



SLOVENSKÁ TECHNICKÁ  
UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY  
A INFORMATIKY

**Ing. Eduard Mráz**

**AUTOREFERÁT DIZERTAČNEJ PRÁCE**

**Robustná lokalizácia pre drony  
lietajúce vo vnútornom priemyselnom prostredí**

*na získanie vedecko-akademickej hodnosti  
philosophiae doctor, PhD.*

v doktorandskom študijnom programe:

Robotika a kybernetika

v študijnom odbore:

Kybernetika

Forma štúdia:

denná

Bratislava, 2024



**SLOVENSKÁ TECHNICKÁ  
UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY  
A INFORMATIKY**

**Ing. Eduard Mráz**

**AUTOREFERÁT DIZERTAČNEJ PRÁCE**

**Robustná lokalizácia pre drony  
lietajúce vo vnútornom priemyselnom prostredí**

*na získanie vedecko-akademickej hodnosti  
philosophiae doctor, PhD.*

v doktorandskom študijnom programe:	Robotika a kybernetika
v študijnom odbore:	Kybernetika
Forma štúdia:	denná

Bratislava, 2024

**Predkladateľ:** Ing. Eduard Mráz  
FEI STU  
Ilkovičova 3, 841 04 Bratislava 1

**Školiteľ:** Ing. Jozef Rodina, PhD.  
FEI STU  
Ilkovičova 3, 841 04 Bratislava 1

**Oponenti:** doc. Ing. Dušan Nemeč, PhD.  
Katedra riadiacich a informačných systémov  
Žilinská univerzita  
Univerzitná 8215/1, 010 26 Žilina

Ing. Ján Zelenka, PhD.  
Ústav informatiky SAV, v. v. i.  
Slovenská akadémia vied  
Dúbravská cesta 9, 845 07 Bratislava 45

Autoreferát bol rozoslaný dňa .....

Obhajoba dizertačnej práce sa koná ..... o .....  
na **FEI STU**, Ilkovičova 3, 841 04 Bratislava 1,  
v miestnosti .....

prof. Ing. Vladimír Kutiš, PhD.  
dekan FEI STU v Bratislave

# Obsah

<b>Úvod</b>	<b>6</b>
<b>1 Tézy dizertačnej práce</b>	<b>7</b>
<b>2 Návrh lokalizačného systému</b>	<b>8</b>
2.1 Časticový filter . . . . .	9
2.1.1 Spracovanie vstupov . . . . .	10
2.1.2 Inicializácia filtra . . . . .	10
2.1.3 Predikcia . . . . .	12
2.1.4 Selektívna aktualizácia váh . . . . .	12
2.1.5 Prevzorkovanie . . . . .	14
2.2 Implementácia . . . . .	15
<b>3 Dosiahnuté výsledky</b>	<b>18</b>
3.1 Dataset 2024-06-28-14-55-53, experiment 4 . . . . .	19
3.2 Dataset 2024-06-06-10-30-30, experiment 2 . . . . .	21
3.3 Výkonnostné testy . . . . .	23
3.4 Zhodnotenie . . . . .	24
<b>4 Zoznam publikácií</b>	<b>25</b>

# Úvod

Dizertačná práca pojednáva o problematike lokalizácie v mobilnej robotike, konkrétne pre bezpilotné lietajúce prostriedky vo vnútorných prostrediach. Práca dáva prízvuk na robustnosť systému, ktorá je v industriálnych prostrediach často vyžadovaná. Robustnosť v tomto ponímaní znamená nielen odolnosť voči poruchám, ale aj schopnosť systému opakovane a konzistentne dosahovať vopred stanovené parametre. Oproti predchádzajúcej práci [1] [2] si však aktuálna práca dáva za cieľ vytvorenie odolnejšieho systému lokalizácie. Tento cieľ má práca ambíciu dosiahnuť pomocou fúzie niekoľkých senzorov, či algoritmov, do jedného výsledného odhadu polohy. Práca poskytuje alternatívu najmä k ekvivalentným metódam fúzie senzorov, ktoré je možno kategorizovať ako metódy využívajúce Kalmanov filter a jeho rôzne podoby. Poskytnutie alternatívy a celková náväznosť na predchádzajúce práce má stále dôvod v dlhoročnej absencii medzinárodne uznávaného štandardu pre lokalizáciu v mobilnej robotike vykonávanú vo vnútorných priestoroch.

Práca navrhuje mechanizmus fúzie senzorov na báze časticového filtra. Práca obsahuje teoretický návrh, praktickú implementáciu aj overenie riešenia. Experimenty sú vyhotovené v porovnaní s vysoko presným referenčným pozičným systémom OptiTrack. Počas testovania je najmä graficky opísaných niekoľko datasetov zozbieraných počas reálnej letovej prevádzky. Datasety obsahujú niekoľko scenárov, ktorých cieľom je dokázať funkčnosť a robustnosť navrhovaného riešenia. Okrem testov lokalizácie sú v práci uvedené aj testy výpočtovej výkonnosti, ktoré zase hovoria o praktickom nasadení navrhovaného systému tak, aby dokázal fungovať na palubnom počítači UAV.

# 1 Tézy dizertačnej práce

1. Implementácia robustného lokalizačného systému na báze časticového filtra, alebo inej príbuznej pravdepodobnostnej metódy.
2. Použitie fúzie viacerých senzorov a metód odhadu polohy za účelom dosiahnutia zvýšenej stability a presnosti lokalizačného systému.
3. Vyhodnotenie fungovania systému na reálnych testoch podľa referenčného systému – UWB, VICON atď.
4. Optimalizácia algoritmu tak, aby dokázal fungovať na palubnom počítači priamo na drone v reálnom čase – počas letu.

