt{2\pi\mathbf{C}_v}}, \quad (5.4)$$

kde \mathbf{C}_v představuje hodnotu kovariance Gaussovského šumu měření.

5.3. FUNKCE PDAF2, PDAFFC, PDAFCB

falseAlarmDistribution Rozložení f_{a_m} měření v prostoru dle trojúhelníkového modelu funkce `evaluateTriangle`.

predictedRange, predictedBearing Vypočtení hodnot $\|\tilde{\mathbf{x}}_n - \mathbf{p}^{(s)}\|$ a $\varphi(\tilde{\mathbf{x}}_n, \mathbf{p}^{(s)})$, tedy $\mathbf{h}_{n,m}^{(s)}(\tilde{\mathbf{x}}_n)$ ze vztahu (5.3).

weights Váhy distribuovaného částicového filtru, tedy lokální LF $f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ odvozená v kapitole 4.1. Zde je vypočítána cyklem přes všechna měření senzoru s jako

$$f(\mathbf{z}_n^{(s)}|\mathbf{x}_n) = \mathbf{1} + \sum_{m=1}^{M_n^{(s)}} \frac{cf}{fa_m} \exp\left(\frac{-1}{2\mathbf{C}_v} (\mathbf{z}_{n,m}^{(s)} - \mathbf{h}_{n,m}^{(s)}(\tilde{\mathbf{x}}_n))^2\right). \quad (5.5)$$

Po tomto kroku se začínají jednotlivé PDAF funkce lišit. Funkce PDAF2 obsahuje originální metodu, váhy `weights` vynásobí přes příslušné částice. Funkce PDAFfc a PDAFCb obsahují zlogaritmování vah `logweights`, nalezení minima `smallLambda` (4.17) a vypočtení `bigLambda`, která představuje funkc $\Lambda^{(s)}(\mathbf{x}_n)$ (4.23). Následuje aproximace funkce $\Lambda^{(s)}(\mathbf{x}_n)$ splajnovým systémem.

approxSplines Jedná se o skupinu funkcí rozlišených čísly. Ačkoli tyto funkce nejsou přímo obsaženy ve funkcích PDAF, je vhodné je sem zařadit. V těchto funkcích probíhá vzorkování splajnového systému (slovníku) a aproximace funkce $\Lambda^{(s)}(\mathbf{x}_n)$ tímto slovníkem dle navržených metod (ortogonální sdružovací metoda a metoda nejmenších čtverců s jejími modifikacemi). Také je zde vazba na vizualizační funkci `approxPlot`.

Dále funkce PDAFfc a PDAFCb pod proměnnou `barBigLambda` obsahují výpočet funkce $\bar{\Lambda}(\mathbf{x}_n)$ (4.21) vynásobením průměrů expanzních koeficientů (které byly spočteny funkcí `approxSplines`) se splajnovým slovníkem. Proměnná `barSmallLambda` obsahuje výpočet průměru $\bar{\lambda}$ (4.20). V případě PDAFfc jsou průměry expanzních koeficientů i průměry $\bar{\lambda}$ spočítány zprůměrováním, v případě PDAFCb je jejich spočtení přenecháno algoritmu konsenzu. Obě dvě funkce obsahují také výpočet globální LF $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ a to podle vztahu (4.34).

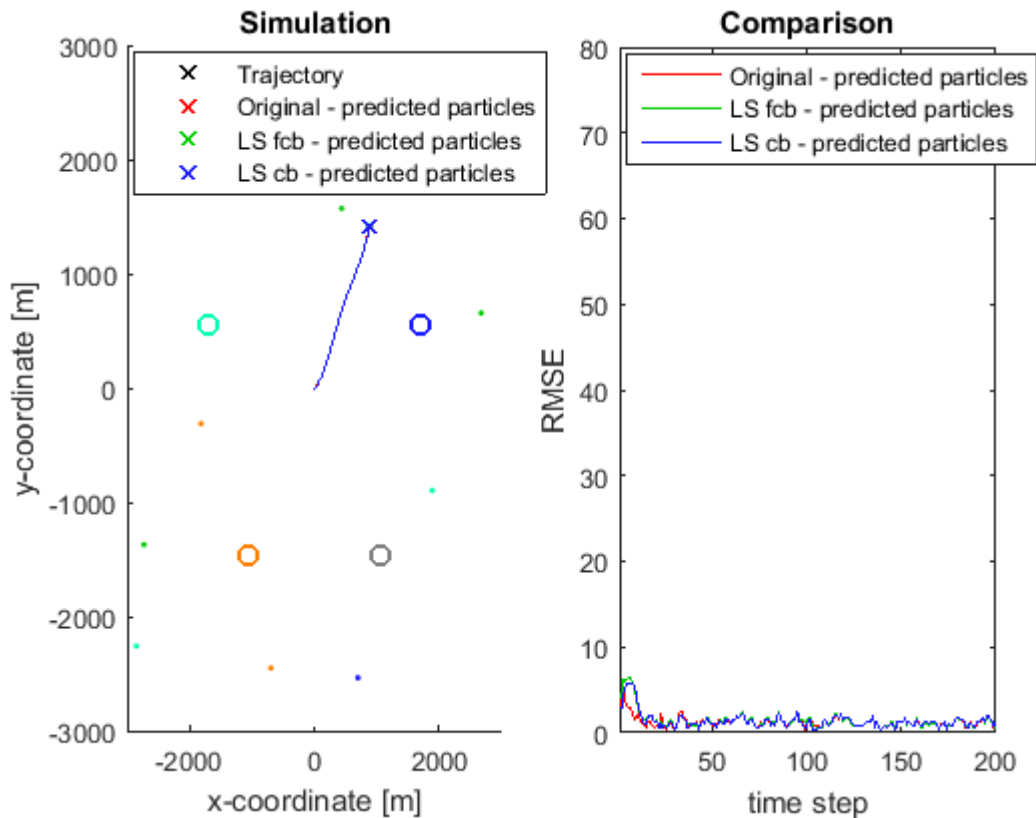
Následující část je opět pro všechny funkce PDAF stejná.

weightsNormalized Aktualizace vah a jejich normalizace (4.48).

estimatedTrajectory Odhad trajektorie sledovaného objektu (4.49).

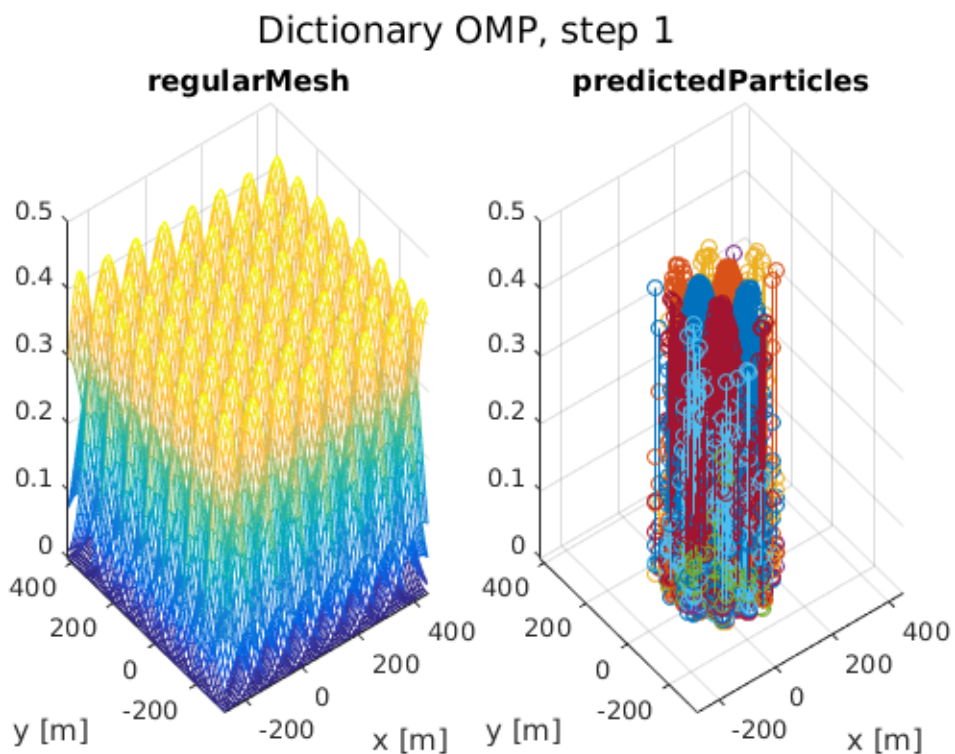
posteriorParticles Využití převzorkování k výpočtu nových částic.

