

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

---

Fakulta elektrotechnická  
Katedra řídicí techniky

## DIPLOMOVÁ PRÁCE



Bc. Martin Paclík

## Navigační systém pro indoor aplikace

**Vedoucí práce:** doc. Ing. Jan Roháč, Ph.D.

**Studijní program:** Otevřená informatika

**Obor:** Počítačové inženýrství

Praha 2019



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Paclík** Jméno: **Martin** Osobní číslo: **420380**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra řídicí techniky**  
Studijní program: **Otevřená informatika**  
Studijní obor: **Počítačové inženýrství**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Navigační systém pro indoor aplikace**

Název diplomové práce anglicky:

**Indoor Navigation System**

Pokyny pro vypracování:

1. Navrhněte a realizujte systém pro "indoor" navigaci bezpilotního prostředku (UAV) sestávající primárně z inerciální měřicí jednotky DMU11 doplněné o LIDAR systém umožňující vyhodnocení vzdáleností od okolních objektů. Jednotlivé části propojte s řídicí a výpočetní jednotkou (např. mikrokontrolér STM32F4xx). Úkolem takto koncipovaného systému je poskytování relativní pozice vůči zadanému počátku souřadnicového systému, rychlosti a orientace v prostoru. Finální řešení by mělo být založeno na fúzi dat realizované pomocí rozšířeného Kalmanova filtru.
2. Proveďte rozbor navigační úlohy z pohledu charakteru pohybu UAV a charakteru okolního prostředí.
3. Realizovanou jednotku otestujte, zkalibrujte a experimentálně ověřte její přesnost.

Seznam doporučené literatury:

- [1] Soták M., Sopata M., Bréda R., Roháč J., Váci L. (ed.): Integrácia navigačných systémov. 1. vyd. Košice: Bréda Róbert, 2006. 344 s. ISBN 80-969619-9-3.
- [2] S. Grewal, P. Andrews: Kalman Filtering - Theory and Practice using Matlab, 3rd edition 2008, Wiley.
- [3] J. A. Farrell, M. Barth: The global positioning system & inertial navigation 2nd edition 1999, Mc graw Hill.
- [4] David Titterton, John Weston: Strapdown Inertial Navigation Tecnology, IEE (Institute of Engineering Technology) 2nd edition 2004.

Jméno a pracoviště vedoucí(ho) diplomové práce:

**doc. Ing. Jan Roháč, Ph.D., katedra měření FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **31.01.2018**

Termín odevzdání diplomové práce: **08.01.2019**

Platnost zadání diplomové práce: **30.09.2019**

\_\_\_\_\_  
doc. Ing. Jan Roháč, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
prof. Ing. Michael Šebek, DrSc.  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



## **Prohlášení**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne .....

Podpis autora: .....

## **Anotace**

Diplomová práce je zaměřena na návrh a realizaci *indoor* navigačního systému založeného na inerciálních senzorech, senzoru barometrického tlaku a čtyřech senzorech vzdálenosti. Součástí práce je rozbor navigační úlohy, návrh navigačního algoritmu založeného na rozšířeném Kalmanově filtru a vyhodnocení výstupů systému zaznamenaných při testovacích experimentech.

## **Annotation**

The diploma thesis is focused on design and realization of indoor navigation system which is based on inertial sensors, barometric pressure sensor and four distance sensors. The thesis includes analysis of the navigation problem, design of the navigation algorithm based on extended Kalman filter and evaluation of system outputs recorded during test experiments.

## **Poděkování**

Děkuji vedoucímu práce doc. Ing. Janu Roháčovi, Ph.D. za konzultace a za čas, který mi věnoval. Dále děkuji Ing. Martinu Šipošovi, Ph.D. za cenné rady nejen z oblasti kalibrace inerciálních senzorů. V neposlední řadě děkuji své rodině za podporu při studiu.





# Obsah

---

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Teoretický rozbor</b>	<b>2</b>
2.1	Souřadnicové systémy a transformace mezi nimi . . . . .	2
2.1.1	Souřadnicový systém navigovaného tělesa . . . . .	2
2.1.2	Nesený tečný souřadnicový systém . . . . .	2
2.1.3	Lokální navigační souřadnicový systém . . . . .	3
2.2	Inerciální senzory . . . . .	4
2.2.1	Akcelerometry . . . . .	4
2.2.2	Senzory úhlové rychlosti . . . . .	5
2.2.3	Chyby inerciálních senzorů . . . . .	6
2.2.4	Kalibrace akcelerometrů . . . . .	7
2.2.5	Kalibrace senzorů úhlové rychlosti . . . . .	8
2.3	Senzory vzdálenosti . . . . .	9
2.3.1	Ultrazvukové senzory . . . . .	9
2.3.2	Laserové senzory . . . . .	9
2.3.3	Infračervené senzory . . . . .	9
2.4	Inerciální navigační systém (INS) . . . . .	10
2.5	Rozšířený Kalmanův filtr (EKF) . . . . .	11
2.6	<i>Indoor</i> navigace . . . . .	14
2.6.1	<i>Simultaneous Localization and Mapping</i> (SLAM) . . . . .	14
2.7	Proložení bodů přímkou pomocí SVD . . . . .	15
2.8	Sběrnice CAN . . . . .	17
<b>3</b>	<b>Hardware</b>	<b>19</b>
3.1	MiniINS v1 . . . . .	19
3.1.1	Inerciální měřicí jednotka (IMU) . . . . .	19
3.1.2	Mikrokontrolér . . . . .	23
3.1.3	Senzor barometrického tlaku . . . . .	23
3.2	TeraRanger Tower . . . . .	23
3.3	Návrh integrace senzorů do jednoho systému . . . . .	25
3.3.1	Zarovnání os a očíslování senzorů . . . . .	26
3.3.2	Výstupy navrženého systému . . . . .	26
<b>4</b>	<b>Návrh navigačního algoritmu</b>	<b>28</b>
4.1	Předpoklady . . . . .	28
4.2	Definice proměnných . . . . .	28
4.3	Geometrie navigační úlohy . . . . .	29

4.4	Předzpracování dat . . . . .	31
4.5	Model EKF . . . . .	32
4.5.1	Model dynamického systému . . . . .	32
4.5.2	Vztah mezi stavovými proměnnými a měřeními . . . . .	33
4.6	Historie bodů . . . . .	34
4.6.1	Extrakce přímk z historie bodů . . . . .	35
4.6.2	Validace parametrů přímk . . . . .	37
4.6.3	Inicializace historie bodů . . . . .	41
4.6.4	Aktualizace historie bodů . . . . .	42
4.7	Navigační algoritmus . . . . .	42
4.7.1	Čekání na inicializaci senzorů . . . . .	42
4.7.2	Sběr dat za klidových podmínek . . . . .	43
4.7.3	Iniciální mapování . . . . .	43
4.7.4	Inicializace EKF . . . . .	45
4.7.5	Běh EKF . . . . .	45
4.8	Omezení a možná rozšíření algoritmu . . . . .	45
4.9	Implementace . . . . .	47
<b>5</b>	<b>Testování</b>	<b>50</b>
5.1	Testovací platforma . . . . .	50
5.2	Závěrečné experimentální ověření . . . . .	51
5.2.1	Výpočet referenční pozice . . . . .	52
5.2.2	Analýza výstupů . . . . .	53
<b>6</b>	<b>Závěr</b>	<b>62</b>
	<b>Seznam použité literatury</b>	<b>64</b>
	<b>Seznam obrázků</b>	<b>67</b>
	<b>Seznam tabulek</b>	<b>69</b>
	<b>Seznam algoritmů</b>	<b>70</b>
	<b>Seznam zkratk</b>	<b>71</b>
	<b>Seznam symbolů</b>	<b>73</b>
	<b>Přílohy</b>	<b>75</b>
A	Kompletní přehled zpráv posílaných po sběrnici CAN . . . . .	75
B	Hodnoty kovariančních matic EKF . . . . .	77
C	Obsah příloženého CD . . . . .	78

# 1 Úvod

---

S rozmachem malých bezpilotních vzdušných prostředků, které mohou létat i uvnitř budov, vznikla potřeba systémů umožňujících jejich lokalizaci právě ve vnitřním prostředí. Pro navigaci mimo budovu lze s výhodou využít některého z globálních navigačních satelitních systémů, které poskytují informace o absolutní pozici navigovaného objektu. Uvnitř budov je ale signál z těchto satelitních systémů slabý kvůli velkému útlumu při průchodu materiálem, a proto není vhodné na něm stavět systém pro *indoor* navigaci.

Ve chvílích, kdy se počítá s provozem bezpilotního prostředku v dané budově, lze rozmístěním různých značek (ať už rádiových nebo obrazových) zajistit možnost získání pozice rozpoznáním těchto značek navigačním systémem. Pokud má být ale navigační systém nezávislý na budově, musí se spolehnout pouze na senzory, které si nese s sebou, a tím pádem se musí obejít i bez informací o absolutní pozici. Takovéto nezávislé systémy se zpravidla spoléhají na inerciální senzory doplněné o senzory vzdálenosti, lidary nebo kamery.

Cílem této diplomové práce je vytvoření *indoor* navigačního systému pro bezpilotní vzdušné prostředky, který je založený na inerciálních senzorech, barometru a čtyřech senzorech vzdálenosti. Aby bylo dosaženo vytyčeného cíle, je třeba učinit několik kroků. Prvním z nich je pospojování senzorů a výpočetní jednotky do jednoho systému a kalibrace tohoto systému. Dalším krokem je rozbor navigační úlohy, navržení navigačního algoritmu založeného na rozšířeném Kalmanově filtru a naprogramování navrženého algoritmu do výpočetní jednotky. Posledním krokem je experimentální ověření funkčnosti a přesnosti vytvořeného navigačního systému.

## 2 Teoretický rozbor

---

### 2.1 Souřadnicové systémy a transformace mezi nimi

V navigaci je používána celá řada souřadnicových systémů pro popis navigačních proměnných. V této části jsou popsány souřadnicové systémy používané realizovaným navigačním systémem a vztahy mezi nimi. Podrobný popis lze najít například v [1] nebo [2].

#### 2.1.1 Souřadnicový systém navigovaného tělesa

Souřadnicový systém navigovaného tělesa (*body coordinate system*) je spjatý s navigovaným objektem. Definován je následovně:

- počátek leží v těžišti tělesa,
- osa  $x^b$  leží v podélné ose tělesa a směřuje kupředu,
- osa  $y^b$  leží v příčné ose tělesa a směřuje na pravobok,
- osa  $z^b$  dotváří pravotočivý souřadnicový systém.

#### 2.1.2 Nesený tečný souřadnicový systém

Nesený tečný souřadnicový systém má:

- počátek v těžišti tělesa,
- osu  $x^t$  rovnou kolmému průmětu osy  $x^b$  do vodorovné roviny procházející těžištěm,
- osu  $y^t$  rovnou kolmému průmětu osy  $y^b$  do vodorovné roviny procházející těžištěm,
- osu  $z^t$  ve směru tíhového zrychlení.

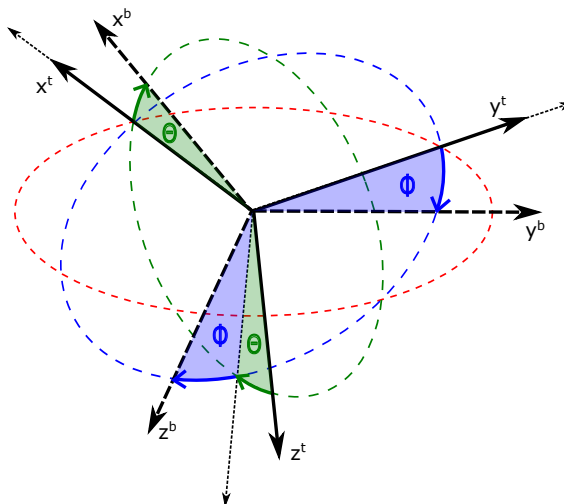
Rovina určená osami  $x^t$  a  $y^t$  je vodorovná. Osa  $x^t$  tečného souřadnicového systému je vlastně kolmým průmětem osy  $x^b$  souřadnicového systému navigovaného tělesa do vodorovné roviny procházející těžištěm. Obdobně osa  $y^t$  je kolmým průmětem osy  $y^b$  do vodorovné roviny procházející těžištěm.

Na obrázku 2.1 je znázorněna orientace souřadnicového systému navigovaného tělesa vůči tečnému. Vzájemná orientace je vyjádřena Eulerovými úhly  $\theta$  (sklon - *pitch*) a  $\phi$  (náklon - *roll*). Vztah mezi vektorem  $\mathbf{v}$  v souřadnicové soustavě navigovaného tělesa (značeno horním indexem  $b$ ) a vektorem  $\mathbf{v}$  v tečné souřadnicové soustavě (značeno horním indexem  $t$ ) je vyjádřen transformační maticí  $\mathbf{C}_b^t$  jako

$$\mathbf{v}^t = \mathbf{C}_b^t \mathbf{v}^b,$$

kde

$$\mathbf{C}_b^t = \begin{bmatrix} \cos \theta & \sin \phi \sin \theta & \cos \phi \sin \theta \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \phi \cos \theta \end{bmatrix}$$



Obrázek 2.1: Orientace souřadnicového systému navigovaného tělesa vůči tečnému, inspirováno obálkou [1]

Pro jakoukoli transformační matici platí, že zpětná transformace se provede pomocí její transpozice. Tedy pro transformační matici  $C_b^t$  platí, že

$$\mathbf{v}^b = C_b^{tT} \mathbf{v}^t$$

### 2.1.3 Lokální navigační souřadnicový systém

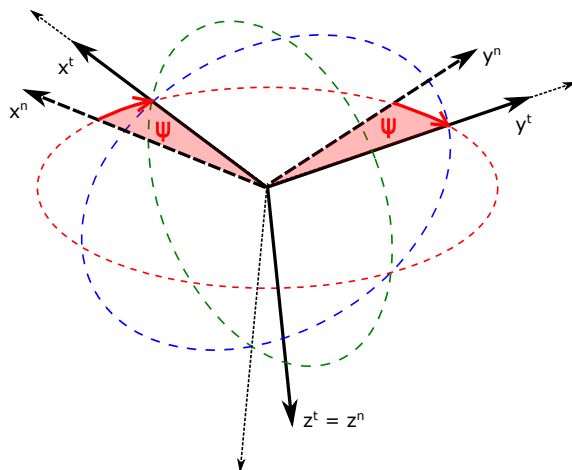
Navigační souřadnicový systém je spjatý se zemskou vztažnou soustavou. Počátek je umístěn v přesně definovaném místě na Zemi a osa  $z^n$  směřuje ve směru, nebo proti směru zemské tíže. Směr osy  $x^n$  a  $y^n$  závisí na konkrétním lokálním navigačním systému. Běžně se používá například systém NED (*North-East-Down*), kdy osa  $x^n$  směřuje na sever a  $y^n$  na východ a osa  $z^n$  ve směru zemské tíže nebo systém ENU (*East-North-Up*), kdy osa  $x^n$  směřuje na východ a  $y^n$  na sever a osa  $z^n$  proti směru zemské tíže. Uvnitř budov nelze snadno určit sever, a proto není v realizované navigační jednotce použit NED ani ENU. Použitý lokální navigační souřadnicový systém bude definovaný později.

Vzájemná orientace tečného a navigačního systému je vyjádřena úhlem  $\psi$  (vybočení - *yaw*). Znázornění je na obrázku 2.2. Transformační matice z tečné do navigační soustavy  $C_t^n$  je

$$C_t^n = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Transformace vektoru  $\mathbf{v}^b$  ze souřadnicového systému navigovaného tělesa do navigačního souřadnicového systému se provede dvěma transformacemi. Nejprve se vektor transformuje maticí  $C_b^t$  do tečné soustavy a následně maticí  $C_t^n$  do navigační soustavy. Tedy

$$\mathbf{v}^n = C_t^n C_b^t \mathbf{v}^b$$



Obrázek 2.2: Orientace tečného souřadnicového systému vůči navigačnímu, inspirováno obálkou [1]

Součin transformačních matic lze vyjádřit jednou transformační maticí  $C_b^n$  jako

$$C_b^n = C_t^n C_b^t = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \phi \sin \theta - \cos \phi \sin \psi & \sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \psi \sin \theta & \cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \phi \cos \theta \end{bmatrix}$$

## 2.2 Inerciální senzory

Tato část je věnovaná inerciálním sensorům. Je zde popsán princip jejich činnosti, měřené hodnoty, chyby a kalibrace.

### 2.2.1 Akcelerometry

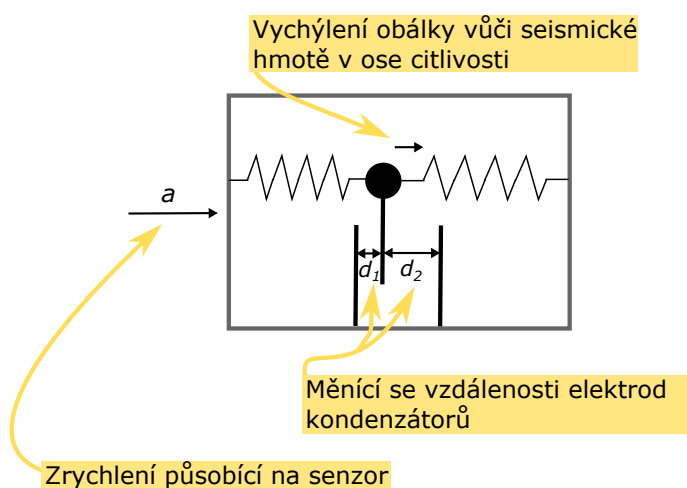
Akcelerometry měří specifickou sílu  $\mathbf{f}_{ib}$ . Dolní index  $ib$  značí specifickou sílu navigovaného tělesa ( $b$  - *body frame*) vůči inerciální vztažné soustavě ( $i$  - *inertial frame*). Naměřená specifická síla se skládá ze zrychlení  $\mathbf{a}_{ib}$  a tíhového zrychlení  $\mathbf{g}$ . V ideálním případě akcelerometr naměří ve volném pádu velikost specifické síly  $\|\mathbf{f}_{ib}\| = 0$  a v klidu  $\|\mathbf{f}_{ib}\| = \|\mathbf{g}\|$ . Platí, že

$$\mathbf{f}_{ib} = \mathbf{a}_{ib} - \mathbf{g}$$

Různých druhů akcelerometrů existuje celá řada. Podrobný popis některých z nich je v [1]. Jelikož v realizovaném navigačním systému je využit MEMS kapacitní akcelerometr, následuje jeho popis.

#### Kapacitní akcelerometry

Princip kapacitního akcelerometru je znázorněn na obrázku 2.3. Základem kapacitního akcelerometru je seismická hmota, které je umožněn pohyb v ose citlivosti. Součástí seismické hmoty je elektroda kondenzátoru. Vnější elektrody jsou uchyceny k obálce akcelerometru. Působí-li na akcelerometr zrychlení, pak dojde k pohybu obálky akcelerometru ve směru zrychlení, zatímco seismická hmota má tendenci zůstat v klidu vůči inerciální



Obrázek 2.3: Princip kapacitního akcelerometru, převzato z [3]

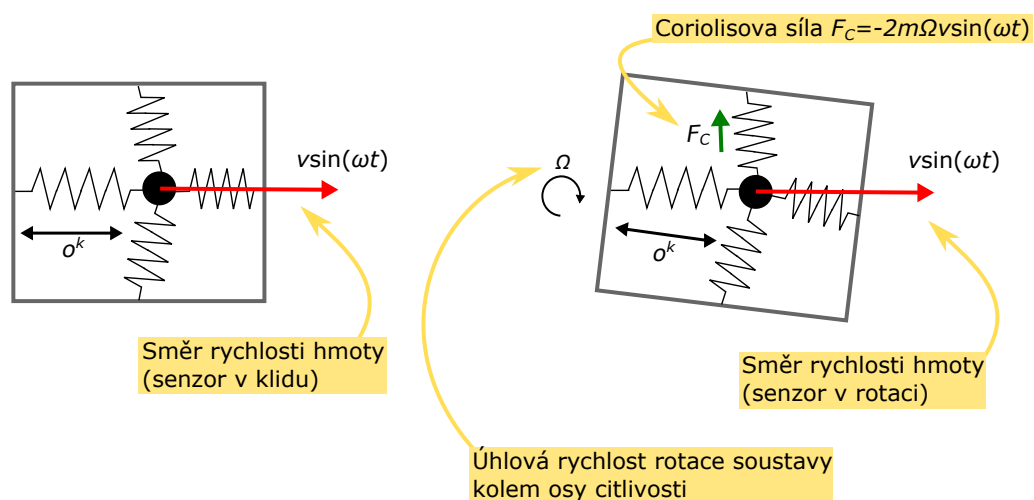
vztažné soustavě. Tím dojde ke změně vzdálenosti elektrod kondenzátoru a ke změně kapacity, podle které se určí výsledná specifická síla.

## 2.2.2 Senzory úhlové rychlosti

Senzory úhlové rychlosti měří, jak už název napovídá, úhlové rychlosti  $\omega_{ib}$ . Opět existuje celá řada druhů. Podrobný popis některých z nich lze najít v [1]. V realizovaném navigačním systému je použit MEMS vibrační senzor úhlové rychlosti, jehož popis následuje.

### Vibrační senzory úhlové rychlosti

Idea vibračního senzoru úhlové rychlosti je znázorněna na obrázku 2.4. V těchto typech senzorů se využívá působení Coriolisovy síly. Coriolisova síla je zdánlivá síla působící na těleso v rotující vztažné soustavě, nepozorovatelná z inerciální vztažné soustavy. Na obrázku 2.4 vlevo je znázorněn model vibračního senzoru úhlové rychlosti v klidu. Černý



Obrázek 2.4: Princip vibračních senzorů úhlové rychlosti, podle [3]

kruh znázorňuje hmotu, která kmitá v ose  $o^k$  kolmé na osu citlivosti senzoru. Vpravo na obrázku je situace při rotaci senzoru. Hmotu koná z pohledu pozorovatele v inerciální vztahné soustavě stále přímočarý pohyb, ale pozorovateli v rotující soustavě přijde, že na hmotu působí síla, která odklání směr rychlosti od osy  $o^k$ . To se projeví vychýlením hmoty v ose kolmé na osu kmitání  $o^k$  a osu citlivosti. Tato výchylka lze měřit například pomocí kondenzátorů, podobně jako v případě kapacitních akcelerometrů (popsáno výše) a lze z ní odvodit úhlovou rychlost.