## 2 Návrh lokalizačného systému

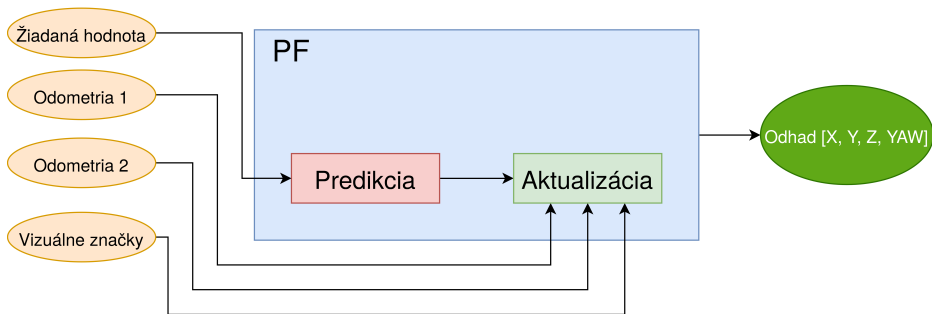
Ideálny lokalizačný systém v kontexte tejto práce by mal mať stanovené nasledovné kritériá:

- Pravdepodobnosť akejkoľvek kolízie , ktorá môže byť spôsobená chybnou lokalizáciou je minimalizovaná.
- Vyžaduje čo najmenšie zásahy do infraštruktúry vnútorného priestoru.
- Pri nasadení do prevádzky vyžaduje čo najmenšie náklady.
- Dokáže fungovať v reálnom čase na palubnom počítači robota (UAV).

Ako vhodná hlavná metóda na implementáciu robustného lokalizačného systému sa javí použitie mechanizmu časticového filtra (z ang. Particle filter, ďalej len PF). PF je uvažovaný najmä z hľadiska schopnosti fúzie viacerých senzorov a metód lokalizácie za účelom presného, no najmä čo najviac robustného odhadu polohy drona vo vnútornom prostredí. Na obrázku 2.1 je znázornený úplne základný náhľad na fungovanie systému.

Dôvod zvolenia si PF ako prvej možnosti spočíva najmä v nelineárnej charakteristike systému. Kalmanov filter (ďalej len KF) aj vo svojich variáciách (EKF, UKF ...) nemá schopnosť optimálne reprezentovať ne-gaussove rozdelenia. Toto je možné riešiť práve použitím PF, ktorý pri rozumnom počte častíc dokáže tento problém prekonať. V neposlednom rade je to taktiež už spomínané vytvorenie alternatívy k prácam a systémom využívajúcich KF, čo má najmä výskumný prínos.





Obr. 2.1: Základná schéma fungovania navrhovaného robustného lokalizačného systému na báze PF.

## 2.1 Časticový filter

Časticový filter je sekvenčná Monte Carlo [3] metóda, ktorá používa veľký počet ováhaných častíc, ktoré štatisticky reprezentujú možné stavy systému. Vo všeobecnosti časticový filter pracuje tak, že najprv propaguje stavy častíc pomocou modelu systému, do ktorého vstupujú reálne riadiace príkazy. Následne sú všetky častice ováhané na základe toho, ako ich stavy zodpovedajú dátam, ktoré boli zozbierané z relevantných senzorov. Odhad stavu sa potom vypočíta ako vážený priemer všetkých (alebo priemer určitej podmnožiny) častíc. Nakoniec nastáva prevzorkovanie, kedy sa tradične niektoré častice odstránia a iné sa naopak vygenerujú.

Princípy PF sú prevzaté z e-knihy autora R. Labbé [4]. Princíp fungovania PF všeobecne pozostáva z nasledovných krokov:

- Náhodné generovanie častíc v priestore - inicializácia
- Predikcia (z ang. Prediction)
- Aktualizácia (z ang. Update)
- Prevzorkovanie (z ang. Resampling)
- Výpočet odhadu (z ang. Estimate)

Podrobný pseudo-kód navrhovaného systému je popísaný v algoritme 1.

V nasledujúcich sekciách budú vysvetlené jednotlivé kroky návrhu robustného lokalizačného systému na báze PF.

### 2.1.1 Spracovanie vstupov

Dáta získané z vizuálnych senzorov musia byť pred použitím v jadre PF predspracované. Toto zahŕňa najmä transformáciu všetkých meraní do spoločného s.s. Takéto spracovanie prebieha pred každou aktualizáciou časticového filtra. Konkrétne v tejto práci je spracovanie vykonávané s frekvenciou 50 Hz. Spracovanie vstupov zahŕňa taktiež zabezpečenie správnosti s.s. senzorov po znovu pripojení pri poruche. Okrem toho sa pri spracovaní vstupov vypočítavajú prírastky odometrie. Prírastky sú potom použité v procese aktualizácie váh v PF namiesto absolútnej polohy. Zámer použitia prírastkov spočíva v čo najväčšej eliminácii naakumulovanej chyby odometrie.

### 2.1.2 Inicializácia filtra

Bezprostredne po vzlete a po dosiahnutí vopred definovanej vzletovej výšky filter čaká na prvé, počiatkové meranie z vizuálnej značky  $p_{A0}$ , resp  $\psi_{A0}$ . Stav (pozície a rotácie) všetkých častíc filtra  $i \in 1..N$  sú inicializované až po bezprostredným získaním takéhoto merania. Tieto častice sú potom náhodne vygenerované vo forme normálnej distribúcie so stredom v hodnotách  $p_{A0}$ ,  $\psi_{A0}$ :

$$p^{(i)} \sim N(p_{A0}, \sigma_{A0}), \quad (2.1)$$

$$\psi^{(i)} \sim N(\psi_{A0}, \sigma_{\psi0}). \quad (2.2)$$

Kovariancie  $\sigma_{A0}$  a  $\sigma_{\psi0}$  ovládajú počiatkové rozloženie častíc, čo v konečnom dôsledku vyjadruje neistotu tohto rozloženia. Keďže počiatková poloha UAV v prostredí je dobre známa, boli zvolené relatívne nízke hodnoty  $\sigma_{A0} = 1.0$  m a  $\sigma_{\psi0} = \pi/4$ . Všetkým časticiam je počas inicializácie priradená rovnaká váha s hodnotou  $w_t^{(i)} = 1/N$ .