5.4. Obrázky k popisu programu simulace

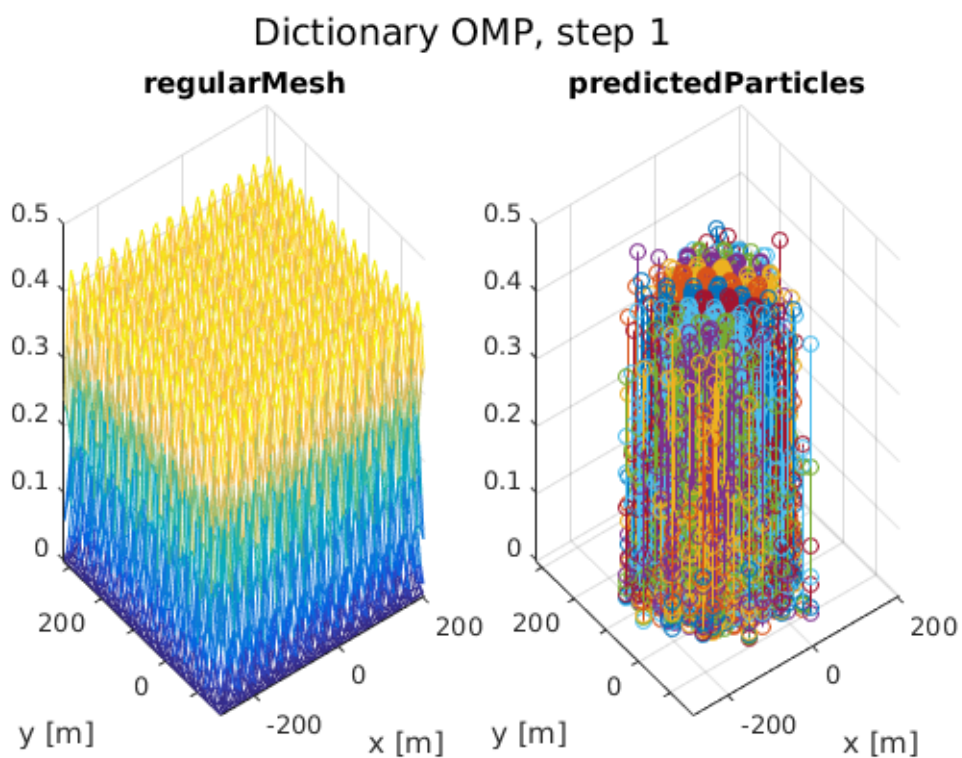


Obrázek 5.1: Vlevo se pod názvem „Trajectory“ nachází vizualizace simulace pohybu objektu v rovině xy a odhad pohybu objektu v této rovině pomocí originální metody a metody nejmenších čtverců (pro distribuovaný PF s fcb - fúzním centrem a s cb - konsenzem). Vidíme, že v tomto případě se původní trajektorie i její odhady překrývají. Kolečka označují polohu senzorů, tečky šum v aktuálním čase. Vpravo je srovnání těchto metod dle střední kvadratické chyby.

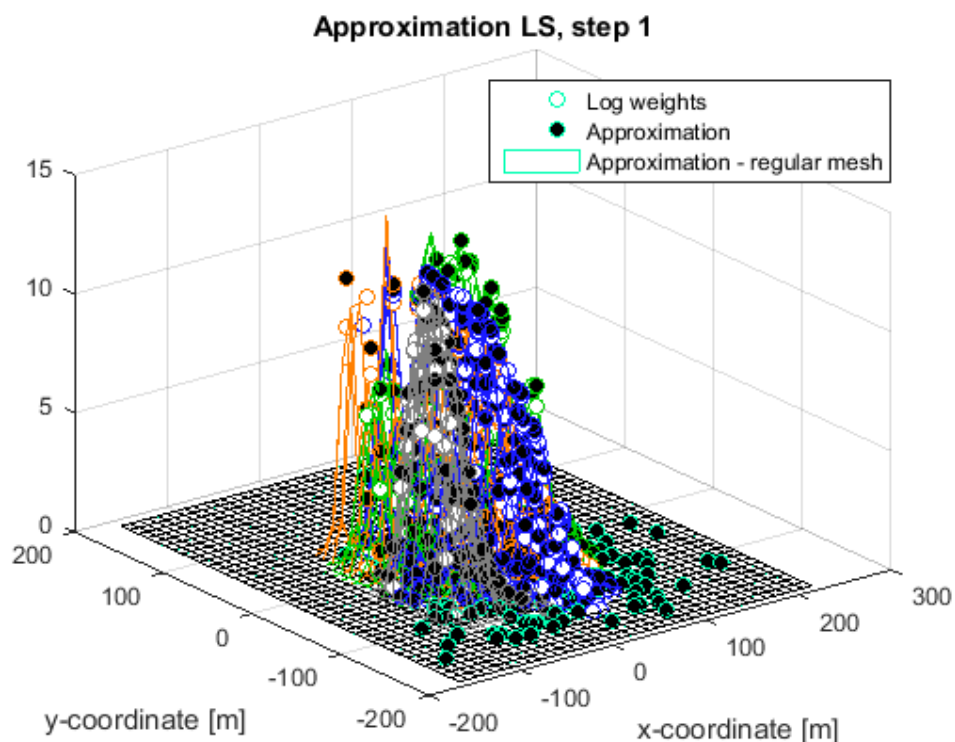
5.4. OBRÁZKY K POPISU PROGRAMU SIMULACE



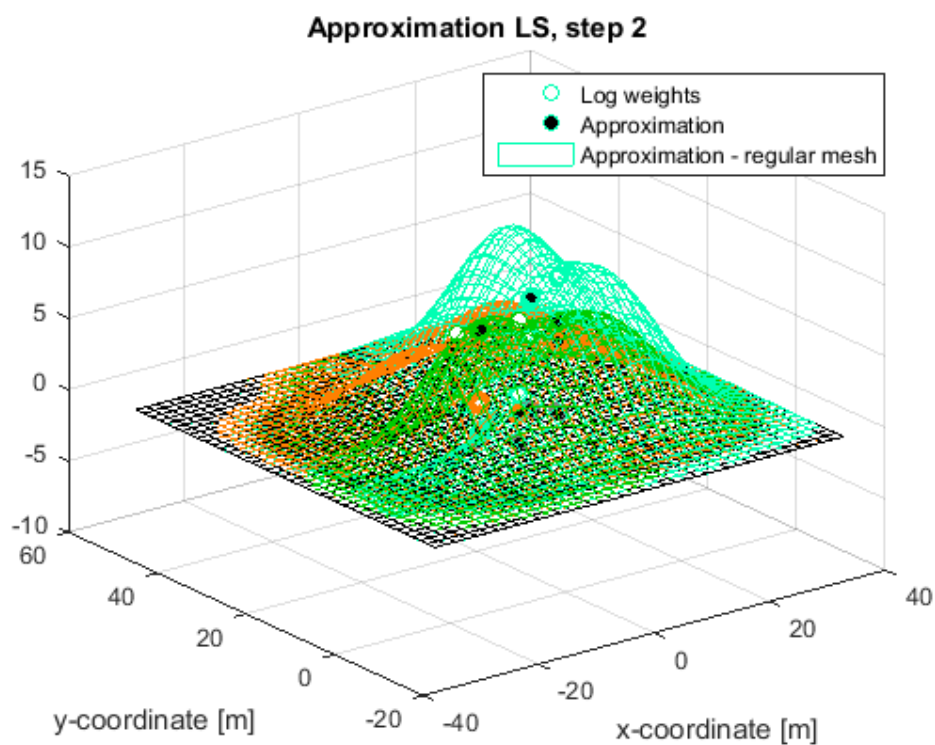
Obrázek 5.2: Příklad slovníku pro parametr dilatace 0,01 a jeho vyčíslení na rovnoměrné mřížce (vlevo) a na pozicích 2000 predikovaných částic (vpravo).



Obrázek 5.3: Příklad slovníku pro parametr dilatace 0,03 a jeho vyčíslení na rovnoměrné mřížce (vlevo) a na pozicích 2000 predikovaných částic (vpravo).

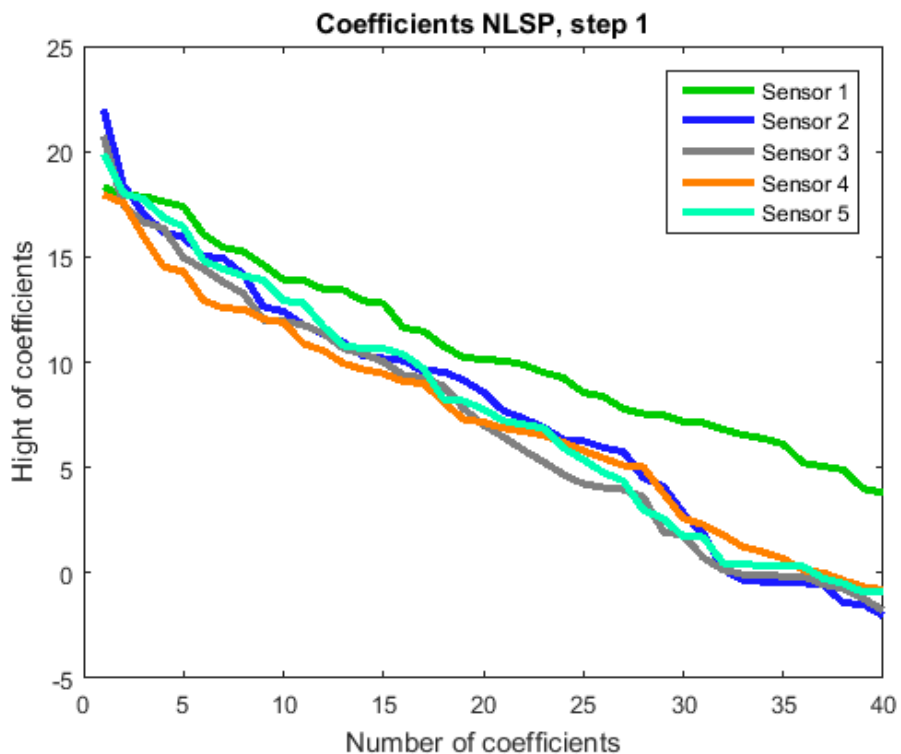


Obrázek 5.4: Vykreslení logaritmu vah na pozicích částic (kruh s bílým vnitřkem), aproximace těchto vah (kruh s černým vnitřkem) a vypočtení aproximace na rovnoměrné síti. Barvy kružnic odpovídají jednotlivým sensorům, černá rovnoměrná síť označuje nulovou hladinu.