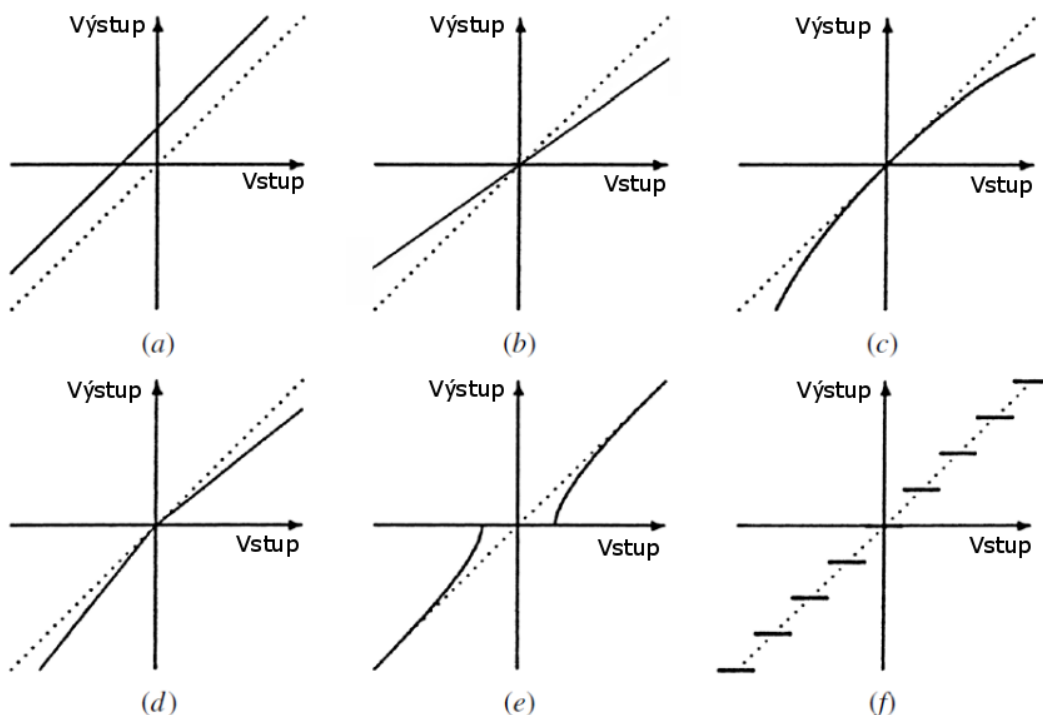
### 2.2.3 Chyby inerciálních senzorů

Měření senzorů jsou zatížena chybami. Některé chyby lze kompenzovat, jiné kompenzovat nelze. Následující popis čerpá z [4] a [5]. Chyby senzorů lze rozdělit na chyby:

1. náhodné,
2. systematické.

Mezi náhodné chyby patří bílý šum, který může mít více příčin. Mezi ně patří například vliv napájení nebo mechanické pnutí v senzoru. Další chybou je šum, který má charakter náhodné procházky (*random-walk*).

Systematické chyby lze na rozdíl od náhodných kompenzovat. Na obrázku 2.5 jsou znázorněny nejběžnější systematické chyby inerciálních senzorů, což jsou:



Obrázek 2.5: Kompensovatelné chyby senzorů: (a) offset, (b) *scale* faktor, (c) nelinearita, (d) asymetrie, (e) hluchá oblast, (f) kvantizace, převzato z [5]



- offset - konstantní chyba měnící se po každém zapojení napájení,
- *scale* faktor,
- nelinearita,
- asymetrie,
- hluchá oblast,
- kvantizace - důsledek digitalizace signálu.

U víceosých senzorů pak přibývá chyba neortogonality, která vzniká tím, že senzory nejsou vůči sobě orientovány zcela přesně kolmo. To má za následek, že výstupy z jednotlivých os víceosého senzoru jsou částečně korelované.

Níže jsou popsány postupy, kterými lze odhadnout některé ze systematických chyb pouze za pomoci náklonné plošiny.

### 2.2.4 Kalibrace akcelerometrů

V [6] a [7] je popsán postup odhadu biasu, chyby neortogonality a *scale* faktoru akcelerometru. Princip tohoto kalibračního postupu je v tom, že velikost specifické síly naměřená akcelerometrem v klidu má být rovna velikosti tíhového zrychlení. Chybový model akcelerometru je zde definován jako

$$\mathbf{u}_a = \mathbf{T}_a \mathbf{S}_a (\mathbf{y}_a - \mathbf{b}_a) = \begin{bmatrix} 1 & 0 & 0 \\ a_{xy} & 1 & 0 \\ a_{zx} & a_{zy} & 1 \end{bmatrix} \begin{bmatrix} s_{ax} & 0 & 0 \\ 0 & s_{ay} & 0 \\ 0 & 0 & s_{az} \end{bmatrix} \left( \begin{bmatrix} y_{ax} \\ y_{ay} \\ y_{az} \end{bmatrix} - \begin{bmatrix} b_{ax} \\ b_{ay} \\ b_{az} \end{bmatrix} \right),$$

kde  $\mathbf{y}_a$  jsou naměřené specifické síly,  $\mathbf{u}_a$  jsou kompenzované specifické síly,  $\mathbf{b}_a$  je vektor biasů,  $\mathbf{S}_a$  je matice se *scale* faktory na diagonále a  $\mathbf{T}_a$  je ortogonalizační matice.

Postup pro odhad parametrů chybového modelu předpokládá možnost namontování kalibrovaného senzoru na náklonnou plošinu, která umožní zafixování jedné z os tak, aby kolem této osy mohla plošina rotovat. Postupně se provede rotace o 360 stupňů kolem každé ze tří os. Během každé rotace je třeba nechat senzor v klidu po dobu několika sekund v alespoň 12 různých polohách (podle [6]). Přesná znalost těchto poloh není potřeba. Doporučené přibližné polohy jsou v [6]. Po třech rotacích je k dispozici 36 několikasekundových úseků měření z 36 různých poloh. Z každého úseku se vypočte střední hodnota, čímž se potlačí vliv šumu na odhad parametrů chybového modelu. Následně se hledá vektor  $\mathbf{x}$ , pro který je funkce  $RMSE(\mathbf{x}, G)$  minimální. Přičemž

$$RMSE(\mathbf{x}, G) = \sqrt{\frac{\sum_{i=1}^m (\|\mathbf{a}_i(\mathbf{x})\| - G)^2}{m}},$$

kde  $\mathbf{x}$  je vektor devíti neznámých parametrů chybového modelu,  $m$  je počet poloh,  $G$  je velikost vektoru tíhového zrychlení a  $\|\mathbf{a}_i(\mathbf{x})\|$  je velikost vektoru specifické síly kompenzované parametry z  $\mathbf{x}$ .

V [6] jsou blíže popsány algoritmy využitelné pro minimalizaci funkce  $RMSE(\mathbf{x}, G)$ . Součástí [7] jsou volně dostupné Matlab kódy<sup>1</sup> pro kalibraci akcelerometrů.

<sup>1</sup>[https://github.com/labliscvut/imu\\_calibration](https://github.com/labliscvut/imu_calibration)

### 2.2.5 Kalibrace senzorů úhlové rychlosti

Popis kalibrace senzoru úhlové rychlosti čerpá z [8], [7] a ze shrnutí postupu kalibrace v [9]. Stejně jako dříve popsaná kalibrace akcelerometru, slouží tento postup k odhadu biasů, chyby neortogonality a *scale* faktorů. Chybový model je definován jako

$$\mathbf{y}_g - \mathbf{b}_g = \mathbf{S}_g \mathbf{T}_g \mathbf{M}_g \mathbf{u}_g = \begin{bmatrix} s_{gx} & 0 & 0 \\ 0 & s_{gy} & 0 \\ 0 & 0 & s_{gz} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ a_{xy} & 1 & 0 \\ a_{zx} & a_{zy} & 1 \end{bmatrix} \begin{bmatrix} r_{g,11} & r_{g,12} & r_{g,13} \\ r_{g,21} & r_{g,22} & r_{g,23} \\ r_{g,31} & r_{g,32} & r_{g,33} \end{bmatrix} \begin{bmatrix} u_{gx} \\ u_{gy} \\ u_{gz} \end{bmatrix},$$

kde  $\mathbf{y}_g$  jsou naměřené úhlové rychlosti,  $\mathbf{u}_g$  jsou kompenzované úhlové rychlosti,  $\mathbf{b}_g$  je vektor biasů,  $\mathbf{S}_g$  je matice se *scale* faktory na diagonále,  $\mathbf{T}_g$  je ortogonalizační matice a  $\mathbf{M}_g$  je rotační matice zarovnávací referenční souřadnicový systém na ortogonalizovaný souřadnicový systém senzoru.

Stejně jako pro kalibraci akcelerometru i zde se využívá náklonná plošina. Navíc je potřeba systém pro přesné určení referenčních úhlů. Po uchycení senzoru na náklonnou plošinu a zapnutí napájení se nechá senzor alespoň 30 vteřin v klidu. Z tohoto statického měření se poté vypočítají průměry. Ty odpovídají biasům. Dále se provede rotace postupně kolem každé z os o určitý referenční úhel. Referenční úhel je potřeba určit. To lze provést například přesným a zkalibrovaným senzorem úhlové rychlosti nebo pomocí akcelerometru (pokud je rotace prováděna ve vertikální rovině).

Od změřených úhlových rychlostí se odečte bias. Integrací takto kompenzovaných úhlových rychlostí se získají úhly  $Y_{g,ij}$ . Po třech rotacích lze zkonstruovat matice  $\mathbf{Y}_g$  a  $\mathbf{U}_g$ . Matice

$$\mathbf{Y}_g = \begin{bmatrix} Y_{g,xx} & Y_{g,xy} & Y_{g,xz} \\ Y_{g,yx} & Y_{g,yy} & Y_{g,yz} \\ Y_{g,zx} & Y_{g,zy} & Y_{g,zz} \end{bmatrix},$$

kde  $Y_{g,ij}$  značí úhel získaný integrací kompenzovaného výstupu  $i$ -tého senzoru při rotaci kolem  $j$ -té osy. Matice

$$\mathbf{U}_g = \begin{bmatrix} U_{g,x} & 0 & 0 \\ 0 & U_{g,y} & 0 \\ 0 & 0 & U_{g,z} \end{bmatrix},$$

kde  $U_{g,i}$  je změřený referenční úhel o který se plošina otočila kolem osy  $i$ . Platí, že

$$\mathbf{Y}_g \mathbf{U}_g^{-1} = \mathbf{S}_g \mathbf{T}_g \mathbf{M}_g.$$

Na levé straně rovnice jsou známé matice. Na pravé straně jsou neznámé parametry chybového modelu. Tyto neznámé se získají z rovnic

$$\begin{aligned} \mathbf{S}_g \mathbf{T}_g &= \text{chol} \left[ (\mathbf{Y}_g \mathbf{U}_g^{-1}) (\mathbf{Y}_g \mathbf{U}_g^{-1})^T \right]^T, \\ [\mathbf{S}_g, \mathbf{T}_g] &= \text{LU}(\mathbf{S}_g \mathbf{T}_g), \\ \mathbf{M}_g &= \mathbf{T}_g^{-1} \mathbf{S}_g^{-1} \mathbf{Y}_g \mathbf{U}_g^{-1}, \end{aligned}$$

kde  $\text{chol}()$  značí Choleského dekompozici a  $\text{LU}()$  značí LU dekompozici.

Podrobnější popis je v [8]. Součástí [7] jsou volně dostupné Matlab kódy<sup>2</sup> a ukázková data pro kalibraci senzorů úhlových rychlostí.

## 2.3 Senzory vzdálenosti

Kromě inerciálních senzorů je v realizovaném systému použit i senzor vzdálenosti. Nejběžnější typy senzorů pro určení vzdálenosti jsou:

- ultrazvukové senzory,
- laserové senzory,
- infračervené senzory.

### 2.3.1 Ultrazvukové senzory

Ultrazvukové senzory generují zvukovou vlnu o vysoké frekvenci a měří čas od vyslání vlny do příjmu vlny odražené. Ze změřeného času a ze známé rychlosti šíření vlny se spočítá vzdálenost. Tento princip měření vzdálenosti se nazývá *Time-of-Flight* (ToF). Další možný způsob jakým ultrazvukové senzory mohou vypočítat vzdálenost je z útlumu amplitudy.

Nevýhodou ultrazvukových senzorů je, že přesnost jejich měření závisí na teplotě a vlhkosti okolního prostředí, protože teplota a vlhkost ovlivňují rychlost šíření zvukové vlny. Jinou nevýhodou je, že schopnost určení vzdálenosti závisí na tvaru objektu. Pakliže odrazová plocha je vůči senzoru orientována o větší úhel, než je úhel kritický, pak senzor nemusí odraženou vlnu vůbec zachytit.

Další vlastností ultrazvukových senzorů je, že oblast pokrytá jedním senzorem má tvar kužele, ne pouze přímky jako u laserových nebo infračervených senzorů. To může být výhoda například v antikolizních systémech, kde nezáleží na přesné poloze objektu, ale pouze na zjištění, zda v dané oblasti nějaký objekt je. Více informací o ultrazvukových senzorech lze najít v [10] a [11].

### 2.3.2 Laserové senzory

Laserové senzory počítají vzdálenost pomocí ToF metody nebo pomocí triangulace. ToF metoda je vysvětlena v předchozí části, triangulace je popsána v článku [12] - následuje její stručné shrnutí. Senzor emituje laserový paprsek, ten se odrazí od překážky a dopadne na detektor v senzoru. Podle přesného místa na detektoru, kam paprsek dopadne, lze určit úhel odrazu a z něj i vzdálenost překážky od senzoru. Laserový paprsek se šíří v přímce, díky tomu lze, na rozdíl od ultrazvukového senzoru, určit polohu překážky vůči senzoru. Výhodou laserového senzoru je vyšší frekvence a větší dosah.

### 2.3.3 Infračervené senzory

Princip fungování infračervených senzorů je podobný jako u těch laserových. Nejběžnější metodou určení vzdálenosti je triangulace nebo ToF. Výhodou oproti laserovým sen-

<sup>2</sup>[https://github.com/labliscvut/imu\\_calibration](https://github.com/labliscvut/imu_calibration)

zorům je nízká hmotnost a to, že nepoškozuje zrak. Nevýhodou je, že citlivost infračervených senzorů je ovlivněna okolním světlem a barvou odrazné plochy.

## 2.4 Inerciální navigační systém (INS)

Základním poznatkem, na kterém je postaven princip inerciálních navigací je, že dvojitou integrací zrychlení se získá pozice a integrací úhlových rychlostí se získají polohové úhly. INS disponuje inerciální měřicí jednotkou (IMU), která obsahuje akcelerometry a senzory úhlové rychlosti, zpravidla trojosé. Z měření IMU lze tedy získat zrychlení a úhlové rychlosti, které je možné integrovat. V ideálním případě, kdy by tyto senzory měřily bez jakékoli chyby a nepůsobily by na ně žádné okolní vlivy mimo síly uvádějící těleso do pohybu, by odhady pozice a úhlů získané integrováním odpovídaly skutečnosti. Měření senzorů jsou ale vždy zatížena chybou a okolní vlivy nelze odstranit. Tyto chyby a důsledky okolních vlivů se rovněž integrují do výsledných odhadů pozice a úhlů, což nevyhnutelně způsobuje postupnou divergenci těchto odhadů od skutečnosti. Některé statické chyby lze odhadnout a následně kompenzovat. Například offset senzoru úhlové rychlosti lze odhadnout jako průměr výstupních hodnot po dobu, kdy byl senzor v klidu. Některé okolní vlivy, jako je rotace Země a odstředivé zrychlení, lze popsat a tím pádem rovněž kompenzovat. Nicméně ne všechny chyby a vlivy lze určit a popsat rovnicemi, a proto je divergence odhadů od skutečnosti u INS, bez spolupráce s dalšími systémy, nevyhnutelná.

Aby se zamezilo divergenci, zpřesňují se odhady pomocí dalších senzorů a systémů, ze kterých lze získat nezávislé informace o pozici, rychlostech, polohových úhlech atd. (dále navigační proměnné). Fúze INS s těmito nezávislými systémy a senzory se často provádí pomocí rozšířeného Kalmanova filtru (EKF), kterému je věnována část 2.5.

INS se dělí podle uchycení základny na:

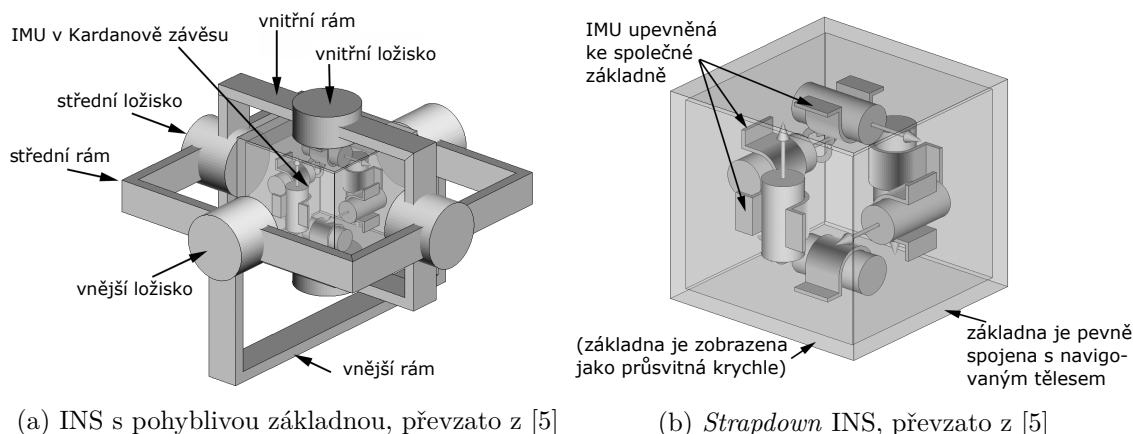
1. systémy s pohyblivou základnou,
2. *strapdown* systémy.

### INS s pohyblivou základnou

INS s pohyblivou základnou je znázorněn na obrázku 2.6a. Tento druh INS je udržován v neměnné orientaci vůči navigační soustavě. Díky této vlastnosti se eliminují některé chyby senzorů závislé na úhlové rychlosti a je umožněno použití přesnějších senzorů [5]. Další výhodou je menší výpočetní náročnost oproti *strapdown* systémům, protože odpadá nutnost transformace změřených hodnot ze souřadnicového systému navigovaného tělesa do navigační soustavy. Nevýhodou je potřeba velmi kvalitní mechanické části zařízení. S tím souvisí také zpravidla vyšší cena a větší rozměry v porovnání se *strapdown* systémy.

### *Strapdown* INS

*Strapdown* INS je zobrazen na obrázku 2.6b. Tento druh INS je pevně spojen s navigovaným tělesem. INS bývá uchycen tak, aby osy citlivosti senzorů odpovídaly osám souřadnicového systému navigovaného tělesa. Více o *strapdown* INS, o mechanizačních rovnicích v různých souřadnicových systémech se lze dočíst například v [13].

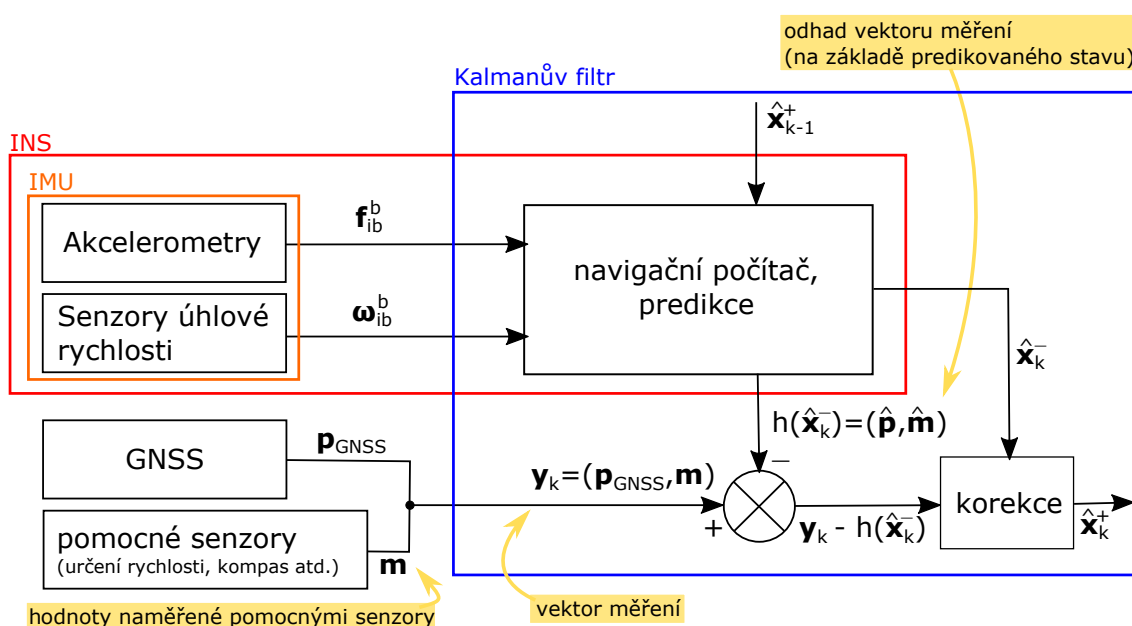


Obrázek 2.6: Druhy INS

## 2.5 Rozšířený Kalmanův filtr (EKF)

Jak bylo napsáno v části 2.4 - odhady samotného INS časem divergují od skutečnosti. Aby se divergenci zabránilo, je třeba fúzovat INS s nezávislými senzory a systémy (fúze INS s dalšími senzory a systémy pomocí Kalmanova filtru je znázorněna na obrázku 2.7). V navigacích pro venkovní užití lze využít některý z globálních družicových navigačních systémů (GNSS) jako je GPS, GLONASS nebo Galileo. Tyto systémy poskytují informace o poloze zpravidla s nižší frekvencí, než je obnovovací frekvence INS, a v běžném módu mají přesnost v řádech metrů, ale mají tu výhodu, že chyba určení pozice neroste s časem měření. Další často používané senzory pro korekci INS jsou senzory barometrického tlaku poskytující nezávislou informaci o výšce a magnetometry, které lze ve venkovních aplikacích využít k určování severu.