---

**Algorithm 1** Pseudo-kód fungovania PF

---

- 1: **Input:** Počet častíc  $N$ , Začiatočná poloha častice  $x_0$  a variancia  $\sigma_0$ , Prahová hodnota  $N_{th}$  pre počet efektívnych častíc  $N_{eff}$
  - 2: Inicializácia množiny častíc  $S = \{x^{(i)}; w^{(i)}\}$ ,  $i \in 1..N$ ,  $x^{(i)} \sim N(x_0, \sigma_0)$ ,  $w^{(i)} = 1/N$
  
  - 3: **loop**
  - 4:     **Input:** Odhady polohy  $p_F, p_B, p_A$  v svetovom s.s. (skr. z súradnicový systém), Rýchlostný riadiaci príkaz  $u$
  - 5:     **for**  $i = 1$  to  $N$  **do** ▷ Predikcia
  - 6:         Šum  $u^* \sim N(0, \sigma_u)$
  - 7:          $x^{(i)} \leftarrow x^{(i)} + (u + u^*)dt$  ▷ Pohyb i-tej častice na základe  
riadiaceho rýchlostného príkazu + šum
  - 8:     **end for**
  - 9:     **for**  $i = 1$  to  $N$  **do** ▷ Selektívna aktualizácia váh
  - 10:         **if**  $p_F, \psi_F$  je k dispozícii **then** ▷ Aktualizácia w.r.t. odhad  
odometrie prednej kamery
  - 11:              $w^{(i)} \leftarrow w^{(i)} * W(p_F, \psi_F, x^{(i)})$
  - 12:         **end if**
  - 13:         **if**  $p_B, \psi_B$  je k dispozícii **then** ▷ Aktualizácia w.r.t. odhad  
odometrie zadnej kamery
  - 14:              $w^{(i)} \leftarrow w^{(i)} * W(p_B, \psi_B, x^{(i)})$
  - 15:         **end if**
  - 16:         **if**  $p_A, \psi_A$  je k dispozícii **then** ▷ Aktualizácia w.r.t. odhad  
polohy z vizuálnej značky
  - 17:              $w^{(i)} \leftarrow w^{(i)} * W(p_A, \psi_A, x^{(i)})$
  - 18:         **end if**
  - 19:     **end for**
  - 20:     **if**  $N_{eff} < N_{th}$  **then**
  - 21:         Prevzorkovanie  $S_t$  podľa váhy  $w^{(i)}$
  - 22:     **end if**
  - 23:     Normalizácia váh  $\sum w^{(i)} = 1$
  - 24:      $\hat{x} \leftarrow estimate(S_t)$  ▷ Výpočet finálneho odhadu stavu
  - 25: **end loop**
-

### 2.1.3 Predikcia

Po inicializácii filtra je proces aktualizácie vykonávaný po každom prijatí správy obsahujúcej rýchlostný príkaz  $u = [v \ \omega_\psi]^T$ , čo ako už bolo spomenuté, sa deje s frekvenciou 50 Hz. V danom časovom kroku  $t$ , pozícia každej častice  $x_t^{(i)} = [p_t^{(i)} \ \psi_t^{(i)}]^T \in S_t$ , je predikovaná pomocou jednoduchého lineárneho pohybového modelu:

$$p_{t+1} = p_t + (v + v^*) dt, \quad (2.3)$$

kde  $v^* \sim N(0, \sigma_{up})$  je vzorka Gaussovho šumu s kovarianciou  $\sigma_{up}$ , ktorá popisuje neistotu v sledovaní žiadaného rýchlostného signálu. Predikcia uhla smerovania funguje podobne:

$$\psi_{t+1} = (\psi_t + (\omega_\psi + \omega_\psi^*) dt) \bmod 2\pi, \quad (2.4)$$

kde vzorka Gaussovho šumu procesu  $\omega_\psi^* \sim N(0, \sigma_{u\psi})$  s kovarianciou  $\sigma_{u\psi}$  simuluje neistotu sledovania žiadanej uhlovej rýchlosti yaw. Modulo operácia zabezpečuje, že predikovaný uhol zostáva v intervale  $\langle 0; 2\pi \rangle$ .

### 2.1.4 Selektívna aktualizácia váh

Ďalším krokom je aktualizácia váh každej častice vzhľadom na viacero zdrojov meraní. Prostredníctvom výpočtu prírastkov odometrie poskytujú algoritmy odometrie počítané z prednej a zadnej kamery odhadované polohy  $\hat{p}_F$ ,  $\hat{p}_B$  a smerovania  $\hat{\psi}_F$ ,  $\hat{\psi}_B$  s príslušnými kovarianciami  $\hat{\sigma}_F$ ,  $\hat{\sigma}_B$ . Pokiaľ je vizuálna značka v danom momente viditeľná kamerou a je aj úspešná jej detekcia, poskytuje odhadovanú polohu  $\hat{p}_A$ , smerovanie  $\hat{\psi}_A$  a kovarianciu  $\hat{\sigma}_A$ . Myšlienkou je selektívne zahrnúť tieto merania len vtedy, keď sú dostupné a dostatočne spoľahlivé.

Funkcia  $W$  je definovaná ako funkcia, ktorá počíta pravdepodobnosť častice  $x_t^{(i)} = [p_t^{(i)} \ \psi_t^{(i)}]^T$  na základe pozorovania polohy  $p$  s kovarianciou  $\sigma_p$  a pozorovania smerovania  $\psi$  s kovarianciou  $\sigma_\psi$ . Táto funkcia je rozdelená na dve čiastkové funkcie  $W_p$  a  $W_\psi$ , ktoré samostatne spracúvajú aktualizáciu polohy a smerovania:

$$W(p, \sigma_p, \psi, \sigma_\psi, x_t^{(i)}) = W_p(p, \sigma_p, x_t^{(i)})W_\sigma(\psi, \sigma_\psi, x_t^{(i)}). \quad (2.5)$$

Funkcia pravdepodobnosti polohy porovnáva pozorovanie polohy  $p$  s polohou častice  $p_t^{(i)}$ :

$$W_p(p, \sigma_p, x_t^{(i)}) = p(p|p_t^{(i)}), \quad (2.6)$$

kde  $p(\cdot)$  je funkcia hustoty pravdepodobnosti (z ang. Probability Density Function, ďalej už len PDF), ktorá vyjadruje normálne rozdelenie s priemerom  $\mu = p_t^{(i)}$  a kovarianciou  $\sigma_p$ .