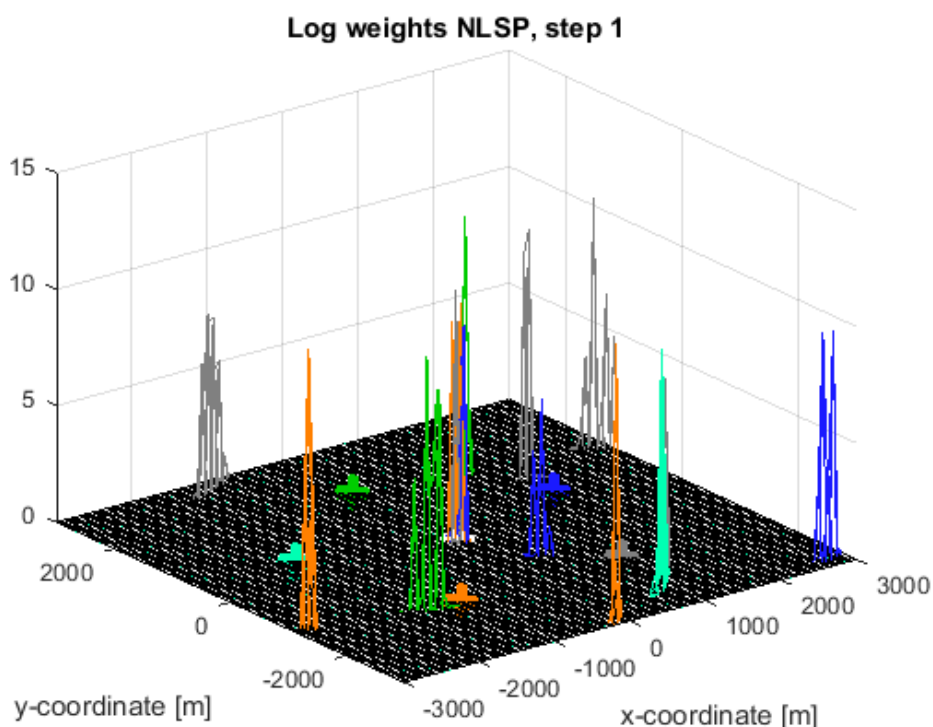


Obrázek 5.5: Vykreslení logaritmu vah na pozicích částic, viz popis obrázku výše.

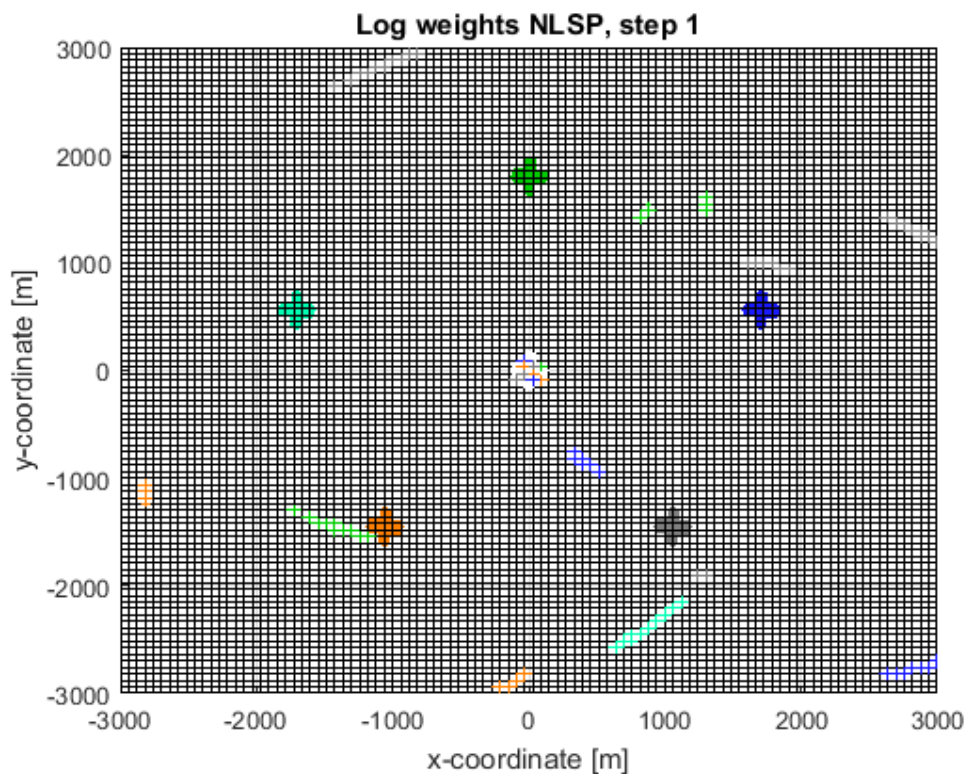
5.4. OBRÁZKY K POPISU PROGRAMU SIMULACE



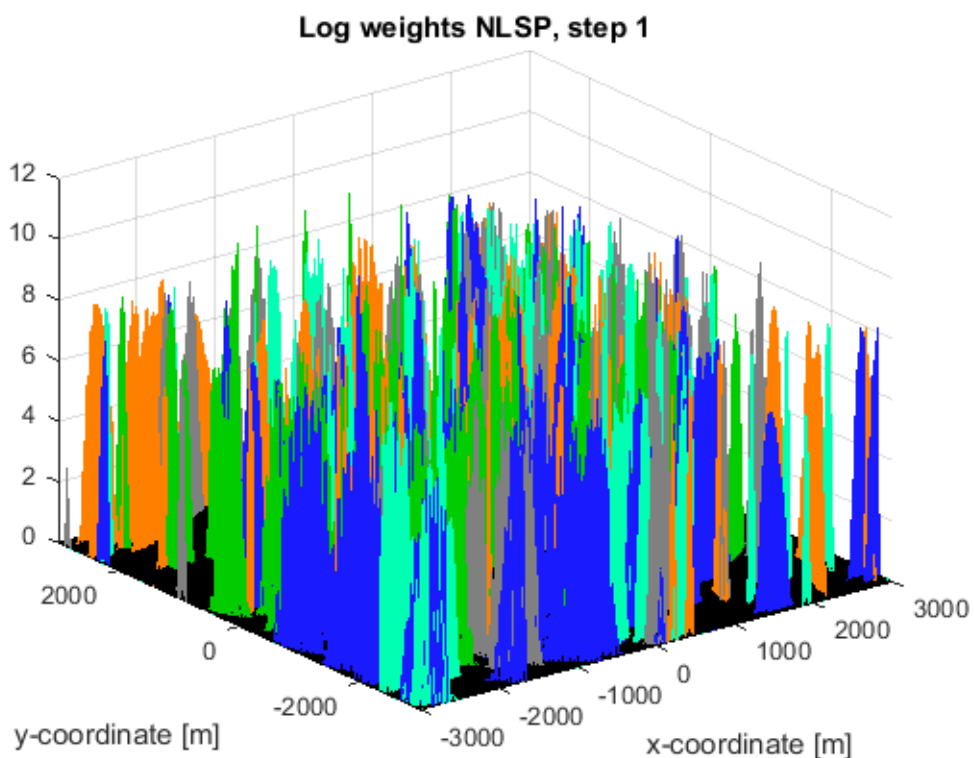
Obrázek 5.6: Zobrazení velikosti expanzních koeficientů a jejich počtu. Nejvyšší možný počet koeficientů určuje parametr numberOfAtoms.



Obrázek 5.7: Zde jsou vykresleny banánové funkce příslušné šumu ($\mu^{(s)} = 7$) i měření senzorů, křížky v hladině nula (černá rovnoměrná mřížka) jsou pozice senzorů. Uprostřed je bíle zobrazena pozice sledovaného cíle.

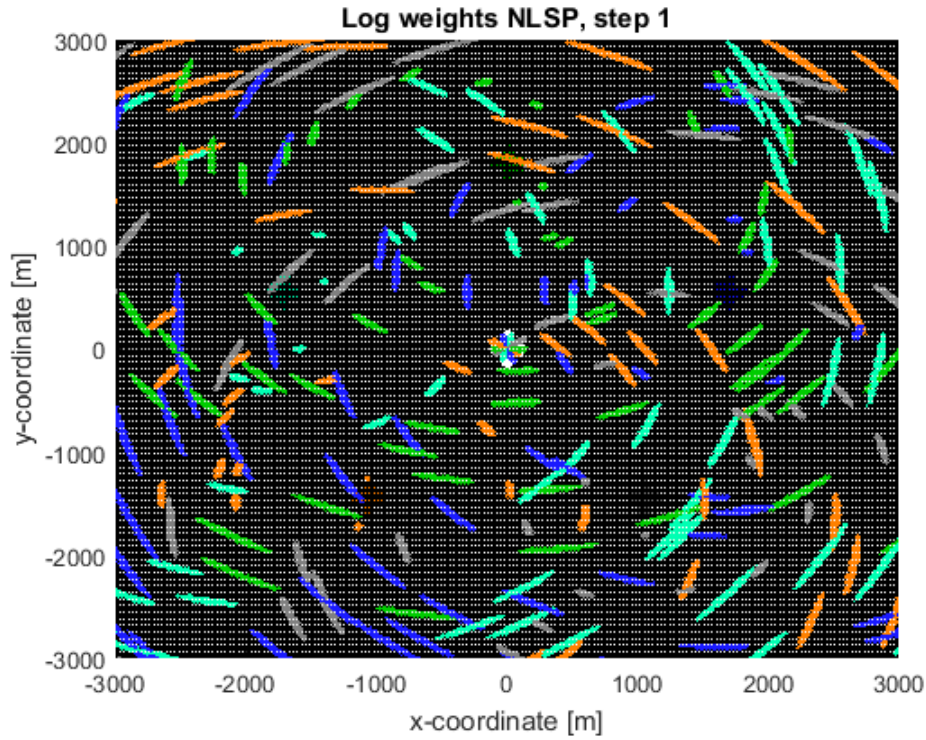


Obrázek 5.8: Pohled na obrázek 5.7 shora, viz popis k tomuto obrázku.