Rozšířený Kalmanův filtr (EKF) je v navigaci často používaným nástrojem pro fúzi



Obrázek 2.7: Fúze INS a dalších systémů, převzato z [3]

dat. Díky fúzi dat z nezávislých měření slouží k přesnému odhadu stavových proměnných a k potlačování šumu, který působí na senzory. EKF je rozšířením standardního Kalmanova filtru na nelineární systémy. Následuje stručný popis EKF. Podrobněji je problematika popsána v [1, 5, 14].

Nelineární diskretní dynamický systém je popsán rovnicí

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \quad (2.1)$$

kde  $\mathbf{x}_k$  je vektor stavů systému,  $\mathbf{f}_{k-1}$  je přechodová funkce,  $\mathbf{u}_k$  je vektor řízení a  $\mathbf{w}_k$  je bílý Gaussovský šum s nulovou střední hodnotou a kovarianční maticí  $\mathbf{Q}$ . To znamená, že autokorelační funkce

$$\mathbf{R}_n = \mathbb{E}[\mathbf{w}_k \mathbf{w}_{k+n}]$$

je nenulová pouze pro posun  $n = 0$  v jakémkoli časovém okamžiku  $k$ . Dále platí, že

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}),$$

kde

$$\mathbf{Q} = \mathbb{E}[\mathbf{w}_k \mathbf{w}_k^T].$$

Vztah mezi vektorem měření  $\mathbf{y}_k$  a stavovým vektorem  $\mathbf{x}_k$  je popsán rovnicí

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad (2.2)$$

kde  $\mathbf{h}_k$  je nelineární funkce a  $\mathbf{v}_k$  je bílý Gaussovský šum s nulovou střední hodnotou a kovarianční maticí  $\mathbf{R} = \mathbb{E}[\mathbf{v}_k \mathbf{v}_k^T]$ .

EKF algoritmus pracuje ve dvou krocích:

1. časová aktualizace (predikce),
2. datová aktualizace (korekce).

EKF si při výpočtech uchovává odhad stavového vektoru  $\hat{\mathbf{x}}_k$  a kovarianční matici odhadů  $\mathbf{P}_k$ , která je definovaná jako

$$\mathbf{P}_k = \mathbb{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T],$$

kde jsou stříškou označeny odhady stavů a bez stříšky jsou jejich skutečné hodnoty. Před začátkem výpočtu EKF potřebuje hodnoty kovariančních matic šumů  $\mathbf{R}$ ,  $\mathbf{Q}$ , iniciální hodnotu odhadu  $\hat{\mathbf{x}}_0$  a iniciální hodnotu kovarianční matice odhadů  $\mathbf{P}_0$ . Dále v textu jsou horním indexem  $+$  označeny proměnné po korekci a horním indexem  $-$  proměnné po predikci.

### Časová aktualizace

Během predikce se linearizuje funkce  $\mathbf{f}_{k-1}$  v bodě posledního odhadu:

$$\mathbf{F}_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}(\mathbf{x}, \mathbf{u}_k)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}^+} \quad (2.3)$$

Následuje diskretizace kovarianční matice  $\mathbf{Q}$ :

$$\mathbf{Q}_{k-1} = \frac{1}{2} dt \left( \mathbf{F}_{k-1} \mathbf{G}_{k-1} \mathbf{Q} \mathbf{G}_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q} \mathbf{G}_{k-1}^T \mathbf{F}_{k-1}^T \right) \quad (2.4)$$

Dále se predikuje stavový vektor podle :

$$\hat{\mathbf{x}}_k^- = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_k) \quad (2.5)$$

Posledním krokem je aktualizace kovarianční matice odhadů  $\mathbf{P}$ :

$$\mathbf{P}_k^- = \mathbf{F}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (2.6)$$

Pakliže by po časové aktualizaci nenásledovala aktualizace datová, nejistota odhadu by neomezeně rostla, což je vidět z poslední rovnice. V části věnované INS bylo zmíněno, že bez měření z nezávislých systémů a senzorů odhad postupně diverguje od skutečnosti. Samotný INS provádí vlastně pouze časovou aktualizaci, přesněji řečeno pouze její část - rovnicí 2.5, neboť ostatní rovnice nemají bez datové aktualizace význam. Datovou aktualizací se nejistota odhadu sníží.

### Datová aktualizace

Při datové aktualizaci (korekci) se linearizuje funkce  $\mathbf{h}_k$  v bodě posledního odhadu:

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \quad (2.7)$$

Dále se spočítá Kalmanovo zesílení:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1} \quad (2.8)$$

Následuje korekce odhadu:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)) , \quad (2.9)$$

Z rovnice je vidět, že korekce odhadu se provede přičtením váhovaného rozdílu vektoru měření  $\mathbf{y}_k$  a predikce vektoru měření  $\mathbf{h}_k(\hat{\mathbf{x}}_k^-)$ , která je učiněna na základě posledního odhadu stavů. Váha je vyjádřena Kalmanovým zesílením  $\mathbf{K}_k$ . Kalmanovo zesílení určuje, jak moc má být výsledný odhad ovlivněn měřením. Jeho hodnota je závislá na poměru kovariančních matic  $\mathbf{P}_k$  a  $\mathbf{R}$ . Čím vyšší je nejistota odhadu v  $\mathbf{P}_k$ , tím větších hodnot nabývá Kalmanovo zesílení a tím větší vliv má měření na výsledný odhad stavových proměnných.

Posledním zbývajícím krokem je aktualizace kovarianční matice odhadu. Základní variantou této aktualizace je

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (2.10)$$

Je-li použita tato verze aktualizace  $\mathbf{P}_k$ , není zajištěna její symetrie. Tu lze následovně zajistit takto:

$$\mathbf{P}_k^+ = \frac{1}{2} (\mathbf{P}_k^+ + \mathbf{P}_k^{+T}) \quad (2.11)$$

Také je možné použít pro aktualizaci  $\mathbf{P}_k$  tzv. Josephův tvar, který zajišťuje symetrii a pozitivní definitnost matice  $\mathbf{P}_k$  [15]:

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (2.12)$$

## 2.6 Indoor navigace

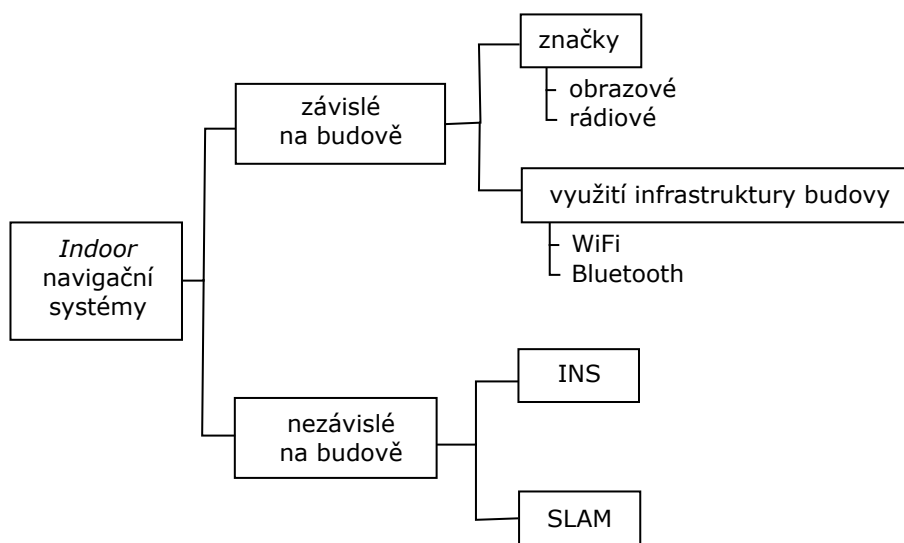
Základní klasifikace navigačních systémů pro aplikace uvnitř budov je na obrázku 2.8. *Indoor* navigace mají oproti vnějším tu nevýhodu, že nelze spoléhat na globální navigační satelitní systém (GNSS). Signál z GNSS je při průchodu střechou a zdmi citelně utlumen. Přijímače se sice neustále zdokonalují a citlivější z nich jsou schopné přijímat i tento utlumený signál, ale také musí čelit dalšímu problému, jímž je přítomnost množství odražených vln od okolních budov a zdí. To může způsobit, že pozice získaná z GNSS je velmi nepřesná a špatně využitelná v *indoor* navigaci, kde je žádoucí poměrně přesné určení polohy.

Možností, jak získat absolutní pozici uvnitř budovy jinak než z GNSS, je více. Jejich popis lze najít v [16]. Jedna z možností je založena na rádiové technologii, kde se využije informace získaná z analýzy signálu nějakého zdroje rádiových vln (WiFi, Bluetooth, RFID - *Radio Frequency Identification...*). Informace o poloze lze získat například metodami ToF (viz 2.3), analýzou rozdílu časů příjmu z více zdrojů rádiových vln (TDoA - *Time Difference of Arrival*) nebo analýzou síly přijatého signálu (RSS - *Received Signal Strength*). Další možností je umístění značek, které navigovaný objekt zachytí kamerou a rozpozná.

Předchozí uvedené možnosti jsou závislé na budově - bez patřičně vybavené budovy není možné provádět navigaci. Nicméně jejich výhodou je, že mohou získat informace o pozici, což systémy na budově nezávislé nemohou. Mezi navigační systém nezávislý na budově lze zařadit dříve prezentovaný INS, jehož odhady ovšem postupně divergují od skutečnosti. Další možností je použít SLAM (*Simultaneous Localization and Mapping*).

### 2.6.1 Simultaneous Localization and Mapping (SLAM)

SLAM (*Simultaneous Localization and Mapping*) je metoda na současné mapování okolního prostředí a určování polohy v tomto prostředí. SLAM se skládá ze tří základních operací [17]:



Obrázek 2.8: Klasifikace *indoor* navigačních systémů



1. predikce nové polohy navigovaného tělesa,
2. detekce významných objektů v okolí pomocí senzorů,
3. korekce polohy navigovaného tělesa a aktualizace mapy významných objektů na základě porovnání mapy s aktuálně detekovanými významnými objekty.

Predikce pozice se obvykle provede pomocí INS nebo odometrie. Způsob detekce významných objektů závisí na senzorech použitých ke snímání okolí. Ke snímání okolí mohou být využity kamery, jednoduché ultrazvukové [18], laserové a infračervené senzory snímající vzdálenost v jednom směru [19], případně laserové snímače pokrývající celý okolní úsek (třeba i celých 360 stupňů) [20]. Ke korekci lze použít například EKF (popsán v části 2.5). Podrobný popis SLAM využívající EKF je v [17].

Dosud byl teoretický rozbor zaměřen na navigační problematiku. V posledních dvou částech je představena metoda prokládání bodů v rovině přímkou za pomoci SVD rozkladu matice, a nakonec je stručně popsána sběrnice CAN. Tyto dvě části nesouvisí přímo s navigací ani se sebou navzájem, ale mají svou roli v níže navrženém navigačním systému, a proto jsou v této kapitole uvedeny.

## 2.7 Proložení bodů přímkou pomocí SVD

V této části je popsán postup pro proložení přímky body v 2D prostoru tak, aby součet čtverců Euklidovských vzdáleností bodů od přímky byl minimální. Dále je zde popsán rozklad matice podle singulárních čísel (SVD), který je součástí tohoto postupu. Následující text čerpá z [21].

### Rozklad matice podle singulárních čísel (SVD)

Jakákoli matice  $\mathbf{A} \in \mathbb{R}^{m \times n}$  lze rozložit jako

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

kde matice  $\mathbf{U} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{V} \in \mathbb{R}^{n \times n}$  a  $\mathbf{S} \in \mathbb{R}^{m \times n}$  je diagonální s diagonálou, na které je seřazeno sestupně  $r$  singulárních čísel doplněných nulami, kde  $r = \text{rank } \mathbf{S} = \text{rank } \mathbf{A}$ .

### Eckart-Youngova věta

Doslovná citace z [21] (kurzívou): *Nechť  $\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T$  je SVD rozklad matice  $\mathbf{A}$ . Nechť  $\mathbf{S}'$  je diagonální matice, která vznikne vynulováním  $r - r'$  nejmenších nenulových diagonálních prvků matice  $\mathbf{S}$ . ... Pak  $\mathbf{A}' = \mathbf{U} \mathbf{S}' \mathbf{V}^T$  je řešením úlohy*

$$\min (\|\mathbf{A} - \mathbf{A}'\|_F \mid \mathbf{A}' \in \mathbb{R}^{m \times n}, \text{rank } \mathbf{A}' = r').$$

Výraz  $\|\mathbf{A} - \mathbf{A}'\|_F$  značí Frobeniovu normu matice  $\mathbf{A} - \mathbf{A}'$ . Platí, že

$$\|\mathbf{A} - \mathbf{A}'\|_F = \sqrt{\sum_{j=1}^n \|\mathbf{a}_j - \mathbf{a}'_j\|^2}, \quad (2.13)$$

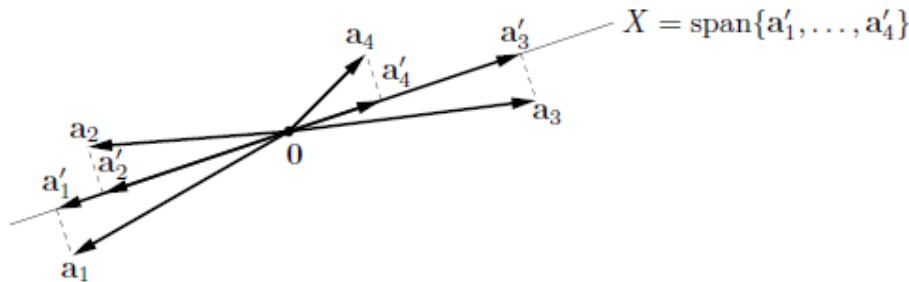
kde  $\mathbf{a}_j$  a  $\mathbf{a}'_j$  jsou  $j$ -té sloupce matic  $\mathbf{A}$  a  $\mathbf{A}'$ . Výraz  $\|\mathbf{a}_j - \mathbf{a}'_j\|$  značí Euklidovskou normu (velikost vektoru).

Jinými slovy Eckart-Youngova věta umožňuje k lineárnímu prostoru  $\text{rng } \mathbf{A}$  o dimenzi  $r$  nalézt lineární podprostor  $\text{rng } \mathbf{A}'$  o dimenzi  $r'$ , takový, že  $\|\mathbf{A} - \mathbf{A}'\|_F$  je minimální.

### Proložení bodů ve 2D přímkou

Eckart-Youngovu větu lze využít pro proložení bodů ve 2D prostoru přímkou tak, aby součet čtverců vzdáleností bodů od přímky byl minimální. Vytvořme matici  $\mathbf{A}$  tak, že jejích  $n$  sloupců tvoří prokládané body z 2D prostoru. Pak  $\mathbf{A} \in \mathbb{R}^{2 \times n}$ ,  $r = 2$  a  $\text{rng } \mathbf{A} = \mathbb{R}^2$ . Hledáme přímkou, tedy lineární podprostor  $X = \text{rng } \mathbf{A}'$  o dimenzi  $r' = 1$  tak, že  $\sqrt{\sum_{j=1}^n \|\mathbf{a}_j - \mathbf{a}'_j\|^2}$  je minimální (tedy Frobeniova norma matice  $\mathbf{A} - \mathbf{A}'$  je minimální). Provedeme SVD rozklad matice  $\mathbf{A}$ . Nyní známe matice  $\mathbf{U}$ ,  $\mathbf{S}$  a  $\mathbf{V}$ . Z matice  $\mathbf{S}$  odstraníme nejmenší singulární číslo a tak získáme matici  $\mathbf{S}'$ . Teď lze dopočítat  $\mathbf{A}' = \mathbf{U}\mathbf{S}'\mathbf{V}^T$ .

Člen  $\mathbf{a}_j$  je  $j$ -tý prokládaný bod z 2D prostoru a  $\mathbf{a}'_j$  je bod z hledaného podprostoru (přímky). Jelikož výraz  $\sqrt{\sum_{j=1}^n \|\mathbf{a}_j - \mathbf{a}'_j\|^2}$  má být minimální, pak všechny Euklidovské vzdálenosti bodu  $\mathbf{a}_j$  od  $\mathbf{a}'_j$  musí být minimální. Z toho plyne, že  $\mathbf{a}'_j$  je kolmou projekcí bodu  $\mathbf{a}_j$  do hledané přímky (podprostoru  $X$ ). Lineární obal sloupců matice  $\mathbf{A}'$  je hledaný lineární podprostor  $X$  (přímka). Pro ilustraci viz obrázek 2.9.



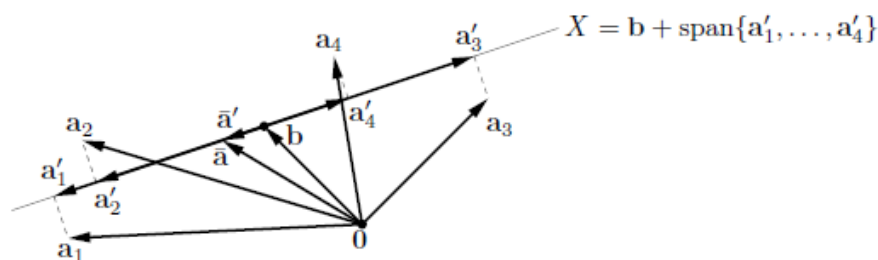
Obrázek 2.9: Body ve 2D a jejich kolmé projekce do přímky procházející počátkem, převzato z [21]

Postupem uvedeným výše se nalezne lineární podprostor, tedy přímka procházející počátkem. Pro nalezení libovolné přímky (afinního podprostoru dimenze 1) je třeba učinit následující:

1. spočítat těžiště bodů  $\bar{\mathbf{a}}$ ,
2. posunout všechny body  $\mathbf{a}_j$  tak, aby těžiště bylo v počátku,
3. aplikovat výše popsany postup pro lineární podprostor.

Afinní podprostor  $X$  (přímka) se pak rovná součtu lineárního obalu sloupců matice  $\mathbf{A}'$  a vektoru  $\mathbf{b} = \bar{\mathbf{a}} - \bar{\mathbf{a}}'$ , kde  $\bar{\mathbf{a}}'$  je těžiště kolmých projekcí  $\mathbf{a}'_j$ . Pro ilustraci viz obrázek 2.10.

Dále je v [21] uvedeno a dokázáno, že báze lineárního obalu sloupců matice  $\mathbf{A}'$  je množina prvních  $r'$  sloupců matice  $\mathbf{U}$  z rozkladu. Jelikož v případě proložení bodů v 2D prostoru přímkou je  $r' = 1$ , tak báze lineárního obalu sloupců matice  $\mathbf{A}'$  je první sloupec matice  $\mathbf{U}$  (označen jako  $\mathbf{u}$ ) a tedy  $X = \mathbf{b} + \text{span}\{\mathbf{u}\}$ .



Obrázek 2.10: Body ve 2D a jejich kolmé projekce do přímky, převzato z [21]

Parametry přímky  $a$  a  $c$  z obecné rovnice přímky ve tvaru  $y = ax + c$  lze dopočítat ze vztahů

$$a = \frac{u_y}{u_x}$$

$$c = \frac{u_y \bar{a}_x - u_x \bar{a}_y}{-u_x}$$

## 2.8 Sběrnice CAN

CAN (*Controller Area Network*) je sběrnice původně vytvořená pro komunikaci senzorů v automobilech. Pro své vlastnosti se ale rozšířil i do jiných odvětví. Protokol CAN definuje pouze linkovou vrstvu ISO/OSI referenčního modelu. Nicméně CAN je součástí standardu ISO 11898, kde je definována i fyzická vrstva. Existuje celá řada používaných standardů pro aplikační vrstvu jako je například CANopen nebo CANaerospace, který byl vytvořen pro potřeby leteckých aplikací.

V síti CAN neexistuje adresace zpráv. Zprávy mají přiřazený identifikátor, který jednoznačně určuje jejich obsah. Uzly pak přijímají ty zprávy, jejichž obsah je zajímavý. Existují dvě varianty standardu CAN lišící se délkou identifikátoru. Varianta A využívá 11-bitové identifikátory pro všechny zprávy. Varianta B může v síti vysílat jak rámce s 11-bitovými identifikátory, tak rámce s rozšířenými 29-bitovými identifikátory současně.

### Fyzická vrstva

Standard CAN definuje dva stavy sběrnice: recesivní a dominantní. Jedním z požadavků na fyzickou vrstvu CAN je, aby sběrnice byla v recesivním stavu, pokud všechny připojené uzly vysílají recesivní bit, a aby byla v dominantním stavu, pokud alespoň jeden uzel vysílá bit dominantní, což je zásadní pro zajištění prioritizace zpráv. Nejčastěji má CAN fyzickou vrstvu podle normy ISO 11898 ve formě diferenciální sběrnice, která je tvořena dvěma vodiči. Logickou úroveň definuje rozdílové napětí mezi těmito vodiči. Sběrnice je zakončena odpory 120  $\Omega$  kvůli potlačení odrazů. Na krátké vzdálenosti je možné dosáhnout přenosové rychlosti až 1 Mbit/s.

### Spojová vrstva

CAN zabezpečuje na spojové vrstvě:

- řízení přístupu k médiu - metoda CSMA/CR (*Carrier Sense Multiple Access with Collision Resolution*),
- prioritní vysílání - pomocí porovnávání vysílaného bitu se stavem sběrnice,
- zabezpečení přenášených dat - 15ti bitový CRC kód, kontrola rámce, vkládání bitů,
- potvrzování správně přijatých rámců,
- hlášení o přetížení uzlu,
- řešení případné vyšší chybovosti komunikace.

Podrobný popis sběrnice CAN lze nalézt v [22, 23].

## 3 Hardware

---

### 3.1 MiniINS v1

MiniINS (viz obrázek 3.1) je inerciální navigační jednotka vytvořená v Laboratoři leteckých informačních systémů Katedry měření na Fakultě elektrotechnické ČVUT v Praze, která byla použita v realizovaném navigačním systému. Její součástí je IMU, magnetometr, senzor barometrického tlaku a mikrokontrolér. V realizovaném navigačním systému je využita IMU jako zdroj inerciálních dat a mikrokontrolér jako hlavní výpočetní část celého systému. Původně měla být v realizovaném systému použita druhá verze MiniINS, ale nakonec nebyla k dispozici, a proto byla použita verze první. Hlavní výhodou druhé verze by byla využitá IMU - viz popis níže.