Pri výpočte rozdielu uhlov smerovania  $\Delta\psi^{(i)}$  medzi pozorovaním smerovania  $\psi$  a smerovaním častice  $\psi_t^{(i)}$  je potrebné ošetriť prechod intervalu medzi 0 a  $2\pi$ . Toto je zabezpečené definovaním funkcie  $D(\alpha_1, \alpha_2)$ , ktorej výstupom je vždy menší možný uhol medzi dvoma smerovaniami, ktoré sú v intervale  $\langle -\pi; \pi \rangle$ :

$$D(\alpha_1, \alpha_2) = \text{atan2}(\sin(\alpha_1 - \alpha_2), \cos(\alpha_1 - \alpha_2)), \quad (2.7)$$

$$\Delta\psi^{(i)} = D(\psi, \psi_t^{(i)}). \quad (2.8)$$

Z toho vyplýva, že častice, ktorých hodnota rozdielu  $\Delta\psi^{(i)}$  sa nachádza v okolí nuly, sú práve tie, ktoré sú najbližšie k pozorovanému uhlu  $\psi$ . To znamená, že ich váha by mala byť vysoká. Pre vysoké hodnoty  $\Delta\psi^{(i)}$  platí opak. Funkcia pravdepodobnosti smerovania  $W_\psi$  vykonáva toto porovnanie výpočtom PDF  $p(\Delta\psi^{(i)})$  s normálnym rozdelením s priemerom  $\mu = 0$  a kovarianciou  $\sigma_\psi$ :

$$W_\sigma(\psi, \sigma_\psi, x_t^{(i)}) = p(\Delta\psi^{(i)}). \quad (2.9)$$

Funkcia aktualizácie váh  $W$  je potom použitá na aktualizáciu predchádzajúcich váh častíc  $w_t^{(i)}$  vzhľadom na všetky dostupné senzorické merania:

$$\begin{aligned}
 w_{t+1}^{(i)} = w_t^{(i)} \cdot & W(\widehat{p}_F, \widehat{\sigma}_F, \widehat{\psi}_F, \widehat{\sigma}_F, x_t^{(i)}) \\
 & \cdot W(\widehat{p}_B, \widehat{\sigma}_B, \widehat{\psi}_B, \widehat{\sigma}_B, x_t^{(i)}) \\
 & \cdot W(\widehat{p}_A, \widehat{\sigma}_A, \widehat{\psi}_A, \widehat{\sigma}_A, x_t^{(i)}).
 \end{aligned} \tag{2.10}$$

Treba poznamenať, že všetky tri zložky funkcie  $W$  nemusia byť počítané v každom kroku. Aktualizácia z odhadu vizuálnej značky  $W(\widehat{p}_A, \dots)$  prebieha len vtedy, keď je značka aktuálne detegovaná a odhad polohy je dostupný. Aktualizácie na základe vizuálnej odometrie  $W(\widehat{p}_F, \dots)$  a  $W(\widehat{p}_B, \dots)$  môžu byť naopak vynechané, pokiaľ je detegovaná porucha prednej alebo zadnej kamery. Táto metóda výpočtu váh sa dá zároveň použiť ako predloha v prípade pridávania ďalších meraní a senzorov do procesu aktualizácie PF.

Nakoniec sú všetky váhy normalizované tak, aby sa získalo pravdepodobnostné rozdelenie, ktoré spĺňa podmienku že  $\sum w^{(i)} = 1$ :

$$w_{t+1}^{(i)} = \frac{w_{t+1}^{(i)}}{\sum_{i=1}^N w_{t+1}^{(i)}}. \tag{2.11}$$

## 2.1.5 Prevzorkovanie

Opakovaným aplikovaním kroku predikcie môže nastať situácia, kedy sa rozptyl častíc časom zväčšuje, čo znamená že v blízkosti skutočného stavu systému je len málo častíc. Tento problém je pomenovaný ako degenerácia filtra [5]. Práve na riešenie degenerácie filtra sa používa prevzorkovanie. Počas tohto procesu sú častice s veľkými váhami duplikované, alebo pribúdajú nové častice v ich blízkom okolí. Častice s nízkymi váhami sú naopak odstránené.

Prevzorkovanie nemusí prebiehať pri každej iterácii PF. Jednak by neustále prevzorkovanie mohlo neblaho vplývať na celý proces štatisticky a jednak by to znamenalo zvýšenú záťaž na výpočtovú jednotku. Na základe oboch týchto dôvodov je v tejto práci využívaná metrika počtu efektívnych častíc  $N_{eff}$  [5], [4]. Táto metrika kvantifikuje počet častíc,

ktoré zmysluplne/efektívne prispievajú k odhadu filtra. Inými slovami,  $N_{eff}$  je metrika, ktorá hovorí o tom, kedy filter začína degenerovať:

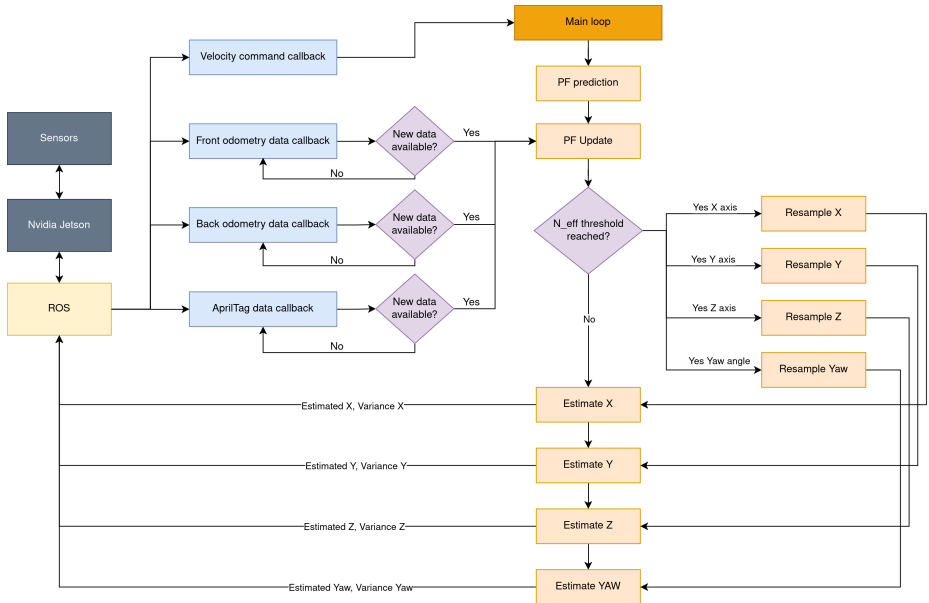
$$N_{eff} = \frac{1}{\sum_{i=1}^N (w_t^{(i)})^2}. \quad (2.12)$$

Prezorkovanie sa spustí vo chvíli, keď  $N_{eff}$  klesne pod prahovú hodnotu  $N_{th}$ , ktorá bola stanovená na hodnotu  $N/4$ . Nová množina častíc  $S_{t+1}$  je generovaná pomocou systematickej metódy prezorkovania (z ang. systematic resampling method) [6], ktorá bola vybraná na základe svojej nízkej algoritmickej zložitosti -  $O(N)$ .