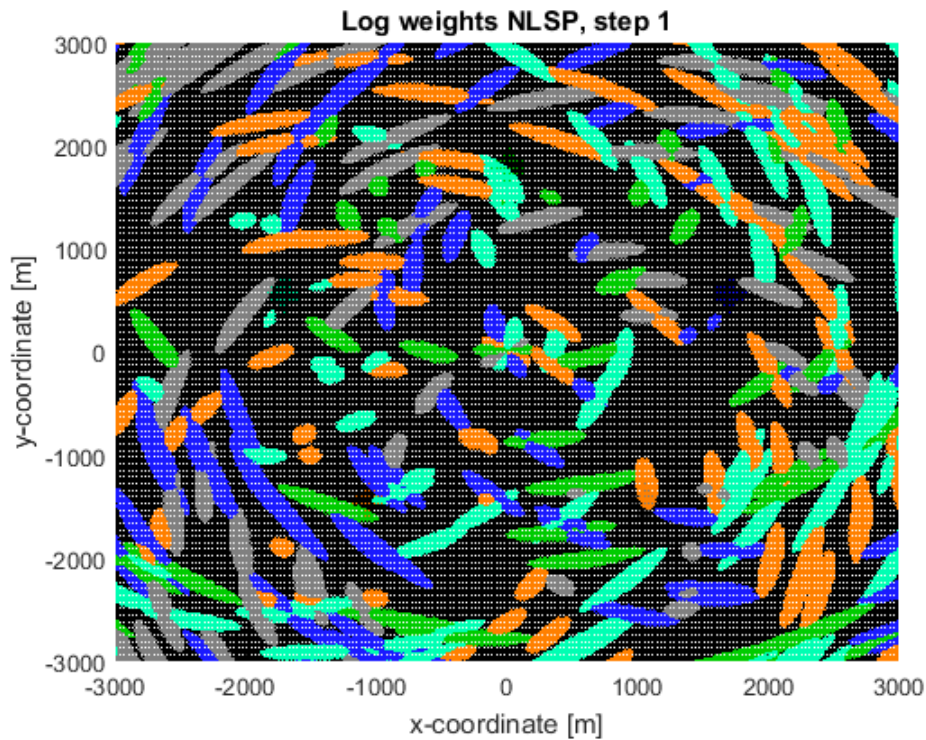


Obrázek 5.9: Viz popis k obrázku 5.7. Pro lepší zobrazení byla použita velmi vysoká hodnota $\mu^{(s)} = 170$.

5.4. OBRÁZKY K POPISU PROGRAMU SIMULACE



Obrázek 5.10: Pohled na obrázek 5.9 shora. Zde jsou již mnohem jasněji vidět banánové funkce příslušné jednotlivým sensorům. Pro tyto funkce jsou použity parametry $\mathbf{C}_v = \text{diag}([5^2, 0, 75])$.



Obrázek 5.11: Příklad banánových funkcí s parametry $\mathbf{C}_v = \text{diag}([15^2, 1, 75])$, pro šum byl použit parametr $\mu^{(s)} = 170$.

6. Výsledky simulace a porovnání aproximačních metod

V této kapitole jsou shrnuty výsledky simulace a jsou zde porovnány metody použité k aproximaci věrohodnostní funkce z hlediska jejich výpočetní složitosti.

6.1. Výsledky

K nastavení simulace je použit popis z kapitoly 5.1. Stejné nastavení je zachováno i v souboru `testbed`.

Je vhodné připomenout, že simulace probíhala pro 200 časových kroků, 2000 částic ve smyslu DPF, pro dilataci 0,08 a pro nastavení nejvýše 40 použitých atomů slovníku (tedy nejvýše 40-řádkového vektoru expanzních koeficientů). Obvyklý počet reálně použitých atomů se pohyboval okolo deseti.

Simulace probíhala pomocí skriptu `testscript`, ze kterého byl 900-krát spuštěn skript `testbed`. Výsledky simulací jsou porovnávány dle průměrné střední kvadratické chyby `meanRMSE`, která vznikla sečtením střední kvadratické chyby pro každou metodu a každé spuštění skriptu `testbed` zvlášť, a která byla následně podělena počtem kroků `timeSteps`. Ke každé metodě je tedy vypočteno 900 průměrů středních kvadratických chyb. Výsledky simulací jsou k vidění na obrázcích 6.1–6.3.

K vykreslení výsledků byly použity krabicové grafy (boxploty). V nich červené horizontální čáry označují medián, tedy kvantil $Q_{0,5}$. Dolní a horní části „krabic“ znázorňují percentily $Q_{0,25}$ a $Q_{0,75}$. Vzdálenost mezi těmito percentily se nazývá mezikvartilové rozpětí. Tzv. vousy vyjadřují variabilitu dat. Hodnoty, které jsou dál než konec vousů jsou obvykle označovány jako odlehle hodnoty (outliery), jsou počítány jako $1,5(Q_{0,75} - Q_{0,25})$ a v Matlabu jsou zobrazovány červenými křížky. Pro interpretaci výsledků simulace jsem jako hraniční hodnotu pro outliery zvolila průměr RMSE jedno sto. Za touto hodnotou leží ztracená pozorování, tedy případy, kdy algoritmus sledování objektu selže. Toto nastává např. je-li šum v globální LF mnohem výraznější než sledovaný objekt.

Pozorování, která nejsou ztracená nazýváme pozorováními správnými. Z těchto je spočítán průměr a směrodatná odchylka, které jsou na obrázcích 6.1–6.3 označeny černými a zelenými křížky.

V tabulkách níže jsou uvedeny tyto charakteristiky včetně hodnot mediánu a ztracených sledování v porovnání s originální metodou. Ta by měla vykazovat nejlepší výsledky, jelikož má nejvíce informací.

DPF s fúzním centrem

Charakteristika	Orig.	OMP fcb	LS fcb	NLS fcb	NLSP fcb
průměr	3,55	3,70	3,11	3,05	3,40
sm. odchylka	7,12	6,63	5,47	5,67	6,50
medián	1,56	1,63	1,53	1,57	1,60
outliery [%]	4,77	6,22	5,33	5,11	4,55

6.2. VÝPOČETNÍ SLOŽITOST

DPF s konsenzem

Charakteristika	Orig.	OMP cb	LS cb	NLS cb	NLSP cb
průměr	3,55	3,57	3,45	3,36	3,23
sm. odchylka	7,12	6,88	6,84	7,08	5,57
medián	1,56	1,60	1,55	1,54	1,59
outliery [%]	4,77	6,55	4,33	4,33	5,11

Z obrázků simulací i z hodnot uvedených v tabulkách plyne, že rozdíly mediánů průměrů RMSE jednotlivých aproximací jsou velmi malé. Pohybují se v rozmezí 1,53 – 1,63, a ačkoli se dle nich jako nejlepší jeví metoda nejmenších čtverců pro fúzní centrum a jako nejhorší ortogonální sdružovací metoda pro fúzní centrum, nelze podle nich jednoznačně určit nejlepší metodu vhodnou k aproximacím. Metody nelze věrohodně rozdělit ani na základě průměru, kde se hodnoty pohybují mezi 3,05 – 3,70, či směrodatné odchylky s hodnotami 5,47 – 7,12, jelikož tyto hodnoty závisí na hranici mezi ztracenými a správnými sledováními objektu.

Viditelné rozdíly vykazuje až procento ztracených sledování 4,33 – 6,55 %, kde se jako nelepší ukazuje metoda nejmenších čtverců a normalizovaná metoda nejmenších čtverců, obě počítány konsenzem. Naopak nejhorší jsou oba typy ortogonální sdružovací metody, kde je 6,55 % ztracených sledování pro konsenzus a 6,22 % ztracených pro fúzní centrum.

6.2. Výpočetní složitost

Jelikož jsou výsledky simulace pro různé metody aproximace podobné, je vhodné rozlišit tyto metody i na základě jejich výpočetní složitosti [6]. Tím získáme představu např. o energetických nárocích výpočtů.

Hrubý časový odhad potřebný k provedení nějakého výpočtu lze jednoduše zjistit sečtením všech základních aritmetických operací (sčítání, odčítání, násobení a dělení). Tyto operace jsou nazývány operacemi s pohyblivou řádovou čárkou (ang. floating point operations, zkr. FLOPs) a to z důvodu uložení čísel v počítači právě ve formátu s pohyblivou řádovou čárkou. Výpočetní složitost operace určuje počet FLOP potřebných k jejímu provedení. Protože čas, který potřebuje počítač k provedení výpočtu, závisí na mnoha faktorech, jsou u počtu FLOP obvykle zanedbávána malá čísla (např. odečtení jedničky). Pro velmi hrubé odhady jsou dokonce ignorovány i číselné násobky (tj. počet $2n$ přejde na n).