Obrázek 3.1: MiniINS v1 použitá v realizovaném navigačním systému

#### 3.1.1 Inerciální měřicí jednotka (IMU)

První verze MiniINS obsahuje IMU MPU-9150 od firmy InvenSense. Součástí MPU-9150 je trojosý MEMS kapacitní akcelerometr (princip popsán v části 2.2.1), trojosý MEMS vibrační senzor úhlové rychlosti (princip popsán v části 2.2.2) a magnetometr. Základní parametry senzoru úhlové rychlosti jsou uvedeny v tabulce 3.1 a parametry akcelerometru v tabulce 3.2. Hodnoty naměřené inerciálními senzory jsou vzorkovány 16 bitovým A/D převodníkem a posílány přes I2C sběrnici.

Druhá verze MiniINS, která měla být původně použita, bude disponovat IMU DMU-11 od firmy Silicon Sensing. Tato IMU rovněž obsahuje MEMS vibrační senzor úhlové rychlosti a MEMS kapacitní akcelerometr. V porovnání s MPU-9150 nabízí lepší parametry (viz [25]).

Parametr	Hodnota	Jednotka
rozsah	$\pm 250, \pm 500, \pm 1000, \pm 2000$	deg/s
šum (RMS, 92 Hz)	0,06	deg/s
nelinearita	typ. 0,2	%
mezosová citlivost	typ. $\pm 2$	%
výstupní frekvence	4 až 8000	Hz
<i>zero rate</i> výstup (25°C)	$\pm 20$	deg/s

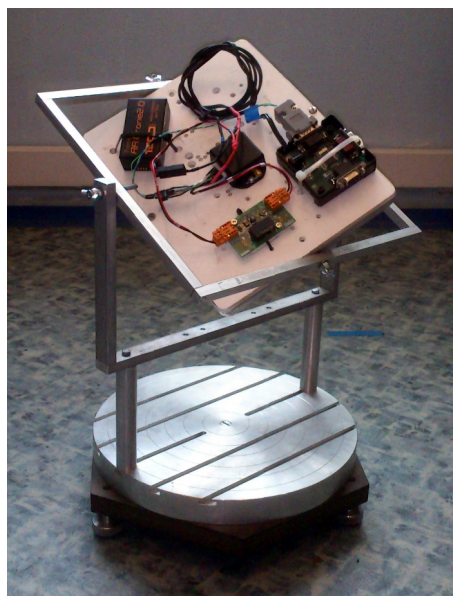
Tabulka 3.1: Základní parametry senzoru úhlové rychlosti v MPU-9150 [24]

Parametr	Hodnota	Jednotka
rozsah akcelerometru	$\pm 2, \pm 4, \pm 8, \pm 16$	g
šum (RMS, 100 Hz)	4	mg
nelinearita	typ. 0,5	%
výstupní frekvence	4 až 1000	Hz
<i>zero g</i> výstup	$\pm 80$ pro osy $x, y$ a $\pm 150$ pro osu $z$	mg

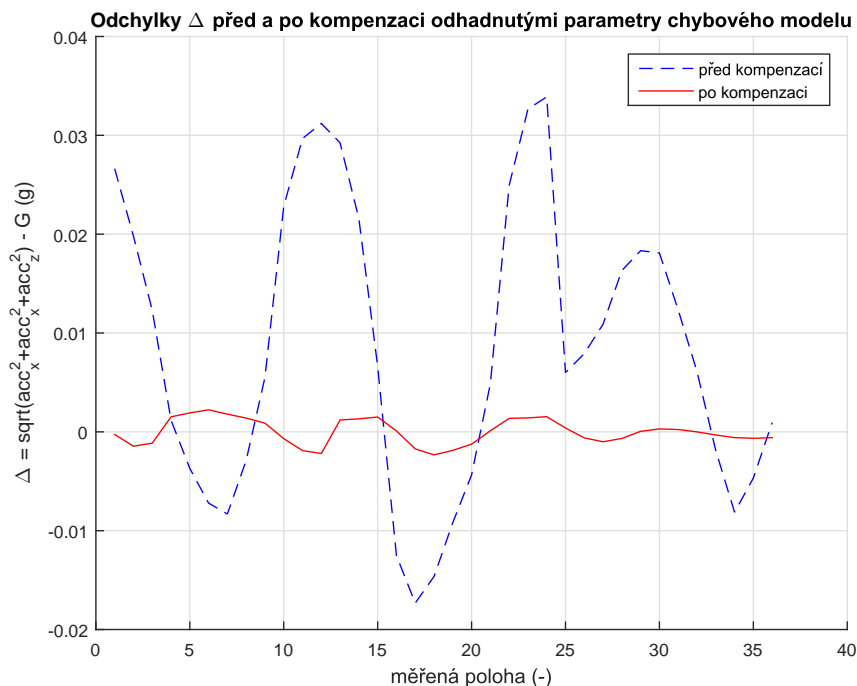
Tabulka 3.2: Základní parametry akcelerometru v MPU-9150 [24]

### Kalibrace akcelerometru

Kalibrační procedura pro akcelerometry byla popsána v části 2.2.4. Na obrázku 3.2 je vyfocena MiniINS na kalibrační otočné plošině spolu se záznamníkem dat. Na obrázku 3.3 jsou vidět odchylky velikosti změřené (modře)/kompenzované (červeně) specifické síly od velikosti vektoru tíhového zrychlení. Odhadnuté parametry chybového modelu jsou v tabulce 3.3.



Obrázek 3.2: MiniINS na kalibrační náklonné plošině



Obrázek 3.3: Odchyly velikosti změřené (modře)/kompenzované (červeně) specifické síly od velikosti vektoru tíhového zrychlení

Parametr	Hodnota	Parametr	Hodnota	Parametr	Hodnota
$a_{xy}$	0.0001	$a_{zx}$	0.0000	$a_{zy}$	-0.0064
$s_{ax}$	0.9963	$s_{ay}$	0.9909	$s_{az}$	0.9902
$b_{ax}$	-0.0111 g	$b_{ay}$	-0.0056 g	$b_{az}$	-0.0209 g

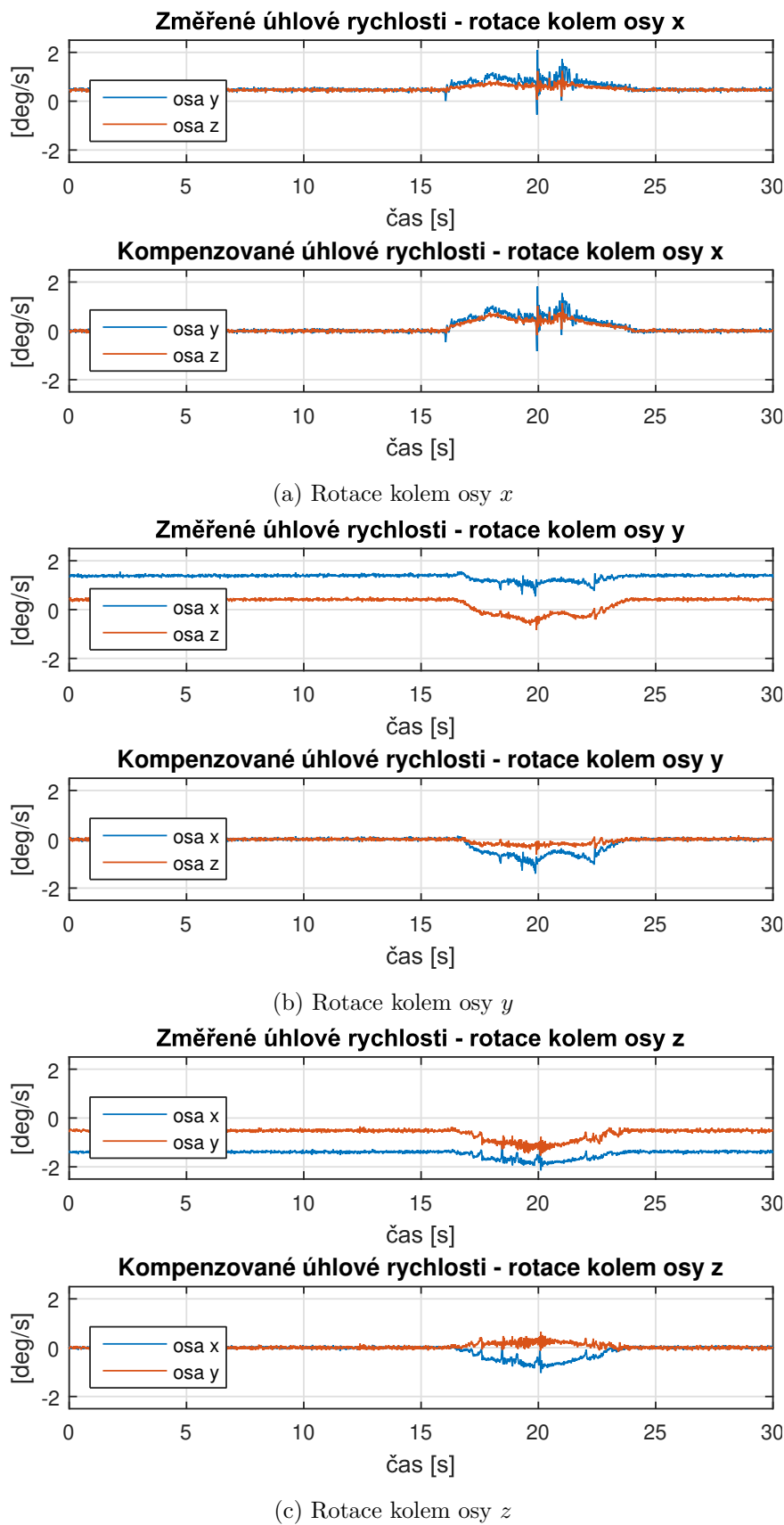
Tabulka 3.3: Odhadnuté parametry chybového modelu akcelerometru

### Kalibrace senzoru úhlové rychlosti

Postup kalibrace senzoru úhlové rychlosti byl popsán v části 2.2.5. Na obrázku 3.4 jsou znázorněny úhlové rychlosti měřené ve dvou osách citlivosti při rotaci kolem třetí zafixované osy před a po kompenzaci. V tabulce 3.4 jsou uvedeny odhadnuté parametry chybového modelu.

Parametr	Hodnota	Parametr	Hodnota	Parametr	Hodnota
$s_{gx}$	1.0053	$s_{gy}$	1.0005	$s_{gz}$	0.9987
$a_{xy}$	0.0027	$a_{zx}$	-0.0016	$a_{zy}$	-0.0190
$r_{g,11}$	1.0000	$r_{g,12}$	0.0070	$r_{g,13}$	0.0034
$r_{g,21}$	-0.0070	$r_{g,22}$	0.9999	$r_{g,23}$	-0.0111
$r_{g,31}$	-0.0035	$r_{g,32}$	0.0111	$r_{g,33}$	0.9999
$b_{gx}$	1.4226 °/s	$b_{gy}$	0.4740 °/s	$b_{gz}$	0.4581 °/s

Tabulka 3.4: Odhadnuté parametry chybového modelu senzoru úhlové rychlosti



Obrázek 3.4: Úhlové rychlosti naměřené při rotacích kolem jednotlivých os



### 3.1.2 Mikrokontrolér

V MiniINS v1 je jako výpočetní jádro použit mikrokontrolér STM32F405. Tento mikrokontrolér obsahuje procesor ARM Cortex-M4. Cortex-M4 je 32-bitový RISC procesor se třístupňovou *pipeline* a predikcí skoků. Podporuje 8/16-bit SIMD (*single instruction, multiple data*) aritmetiku a jednocyklové MAC (*multiply and accumulate*) instrukce. Verze v STM32F405 navíc disponuje FPU (*floating point unit*) pro výpočty v plovoucí řádové čárce v jednoduché přesnosti podle IEEE 754 [26].

Mikrokontrolér STM32F405 nabízí velké množství periférií. Disponuje komunikačními rozhraními USART (6x), SPI (3x), I2C (3x), CAN (2x), USB OTG (2x). Navíc nabízí 2 12-bitové D/A převodníky, 3 12-bitové A/D převodníky a až 17 16/32-bitových časovačů [27].

### 3.1.3 Senzor barometrického tlaku

Součástí MiniINS v1 je i senzor barometrického tlaku MS5611 od firmy TE Connectivity. Základní parametry senzoru jsou v tabulce 3.5.

Parametr	Hodnota	Jednotka
rozsah senzoru	10 až 1200	mbar
rozlišení ve výšce	10	cm
přesnost (25°C, 750 mbar)	min: -1,5; max: 1,5	mbar
dlouhodobá stabilita	typ. $\pm 1$	mbar/rok

Tabulka 3.5: Základní parametry senzoru barometrického tlaku MS5611, podle [28]

## 3.2 TeraRanger Tower

TeraRanger Tower od firmy Terabee je zařízení pro měření vzdáleností, které bylo použito v realizovaném navigačním systému. Skládá se až z osmi infračervených senzorů vzdálenosti TeraRanger One a desky TeraRanger Hub, která sbírá měření od všech senzorů a poskytuje je v ucelených zprávách. TeraRanger One funguje na principu ToF (popsáno v části 2.3). Blíže je TeraRanger One představen v [11]. Na obrázku 3.5 je ukázán Tera-



Obrázek 3.5: Terabee TeraRanger Tower použitá v realizovaném navigačním systému, převzato z [29]

Ranger Tower se čtyřmi senzory ve třech různých konfiguracích. V realizovaném systému je využita konfigurace zobrazená na obrázku 3.5 uprostřed. Parametry zařízení Teraranger Tower jsou v tabulce 3.6.

Parametr	Hodnota
napájení	12 V
rozsah	až 14 m (závislé na okolním světle)
váha	130 g (při plném osazení 8 senzory)
rozlišení	0,5 cm
přesnost	$\pm 4$ cm (v přesném módu)
obnovovací frekvence	až 270 Hz (při zapnutém <i>crosstalk avoidance</i> )
zorné pole jednoho senzoru	3 stupně

Tabulka 3.6: Základní parametry zařízení TeraRanger Tower, podle [30]

Zařízení může pracovat ve dvou módech:

1. přesný mód,
2. rychlý mód.

V rychlém módu je při zapnutém *crosstalk avoidance* garantovaná frekvence 270 Hz, ale maximální dosah je redukován na 4,5 metru. V přesném módu je garantovaná přesnost  $\pm 4$  cm a frekvence závisí na vzdálenosti překážek (viz tabulku 3.7).

Vzdálenost [cm]	Frekvence [Hz]
50	514,9
100	444,5
300	194,2
500	66,3
800	18,2
1000	9,3
1300	4,6

Tabulka 3.7: TeraRanger Tower - závislost frekvence na vzdálenosti (přesný mód při vypnutém *crosstalk avoidance*), podle [11]

Naměřená data jsou vysílána přes USB a přes UART. UART rozhraní používá TTL logické úrovně a je konfigurované na rychlost 921600 Bd, 8 datových bitů, jeden stop bit, žádný paritní bit. Zprávy mohou být posílány buď v binární nebo ASCII podobě.

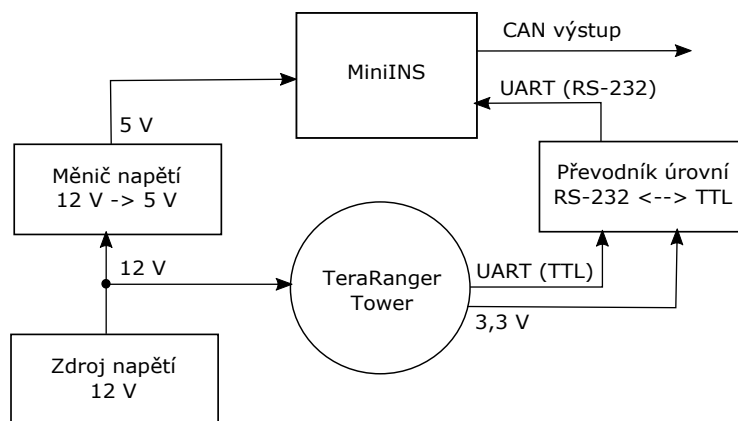
V realizovaném systému je využíván přesný mód zařízení. *Crosstalk avoidance* je vypnut, protože při použití konfigurace se čtyřmi senzory, jejichž zorná pole jsou od sebe pootočena o  $90^\circ$ , přeslechy nejsou pravděpodobné. Měřená data jsou přijímána v binární podobě přes rozhraní UART. Binární zpráva je dlouhá 19 bytů a má následující formát [30]: THXXXXXXXXXXXXXXXXXCRCS, kde:

- TH je hlavička zprávy (2 byty),
- 8 dvojic XX - každá dvojice značí 2 byty se zakódovanou vzdáleností jednoho senzoru od překážky v milimetrech,
- CRC8 - kontrolní součet (1 byte).

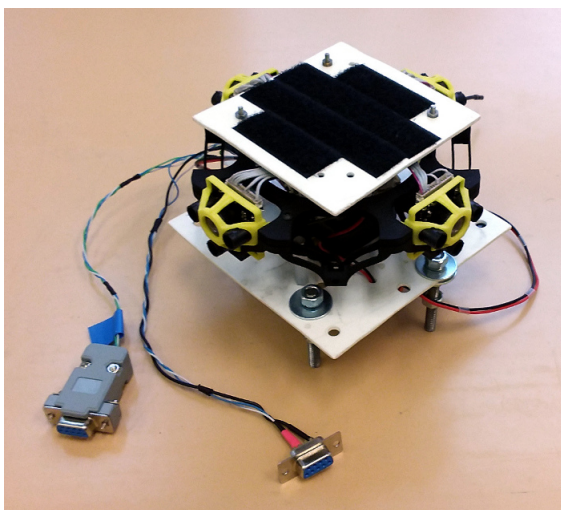
Pokud není některý z 8 senzorů zapojen, odpovídající hodnota vzdálenosti ve zprávě je nastavena na 0xFFFF.

### 3.3 Návrh integrace senzorů do jednoho systému

V této části je popsáno, jakým způsobem byly výše představené senzory integrovány do jednoho systému. Na obrázku 3.6 je znázorněn návrh propojení MiniINS a senzoru vzdáleností. Na obrázku 3.7 je fotografie vytvořeného systému. Mikrokontrolér v MiniINS je použit jako hlavní výpočetní jednotka. Kromě měřených dat z IMU, která je součástí MiniINS, sbírá i data ze senzorů vzdálenosti přes UART a 100krát za sekundu počítá nové hodnoty navigačních proměnných a posílá je na výstup. Parametry rozhraní UART jsou nastaveny na: baud rate 921600 bit/s, 1 stop bit, 8 datových bitů, žádný paritní bit.



Obrázek 3.6: Návrh propojení senzorů do jednoho systému



Obrázek 3.7: Fotografie vytvořeného navigačního systému

### 3.3.1 Zarovnání os a očíslování senzorů

Osy souřadnicového systému navigovaného tělesa jsou sesouhlaseny s osami MiniINS následovně:

- osa  $x^b$  = osa citlivosti  $y$  inerciální měřicí jednotky,
- osa  $y^b$  = osa citlivosti  $-x$  inerciální měřicí jednotky,
- osa  $z^b$  = osa citlivosti  $z$  inerciální měřicí jednotky.

TeraRanger Tower je orientován tak, že senzor vzdálenosti

- č. 1 míří v kladném směru osy  $x^b$ ,
- č. 2 míří v záporném směru osy  $x^b$ ,
- č. 3 míří v kladném směru osy  $y^b$ ,
- č. 4 míří v záporném směru osy  $y^b$ .

Takto definované očíslování senzorů vzdálenosti je dále v textu používáno. Očíslování používané zařízením Teraranger Tower se však liší. Platí, že:

- senzor č. 1 podle očíslování definovaného touto prací = senzor č. 2 podle očíslování zařízením Teraranger Tower,
- senzor č. 2 podle očíslování definovaného touto prací = senzor č. 6 podle očíslování zařízením Teraranger Tower,
- senzor č. 3 podle očíslování definovaného touto prací = senzor č. 8 podle očíslování zařízením Teraranger Tower,
- senzor č. 4 podle očíslování definovaného touto prací = senzor č. 4 podle očíslování zařízením Teraranger Tower.

Senzory 1, 3, 5, 7 podle očíslování zařízením Teraranger Tower nejsou do navigačního systému připojeny.

### 3.3.2 Výstupy navrženého systému

Výstupní zprávy s informacemi o naměřených datech a odhadech navigačních proměnných se posílají po sběrnici CAN. Identifikátory zpráv s odhady navigačních proměnných jsou v tabulce 3.8. Úplný seznam posílaných zpráv, je v tabulce A.1 v příloze A. Výstupní frekvence je 100 Hz. Zprávy jsou posílány v režimu s 11-bitovým identifikátorem a 8 datovými byty. Hodnoty jsou v datovém poli uloženy v *little endian* formátu.

ID	Rozsah indexů v datovém poli	Datový typ	Popis
0x600	0-3	float	pozice - osa $x^n$ [m]
	4-7	float	pozice - osa $y^n$ [m]
0x601	0-3	float	pozice - osa $z^n$ [m]
	4-7	float	rychlost - osa $x^n$ [m/s]

*Pokračování tabulky na další stránce →*

→ Pokračování tabulky z předchozí strany

ID	Rozsah indexů v datovém poli	Datový typ	Popis
0x602	0-3	float	rychlost - osa $y^n$ [m/s]
	4-7	float	rychlost - osa $z^n$ [m/s]
0x603	0-3	float	Eulerův úhel - $\phi$ [rad]
	4-7	float	Eulerův úhel - $\theta$ [rad]
0x604	0-3	float	Eulerův úhel - $\psi$ [rad]
	4-7	float	bias senzoru úhlové rychlosti - osa $x^b$ [rad/s]
0x605	0-3	float	bias senzoru úhlové rychlosti - osa $y^b$ [rad/s]
	4-7	float	bias senzoru úhlové rychlosti - osa $z^b$ [rad/s]
0x606	0-3	float	bias akcelerometru - osa $x^b$ [m/s <sup>2</sup> ]
	4-7	float	bias akcelerometru - osa $y^b$ [m/s <sup>2</sup> ]
0x607	0-3	float	bias akcelerometru - osa $z^b$ [m/s <sup>2</sup> ]
	4-7	float	nula
0x608	0-3	float	kolmá vzdálenost k překážce č. 1 [m]
	4-7	float	kolmá vzdálenost k překážce č. 2 [m]
0x609	0-3	float	kolmá vzdálenost k překážce č. 3 [m]
	4-7	float	kolmá vzdálenost k překážce č. 4 [m]

Tabulka 3.8: Seznam zpráv s odhady navigačních proměnných posílaných po sběrnici CAN

## 4 Návrh navigačního algoritmu

---

Navigační algoritmus integruje INS (inerciální navigační systém) s měřením senzorů vzdálenosti. Jelikož uvnitř budov převažují rovné plochy (zdi, skříně, stoly...), vytvořený algoritmus předpokládá existenci těchto ploch, snaží se je detekovat a využít ve svých výpočtech, jak je podrobněji uvedeno dále.