## 2.2 Implementácia

Implementácia PF je v programovacom jazyku Python, ktorý bol zvolený vďaka možnosti rýchleho tzv. rapidného prototypovania (z ang. rapid prototyping). Keďže nemalú časť práce tvorila práve implementácia riešenia navrhnutého v tejto práci, je vhodné uviesť aspoň základné programátorské postupy, ktoré tu boli použité. Diagram naznačujúci implementáciu je vyobrazený na obrázku 2.2.

Ako je možné pozorovať, implementácia PF obsahuje niekoľko tzv. callback funkcií, ktoré sa vykonávajú asynchrónne. Vykonávajú sa v závislosti na tom, či sú dostupné dáta zo senzorov, alebo algoritmov (napríklad vizuálna odometria). Toto zabezpečuje efektívnosť vo vykonávaní jednotlivých krokov. Hlavná programová slučka je vykonávaná v callbacku, kde sú prijímané žiadané rýchlosti priamo z letového kontroléra. Keď je na základe žiadaných rýchlostí vykonaný krok predikcie, prebehne kontrola, ktoré senzorové dáta sú pripravené na to, aby podľa nich mohol prebehnúť krok aktualizácie. Efektívnosť algoritmu je teda v tom, že celá hlavná programová slučka prebehne aj vtedy, pokiaľ sú dostupné iba dáta obsahujúce žiadanú rýchlosť. Síce by odhad opierajúci sa čisto o takéto dáta z dlhodobého hľadiska bol veľmi nepresný, no pri riadiacich algoritmoch je dôležité, aby dáta z modulov, akým je napríklad lokalizácia, prichádzali pravidelne, dostatočne rýchlo a aby boli aktuálne.



Obr. 2.2: Navrhnutý sekvenčný diagram implementácie PF použitého v tejto práci.

Mechanizmus callback funkcií je zaujímavejší napríklad pri už spomínanom odpojení a znovupripojení niektorého zo senzorov. Ak je v jednom momente dostupné meranie len z jedného senzora, tak aktualizácie na základe ostatných senzorov sa jednoducho nepoužijú a nemusí sa na ne čakať, čo by v opačnom prípade mohlo spôsobiť tzv. deadlock (znefunkčnenie) celého systému. Pri znovupripojení jednej z odometrií sú kvôli použitiu prírastkov namiesto používania absolútnej polohy potrebné aspoň dve vzorky dát na to, aby bolo možné prírastok vypočítať. Pri spomínaných frekvenciách toto nespôsobuje žiadne problémy. Posielanie dát medzi jednotlivými procesmi, ako aj ich asynchrónne spracovávanie je vo veľkej miere zabezpečené použitím systému ROS [7].

Optimalizácia algoritmu pôvodne uvažovala s paralelizáciou na CUDA [8] jadrách. GPU sa však hojne používa aj pri vysokoúrovňových aplikáciách, akými sú napríklad autonómne inventarizácie skladov. UAV tu dokáže zúžitkovať kapacitu GPU na detekciu skladových tovarových



štítkov. Preto je rozumné nechať čo najviac GPU výkonu pre aplikácie, ktoré profitujú z využitia GPU viac, ako PF. Optimalizácia algoritmu však bola potrebná. Na štatistické funkcie sa používa knižnica *Filterpy*[9] a na všeobecné matematické operácie knižnica *Numpy*[10]. *Numpy* je masovo využívaná knižnica (tvrdenie na základe GitHub štatistík) a počas profilovania (z ang. profiling) to nebol prvok, ktorý by algoritmus výrazne spomaľoval. To isté platí pre *Filterpy*. Čo sa ukázalo ako najväčší konzument výpočtového času bolo kopírovanie dát prijatých v callback slučkách do premenných programovej triedy *ParticleFilter*, ktorá implementuje celú funkcionality navrhnutú v tejto práci. Niektoré dáta treba napríklad ukladať a následne k nim pristupovať v budúcnosti. Na zistenie týchto úzkych miest celého procesu bol použitý pomerne známy *FlameGraph*[11] a jeho rozšírenie do programovacieho jazyka Python - *python-flamegraph*[12]. Po odstránení tohto problému sa preukázala implementácia PF ako dostatočne efektívna.

### 3 Dosiiahnuté výsledky

Spoločný znak všetkých experimentov je, že sa uskutočnili v priestoroch laboratória bezpilotných lietajúcich systémov nachádzajúceho sa v pešej dostupnosti FEI STU. Laboratórium je vybavené vysoko presným kamerovým pozičným systémom OptiTrack [13]. Výrobca udáva odchýlku nepresahujúcu 0.3 mm pri odhade pozície a  $0.05^\circ$  pri odhade natočenia.

Kvôli charakteristike a obmedzeniam laboratória bola zvolená maximálna možná trajektória s rozmermi 3x3 metre. Rýchlosť pohybu bola nastavená na hodnotu 0.25 m/s počas celej trajektórie v rámci všetkých datasetov. V rohoch trajektórie UAV vždy zastal a znovu akceleroval na požadovanú rýchlosť. Dáta boli zozbieranú priamo počas letu a neboli žiadnym spôsobom upravené. Následne boli spracované pre účely výpočtu odhadov pomocou PF. Pôvodne bolo uvažované (mimo téz dizertačnej práce) o priamom napojení odhadu polohy z implementácie PF do CUBE FMU. Toto však vyžaduje viacero netriviálnych zásahov do firmvéru zariadenia CUBE, čo výrazne presahuje zadanie práce a takýto úkon je časovo náročný. Ďalej teda nemalo žiadny význam počítať odhad polohy počas letu. Kvôli efektívnosti vyhodnotenia výsledkov bol výpočet napokon uskutočnený na štandardnom notebooku. Aby boli dodržané tézy dizertačnej práce, konkrétne:

Optimalizácia algoritmu tak, aby dokázal fungovať na palubnom počítači priamo na drone v reálnom čase – počas letu.

boli uskutočnené záťažové testy na systéme Airvolute DroneCore. Na základe týchto testov je možné konštatovať, že téza je splnená.