V následujících tabulkách jsou uvedeny vybrané operace a počty FLOP potřebných k jejich provedení. Malá čísla jsou zanedbána. Symboly \mathbf{x} a \mathbf{y} jsou označeny vektory délky n , a značí skalár a $\mathbf{A}^{m \times n}$, $\mathbf{B}^{m \times n}$, $\mathbf{C}^{n \times p}$, $\mathbf{D}^{n \times n}$ jsou matice daných typů.

Vektorové operace

Operace	Počet FLOP	Vysvětlení
$a\mathbf{x}$	n	n -krát operace sčítání (= 1 FLOP)
$\mathbf{x} + \mathbf{y}$	n	n -krát sčítání
$\mathbf{x}^T \mathbf{y}$	$2n$	n -krát násobení + $(n - 1)$ -krát sčítání
$\ \mathbf{x}\ $	$2n$	n -krát násobení + $(n - 1)$ -krát sčítání
$\ \mathbf{x} - \mathbf{y}\ $	$3n$	n -krát odečítání + n -krát násobení + $(n - 1)$ -krát sčítání

Maticové operace

Operace	Počet FLOP	Vysvětlení
$a\mathbf{A}$	mn	mn -krát sčítání
$\mathbf{A} + \mathbf{B}$	mn	mn -krát sčítání
$\mathbf{A}\mathbf{x}$	$2mn$	$(n$ -krát násobení + $(n - 1)$ -krát sčítání) m -krát
$\mathbf{A}\mathbf{C}$	$2mnp$	$(n$ -krát násobení + $(n - 1)$ -krát sčítání) mp -krát
$\mathbf{A}^T\mathbf{A}$	mn^2	$(m$ -krát násobení + $(m - 1)$ -krát sčítání) n^2 -krát
\mathbf{D}^{-1}	$3n^3$	vychází z QR rozkladu [6]
\mathbf{A}^+	$3mn^2$	vychází z QR rozkladu [6]

Nyní již lze přistoupit k porovnání aproximačních metod dle jejich výpočetní složitosti.

Originální metoda Zde probíhá jen vynásobení vah (součin lokálních LF) odpovídajících příslušným částicím přes všechny senzory S . Pro J částic a S senzorů je složitost výpočtu $J(S - 1)$, tedy JS .

Metoda nejmenších čtverců Nejdříve vztah pro výpočet řešení metodou nejmenších čtverců $\hat{\alpha}^{(s)} = (\Psi^{(s)T}\Psi^{(s)})^{-1}\Psi^{(s)T}\beta^{(s)}$ rozdělíme na dílčí části. Připomeňme, že $\hat{\alpha}^{(s)}$ je typu $n_\psi \times 1$, $\Psi^{(s)}$ je typu $J \times n_\psi$ a $\beta^{(s)}$ je typu $J \times 1$.

- $\Psi^{(s)T}\Psi^{(s)}$ násobení matic typu $n_\psi \times J$ a $J \times n_\psi$, výsledná složitost je tedy Jn_ψ^2 ;
- $(\Psi^{(s)T}\Psi^{(s)})^{-1}$ invertování matice typu $n_\psi \times n_\psi$ má složitost $3n_\psi^3$;
- Násobení výrazů $(\Psi^{(s)T}\Psi^{(s)})^{-1}$ a $\Psi^{(s)T}$, tedy násobení matic typu $n_\psi \times n_\psi$ a $n_\psi \times J$, má složitost Jn_ψ^2 ;
- Násobení výrazů $(\Psi^{(s)T}\Psi^{(s)})^{-1}\Psi^{(s)T}$ a $\beta^{(s)}$, tedy násobení matic typu $n_\psi \times J$ a $J \times 1$, má složitost Jn_ψ .

Složitost celého výrazu je vyjádřena součtem $Jn_\psi + 2Jn_\psi^2 + 3n_\psi^3$. Pro získání složitosti k -řádkého řešení je třeba vzít v potaz i seřazení expanzních koeficientů dle velikosti, např. quicksort má složitost n_ψ^2 . Výslednou složitost určuje výraz

$$3n_\psi^3 + n_\psi^2 + 2Jn_\psi^2 + Jn_\psi, \quad (6.1)$$

který pro velká J a n_ψ a pro $J > n_\psi$ přejde na Jn_ψ^2 .

Normalizovaná metoda nejmenších čtverců Tedy metoda nejmenších čtverců s normalizovaným slovníkem.

- Normalizujeme jednotlivé sloupce slovníku $\Psi^{(s)}$, kde normalizace jednoho atomu má složitost $2J$ a normalizujeme n_ψ atomů. Získanými normami sloupců podělíme odpovídající sloupce matice, složitost je Jn_ψ . Výsledná složitost pak $3Jn_\psi$;
- Metoda nejmenších čtverců k -řádkého řešení má složitost $Jn_\psi + 2Jn_\psi^2 + n_\psi^2 + 3n_\psi^3$.

Složitost celého výrazu je

$$3n_\psi^3 + n_\psi^2 + 2Jn_\psi^2 + 4Jn_\psi, \quad (6.2)$$

který pro velká J a n_ψ a pro $J > n_\psi$ přejde na Jn_ψ^2 .

6.3. OBRÁZKY K VÝSLEDKŮM SIMULACE

Normalizovaná metoda nejmenších čtverců s přefitováním Normalizovaná metoda nejmenších čtverců, kde je pro k nejvýznamnějších atomů příslušných k největším expanzním koeficientům provedená opět metoda nejmenších čtverců.

- Normalizace sloupců slovníku má složitost $3Jn_\psi$;
- Metoda nejmenších čtverců k -řádkého řešení má složitost $Jn_\psi + 2Jn_\psi^2 + n_\psi^2 + 3n_\psi^3$;
- Metoda nejmenších čtverců počítána jen s k nejvýznamnějšími atomy má složitost $Jk + 2Jk^2 + k^2 + 3k^3$.

Výsledná složitost algoritmu je vyjádřena součtem

$$3n_\psi^3 + n_\psi^2 + 2Jn_\psi^2 + 4Jn_\psi + Jk + 2Jk^2 + k^2 + 3k^3, \quad (6.3)$$

který pro velká J a n_ψ a pro $J > n_\psi$ přejde na Jn_ψ^2 .

Ortogonální sdružovací metoda Jednotlivé kroky rozdělíme opět na dílčí části. Písmenem i značíme i -tou iteraci algoritmu, matice \mathbf{S} je typu $J \times i$

- Nejdříve hledáme $b_0 = \arg \max_b \left| \left[\psi_b(\mathbf{x}_n^{(s,1)}), \dots, \psi_b(\mathbf{x}_n^{(s,J)}) \right]^T \mathbf{r} \right|$. Je provedeno n_ψ operací součinu dvou vektorů, pro který je potřeba $2J$ operací. Následně je vybrán největší ze součinů, pro quicksort je složitost n_ψ^2 . Nalezení b_0 má tedy složitost $2Jn_\psi + n_\psi^2$;
- Metodou nejmenších čtverců řešíme $\hat{\mathbf{c}} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \boldsymbol{\beta}^{(s)}$, jehož složitost je $Ji + 2Ji^2 + 3i^3$;
- Výraz $\mathbf{r} = \boldsymbol{\beta}^{(s)} - \mathbf{S}\hat{\mathbf{c}}$ je složen z násobení matice a vektoru se složitostí $2Ji$ a z odečítání dvou vektorů se složitostí J . Výsledná složitost je tedy $J + 2Ji$;
- Norma $\epsilon = \|\mathbf{r}\|_2$ má složitost $2J$.

Složitost posledních tří operací (navýšení počtu iterací a porovnání dvou ukončovacích kritérií) lze zanedbat. OMP má složitost