### 4.1 Předpoklady

Níže představený navigační algoritmus má tyto předpoklady a omezení:

- Předpokládá se existence rovných ploch v okolním prostředí - ve standardní budově většinou splněno.
- Dále se předpokládá, že vertikální plochy jsou kolmé na vodorovnou rovinu - u stěn splněno, ale v prostředí mohou existovat i plochy, které toto nesplňují (např. desky opřené o zeď) a při případném budoucím rozšiřování algoritmu by toto omezení bylo dobré odstranit.

Navigační souřadnicový systém je shodný s počátečním neseným tečným souřadnicovým systémem.

### 4.2 Definice proměnných

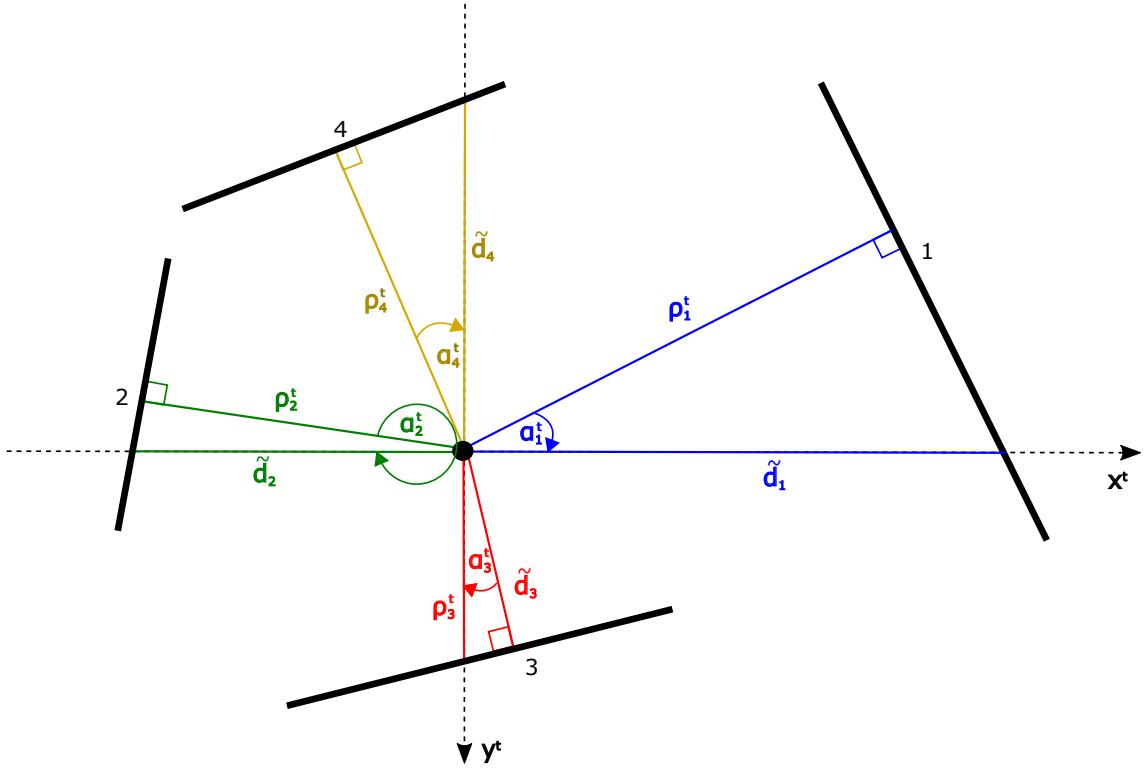
Nejdříve ze všeho je třeba očíslovat směry:

- směr č. 1 míří v kladném směru osy  $x$  daného souřadnicového systému,
- směr č. 2 míří v záporném směru osy  $x$  daného souřadnicového systému,
- směr č. 3 míří v kladném směru osy  $y$  daného souřadnicového systému,
- směr č. 4 míří v záporném směru osy  $y$  daného souřadnicového systému.

Toto očíslování je v souladu s očíslováním senzorů vzdálenosti z části 3.3. Senzor vzdálenosti č.  $i$  míří v  $i$ -tém směru souřadnicového systému navigovaného tělesa. Stejně tak jsou očíslovány i překážky. Senzor č.  $i$  „vidí“  $i$ -tou překážku. Toto očíslování je podstatné pro popis navigačního algoritmu.

Na obrázku 4.1 je znázorněný význam dále používaných proměnných. Proměnné  $\tilde{d}_i$  značí průmět vzdálenosti změřené  $i$ -tým senzorem do vodorovné roviny (tedy vzdálenosti kompenzované o náklon a sklon navigovaného objektu). V případě, že vertikální plochy jsou kolmé na vodorovnou rovinu (předpoklad) platí, že

$$\begin{aligned}\tilde{d}_1 &= d_1 \cos \theta, \\ \tilde{d}_2 &= d_2 \cos \theta, \\ \tilde{d}_3 &= d_3 \cos \phi, \\ \tilde{d}_4 &= d_4 \cos \phi,\end{aligned}\tag{4.1}$$



Obrázek 4.1: Definice proměnných

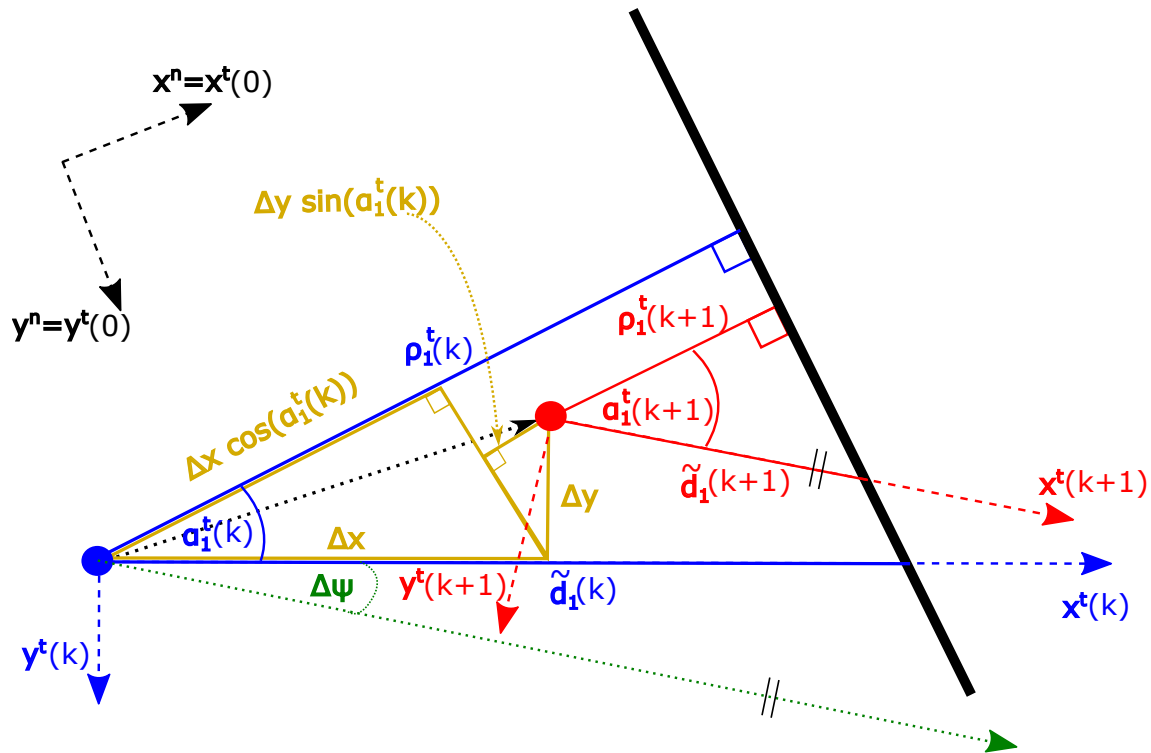
kde  $d_i$  je vzdálenost změřená  $i$ -tým senzorem,  $\theta$  značí sklon (*pitch*) a  $\phi$  značí náklon (*roll*).

Proměnná  $\rho_i^t$  je kolmá vzdálenost navigovaného objektu od  $i$ -té stěny (překážky). Proměnná  $\alpha_i^t$  je úhel, který svírá  $i$ -tý směr neseného tečného souřadnicového systému s kolmicí na  $i$ -tou stěnu (viz obrázek 4.1). Úhel  $\alpha_i^t$  může nabývat hodnot z intervalu  $\langle 0^\circ, 90^\circ \rangle \cup \langle 270^\circ, 360^\circ \rangle$ . Obdobně je definovaná vzdálenost  $\rho_i^n$  a úhel  $\alpha_i^n$ . Proměnná  $\rho_i^n$  je kolmá vzdálenost počátku navigačního souřadnicového systému od  $i$ -té překážky. Proměnná  $\alpha_i^n$  je úhel, který svírá  $i$ -tý směr navigačního souřadnicového systému s kolmicí na  $i$ -tou stěnu. Úhel  $\alpha_i^n$  může nabývat hodnot z intervalu  $\langle 0^\circ, 360^\circ \rangle$ .

### 4.3 Geometrie navigační úlohy

Na obrázku 4.2 je znázorněna geometrie navigační úlohy pro senzor vzdálenosti 1 (senzor míří v kladném směru osy  $x^b$  - očíslování senzorů definováno v části 3.3). Modře je znázorněna situace v čase  $k$ , červeně situace v čase  $k + 1$ . Mezi časem  $k$  a  $k + 1$  došlo k rotaci a posunu objektu. Z obrázku je vidět, že

$$\begin{aligned}\Delta\rho_1^t &= \rho_1^t(k+1) - \rho_1^t(k) = -\Delta x \cos(\alpha_1^t(k)) + \Delta y \sin(\alpha_1^t(k)), \\ \Delta\alpha_1^t &= \alpha_1^t(k+1) - \alpha_1^t(k) = \Delta\psi.\end{aligned}$$



Obrázek 4.2: Geometrie navigační úlohy

Pro senzory vzdálenosti 2, 3 a 4 se rovnice liší o znaménka kvůli odlišné orientaci os. Rovnice pro změny všech čtyř kolmých vzdáleností jsou

$$\begin{aligned}
 \Delta \rho_1^t &= \rho_1^t(k+1) - \rho_1^t(k) = -\Delta x \cos(\alpha_1^t(k)) + \Delta y \sin(\alpha_1^t(k)), \\
 \Delta \rho_2^t &= \rho_2^t(k+1) - \rho_2^t(k) = \Delta x \cos(\alpha_2^t(k)) - \Delta y \sin(\alpha_2^t(k)), \\
 \Delta \rho_3^t &= \rho_3^t(k+1) - \rho_3^t(k) = -\Delta y \cos(\alpha_3^t(k)) - \Delta x \sin(\alpha_3^t(k)), \\
 \Delta \rho_4^t &= \rho_4^t(k+1) - \rho_4^t(k) = \Delta y \cos(\alpha_4^t(k)) + \Delta x \sin(\alpha_4^t(k)).
 \end{aligned} \tag{4.2}$$

Pro všechny úhly  $\alpha_i^t$  ( $i \in \{1, 2, 3, 4\}$ ) platí rovnost

$$\Delta \alpha_i^t = \alpha_i^t(k+1) - \alpha_i^t(k) = \Delta \psi. \tag{4.3}$$

Dále platí, že

$$\tilde{d}_i(k+1) = \frac{\rho_i^t(k+1)}{\cos(\alpha_i^t(k+1))} = \frac{\rho_i^t(k+1)}{\cos(\alpha_i^t(k) + \Delta \psi)} \tag{4.4}$$

pro každý senzor  $i$ , v každém čase  $k$ .

Z rovnic (4.2) a (4.3) vyplývá vztah mezi  $\alpha_i^t$ ,  $\alpha_i^n$ ,  $\rho_i^t$  a  $\rho_i^n$ . Platí, že

$$\begin{aligned}
 \rho_1^t &= \rho_1^n - p_x \cos(\alpha_1^n) + p_y \sin(\alpha_1^n), \\
 \rho_2^t &= \rho_2^n + p_x \cos(\alpha_2^n) - p_y \sin(\alpha_2^n), \\
 \rho_3^t &= \rho_3^n - p_y \cos(\alpha_3^n) - p_x \sin(\alpha_3^n), \\
 \rho_4^t &= \rho_4^n + p_y \cos(\alpha_4^n) + p_x \sin(\alpha_4^n),
 \end{aligned} \tag{4.5}$$

a dále, že

$$\alpha_i^t = \alpha_i^n + \psi; \quad i \in \{1, 2, 3, 4\}. \tag{4.6}$$



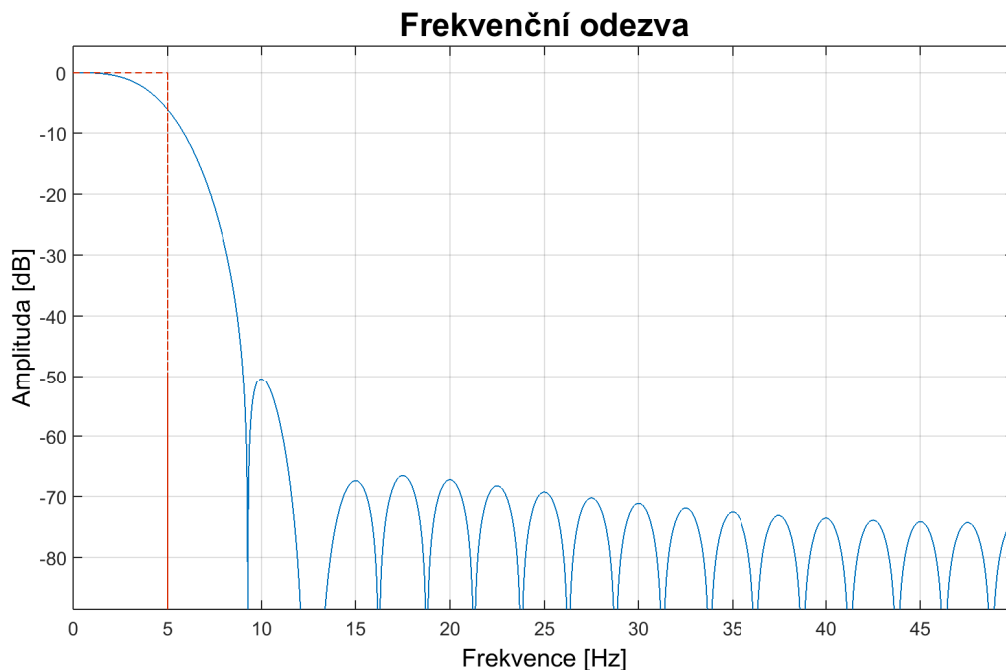
Dvojice  $(p_x, p_y)$  z rovnice (4.5) je pozice navigovaného objektu v horizontální rovině v navigačním souřadnicovém systému a  $\psi$  z rovnice (4.6) je úhel, o který se objekt od začátku pohybu otočil.

## 4.4 Předzpracování dat

IMU MPU-9150 poskytuje inerciální data na frekvenci 1000 Hz. Každých 10 vzorků je zprůměrováno a tento průměr vstupuje následně do *lowpass* FIR filtru, kvůli odstranění vysokofrekvenčního šumu. Barometr MS5611 poskytuje data na 100 Hz. Výstupy z něj jsou filtrovány stejným filtrem jako data z IMU. Frekvenční odezva filtru je znázorněna na obrázku 4.3. Parametry filtru jsou:

- $F_{cutoff} = 5$  Hz,
- $F_{sample} = 100$  Hz,
- řád filtru = 40.

Teraranger Tower je nakonfigurován pro práci v přesném módu s vypnutým *crosstalk avoidance*. V této konfiguraci není obnovovací frekvence pevně daná a je závislá na vzdálenosti od překážky (viz tab. 3.7). Navigační algoritmus ve stávající podobě předpokládá přísun nových dat na 100 Hz. Do algoritmu tedy vstupuje průměr vzdáleností za posledních 10 ms. Pokud za posledních 10 ms nebyly vzdálenosti změřeny, do algoritmu vstupuje odhad získaný lineární interpolací. Z toho vyplývá následující omezení: vzdálenost senzoru od překážky by neměla přesáhnout 5 metrů. Pokud je vzdálenost senzoru od překážky větší



Obrázek 4.3: Frekvenční odezva dolní propusti

než 5 metrů, frekvence měření padá pod 60 Hz (viz tabulku 3.7) a roste podíl interpolovaných vzdáleností vstupujících do algoritmu, což ovlivňuje jeho přesnost.

Jelikož jsou inerciální a barometrická data filtrována FIR filtrem řádu 40, je nutné vzdálenostní data před vstupem do EKF zpozdít o 20 vzorků a tím je zarovnat s daty z IMU a tlakového senzoru.

## 4.5 Model EKF

Rozšířený Kalmanův filtr (viz kapitola 2.5) je páteří celého algoritmu. Predikce běží na 100 Hz. Korigován je pomocí informací získaných ze senzorů vzdálenosti na frekvenci 10 Hz.

### 4.5.1 Model dynamického systému

Stavový vektor

$$\mathbf{x} = \left[ \mathbf{p}^n \quad \mathbf{v}^n \quad \boldsymbol{\epsilon} \quad \mathbf{b}_g^b \quad \mathbf{b}_a^b \quad \boldsymbol{\rho}^t \right]^T,$$

kde  $\mathbf{p}^n$  je vektor pozic v navigačním souřadnicovém systému,  $\mathbf{v}^n$  je vektor rychlostí v navigačním souřadnicovém systému,  $\boldsymbol{\epsilon} = [\phi, \theta, \psi]^T$  je vektor Eulerových úhlů ( $\phi$  = příčný náklon (*roll*),  $\theta$  = podélný sklon (*pitch*),  $\psi$  = vybočení (*yaw*)),  $\mathbf{b}_g^b$  je vektor biasů senzoru úhlové rychlosti,  $\mathbf{b}_a^b$  je vektor biasů akcelerometru a  $\boldsymbol{\rho}^t$  je vektor kolmých vzdáleností navigovaného objektu od překážek.

Diferenciální rovnice popisující systém jsou následující:

$$\begin{aligned} \dot{\mathbf{p}}^n &= \mathbf{v}^n \\ \dot{\mathbf{v}}^n &= \mathbf{g}^n + \mathbf{C}_b^n \left[ (\mathbf{f}_{ib}^b - \mathbf{b}_a^b) + (\mathbf{C}_n^b \mathbf{v}^n) \times (\boldsymbol{\omega}_{ib}^b - \mathbf{b}_g^b) \right] \\ \dot{\boldsymbol{\epsilon}} &= \mathbf{R}(\boldsymbol{\omega}_{ib}^b - \mathbf{b}_g^b) \\ \dot{\mathbf{b}}_g^b &= \mathbf{0} \\ \dot{\mathbf{b}}_a^b &= \mathbf{0} \\ \dot{\rho}_1^t &= -v_x^n \cos \alpha_1^n + v_y^n \sin \alpha_1^n \\ \dot{\rho}_2^t &= v_x^n \cos \alpha_2^n - v_y^n \sin \alpha_2^n \\ \dot{\rho}_3^t &= -v_y^n \cos \alpha_3^n - v_x^n \sin \alpha_3^n \\ \dot{\rho}_4^t &= v_y^n \cos \alpha_4^n + v_x^n \sin \alpha_4^n \end{aligned}$$

kde  $\boldsymbol{\omega}_{ib}^b$  je vektor změřených úhlových rychlostí,  $\mathbf{f}_{ib}^b$  je vektor změřených specifických sil,  $\mathbf{C}_n^b \mathbf{v}^n \times (\boldsymbol{\omega}_{ib}^b - \mathbf{b}_g^b)$  je člen kompenzující odstředivé zrychlení podle [9] a

$$\mathbf{R} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \theta \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \theta \sec \theta \end{bmatrix}.$$

Rovnice pro změnu  $\rho_i^t$  vychází z (4.2) odvozené v části „Geometrie navigační úlohy“.

### 4.5.2 Vztah mezi stavovými proměnnými a měřením

Vektor měření

$$\mathbf{y} = \begin{bmatrix} \tilde{\mathbf{p}}^n & \tilde{\mathbf{d}} & p_{baro} \end{bmatrix}^T,$$

kde  $\tilde{\mathbf{p}}^n = [\tilde{p}_x^n, \tilde{p}_y^n]^T$  je odhad pozice získané pomocí metody nejmenších čtverců a  $\tilde{\mathbf{d}}$  je vektor průmětů změřených vzdáleností do vodorovné roviny a  $p_{baro}$  je tlak měřený senzorem barometrického tlaku.

Vztah mezi měřením a stavy je

$$\begin{aligned} \tilde{p}_x^n &= p_x^n \\ \tilde{p}_y^n &= p_y^n \\ \tilde{d}_1 &= \frac{\rho_1^t}{\cos(\alpha_1^n + \psi)} \\ \tilde{d}_2 &= \frac{\rho_2^t}{\cos(\alpha_2^n + \psi)} \\ \tilde{d}_3 &= \frac{\rho_3^t}{\cos(\alpha_3^n + \psi)} \\ \tilde{d}_4 &= \frac{\rho_4^t}{\cos(\alpha_4^n + \psi)} \\ p_{baro} &= p_{ref} [1 - k_1 (-p_z^n)]^{k_2} \end{aligned}$$

kde rovnice pro  $\tilde{d}_i$  vychází ze vztahu (4.4) odvozeného v části „Geometrie navigační úlohy“,  $p_{ref}$  je tlak ve výšce  $p_z^n = 0$ , konstanta  $k_1 = 2.2558 \cdot 10^{-5}$  a konstanta  $k_2 = 5.2559$ .