### 3.1 Dataset 2024-06-28-14-55-53, experiment 4

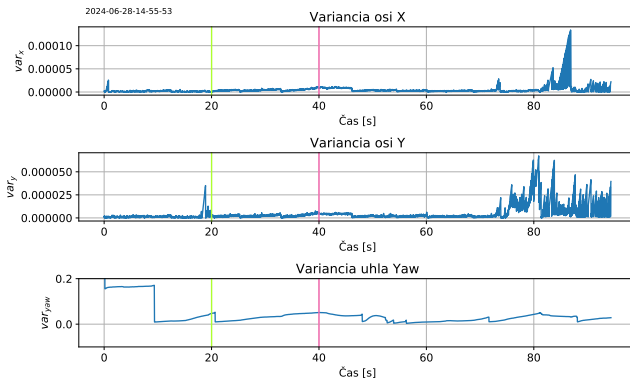
Navrhovaný systém by sa mal vedieť vysporiadať aj so znovupripojením niektorého zo zdrojov lokalizácie. Znovupripojenie benefituje z použitia inkrementov zo zdroja lokalizácie namiesto absolútnej polohy. Po znovupripojení lokalizačný zdroj pokračuje z posledného odhadu PF. Znovu je nastavenie parametrov zachované, tabuľka 3.1.

	X	Y	Yaw	Škála
$\sigma_A$	-	-	-	-
$\sigma_{FS}$	0.7	0.7	1.0	1.5
$\sigma_{BS}$	0.7	0.7	1.0	1.0
$\sigma_{upS}$	1.0	1.0	-	5e-2
$\sigma_{\psi S}$	-	-	0.1	5e-2

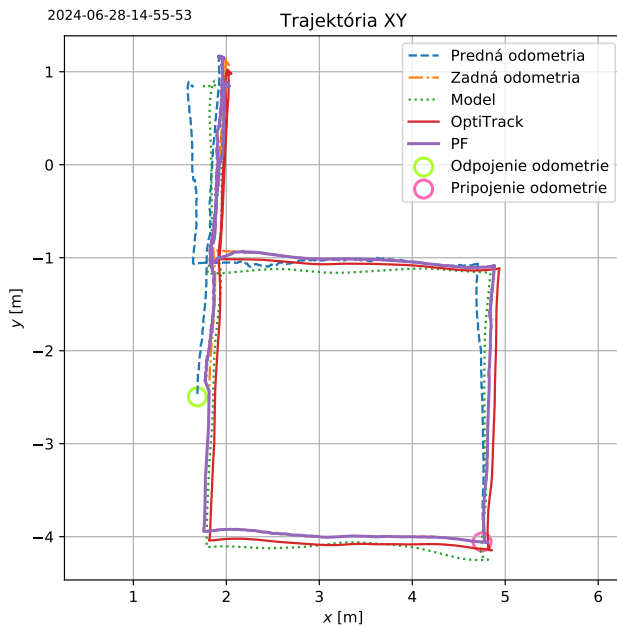
Tabuľka 3.1: Parametre pre dataset 2024-06-28-14-55-53, experiment 4.

Dáta na obrázku 3.2 ukazujú schopnosť navrhovaného PF systému kedykoľvek prijať do procesu odhadu polohy nový lokalizačný zdroj. Čo je možno považovať za nedostatok je fakt, že pokiaľ sa niektorý zo zdrojov znovu pripojí, získava miernu výhodu. A to tú, že jeho poloha sa nachádza v mieste, kde sú v tom čase sústredené častice s najväčšími váhami. Na obrázku 3.2 je to vidno na základe takmer dokonalého prekryvu zadnej odometrie a odhadu PF.

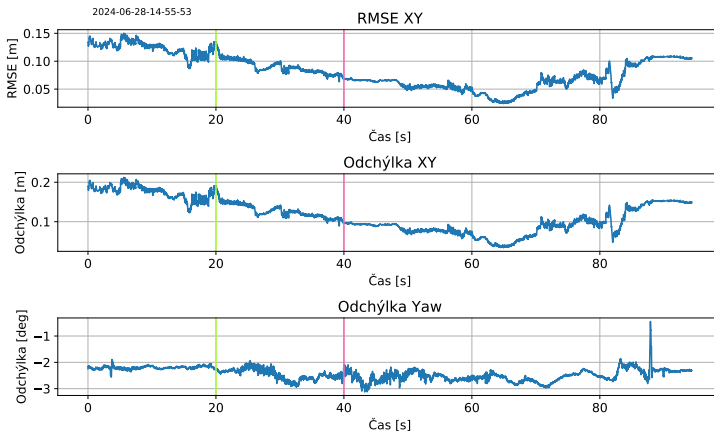
Pri zvyšných nameraných dátach (obrázky 3.1 a 3.3) len možno konštatovať, že filter nedegradoval a fungoval správne po odpojení aj po znovu pripojení lokalizačného zdroja.



Obr. 3.1: Variansia (rozptyl) PF pre osi X, Y a uhol Yaw. Dataset 2024-06-28-14-55-53, experiment 4.



Obr. 3.2: Priebeh odhadu polohy PF v osiach X, Y. Dataset 2024-06-28-14-55-53, experiment 4.



Obr. 3.3: Priebeh odchýlok pre osi X, Y a pre uhol YAW. RMSE pre osi X, Y. Dataset 2024-06-28-14-55-53, experiment 4.