$$n_\psi^2 + 2Jn_\psi + 2Ji^2 + 3i^3 + 3Ji + 3J, \quad (6.4)$$

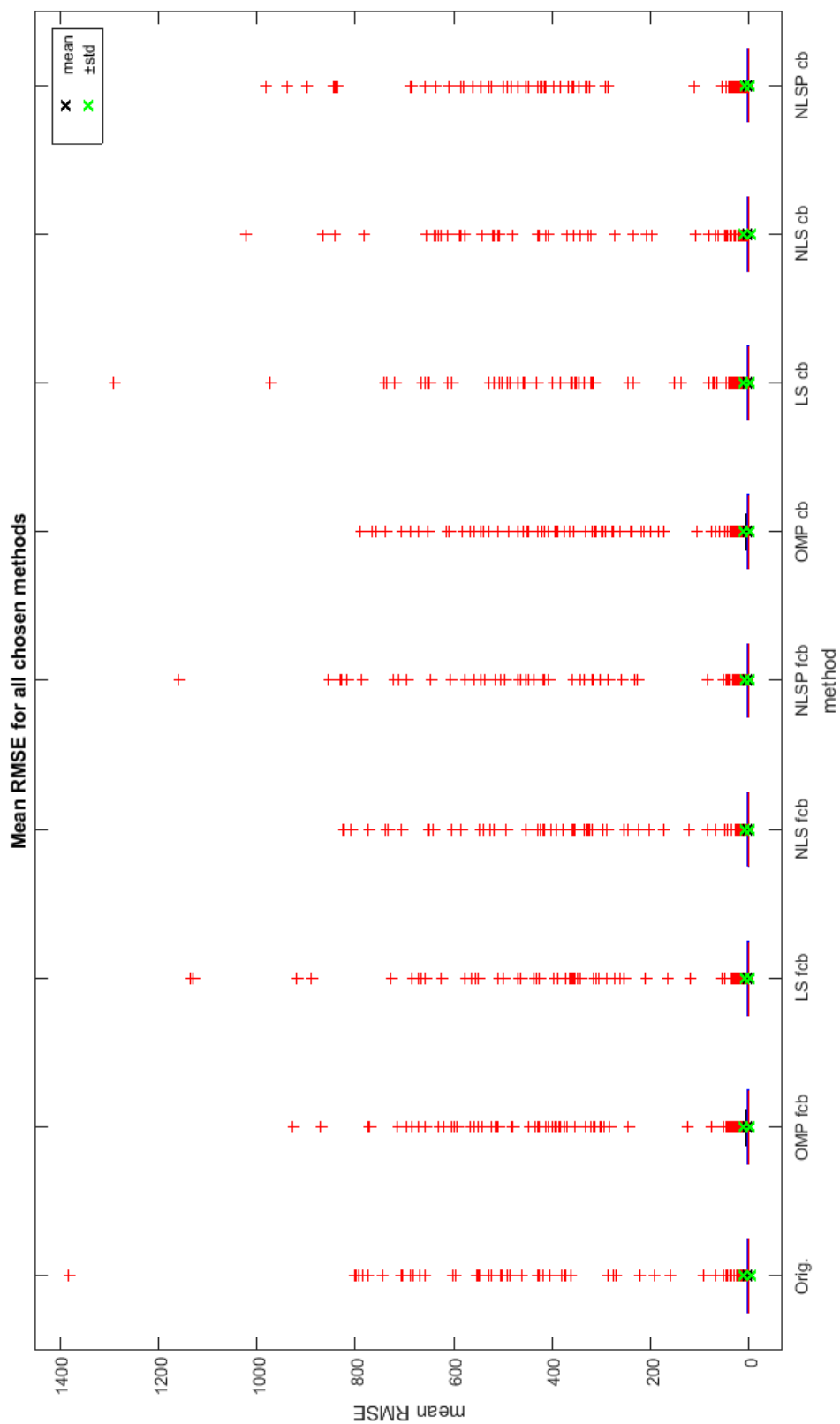
který pro velká J a n_ψ , pro $J > n_\psi$ a pro nejhorší možný případ, kdy $i = n_\psi$, tedy pro n_ψ iterací, přejde na Jn_ψ^3 .

Je zřejmé, že ortogonální sdružovací metoda je výpočetně nejnáročnější, na rozdíl od různých modifikací metody nejmenších čtverců, které jsou téměř srovnatelné. Naopak nejméně náročná je originální metoda, ve které však neprobíhá žádná aproximace. Tato metoda se navíc nedá kvůli velkému množství dat použít pro výpočet pomocí DPF.

6.3. Obrázky k výsledkům simulace

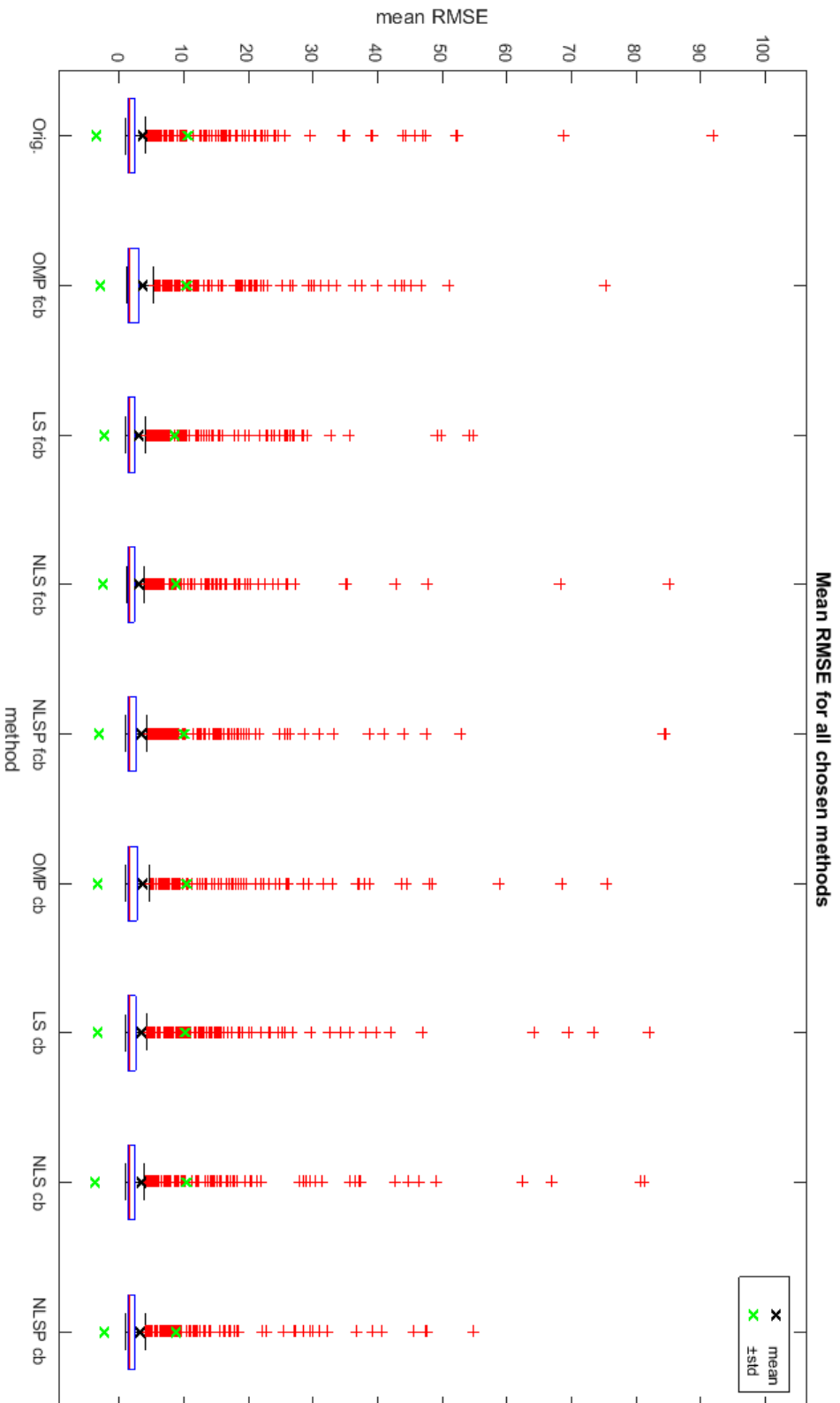
Do této kapitoly jsou zařazeny tři obrázky s krabicovými grafy simulací obsahující jednotlivé metody.

6. VÝSLEDKY SIMULACE A POROVNÁNÍ APROXIMAČNÍCH METOD

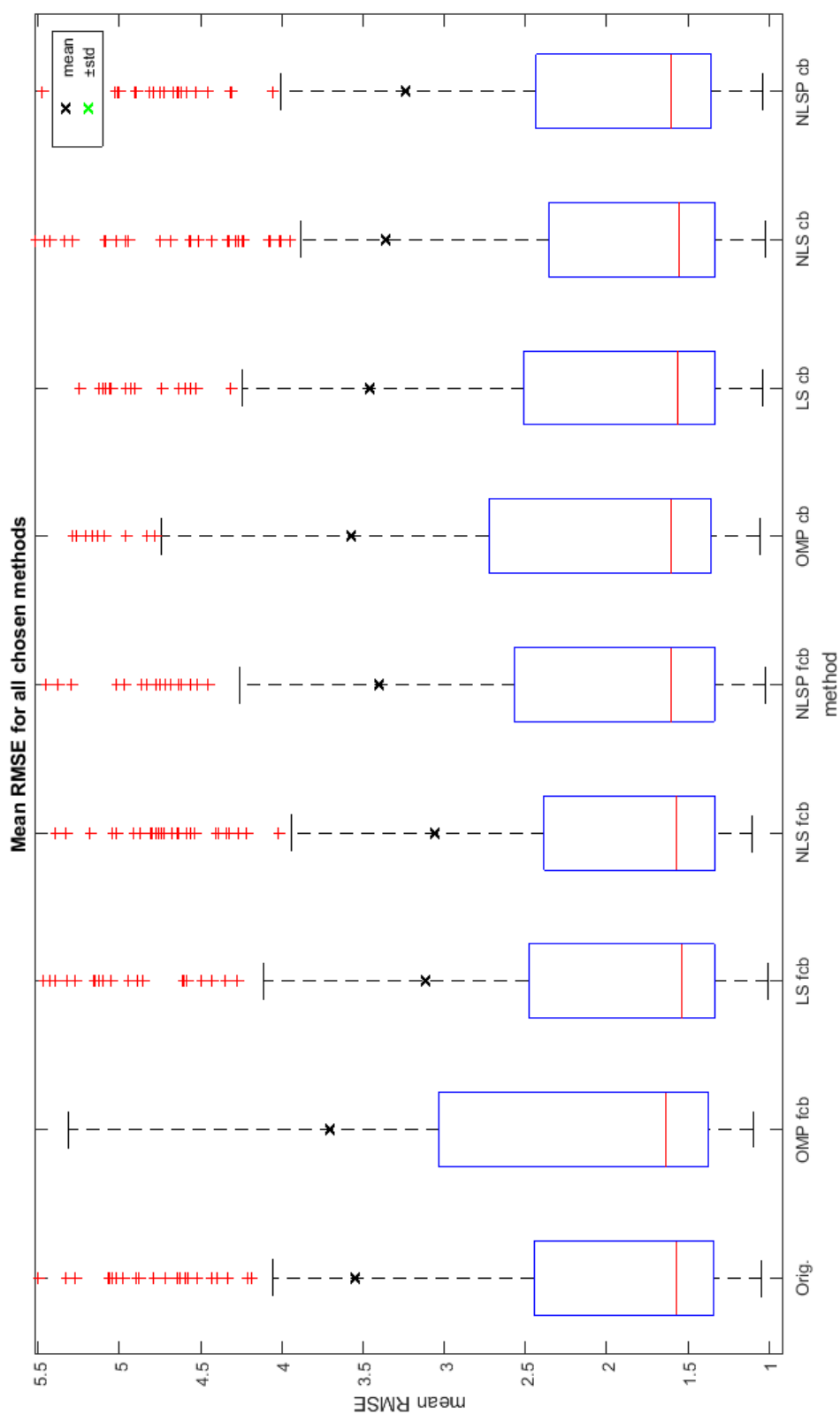


Obrázek 6.1: Zde jsou vykresleny všechny výsledky simulací, jedná se o 900 spuštění pro každou metodu. Na svislé ose je průměrná střední kvadratická chyba, na vodorovné ose leží jednotlivé aproximační metody.