V následujícím textu je popsán odhad pozice  $\tilde{\mathbf{p}}^n$  pomocí metody nejmenších čtverců. Jak z popisu vyplývá,  $\tilde{\mathbf{p}}^n$  a  $\tilde{\mathbf{d}}$  z vektoru měření jsou závislé. Vyzkoušené modely bez  $\tilde{\mathbf{p}}^n$  ve vektoru měření byly méně stabilní. Proto byl odhad  $\tilde{\mathbf{p}}^n$  ve vektoru měření ponechán a hodnoty v kovarianční matici šumu měření  $\mathbf{R}$  nastaveny výše.

#### Odhad pozice pomocí metody nejmenších čtverců

Jestliže se za  $\rho_i^t$  v rovnicích (4.5) dosadí  $\tilde{d}_i \cos(\alpha_i^n + \psi)$  podle vztahu (4.4), pak po převedení proměnných na jednu stranu se získají rovnice

$$\begin{aligned} 0 &= -\tilde{d}_1 \cos(\alpha_1^n + \hat{\psi}) + \rho_1^n - \tilde{p}_x^n \cos(\alpha_1^n) + \tilde{p}_y^n \sin(\alpha_1^n), \\ 0 &= -\tilde{d}_2 \cos(\alpha_2^n + \hat{\psi}) + \rho_2^n + \tilde{p}_x^n \cos(\alpha_2^n) - \tilde{p}_y^n \sin(\alpha_2^n), \\ 0 &= -\tilde{d}_3 \cos(\alpha_3^n + \hat{\psi}) + \rho_3^n - \tilde{p}_y^n \cos(\alpha_3^n) - \tilde{p}_x^n \sin(\alpha_3^n), \\ 0 &= -\tilde{d}_4 \cos(\alpha_4^n + \hat{\psi}) + \rho_4^n + \tilde{p}_y^n \cos(\alpha_4^n) + \tilde{p}_x^n \sin(\alpha_4^n), \end{aligned} \tag{4.7}$$

kde  $\hat{\psi}$  je odhad úhlu vybočení z EKF. Proměnné  $\rho_i^n$  a  $\alpha_i^n$  lze dopočítat z historie bodů, jak je popsáno níže a proměnné  $\tilde{d}_i$  jsou změřené vzdálenosti kompenzované o odhadnutý náklon a sklon.

Pro neznámý vektor pozice  $\tilde{\mathbf{p}}^n = [\tilde{p}_x^n, \tilde{p}_y^n]^T$  lze rovnice vyjádřit v maticové formě následovně:

$$\mathbf{A} \tilde{\mathbf{p}}^n = \mathbf{b}, \tag{4.8}$$

kde

$$\mathbf{A} = \begin{bmatrix} -\cos(\alpha_1^n) & \sin(\alpha_1^n) \\ \cos(\alpha_2^n) & -\sin(\alpha_2^n) \\ -\sin(\alpha_3^n) & -\cos(\alpha_3^n) \\ \sin(\alpha_4^n) & \cos(\alpha_4^n) \end{bmatrix} \quad (4.9)$$

a

$$\mathbf{b} = \begin{bmatrix} \tilde{d}_1 \cos(\alpha_1^n + \hat{\psi}) - \rho_1^n \\ \tilde{d}_2 \cos(\alpha_2^n + \hat{\psi}) - \rho_2^n \\ \tilde{d}_3 \cos(\alpha_3^n + \hat{\psi}) - \rho_3^n \\ \tilde{d}_4 \cos(\alpha_4^n + \hat{\psi}) - \rho_4^n \end{bmatrix}. \quad (4.10)$$

Řešením úlohy

$$\min_{\tilde{\mathbf{p}}^n \in \mathbb{R}^{2 \times 1}} \|\mathbf{A}\tilde{\mathbf{p}}^n - \mathbf{b}\|^2$$

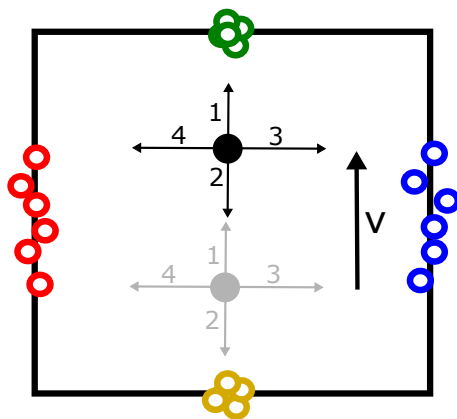
se získá přibližné řešení soustavy (4.7) ve smyslu nejmenších čtverců. Takto získané odhady pozice pak vstupují do vektoru měření.

## 4.6 Historie bodů

V předchozí části vystupuje jak v popisu dynamického systému, tak v popisu vztahu mezi měřeními a stavy proměnná  $\alpha_i^n$ . Pro připomenutí je to úhel, který svírá  $i$ -tý směr navigačního souřadnicového systému s kolmicí na  $i$ -tou překážkou. Aby mohl být tento parametr spočítán, je třeba znát polohu překážky. Pro určení polohy překážky je potřeba uchovávat historii bodů dříve „viděných“ senzory vzdálenosti v navigačním souřadnicovém systému. Na obrázku 4.4 je pro ilustraci znázorněna historie „viděných“ bodů při translačním pohybu v jedné ose.

Navigovaný objekt se na obrázku pohybuje pouze ve směru č. 1. Ve směru č. 3 a 4 je tak možné proložit naposledy „viděné“ body přímkou a určit parametry  $\alpha_i^n$  a  $\rho_i^n$ . Ve směru 1 a 2 ale tyto parametry určit v tomto ilustračním příkladě nelze. Senzor vzdálenosti 1 a 2 „vidí“ stále ten samý bod, z něhož nelze vyčíst polohu překážky.

Historie bodů uchovává naposledy „viděné“ body v navigačním souřadnicovém systému. Dále v textu je jako  $H_i$  značena historie bodů „viděných“  $i$ -tým senzorem vzdále-



Obrázek 4.4: Ilustrační příklad bodů „viděných“ senzory při pohybu v jedné ose

nosti, tedy bodů patřících  $i$ -té překážce. Vhodnou inicializací a aktualizací mapy je dbáno na to, aby uložené body nebyly koncentrovány v jednom místě, ale rozmístěny tak, aby jimi bylo možné proložit přímkou a dopočítat její parametry  $\alpha_i^n$  a  $\rho_i^n$ .

Historie bodů uchovává 15 bodů pro každý senzor vzdálenosti. Může se stát, že uchovávané body jednoho senzoru vzdálenosti nepatří k jedné stěně/ploše (např. v rozích). Pak přímkou proložená těmito body neodpovídá poloze překážky a spočítané parametry  $\alpha_i^n$ ,  $\rho_i^n$  přímkou proložené těmito body nejsou validní. V tom případě tyto nevalidní parametry nevstupují do výpočtu EKF.

Bod  $\mathbf{b}_i^n$  v navigačním souřadnicovém systému „viděný“  $i$ -tým senzorem se získá z průmětů měřených vzdáleností do vodorovné roviny  $\tilde{d}_i$  následovně:

$$\begin{aligned}
 \mathbf{b}_1^n &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}_t^n \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{d}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} p_x^n \\ p_y^n \end{bmatrix} \\
 \mathbf{b}_2^n &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}_t^n \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\tilde{d}_2 \\ 0 \end{bmatrix} + \begin{bmatrix} p_x^n \\ p_y^n \end{bmatrix} \\
 \mathbf{b}_3^n &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}_t^n \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \tilde{d}_3 \end{bmatrix} + \begin{bmatrix} p_x^n \\ p_y^n \end{bmatrix} \\
 \mathbf{b}_4^n &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{C}_t^n \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -\tilde{d}_4 \end{bmatrix} + \begin{bmatrix} p_x^n \\ p_y^n \end{bmatrix}
 \end{aligned} \tag{4.11}$$

#### 4.6.1 Extrakce přímek z historie bodů

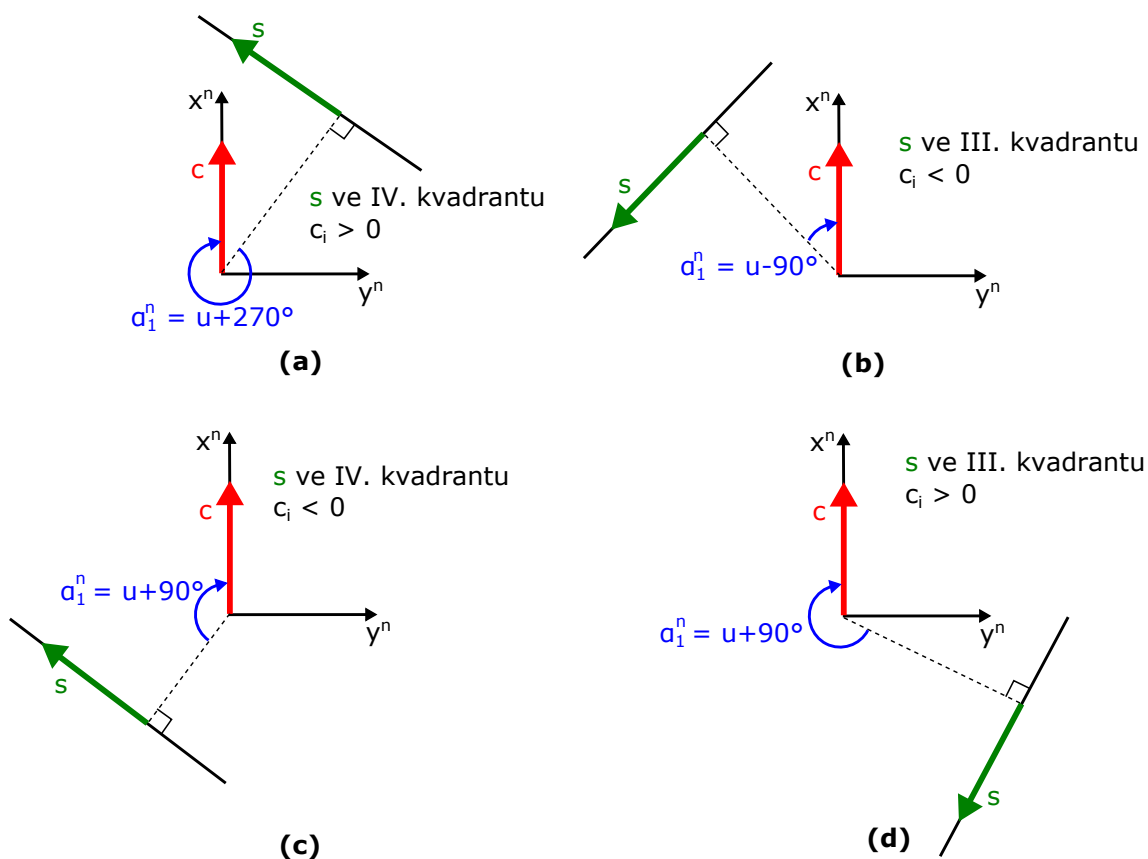
Obecné rovnice přímek ve tvaru  $y = a_i x + c_i$  se z historie bodů dostanou za pomoci SVD rozkladu postupem, který je popsán v části 2.7. Postup pro získání parametrů  $\alpha_i^n$ ,  $\rho_i^n$  z obecné rovnice přímkou je v algoritmu 1. Algoritmus nejdříve spočítá parametry pro směr č. 1 navigačního souřadnicového systému ( $\rho_1^n$  a  $\alpha_1^n$ ) a následně k nim přičte 0/90/180/270 stupňů podle toho, jak je orientován senzor č.  $i$ , který „vidí“ danou stěnu, vůči senzoru č. 1. Vysvětlující obrázek k algoritmu 1 je na obrázku 4.5.

V algoritmu 1 se počítá vzdálenost bodu od přímkou a úhel mezi vektory. Vzdálenost bodu  $[x, y]$  od přímkou  $p : y = ax + c$  se počítá jako

$$\frac{|ax - y + c|}{\sqrt{a^2 + 1}}$$

a úhel mezi vektory  $\mathbf{u}$  a  $\mathbf{v}$  se počítá jako

$$\arccos \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}.$$



Obrázek 4.5: Obrázek k algoritmu 1

**Algoritmus 1** Výpočet  $\alpha_i^n, \rho_i^n$  z obecné rovnice přímky

**Vstup:** parametry obecné rovnice přímky  $a_i$  a  $c_i$  ( $y = a_i x + c_i$ ) „viděné“ senzorem č.  $i$ ,  
senzor  $i \in \{1, 2, 3, 4\}$

**Výstup:** parametry  $\alpha_i^n, \rho_i^n$

- 1:  $\mathbf{c} \leftarrow [1 \ 0]^T$  {směrový vektor osy  $x$ }
- 2:  $\mathbf{s} \leftarrow [-1/a_i \ -1]^T$  {směrový vektor přímky  $\mathbf{s}$  je ve III. nebo IV. kvadrantu}
- 3:  $u \leftarrow$  úhel mezi vektory  $\mathbf{s}$  a  $\mathbf{c}$
- 4:
- 5: **if**  $s_1 > 0$  **and**  $c_i > 0$  **then**
- 6:    $\alpha_1^n \leftarrow u + 270^\circ$  {viz obrázek 4.5a}
- 7: **else if**  $s_1 > 0$  **and**  $c_i \leq 0$  **then**
- 8:    $\alpha_1^n \leftarrow u + 90^\circ$  {viz obrázek 4.5c}
- 9: **else if**  $s_1 \leq 0$  **and**  $c_i > 0$  **then**
- 10:    $\alpha_1^n \leftarrow u + 90^\circ$  {viz obrázek 4.5d}
- 11: **else if**  $s_1 \leq 0$  **and**  $c_i \leq 0$  **then**
- 12:    $\alpha_1^n \leftarrow u - 90^\circ$  {viz obrázek 4.5b}
- 13: **end if**
- 14: {Pokračování algoritmu na další stránce  $\rightarrow$ }

---

```

15: {→ Pokračování algoritmu z předchozí strany}
16: switch ( $i$ )
17:   case 1:  $\alpha_i^n \leftarrow \alpha_1^n$ 
18:   case 2:  $\alpha_i^n \leftarrow \alpha_1^n + 180^\circ + r \cdot 360^\circ$ ,  $r \in \mathbb{Z}$  tak, aby  $\alpha_i^n \in \langle 0^\circ, 360^\circ \rangle$ 
19:   case 3:  $\alpha_i^n \leftarrow \alpha_1^n + 90^\circ + r \cdot 360^\circ$ ,  $r \in \mathbb{Z}$  tak, aby  $\alpha_i^n \in \langle 0^\circ, 360^\circ \rangle$ 
20:   case 4:  $\alpha_i^n \leftarrow \alpha_1^n + 270^\circ + r \cdot 360^\circ$ ,  $r \in \mathbb{Z}$  tak, aby  $\alpha_i^n \in \langle 0^\circ, 360^\circ \rangle$ 
21: end switch
22:
23:  $\rho_i^n \leftarrow$  vzdálenost bodu  $[0, 0]$  od přímky  $y = a_i x + c_i$ 

```

---

#### 4.6.2 Validace parametrů přímek

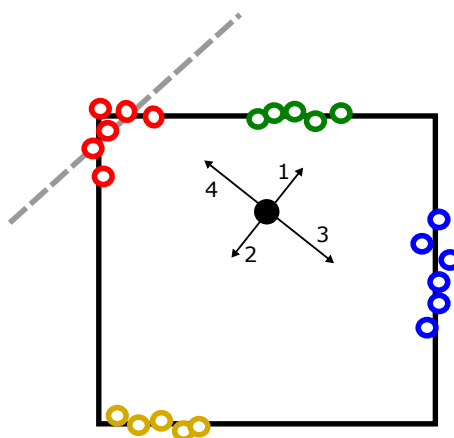
V předchozí části byl popsán způsob výpočtu parametrů  $\alpha_i^n$ ,  $\rho_i^n$  přímek odhadnutých pomocí bodů uchovaných v historii senzoru č.  $i$ . Nicméně přímka proložená body z historie nemusí odpovídat reálné poloze překážky. Ilustrační příklad je na obrázku 4.6. Zatímco přímky proložené body z historií senzorů vzdálenosti č. 1, 2 a 3 přibližně budou odpovídat reálné poloze překážky, přímka proložená body z historie senzoru č. 4 (naznačena přerušovaně) odpovídat nebude. Proto parametr  $\alpha_4^n$  spočítaný na základě takto proložené přímky nesmí být použit v korekčním kroku EKF.

Postup pro validaci parametrů odhadnuté přímky je v algoritmu 2. V tomto algoritmu je jako  $S_{invalid}$  označena množina překážek podezřelých z toho, že jejich poloha neodpovídá proložené přímce. Překážka  $i \in S_{invalid}$  právě tehdy, když parametry  $\alpha_i^n$ ,  $\rho_i^n$  nejsou validní.

V algoritmu 2 jsou na vstupu dosud nedefinované proměnné:

- $e_{sse,i}$  - chyba SSE (*Error Sum of Squares*),
- $e_{dev,i}$  - vzdálenost právě „viděného“ bodu od proložené přímky,
- $\delta_{est,i}$  - odchylka předpokládané vzdálenosti od skutečně změřené,
- $\sigma_i$  - diference rozdílů mezi právě změřenou a zpožděnou vzdáleností.

Tyto klasifikační proměnné slouží pro rozhodnutí, zda jsou parametry  $\alpha_i^n$ ,  $\rho_i^n$  proložené přímky validní, či nikoliv. Každá klasifikační proměnná má nastaveny prahy, jejichž pře-



Obrázek 4.6: Příklad možného stavu historie bodů

**Algoritmus 2** Validace parametrů  $\alpha_i^n, \rho_i^n$ 

**Vstup:** množina podezřelých překážek  $S_{invalid}$ ,  $\sigma_i$ ,  $e_{sse,i}$ ,  $\delta_{est,i}$ ,  $e_{dev,i}$ ,  $\alpha_i^n$ , překážka  $i$ , odhad vybočení  $\hat{\psi}$

**Výstup:** aktualizovaná množina podezřelých překážek  $S_{invalid}$

- 1:  $\alpha_i^t \leftarrow \alpha_i^n + \hat{\psi} + r \cdot 360^\circ$ ,  $r \in \mathbb{Z}$  tak, aby  $\alpha_i^t \in \langle 0^\circ, 360^\circ \rangle$
- 2: **if**  $i \notin S_{invalid}$  **and** ( $\sigma_i > 5$  cm **or**  $\delta_{est,i} > 5$  cm **or**  $e_{sse,i} > 3$  cm **or**  $\cos(\alpha_i^t) < \cos(80^\circ)$ ) **then**
- 3:  $S_{invalid} \leftarrow S_{invalid} \cup \{i\}$
- 4: **else if**  $i \in S_{invalid}$  **and**  $e_{dev,i} < 5$  cm **and**  $\sigma_i < 3$  cm **and**  $e_{sse,i} < 5$  mm **and**  $\cos(\alpha_i^t) > \cos(70^\circ)$  **then**
- 5:  $S_{invalid} \leftarrow S_{invalid} \setminus \{i\}$
- 6: **end if**

kročení invaliduje parametry odhadnuté přímkou. Tyto prahy byly zjištěny experimentálně opakovaným spouštěním algoritmu.

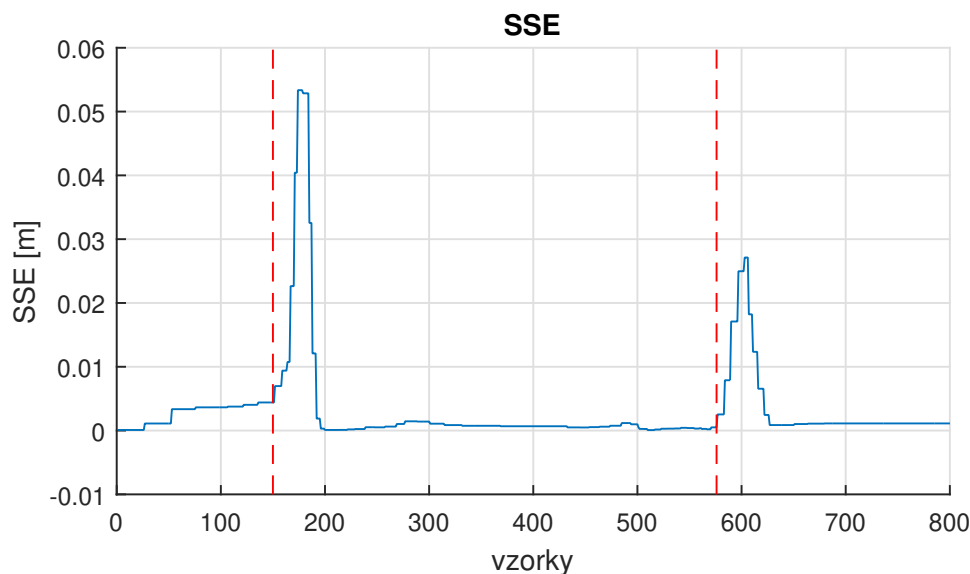
Po opětovné validaci parametrů přímkou proložených historií bodů senzoru č.  $i$  je nastaven odhad kolmé vzdálenosti ke stěně  $\hat{\rho}_i^t = \tilde{d}_i \cdot \cos(\alpha_i^n + \hat{\psi})$ .

**Error Sum of Squares** ( $e_{sse,i}$ )

Pro chybu SSE platí, že

$$e_{sse,i} = \sum_{\mathbf{b}_i^n \in H_i} \|\mathbf{b}_i^n - \bar{\mathbf{b}}_i^n\|^2,$$

kde  $\mathbf{b}_i^n$  je bod z historie senzoru  $H_i$ ,  $\bar{\mathbf{b}}_i^n$  značí kolmý průmět bodu  $\mathbf{b}_i^n$  do přímky proložené touto historií (tato přímka dále značena jako  $l_i$ ). Na obrázku 4.7 je vidět průběh SSE při přechodu přes rohy místnosti. Červeně jsou na obrázku znázorněny časy, kdy došlo k přechodu přes roh. Je vidět, že po přechodu přes roh SSE rychle roste. Maximální je



Obrázek 4.7: Vývoj SSE při přechodech přes roh místnosti



SSE ve chvíli, kdy historie „viděných“ bodů obsahuje body patřící různým překážkám přibližně rovným dílem (jako je například ilustrováno na obrázku 4.6 červeně). SSE roste po přechodu přes roh výrazně, ale ne dostatečně rychle, aby se na něj dalo výhradně spoléhat při invalidaci parametrů odhadnuté přímky. SSE slouží hlavně k opětovné validaci parametrů, poté co jeho hodnota klesne pod 5 mm.