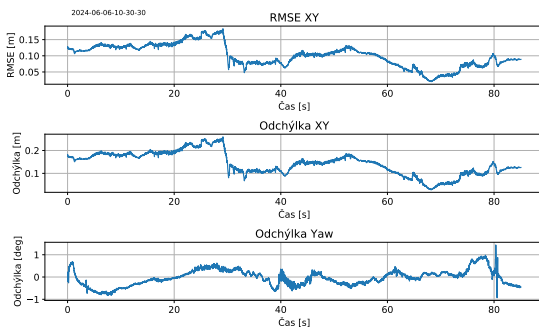
## 3.2 Dataset 2024-06-06-10-30-30, experiment 2

Zmena, ktorú možno badať v parametroch v tabuľke 3.2 je tá, že sú vyplnené hodnoty pre AprilTag. Vplyv týchto parametrov však nie je na prvý pohľad markantný, keďže poloha oboch odometrií sa automaticky nastavuje na polohu vizuálnej značky. Napriek tomu sa aktualizácia váh opiera aj o merania z vizuálnych značiek. Je tomu tak z dôvodu, aby boli posilnené váhy častíc nachádzajúcich sa práve v okolí meraní z vizuálnych značiek. Aktualizácia PF na základe AprilTag značiek prebehla iba v osiach X a Y. Yaw aktualizácia započítaná nebola.

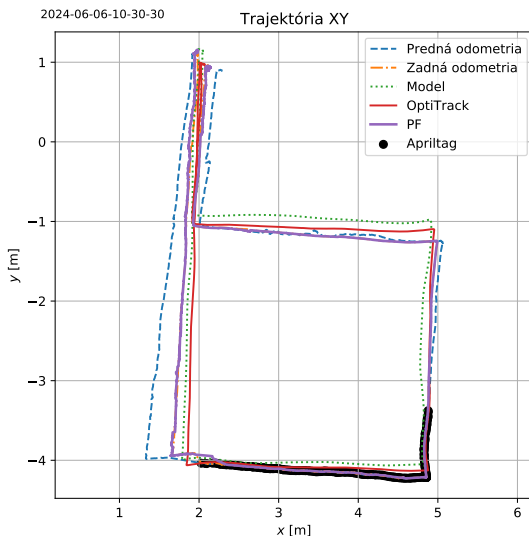
	X	Y	Yaw	Škála
$\sigma_A$	0.1	0.1	-	1.0
$\sigma_{FS}$	0.7	0.7	1.0	1.5
$\sigma_{BS}$	0.7	0.7	1.0	1.0
$\sigma_{upS}$	1.0	1.0	-	5e-2
$\sigma_{\psi S}$	-	-	0.1	5e-2

Tabuľka 3.2: Parametre pre dataset 2024-06-06-10-30-30, experiment 2.

V tomto prípade na obrázku 3.5 možno pozorovať dáta namerané zo značiek AprilTag. Taktiež možno pozorovať nastavenie polohy oboch odometrií na polohu meraní vizuálnych značiek AprilTag.



Obr. 3.4: Priebeh odchýlok pre osi X, Y a pre uhol YAW. RMSE pre osi X, Y. Dataset 2024-06-06-10-30-30, experiment 2.



Obr. 3.5: Priebeh odhadu PF pre osi X, Y. Dataset 2024-06-06-10-30-30, experiment 2.

### 3.3 Výkonnostné testy

Testy sú vykonané na rovnakých dátach, ako experimenty. Testy prebiehali offline výpočtom polohy z reálnych nameraných dát. V tabuľke 3.3 sú uvedené výsledky. Ľavý stĺpec označený  $R_f$  označuje počet cyklov algoritmu PF, za ktoré sa periodicky vykoná prevzorkovanie. Počas testov a práce s algoritmom sa ukázalo, že prevzorkovanie má najviac premenlivý vplyv na výpočtovú záťaž algoritmu PF. To znamená, že prevzorkovanie môže pri niektorom datasete prebiehať každý desiaty cyklus, no v niektorých momentoch aj každý cyklus. V priemere bolo zaznamenané prevzorkovanie každý štvrtý cyklus. Preto je v tabuľke 3.3  $R_f = 4$  zvýraznené. Stĺpec  $R_f = 1$  predstavuje najhorší možný scenár.

$R_f$	Dataset		
	06-10-30-30	28-14-55-53	06-10-34-21
1	23.96 ms	24.21 ms	23.20 ms
<b>4</b>	10.26 ms	11.93 ms	11.94 ms
8	9.86 ms	9.63 ms	9.91 ms

Tabuľka 3.3: Výkonnostné testy na zariadení Nvidia Jetson Xavier NX.

*Pozn.: tabuľka je oproti dizertačnej práci v kompaktnom formáte.*

Z tabuľky 3.3 je možné pozorovať, že ani pri najhoršom scenári algoritmus neklesol pod frekvenciu 40 Hz. Je možné konštatovať, že pri nastavení frekvencii filtra na bezpečnú hodnotu 30 Hz nevyťažší algoritmus naplno ani jedno zo šiestich jadier CPU palubného počítača. 30 Hz je hodnota, ktorá sa aj doposiaľ používala v rôznych predchádzajúcich prácach. Pri týchto prácach sa používala vizuálna odometria, ktorá je počítaná s obnovovacou frekvenciou stereo kamier, čo je taktiež 30 Hz. Tým pádom algoritmus PF necháva pre ostatné aplikácie dostatok výpočtového výkonu. Navyše vizuálnu odometriu, či detekciu AprilTag značiek je možné vykonávať na GPU.

## 3.4 Zhodnotenie

Na základe dosiahnutých výsledkov je možné konštatovať, že tézy dizertačnej práce boli splnené v plnom rozsahu. Implementácia a návrh robustnej lokalizačnej metódy prebehol v poriadku. Metóda zahŕňa použitie troch senzorov, resp. metód lokalizácie. Zvýšená stabilita bola priamo dokázaná v experimentoch. Čo sa týka zvyšovania presnosti, tá je dosiahnutá implicitne. Pokiaľ existuje len jeden zdroj lokalizácie, ktorý má poruchu, tak presnosť je v princípe nekonečne vysoká. Systém sa podarilo porovnať s referenčným systémom OptiTrack, ktorý je považovaný za jeden zo štandardných referenčných polohových systémov v robotike. Nakoniec bola overená aj optimalizácia algoritmu na palubnom počítači UAV. Na základe takéhoto overenia možno konštatovať, že systém dokáže plne fungovať počas letu na palube UAV.