6.3. OBRÁZKY K VÝSLEDKŮM SIMULACE



Obrázek 6.2: Po přiblížení obrázku 6.1 si lze všimnout, že většina hodnot je koncentrována pod průměrem střední kvadratické chyby s hodnotou 100. Z toho důvodu byla tato hodnota zvolena jako hraniční hodnota mezi správnými a ztracenými sledováními objektu. Hodnoty průměru a směrodatné odchylky jsou počítány pouze ze správných hodnot.



Obrázek 6.3: Další přiblížení obrázku 6.1, zde jsou již dobře vidět krabicové grafy. Z obrázku plyne, že mediány hodnot průměrů středních kvadratických chyb se pro různé metody téměř neliší. Jednotlivé metody se dle tohoto grafu liší hlavně v průměrech spočtených ze správných sledování a v hodnotách percentilů $Q_{0,75}$

7. Závěr

Práce pojednává o využití řídké reprezentace signálů v distribuovaném částicovém filtru. Konkrétněji se jedná o sledování objektu v decentralizované sensorové síti. V tomto modelu sledování objektu jsou zahrnuty dva typy „nejistot“ měření. Prvním je falešné měření neboli šum, druhým je případ, kdy sensor objekt vůbec nezachytí. Statistickou závislost mezi vektory měření a reálným stavem objektu vyjadřuje věrohodnostní funkce. O této funkci víme, že je řídká, lze tedy k její aproximaci využít řídkých reprezentací signálů.

V této práci byla k aproximaci věrohodnostní funkce použita metoda nejmenších čtverců (s nenormalizovaným i normalizovaným slovníkem a následně i s využitím přefitování) a ortogonální sdružovací metoda. Jako atomy slovníku použitého k aproximaci byly zvoleny centrální kubické bázové splajny, tzv. B-splajny.

Model sledování objektu i se zmíněnými aproximacemi je zprogramován v simulaci, která je přílohou této práce a jejíž popis lze najít v kapitole 5. Z výsledků simulací je patrné, že pro zvolené parametry nastavení simulace jsou metody pro různé charakteristiky velmi podobné. Významnější rozdíl vykazuje jen procento ztracených sledování, kde metoda nejmenších čtverců spolu s normalizovanou metodou nejmenších čtverců, obě počítány konsenzem, mají jen 4,33 % ztracených sledování, nanrozdíl od ortogonálních sdružovacích metod, kde je 6,55 % ztracených sledování pro konsenzus a 6,22 % ztracených sledování pro fúzní centrum.

Jelikož nelze na základě informací ze simulace jednoznačně určit nejlepší metodu aproximace, byly tyto metody porovnány i z hlediska výpočetní náročnosti. Zde se pro hrubé odhady jako nejhorší ukazuje ortogonální sdružovací metoda, zatímco všechny modifikace metody nejmenších čtverců jsou si v tomto ohledu rovny.

Pro další výzkum navrhuji spuštění nových simulací, kdy pro jiné volby parametrů mohou navržené metody vykazovat vyšší rozdíly. Dále je vhodné vylepšit přiložený program např. o další možnosti volby slovníků (zmíněných Gaussovských atomů) nebo o automatické určení matice sousednosti pro zvolený parametr vzdálenosti a pro libovolný počet sensorů.

V modelu popsaném v této práci je pro zjednodušení uvažováno, že na sledované oblasti je jen jeden objekt. O něm navíc víme, že se na oblasti nachází. Přirozeným dalším zobecněním je neznalost existence objektu na sledované oblasti a dále i větší množství sledovaných objektů.

Literatura

- [1] AGHAJAN, Hamid a Andrea CAVALLARO. *Multi-camera networks principles and applications*. Amsterdam: Elsevier, AP, 2009. ISBN 978-008-0878-003.
- [2] ANĎĚL, Jiří. *Základy matematické statistiky*. Vyd. 3. Praha: Matfyzpress, 2011. ISBN 978-80-7378-162-0.
- [3] 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*. IEEE, 2002, 50(2), 174-188. DOI: 10.1109/78.978374. ISSN 1053587x. Dostupné také z: <http://ieeexplore.ieee.org/document/978374/>
- [4] BAR-SHALOM, Yaakov a Xiao-Rong LI. *Multitarget-multisensor tracking: principles and techniques*. 3rd printing. Storrs, Conn: YBS, 1995. ISBN 978-096-4831-209.
- [5] BJORCK, Ake. *Numerical methods for least squares problems*. Philadelphia: SIAM, c1996. ISBN 08-987-1360-9.
- [6] BOYD, S. a L. VANNDENBERGE. Vectors, Matrices, and Least Squares [online]. Stanford: Stanford University [cit. 2017-05-25]. Dostupné z: <https://stanford.edu/class/ee103/mma.pdf>, in preparation.
- [7] BORDIN, Claudio J. a Marcelo G. S. BRUNO. Consensus-based distributed particle filtering algorithms for cooperative blind equalization in receiver networks. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2011, 3968-3971. DOI: 10.1109/ICASSP.2011.5947221. ISBN 978-1-4577-0538-0. Dostupné také z: <http://ieeexplore.ieee.org/document/5947221/>
- [8] BRUCKSTEIN, Alfred M., David L. DONOHO a Michael ELAD. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. *SIAM Review*. 2009, 51(1), 34-81. DOI: 10.1137/060657704. ISSN 0036-1445. Dostupné také z: <http://epubs.siam.org/doi/10.1137/060657704>
- [9] BULLO, Francesco, Jorge CORTÉS a Sonia MARTÍNEZ. *Distributed control of robotic networks a mathematical approach to motion coordination algorithms*. Princeton: Princeton University Press, 2009. ISBN 978-140-0831-470.
- [10] CAPPE, Olivier, Simon J. GODSILL a Eric MOULINES. An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proceedings of the IEEE*. IEEE, 2007, 95(5), 899-924. DOI: 10.1109/JPROC.2007.893250. ISSN 0018-9219. Dostupné také z: <http://ieeexplore.ieee.org/document/4266870/>
- [11] Convolution. *Wolfram MathWorld* [online]. Long Hanborough: Wolfram Research, c1999-2017 [cit. 2017-04-09]. Dostupné z: <http://mathworld.wolfram.com/Convolution.html>
- [12] DAŇKOVÁ, Marie. *Komprimované snímání v perfuzním zobrazování pomocí magnetické rezonance*. Brno: Vysoké učení technické v Brně, Fakulta strojíního inženýrství, 2014. 56 s. Vedoucí Mgr. Pavel Rajmic, Ph.D.

LITERATURA

- [13] DJURIC, P. M., J. H. KOTTECHA, J. ZHANG, Y. HUANG, T. GHIRMAI, M. F. BURGALLO a J. MIGUEZ. Particle Filtering. *IEEE Signal Processing Magazine*. IEEE, 2003, 20(5), 19-38. DOI: 10.1109/MSP.2003.1236770. ISSN 1053-5888. Dostupné také z: <http://ieeexplore.ieee.org/document/1236770/>
- [14] ELAD, Michael. *Sparse and redundant representations: from theory to applications in signal and image processing* [online]. New York: Springer, c2010 [cit. 2017-04-02]. ISBN 978-1-4419-7010-7.
- [15] FARAHMAND, Shahrokh, Stergios I. ROUMELIOTIS a Georgios B. GIANNAKIS. Set-Membership Constrained Particle Filter: Distributed Adaptation for Sensor Networks. *IEEE Transactions on Signal Processing*. IEEE, 2011, 59(9), 4122-4138. DOI: 10.1109/TSP.2011.2159599. ISSN 1053-587x. Dostupné také z: <http://ieeexplore.ieee.org/document/5876336/>
- [16] GORDON, N. J., D. J. SALMOND a A. F. M. SMITH. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*. IET, 1993, 140(2), 107-113. DOI: 10.1049/ip-f-2.1993.0015. ISSN 0956375x. Dostupné také z: <http://digital-library.theiet.org/content/journals/10.1049/ip-f-2.1993.0015>
- [17] HLINKA, Ondrej, Franz HLAWATSCH a Petar M. DJURIC. Consensus-based Distributed Particle Filtering With Distributed Proposal Adaptation. *IEEE Transactions on Signal Processing*. IEEE, 2014, 62(12), 3029-3041. DOI: 10.1109/TSP.2014.2319777. ISSN 1053-587x. Dostupné také z: <http://ieeexplore.ieee.org/document/6804018/>
- [18] HLINKA, Ondrej, Franz HLAWATSCH a Petar 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/>
- [19] HLINKA, Ondrej, Franz HLAWATSCH a Petar M. DJURIC. Likelihood consensus-based distributed particle filtering with distributed proposal density adaptation. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2012, , 3869-3872. DOI: 10.1109/ICASSP.2012.6288762. ISBN 978-1-4673-0046-9. Dostupné také z: <http://ieeexplore.ieee.org/document/6288762/>
- [20] HLINKA, Ondrej, Ondrej SLUCIAK, Franz HLAWATSCH, Petar M. DJURIC a Markus RUPP. Likelihood Consensus and Its Application to Distributed Particle Filtering. *IEEE Transactions on Signal Processing*. IEEE, 2012, 60(8), 4334-4349. DOI: 10.1109/TSP.2012.2196697. ISSN 1053-587x. Dostupné také z: <http://ieeexplore.ieee.org/document/6190768/>
- [21] HRBÁČEK, Radek, Pavel RAJMIC, Vítězslav VESELÝ a Jan ŠPIŘÍK. Řídké reprezentace signálů: komprimované snímání. *Elektrorevue* [online]. 2011, 13(67), 1-8. ISSN 12131539.
- [22] HRBÁČEK, Radek, Pavel RAJMIC, Vítězslav VESELÝ a Jan ŠPIŘÍK. Řídké reprezentace signálů: úvod do problematiky. *Elektrorevue* [online]. 2011, 13(50), 1-10. ISSN 12131539.