### Odchyłka předpokládané vzdálenosti od skutečně změřené ( $\delta_{est,i}$ )

$\delta_{est,i}$  je odchyłka vzdálenosti  $d_{est,i}$ , která je odhadnutá na základě  $\hat{\rho}_i^t$ , od  $\tilde{d}_i$ . Strážka nad proměnnou značí, že jde o odhad z EKF. Platí, že

$$\delta_{est,i} = |d_{est,i} - \tilde{d}_i|,$$

přičemž

$$d_{est,i} = \frac{\hat{\rho}_i^t}{\cos(\alpha_i^n + \hat{\psi})}.$$

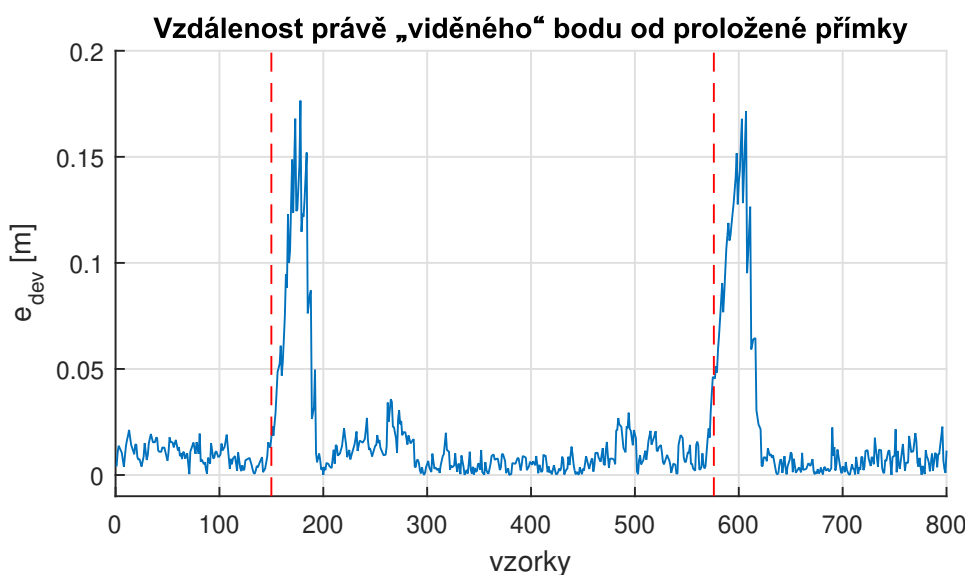
Když se tato odchyłka začne zvyšovat, značí to nevalidní parametr  $\alpha_i^n$  a tím pádem i  $\rho_i^n$ .

### Vzdálenost právě „viděného“ bodu od proložené přímky ( $e_{dev,i}$ )

Právě „viděné“ body  $\mathbf{b}_i^n = [b_{i,x}^n, b_{i,y}^n]$  lze vypočítat z rovnic (4.11). Platí, že

$$e_{dev,i} = \frac{|a_i b_{i,x}^n - b_{i,y}^n + c_i|}{\sqrt{a_i^2 + 1}},$$

kde  $a_i$  a  $c_i$  jsou parametry obecného tvaru přímky  $l_i : y = a_i x + c_i$ .  $e_{dev,i}$  je vlastně Euklidovská vzdálenost právě „viděného“ bodu  $\mathbf{b}_i^n$  od přímky  $l_i$ . Vývoj  $e_{dev,i}$  při přechodu přes roh je vyobrazen na obrázku 4.8.



Obrázek 4.8: Vývoj vzdálenosti právě „viděného“ bodu od proložené přímky při přechodech přes roh místnosti

### Diference rozdílů mezi právě změřenou a zpožděnou vzdáleností ( $\sigma_i$ )

Data vstupující do EKF jsou zpožděna o 200 ms kvůli filtraci barometrických a inerciálních dat. Kvůli zarovnání jsou uměle zpožděována i vzdálenostní data o 20 vzorků. Ne-zpožděná naměřená data o vzdálenostech pak navigačnímu algoritmu slouží jako „pohled do budoucnosti“. Dále je tedy jako  $d_i(k-20)$  označena vzdálenost, která vstupuje do EKF a jako  $d_i(k)$  právě změřená data, která budou do EKF vstupovat o 200 ms později.

Před samotným výpočtem  $\sigma_i$  je třeba spočítat rozdíl mezi právě změřenou a zpožděnou vzdáleností:

$$\xi_i(k) = d_i(k) - d_i(k-20).$$

Zároveň se počítá plovoucí průměr těchto rozdílů:

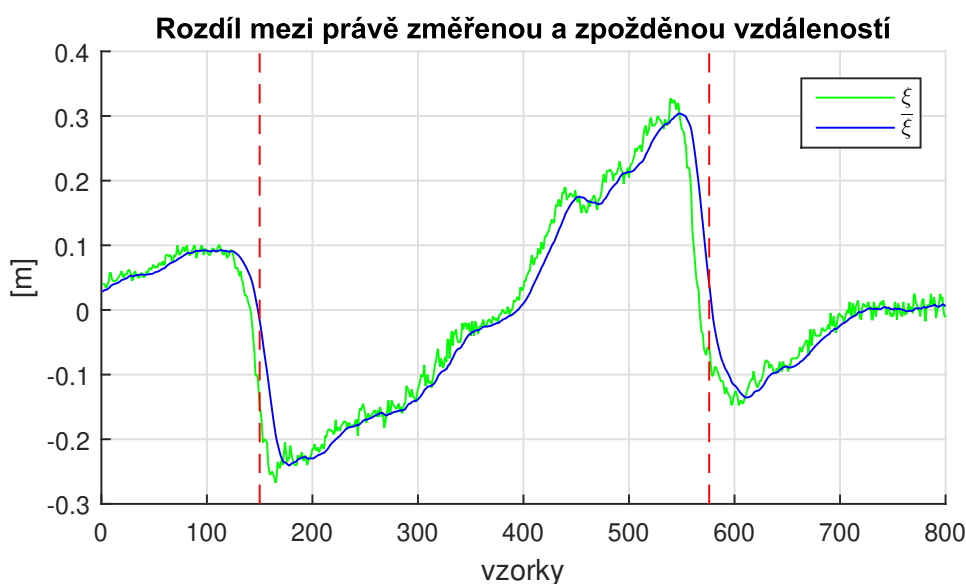
$$\bar{\xi}_i(k) = \frac{1}{21} \sum_{j=0}^{20} \xi_i(k-j).$$

Na obrázku 4.9 je zobrazen vývoj  $\xi_i$  (zeleně) a vyhlazeného  $\bar{\xi}_i$  (modře). Červeně jsou opět znázorněny časy, ve kterých došlo k přechodu přes rohy. Z obrázku je vidět, že při přechodu přes rohy dochází ke strmé změně signálů  $\xi_i$  a  $\bar{\xi}_i$ . Diference

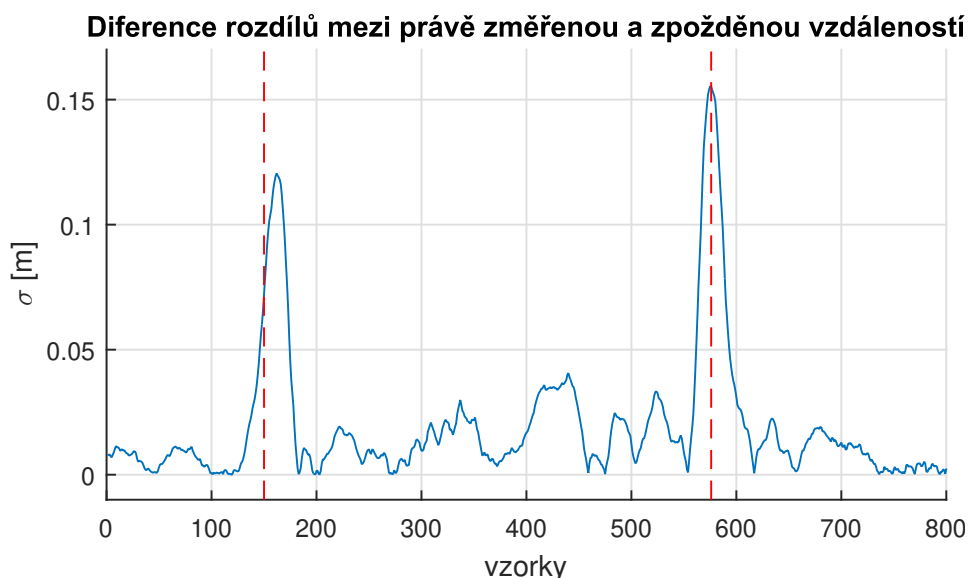
$$\sigma_i(k) = \left| \bar{\xi}_i(k-10) - \bar{\xi}_i(k) \right|$$

dosahuje v těchto strmých oblastech maxima. Diference rozdílů mezi právě změřenou a zpožděnou vzdáleností je zobrazena na obrázku 4.10. Jak je z obrázku patrné, přechody přes rohy mohou být díky klasifikační proměnné  $\sigma_i$  detekovány včas. Při druhém přechodu došlo dokonce k ideálnímu případu, kdy v tom okamžiku dosáhla  $\sigma_i$  maxima.

Nicméně tato včasná metoda invalidace parametrů odhadnuté přímky je vykoupena tím, že detekuje v signálu  $d_i$  nejenom nelinearity sledované překážky, ale i nelinearity způsobené výraznější rotací nebo zrychlováním navigovaného objektu. Navigační algoritmus



Obrázek 4.9: Vývoj rozdílů mezi právě změřenou a zpožděnou vzdáleností při přechodech přes roh místnosti



Obrázek 4.10: Vývoj difference rozdílů mezi právě změřenou a zpožděnou vzdáleností při přechodech přes roh místnosti

je určen pro kvadroptéry a jiné malé bezpilotní prostředky schopné pohybu uvnitř budov a předpokládá se, že výraznější zrychlování a rotace trvá krátkou dobu, po které následuje (i kdyby na krátký čas) víceméně lineární pohyb.

### 4.6.3 Inicializace historie bodů

Cílem inicializace historie bodů je ještě před spuštěním EKF zajistit, aby historie bodů obsahovala pro každý senzor vzdálenosti body, kterými lze proložit přímku. Jinými slovy inicializace zabraňuje situaci znázorněné v ilustračním příkladu na obrázku 4.4, kde ve směru č. 1 a 2 jsou body shluknuty do jednoho místa a přímka proložená těmito body by se vůbec nemusela shodovat s reálnou polohou stěny.

Pro správnou inicializaci historie bodů se před samotným spuštěním EKF provádí mapovací rotace (jeden z kroků algoritmu - jednotlivé kroky popsány níže). Předpokládá se, že při mapovací rotaci navigovaný objekt drží konstantní nulovou pozici a pouze rotuje kolem vertikální osy. Vkládání bodů do iniciální historie bodů popisuje algoritmus 3.

---

#### Algoritmus 3 Aktualizace iniciální historie bodů

---

**Vstup:** „viděné“ body  $b_1^n, b_2^n, b_3^n, b_4^n$ , iniciální historie bodů  $H_1, H_2, H_3, H_4$

**Výstup:** aktualizované iniciální historie bodů  $H_1, H_2, H_3, H_4$

- 1: **for** senzor č.  $i \in \{1, 2, 3, 4\}$  **do**
  - 2:   **if** vzdálenost bodu  $b_i^n$  od posledního bodu přidávaného do  $H_i > 5$  cm **then**
  - 3:     **if** počet bodů v  $H_i = 15$  **then**
  - 4:       odeber nejstarší bod z  $H_i$
  - 5:     **end if**
  - 6:     přidej bod  $b_i^n$  do  $H_i$
  - 7:   **end if**
  - 8: **end for**
-

Body  $p_i$  na vstupu algoritmu jsou spočteny podle rovnic (4.11). Jelikož EKF ještě není spuštěn, je transformační matice  $C_t^n$  v těchto rovnicích spočítána podle  $\psi$  získaného čistě integrací měřených úhlových rychlostí kompenzovaných o odhad biasu z klidových podmínek. Z toho vyplývá, že mapovací rotace musí trvat co nejkratší dobu, aby mohl být bias senzoru úhlové rychlosti považován za konstantní.

#### 4.6.4 Aktualizace historie bodů

Aktualizace historie bodů po spuštění EKF probíhá trochu odlišně než při iniciálním mapování, jak bylo popsáno v algoritmu 3. Cílem je zajistit, aby se ukládané body nekoncentrovaly do jednoho místa. Proto je definována minimální vzdálenost vkládaného bodu od posledního vloženého bodu. Navíc režim vkládání nových bodů do historie je závislý na tom, zda jsou parametry odhadnuté přímky validní, či nikoliv. Pokud validní nejsou, je počet bodů v historii pro daný směr omezen na 10 a minimální vzdálenost vkládaného bodu od posledně vloženého je poloviční. To má za následek rychlejší naplnění historie novými body a opětovnou validaci parametrů přímky. Algoritmus aktualizace historie bodů při běhu EKF je popsán v algoritmu 4.

---

#### Algoritmus 4 Aktualizace historie bodů

---

**Vstup:** viděné body  $b_1^n, b_2^n, b_3^n, b_4^n$ , historie bodů  $H_1, H_2, H_3, H_4$ , množina podezřelých bodů  $S_{invalid}$

**Výstup:** historie bodů  $H_1, H_2, H_3, H_4$

```

1: for senzor č.  $i \in \{1, 2, 3, 4\}$  do
2:   if  $i \in S_{invalid}$  and vzdálenost  $b_i^n$  od posledního bodu přidaného do  $H_i > 5$  cm then
3:     přidej bod  $b_i^n$  do  $H_i$ 
4:   else if vzdálenost bodu  $b_i^n$  od posledního bodu přidaného do  $H_i > 10$  cm then
5:     if počet bodů v  $H_i = 15$  then
6:       odeber nejstarší bod z  $M_i$ 
7:     end if
8:     přidej bod  $b_i^n$  do  $H_i$ 
9:   end if
10: end for

```

---

$S_{invalid}$  ve vstupu algoritmu 4 značí množinu překážek podezřelých z toho, že jejich poloha neodpovídá proložené přímce.

## 4.7 Navigační algoritmus

Fáze navigačního algoritmu jsou popsány na obrázku 4.11.

### 4.7.1 Čekání na inicializaci senzorů

Ze všeho nejdříve je po zapnutí napájení nutno počkat, až se detekují senzory vzdálenosti. Přípravenost senzorů je indikována zablikáním zelené ledky na zařízení Teraranger Tower. Zelená ledka zabliká tolikrát, kolik senzorů vzdálenosti bylo detekováno. Pokud



Obrázek 4.11: Fáze navigačního algoritmu

zabliká méně než čtyřikrát, je třeba vypnout a znovu zapnout napájení, protože s méně než čtyřmi senzory vzdálenosti nebude algoritmus pracovat správně.

#### 4.7.2 Sběr dat za klidových podmínek

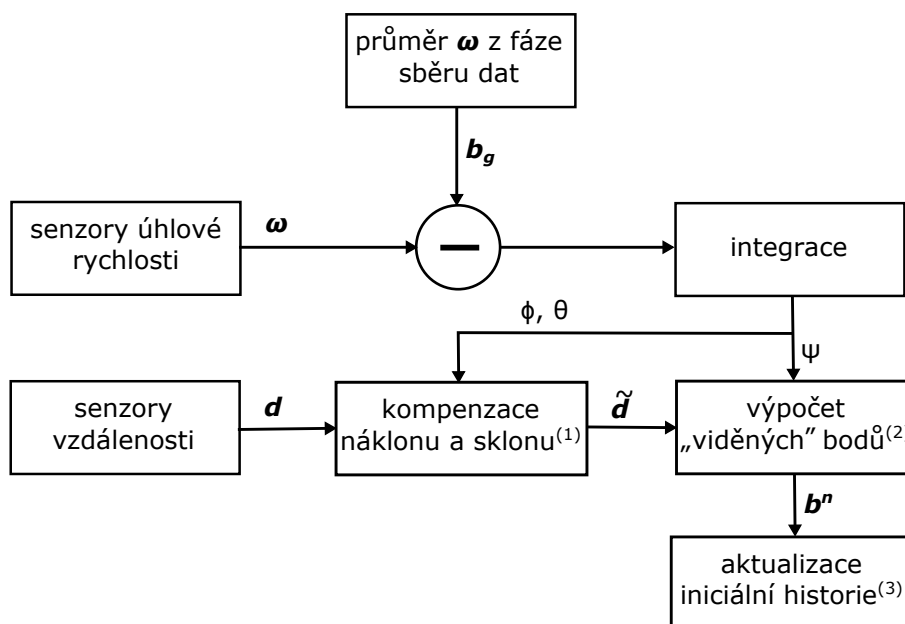
Vstup do fáze sběru dat za klidových podmínek (dále jen sběr dat) je na MiniINS signalizován zhasnutím oranžové ledky. Po celou dobu sběru dat musí být navigovaný objekt v klidu. V této fázi se neděje nic jiného, než že se průměrují data ze senzorů. Průměr úhlových rychlostí se později použije jako iniciální odhad biasu úhlových rychlostí a průměr specifických sil se využije při výpočtu iniciálních hodnot náklonu (*roll*) a sklonu (*pitch*) pro EKF. Průměrný tlak z fáze sběru dat je dále v barometrických rovnicích používán jako tlak referenční.

Sběr dat je automaticky ukončeno, jakmile je detekováno porušení podmínky nehybnosti. Podmínka nehybnosti je považována za porušenou, když některá ze složek vektoru  $|\boldsymbol{\omega}_{ib} - \bar{\boldsymbol{\omega}}_{ib}|$  je větší než  $0,5 \text{ rad/s}$ , nebo když některá ze složek vektoru  $|\mathbf{f}_{ib} - \bar{\mathbf{f}}_{ib}|$  je větší než  $0,1 \text{ m}\cdot\text{s}^{-2}$ . Přičemž  $\bar{\boldsymbol{\omega}}_{ib}$  značí průměr úhlových rychlostí a  $\bar{\mathbf{f}}_{ib}$  značí průměr specifických sil od vstupu algoritmu do fáze sběru dat.

#### 4.7.3 Iniciální mapování

Iniciální mapování slouží k naplnění historií bodů „viděných“ senzory vzdálenosti tak, jak je popsáno v části 4.6.3. Průběh je znázorněn na obrázku 4.12. Tato fáze je ukončena, pokud všechny 4 historie obsahují 15 bodů.

Při provádění iniciálního mapování se provádí pouze rotace navigovaného objektu. Pozice je považována za konstantní. Pro potřeby výpočtu „viděných“ bodů podle (4.11) je prováděna integrace změřených úhlových rychlostí do Eulerových úhlů. Integrované úhlové rychlosti jsou kompenzované o bias vzatý jako průměrná hodnota úhlových rychlostí

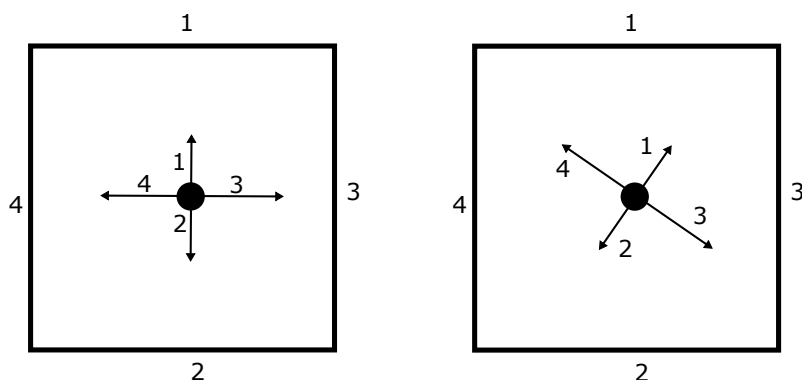


Obrázek 4.12: Průběh iniciálního mapování <sup>(1)</sup> podle rovnice (4.1); <sup>(2)</sup> podle rovnice (4.11); <sup>(3)</sup> podle algoritmu 3)

z fáze sběru dat. Bias není v této fázi nijak dále odhadován. Proto musí mapovací rotace proběhnout co nejrychleji, aby mohl být bias považován za konstantní.

Dále je třeba uvést, že mapovací rotací se nemyslí rotace o celých 360 stupňů. Cílem této rotace je pouze naplnit historii bodů 15 vzorky. K tomu může stačit rotace pouze o 10 stupňů a zpět (v závislosti na vzdálenosti od překážky). Při rotaci je třeba dbát na to, aby šly následně historiemi bodů proložit alespoň 2 validní nerovnoběžné přímky. V opačném případě by odhady EKF po spuštění divergovaly od skutečnosti.

Příklady polohy před zahájením iniciálního mapování jsou na obrázku 4.13. Vlevo je ideální poloha. Iniciální rotace může probíhat po i proti směru hodinových ručiček. Pokud se při rotaci nepřejde paprskem senzoru vzdálenosti přes roh, po ukončení mapovací rotace půjde všemi čtyřmi historiemi bodů proložit validní přímka. Vpravo už není situace tak optimální. Pokud se provede rotace po směru hodinových ručiček, hrozí, že v historiích



Obrázek 4.13: Příklad možných poloh před iniciálním mapováním (vlevo - dobře, vpravo - špatně)

bodů bude navzorkovaný roh místnosti. Příмка proložená takovou historií by byla klasifikována jako nevalidní, což by způsobilo odstranění některých vzdáleností z vektoru měření a v nejhorším případě i divergenci odhadů od skutečnosti.

#### 4.7.4 Inicializace EKF

V této fázi algoritmus naplní iniciální hodnoty do matic a vektorů potřebných pro běh Kalmanova filtru. Inicialní stavový vektor

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} \mathbf{0}^{1 \times 6} & \hat{\boldsymbol{\epsilon}}_0 & \mathbf{b}_{g,0} & \mathbf{0}^{1 \times 3} & \hat{\boldsymbol{\rho}}_0^t \end{bmatrix}^T,$$

přičemž

- $\hat{\boldsymbol{\epsilon}}_0$  je vektor Eulerových úhlů naintegrovaných při rotaci ve fázi iniciálního mapování,
- $\mathbf{b}_{g,0}$  je vektor průměrů úhlových rychlostí získaný ve fázi sběru dat,
- $\hat{\boldsymbol{\rho}}_0^t$  se rovná vektoru  $\boldsymbol{\rho}^n$  spočteného z iniciální mapy jak je popsáno v části 4.6.1.