## 4 Zoznam publikácií

- **Mráz, Eduard** – Trizuljak, Adam – Sedláček, Martin – Rajchl, Matej – Štec, Filip – Stanko, Jaromír – Rodina, Jozef. Multi-Sensor Fusion for Robust Localization of Industrial UAVs Using Particle Filter. Journal of Electrical Engineering. eISSN: 1339-309X. [v tlači].
- Sedláček, Martin, **Mráz, Eduard** - Rajchl, Matej - Rodina, Jozef. Environment for UAV Education. In: BALOGH, Richard; OBRŽÁLEK, David; CHRISTOFOROU, Eftychios. Robotics in Education : RiE 2023. Cham, Švajčiarsko: Springer, 2023, s. 257–269. ISBN 978-3-031-38453-0.
- Sedláček, Martin - Rajchl, Matej - **Mráz, Eduard** - Rodina, Jozef. Improvements in indoor UAV localization. In: KOZÁKOVÁ, Alena. ELITECH '23. Bratislava: Vydavateľstvo Spektrum STU, 2023, ISBN 978-80-227-5298-5.
- Duchoň, František - Čornák, Marek - Tölgyessy, Michal - **Mráz, Eduard** - Štec, Filip - Trebuľa, Marek - Chovanec, Ľuboš. Automatica 2022 očami návštevníkov. ATP Journal, 29. s. 58–60. 2022.
- Babinec, Andrej - Rodina, Jozef - **Mráz, Eduard** - Stanko, Jaromír. Using RTAB-Map as a 3D mapping tool for drones. In: KOZÁKOVÁ, Alena. ELITECH '20. Bratislava: Vydavateľstvo Spektrum STU, 2020, ISBN 978-80-227-5001-1.
- **Mráz, Eduard** - Babinec, Andrej - Rodina, Jozef. Using fiducial markers to improve localization of a drone. In: ISMCR 2020. Piscataway: IEEE, 2020, ISBN 978-0-7381-4269-2.
  - 9 citácií v databáze Scopus [14].

– 18 citácií celkovo v Google Scholar [15].

- Babinec, Andrej - Rodina, Jozef - **Mráz, Eduard**. Systém lokalizácie a mapovania pre drony. ATP Journal plus, s. 36–40. 2019.

## **Prednášky**

- DRONTEX 2022. Projekt UAVLIFE - Výskum a vývoj využiteľnosti UAS v boji proti pandémie COVID-19.

## **Projektová činnosť**

- Robustná lokalizácia pre drony v priemysle 4.0. 1/0599/20. 2020 – 2022.
- Navigačný stack pre autonómne drony v priemyselnom prostredí. APVV-21-0352. 2022 – 2024.
- Výskum a vývoj využiteľnosti autonómnych lietajúcich prostriedkov v boji proti pandémie spôsobenej COVID-19. 313011ATR9. 2020 – 2023.

## **Pedagogická činnosť**

- B-AP - Architektúra počítačov 2019/2020 - 2023/2024.
- I-LRS - Lietajúce robotické systémy, 2021/2022.
- 3 obhájené bakalárske práce.

# Bibliografia

1. MRÁZ, Eduard. *Vizuálny systém pre sledovanie objektov lietajúcimi robotmi*. 2017. Bakalárska práca. Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave.
2. MRÁZ, Eduard. *Systém lokalizácie dronu vo vnútornom prostredí priemyselného objektu*. 2019. Diplomová práca. Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave. Dostupné na <https://opac.crzp.sk/?fn=detailBiblioForm&sid=64F3CCE308D9C9BF41EBED333CB3>.
3. FOX, Dieter; BURGARD, Wolfram; DELLAERT, Frank; THRUN, Sebastian. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In: 01. 1999, s. 343–349.
4. LABBE, Roger R. *Kalman and Bayesian Filters in Python* [online]. [cit. 14. 06. 2024]. Dostupné z : <https://github.com/rllabbe/Kalman-and-Bayesian-Filters-in-Python>.
5. YANG, Ning; TIAN, Wei Feng; JIN, Zhi Hua; ZHANG, Chuan Bin. Particle filter for sensor fusion in a land vehicle navigation system. *Measurement Science and Technology*. 02. 2005, roč. 16, č. 3, s. 677–681. Dostupné z doi: 10.1088/0957-0233/16/3/008.
6. LI, Tiancheng; BOLIC, Miodrag; DJURIC, Petar. Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *Signal Processing Magazine, IEEE*. 05. 2015, roč. 32, s. 70–86. Dostupné z doi: 10.1109/MSP.2014.2330626.
7. *ROS.org | About ROS* [online]. [cit. 25. 03. 2019]. Dostupné z : <http://www.ros.org/about-ros/>.
8. NVIDIA. *CUDA | GeForce* [online]. [cit. 14. 06. 2024]. Dostupné z : <https://www.nvidia.com/en-gb/geforce/technologies/cuda/>.

9. *FilterPy — FilterPy 1.4.4 documentation* [online]. [cit. 30.06.2024]. Dostupné z : <https://filterpy.readthedocs.io/en/latest/>.
10. *Numpy* [online]. [cit. 30.06.2024]. Dostupné z : <https://numpy.org/>.
11. *brendangregg/FlameGraph: Stack trace visualizer* [online]. [cit. 30.06.2024]. Dostupné z : <https://github.com/brendangregg/FlameGraph>.
12. *evanhempel/python-flamegraph: Statistical profiler which outputs in format suitable for FlameGraph* [online]. [cit. 30.06.2024]. Dostupné z : <https://github.com/evanhempel/python-flamegraph>.
13. *OptiTrack - Motion Capture for Robotics* [online]. [cit. 30.06.2024]. Dostupné z : <https://optitrack.com/applications/robotics/>.
14. *Mráz, Eduard - Author details - Scopus Preview* [online]. [cit. 09.07.2024]. Dostupné z : <https://www.scopus.com/authid/detail.uri?authorId=57221595835>.
15. *Mráz: Using fiducial markers to improve localization... - Google Scholar* [online]. [cit. 09.07.2024]. Dostupné z : [https://scholar.google.com/scholar?cites=15985718450837074966&as\\_sdt=2005&scioldt=0,5&hl=en](https://scholar.google.com/scholar?cites=15985718450837074966&as_sdt=2005&scioldt=0,5&hl=en).