- [23] CHENG, Qi a Pramod K. VARSHNEY. Joint State Monitoring and Fault Detection using Distributed Particle Filtering. *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*. IEEE, 2007, , 715-719. DOI: 10.1109/ACSSC.2007.4487308. ISBN 978-1-4244-2109-1. ISSN 1058-6393. Dostupné také z: <http://ieeexplore.ieee.org/document/4487308/> .
- [24] KARÁSEK, Jiří a Ladislav SKULA. *Lineární algebra: Teoretická část*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3100-8.
- [25] KO, Jeong G., Chenyang LU, Mani B. SRIVASTAVA, John A. STANKOVIC, Andreas TERZIS a Matt WELSH. Wireless Sensor Networks for Healthcare. *Proceedings of the IEEE*. 2010, 98(11), 1947-1960. DOI: 10.1109/JPROC.2010.2065210. ISSN 0018-9219. Dostupné také z: <http://ieeexplore.ieee.org/document/5570866/>
- [26] KOTTECHA, J. H. a P. M. DJURIC. Gaussian particle filtering. *IEEE Transactions on Signal Processing*. 2003, 51(10), 2592-2601. DOI: 10.1109/TSP.2003.816758. ISSN 1053-587x. Dostupné také z: <http://ieeexplore.ieee.org/document/1232326>
- [27] MAHLER, Ronald P. S. *Statistical multisource-multitarget information fusion*. Boston: Artech House, c2007. ISBN 978-159-6930-926.
- [28] MOSK-AOYAMA, Damon a Devavrat SHAH. Fast Distributed Algorithms for Computing Separable Functions. *IEEE Transactions on Information Theory*. IEEE, 2008, 54(7), 2997-3007. DOI: 10.1109/TIT.2008.924648. ISSN 0018-9448. Dostupné také z: <http://ieeexplore.ieee.org/document/4544980/>
- [29] OLFATI-SABER, R., J. A. FAX a R. M. MURRAY. Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE*. IEEE, 2007, 95(1), 215-233. DOI: 10.1109/JPROC.2006.887293. ISSN 0018-9219. Dostupné také z: <http://ieeexplore.ieee.org/document/4118472/>
- [30] OLSHEVSKY, Alex a John N. TSITSIKLIS. Convergence Speed in Distributed Consensus and Averaging. *SIAM Journal on Control and Optimization*. 2009, 48(1), 33-55. DOI: 10.1137/060678324. ISSN 0363-0129. Dostupné také z: <http://epubs.siam.org/doi/10.1137/060678324>
- [31] PAPA, Giuseppe, Franz MEYER, Paolo BRACA a Franz HLAWATSCH. *Distributed Bernoulli filtering using likelihood consensus*. *IEEE Trans. Signal Inf. Process. Netw.* 2016, in preparation.
- [32] RAJMÍČ, Pavel, Florian MEYER, Giuseppe PAPA a Franz HLAWATSCH. *Distributed Object Tracking with Sparsity-Exploiting Clutter Rejection*. in preparation.
- [33] RAJMÍČ, Pavel *Řídké a nízkohodnostní reprezentace signálů s aplikacemi*. Brno: 2014. s. 1-154.
- [34] SHIMA, Tal a Steven RASMUSSEN. *UAV cooperative decision and control challenges and practical approaches*. Philadelphia: Society for Industrial and Applied Mathematics, 2009. ISBN 978-089-8718-584.

LITERATURA

- [35] SCHOENBERG, Isaac Jacob. *Cardinal spline interpolation*. 2. Philadelphia: Society for Industrial and Applied Mathematics, 1993. ISBN 978-089-8710-090.
- [36] SLUCIAK, Ondrej, Ondrej HLINKA, Markus RUPP, Franz HLAWATSCH a Petar M. DJURIC. Sequential likelihood consensus and its application to distributed particle filtering with reduced communications and latency. *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers*. IEEE, 2011, 1766-1770. DOI: 10.1109/ACSSC.2011.6190324. ISBN 978-1-4673-0323-1. Dostupné také z: <http://ieeexplore.ieee.org/document/6190324/>
- [37] ŠPIŘÍK, Jan, Pavel RAJMIC a Vítězslav VESELÝ. Reprezentace signálů: od báží k framům. *Elektrorevue* [online]. 2010, 12(111), 1-10. ISSN 12131539.
- [38] *Teorie grafů* [online]. Praha: MFF UK, 2008 [cit. 2017-05-25]. Dostupné z: <http://teorie-grafu.cz/>
- [39] UNSER, Michael. Splines: a perfect fit for signal and image processing. *IEEE*. 1999, 16(6), 22-38. DOI: 10.1109/79.799930. ISBN 10.1109/79.799930. Dostupné také z: <http://ieeexplore.ieee.org/document/799930/>
- [40] USTEBAY, Deniz, Mark COATES a Michael RABBAT. Distributed auxiliary particle filters using selective gossip. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2011, , 3296-3299. DOI: 10.1109/ICASSP.2011.5946726. ISBN 978-1-4577-0538-0. Dostupné také z: <http://ieeexplore.ieee.org/document/5946726/>
- [41] VEMULA, M., M. F. BUGALLO a P. M. DJURIC. Target Tracking in a Two-Tiered Hierarchical Sensor Network. 2006 *IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*. IEEE, 2006, 969-972. DOI: 10.1109/ICASSP.2006.1661132. ISBN 1-4244-0469-X. Dostupné také z: <http://ieeexplore.ieee.org/document/1661132/>
- [42] VESELÝ, Vítězslav a Pavel RAJMIC. *Funkcionální analýza s aplikacemi ve zpracování signálů*. Brno: Vysoké učení technické v Brně, 2015. 164 s. ISBN 978-80-214-5186-5.
- [43] XIAO, L., S. BOYD a S. LALL. A scheme for robust distributed sensor fusion based on average consensus. *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/>
- [44] ZHAO, Feng a Leonidas J. GUIBAS. *Wireless sensor networks an information processing approach*. Amsterdam: Morgan Kaufmann, 2004. ISBN 978-008-0521-725.
- [45] ZHAO, Tong a Arye NEHORAI. Distributed Sequential Bayesian Estimation of a Diffusive Source in Wireless Sensor Networks. *IEEE Transactions on Signal Processing*. 2007, 55(4), 1511-1524. DOI: 10.1109/TSP.2006.889975. ISSN 1053-587x. Dostupné také z: <http://ieeexplore.ieee.org/document/4133054/>

- [46] ZUO, L., K. MEHROTRA, P.K. VARSHNEY a C.K. MOHAN. Bandwidth-Efficient Target Tracking In Distributed Sensor Networks Using Particle Filters. *2006 9th International Conference on Information Fusion*. IEEE, 2006, 1-4. DOI: 10.1109/ICIF.2006.301692. ISBN 1-4244-0953-5. Dostupné také z: <http://ieeexplore.ieee.org/document/4085978/>
- [47] ZVÁRA, Karel a Josef ŠTĚPÁN. *Pravděpodobnost a matematická statistika*. Vyd. 2. Praha: Matfyzpress, 2001. ISBN 80-858-6376-6.

A. Obsah příloženého CD

Na příloženém CD je složka Program obsahující funkce a skripty k simulaci, její výsledky a pomocné funkce a skripty k vykreslování splajnových systémů. Zde je abecední seznam souborů ve zmíněné složce.

Program

approxMenu
approxPlot
approxSplines2
approxSplines3
approxSplines4
approxSplines5
axisXY
basisSpline
bicubic_spline
combineWeights_angConsensus
cubic_spline
evaluate_spline_system2
evaluateTriangle
Fig_basisSplinesMore
Fig_basisSplinesOne
Fig_dVariateFunctions
generateClutteredMeasurements
generateTrajectory
getSensorPosition
getStartStates
getTransitionMatrices
meanRMSE
normalize_dict
omp
particleFiltering
particles
PDAF2
PDAFcb
PDAFfcb
plotAll
resampleSystematic
sampleFromTriangle
testbed
testscript
throwOut
wrapTo180
wrapTo360