Hodnoty kovariančních matic  $\mathbf{R}$ ,  $\mathbf{Q}$  a  $\mathbf{P}_0$  jsou v příloze B. Tyto hodnoty byly zvoleny po dlouhé fázi experimentování a ladění EKF.

#### 4.7.5 Běh EKF

Tok dat mezi algoritmy a výpočty výše popsanými je znázorněn na obrázku 4.14. Poslední, co je potřeba upřesnit, je modifikace vektoru měření. Výstup algoritmu 2 je množina překážek  $S_{invalid}$ , které jsou podezřelé z toho, že jejich poloha neodpovídá proložené přímce. Na obrázku 4.14 vstupuje tato množina jak do EKF, tak do výpočtu metodou nejmenších čtverců (LSE - *Least Squares Estimator*). V algoritmu 5 je popsáno, jak tato množina ovlivňuje výpočet LSE a konstrukci vektoru měření.

---

#### Algoritmus 5 Konstrukce vektoru měření

---

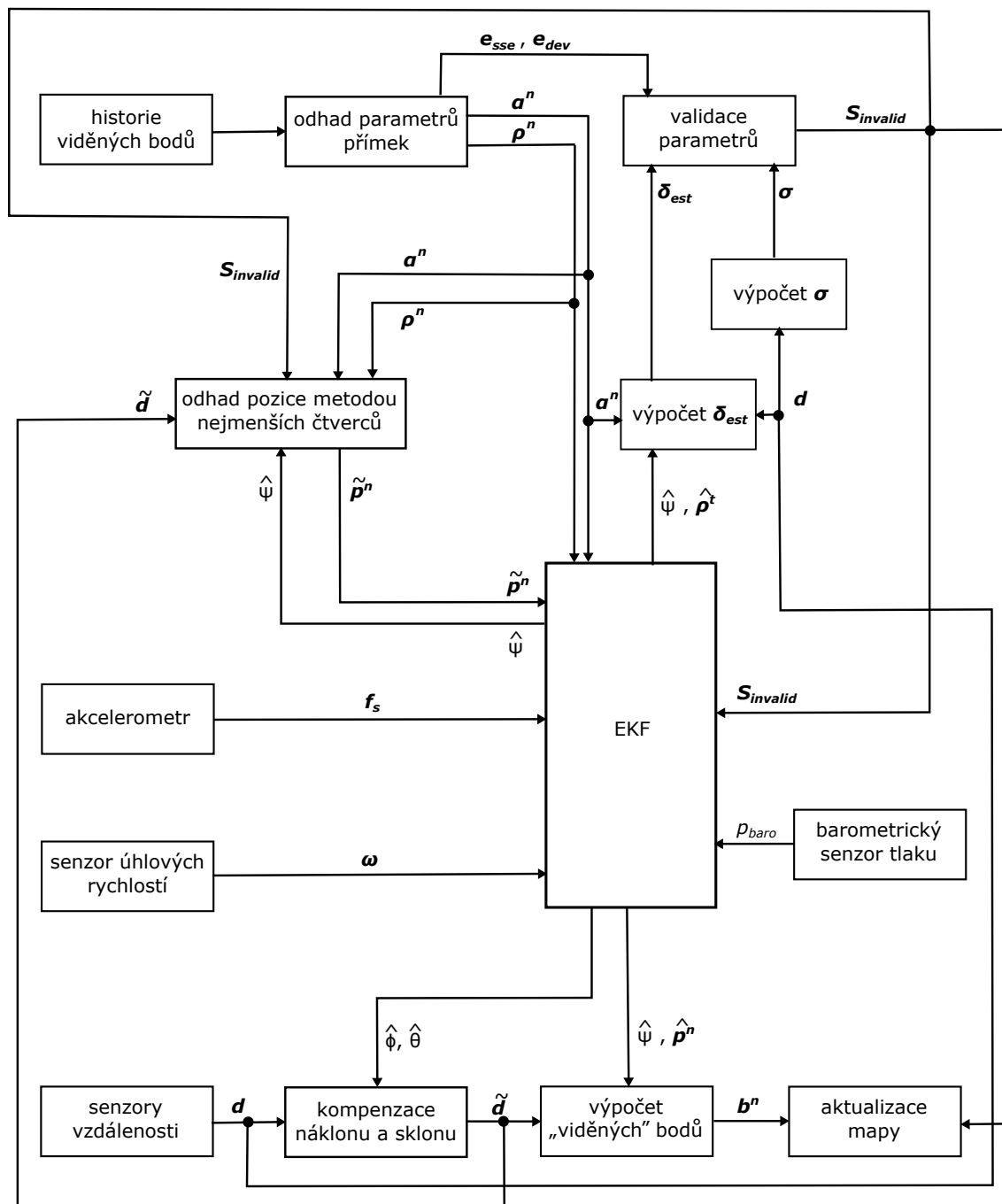
**Vstup:** množina podezřelých překážek  $S_{invalid}$ , matice  $\mathbf{A}$  a  $\mathbf{b}$  pro LSE (vysvětleno v části 4.5.2), kompenzované vzdálenosti  $\tilde{\mathbf{d}}$  a barometrický tlak  $p_{baro}$

**Výstup:** vektor měření  $\mathbf{y}_k$

- 1: **for**  $i \in S_{invalid}$  **do**
  - 2:   odstraň řádek  $i$  z matice  $\mathbf{A}$
  - 3:   odstraň řádek  $i$  z matice  $\mathbf{b}$
  - 4:   odstraň prvek  $\tilde{d}_i$  z vektoru  $\tilde{\mathbf{d}}$
  - 5: **end for**
  - 6:  $\tilde{\mathbf{p}}^n \leftarrow$  řešení úlohy (4.8) {*odhad pozice pomocí metody nejmenších čtverců*} ,
  - 7:  $\mathbf{y}_k = \begin{bmatrix} \tilde{\mathbf{p}}^n \\ \tilde{\mathbf{d}} \\ p_{baro} \end{bmatrix}$
- 

## 4.8 Omezení a možná rozšíření algoritmu

1. Vertikální kanál je korigován pouze barometrem. Jako referenční tlak je brán ve výpočtech barometrických rovnic napevno průměr z fáze sběru dat. Atmosferický tlak



Obrázek 4.14: Tok dat v algoritmu



ale není konstantní a tím, že se s ním jako s konstantním počítá se do výpočtu vnáší chyba. Při případném rozšiřování systému by bylo dobré použít další distanční senzor, který by směřoval dolů a který by umožnil referenční tlak odhadovat pomocí EKF.

2. Jak už bylo uvedeno v předpokladech: pro správnou funkci algoritmu musí v okolním prostředí existovat rovné vertikální plochy, což bývá ve standardní budově (stěny, skříně...) splněno.
3. Současná podoba algoritmu předpokládá, že tyto vertikální plochy jsou kolmé na vodorovnou rovinu. Na tomto předpokladu je postavena kompenzace změřených vzdáleností o náklon a sklon.
4. Dalším omezením je, že algoritmus v této podobě předpokládá distanční data na 100 Hz. Když nejsou na této frekvenci data poskytována, jsou dopočítána lineární interpolací. Pokud je vzdálenost senzoru od překážky větší než 5 metrů, frekvence měření klesá pod 60 Hz (viz tabulku 3.7) a roste podíl interpolovaných vzdáleností vstupujících do algoritmu, což ovlivňuje jeho přesnost. Dále pokud je vzdálenost senzoru od překážky větší než 13 m, může se stát, že data přestanou být poskytována zcela - to současný algoritmus neošetřuje.
5. Ke zrobustnění algoritmu by mohlo pomoci uchování dlouhodobé mapy a větší využívání informace o tom, co bylo již dříve „viděno“.

## 4.9 Implementace

Výše popsaný algoritmus byl vyvíjen a laděn ve vývojovém prostředí Matlab. Následně byl přepsán do jazyka C a nakonec byl tento zdrojový C kód zakomponován do kódu mikrokontroléru.

### Matlab

Matlab kódy jsou na přiloženém CD ve složce `./Matlab`. Jako základ byla využita šablona EKF používaná v Laboratoři leteckých informačních systémů Katedry měření (LIS). Pro kalibraci akcelerometru a senzoru úhlové rychlosti byly využity volně dostupné kódy<sup>3</sup> k článku [7].

### Implementace algoritmu v jazyce C

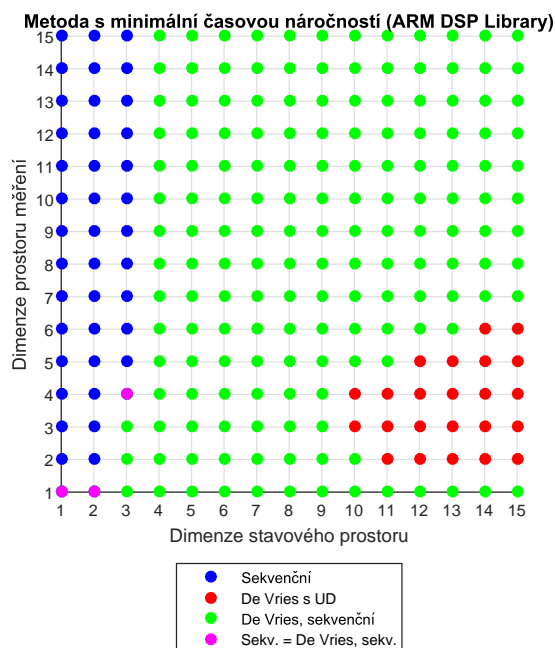
C kódy algoritmu jsou na přiloženém CD ve složce `./EKF_C`. Při přepisu algoritmu do jazyka C byla rovněž využita jako základ EKF šablona z LIS, dříve vytvořena autorem této DP. Dále bylo u některých částí algoritmu využito programu Matlab Coder pro vygenerování C kódů z Matlab kódů. Takto vygenerované kódy jsou na CD ve složce `./EKF_C/matlab_gen`.

Použitá šablona EKF umožňuje výběr z 5 možných metod pro provedení korekčního kroku EKF. O různých postupech provedení korekčního kroku se lze dočíst v [15]. V [3]

<sup>3</sup>[https://github.com/labliscvut/imu\\_calibration](https://github.com/labliscvut/imu_calibration)

byla porovnána efektivita těchto postupů na procesoru ARM Cortex-M4, který je využit i v realizovaném systému jako výpočetní jednotka.

Na obrázku 4.15 jsou znázorněny metody korekce s nejmenší časovou náročností pro danou dimenzi stavového prostoru a prostoru měření při použití ARM DSP knihovny pro maticové výpočty. Ve výše navrženém navigačním algoritmu je dimenze stavového prostoru 19 a dimenze vektoru měření se pohybuje mezi 1 až 7, jak bylo popsáno v algoritmu 5. Pro tyto dimenze není na obrázku znázorněna nejefektivnější metoda korekce. Nicméně z trendu růstu barevných oblastí lze vyčíst, že pro hledané dimenze patří k nejrychlejším metoda znázorněná červeně, což je De Vriesova metoda s UD dekompozicí (viz [15]), a proto byla tato metoda nastavena pro korekční krok EKF.



Obrázek 4.15: Metody korekce s minimální časovou náročností při použití ARM DSP knihovny, převzato z [3]

Ve složce `./EKF_C` na příloženém CD je uložen projekt volně dostupného vývojového prostředí `Code::Blocks`, ve kterém vznikl zdrojový kód algoritmu v jazyce C. Dále je zde uložena dokumentace vygenerovaná programem `Doxygen` z komentovaného zdrojového kódu.

## Mikrokontrolér

Podstatná část kódu pro mikrokontrolér byla převzata ze stávajících zdrojových kódů pro MiniINS. Mezi takto převzaté, nebo mírně upravené části patří inicializace HW periferií, vyčítání dat z IMU a senzoru barometrického tlaku. Nově vznikl kód pro vyčítání dat ze zařízení Teraranger Tower. Zásadní změnou pak samozřejmě bylo zakomponování výše navrženého navigačního algoritmu.

Pro maticové výpočty a FIR filtraci je využívána ARM DSP knihovna<sup>4</sup>, která maximálně využívá SIMD a DSP instrukce poskytované procesorem ARM Cortex-M4. Naprostá většina výpočtů v plovoucí řádové čárce probíhá v jednoduché přesnosti, aby byla maximálně využita jednotka FPU. Veškeré zdrojové kódy pro mikrokontrolér jsou uloženy na příloženém CD ve složce `./Microcontroller`. V této složce je uložen i projekt volně dostupného vývojového prostředí EmBitz, které bylo použito při psaní kódu pro mikrokontrolér.

---

<sup>4</sup><http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>

## 5 Testování

---

Testování vytvořeného navigačního systému probíhalo v Laboratoři bezpilotních prostředků Katedry měření. Tato laboratoř poskytla dostatek prostoru pro realizaci různých testovacích scénářů. Některé scénáře předpokládaly idealizované okolní prostředí, které bylo vytvořeno pomocí stolů položených na bok. Další scénáře pak operovaly s okolím s vloženými překážkami, nebo s okolním prostředím zcela bez úprav, včetně regálů, otevřených dveří, stojících stolů a židlí.

Bylo definováno 7 testovacích scénářů:

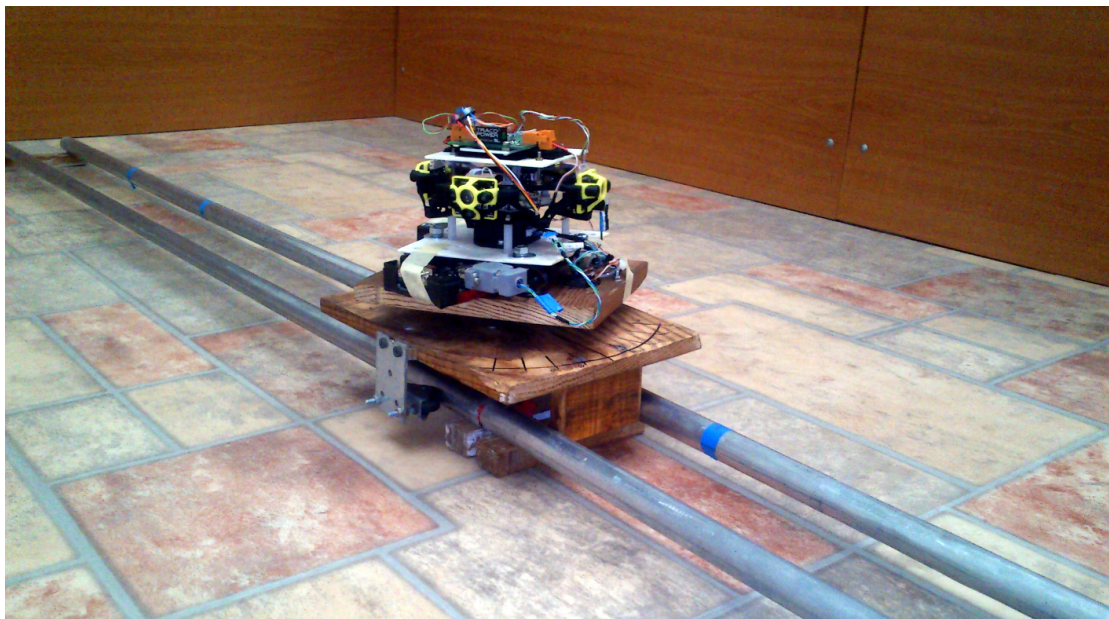
1. idealizované prostředí bez překážek, žádný přechod přes rohy, s rotací, na testovací platformě, pohyb kolmo ke stěně,
2. prostředí s vloženou překážkou, přechod přes překážku, bez rotace, na testovací platformě, pohyb kolmo ke stěně,
3. prostředí bez překážek, přechod přes roh, bez rotace, na testovací platformě, pohyb šikmo ke stěně,
4. prostředí bez překážek, přechod přes 2 rohy, s rotací, na testovací platformě, pohyb šikmo ke stěně,
5. prostředí s vloženou překážkou, přechod přes překážku, s rotací, na testovací platformě, pohyb kolmo ke stěně,
6. prostředí bez překážek, rotace 360 stupňů, na testovací platformě,
7. neupravené prostředí laboratoře, čtvercová trajektorie, rotace v rozích, mimo testovací platformu.

Každý ze scénářů 1 až 6 měl za cíl otestovat typické situace, ve kterých se může navigovaný objekt vyskytnout, a pomoci při návrhu navigačního algoritmu. Při těchto testech se neodchylovala odhadnutá trajektorie od skutečné o více než 10 cm. Scénář č. 7 měl za cíl experimentálně otestovat chování systému v neupravovaném prostředí při pohybu složeném z translací i rotací. V této kapitole je analyzován výstup testu č. 7, který je nejvíce komplexní. Výstupy všech scénářů lze nalézt na příloženém CD ve složce `./Matlab/output`.

### 5.1 Testovací platforma

Některé testy probíhaly na platformě, pro tyto účely speciálně vytvořené. Na obrázku 5.1 je fotografie navigačního systému uchyceného k testovací platformě společně s baterií a záznamníkem dat. Testovací platforma se skládá z kolejničky, vozíku a plošiny a umožňuje pohyb navigovaného objektu v jedné pevně dané ose a rotaci v horizontální rovině. Díky této platformě lze určit referenční trajektorii a orientačně i úhel  $\psi$ .

Záběry některých testovacích scénářů využívajících testovací platformu v idealizovaném prostředí jsou na obrázku 5.2. Vlevo je konkrétně vyfocena poloha testovací platformy při scénáři č. 1 a vpravo při scénáři č. 4.



Obrázek 5.1: Vytvořený navigační systém upevněný na testovací platformě

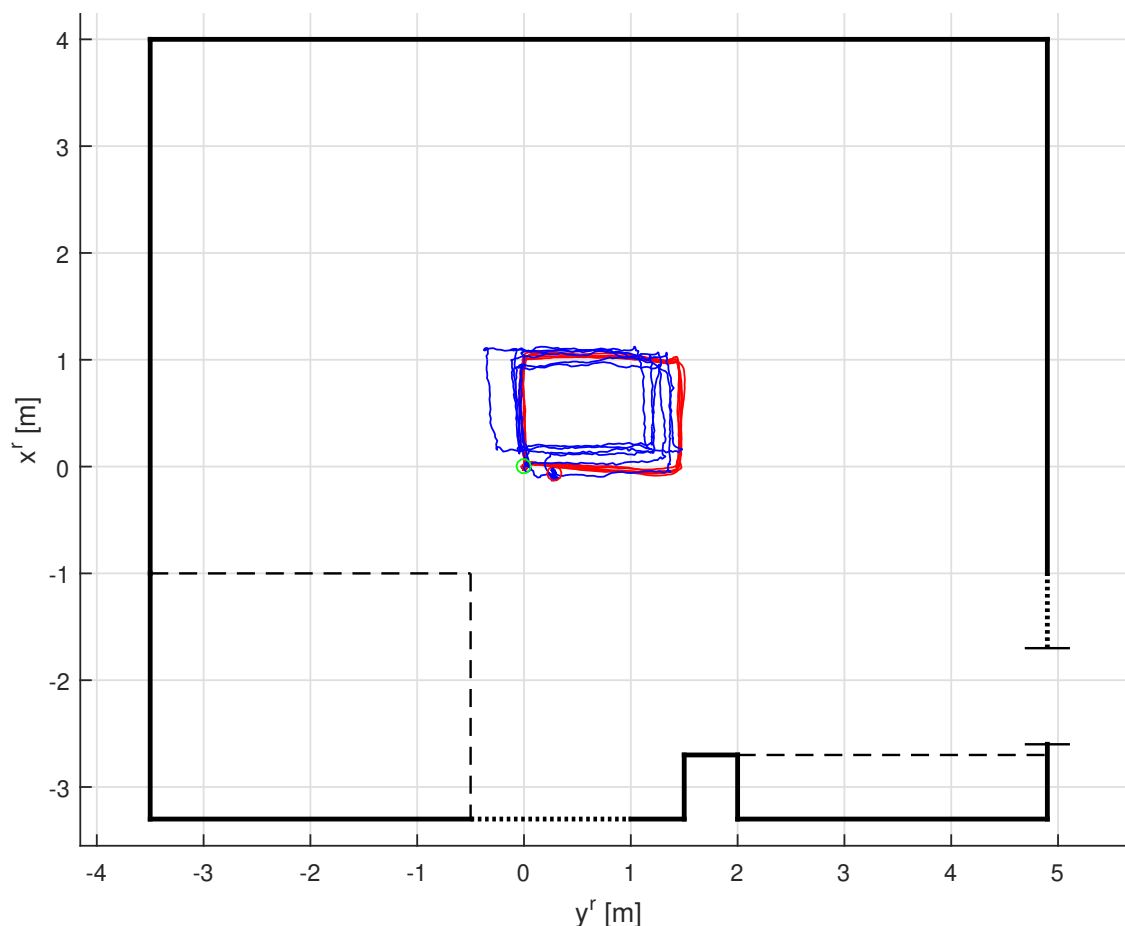


Obrázek 5.2: Záběry některých testovacích scénářů využívajících testovací platformu v idealizovaném prostředí

## 5.2 Závěrečné experimentální ověření

Scénář č. 7, definovaný v úvodu kapitoly č. 5, měl za cíl otestovat navigační systém v neupraveném prostředí Laboratoře bezpilotních prostředků. Trajektorie navigačního systému opisovala obdélník o rozměrech  $1 \times 1,43$  m. Obdélník byl při pohybu opsán celkem 6krát. Pohyb trval necelé 2 minuty. V rozích opisovaného obdélníku vždy došlo k rotaci o 90 stupňů. To znamená, že celkem 24krát došlo během pohybu díky rotaci k invalidaci všech odhadnutých přímek téměř najednou (kvůli navzorkování rohů místností do historií všech senzorů vzdálenosti), což mělo za následek nekorigování odhadu horizontální pozice. K obnově korekcí horizontální pozice pak došlo po 0,7 až 0,9 vteřinách. Po tuto dobu se navigační systém spoléhal pouze na inerciální senzory.

Na obrázku 5.3 je černě znázorněn obrys místnosti, tečkovaně jsou označeny plochy s občasnými nerovnostmi a přerušovanou čarou jsou vymezeny zastavěné oblasti (oblasti,



Obrázek 5.3: Testovací scénář č. 7 (překážky černě, skutečná trajektorie červeně, odhadnutá trajektorie modře)

kde se nacházejí regály a na sebe naskládané stoly a židle). Červená na obrázku označuje referenční trajektorii a modrá její odhad. Zelený kroužek značí místo začátku a konce pohybu a červený kroužek zase odhadnutou cílovou pozici. Vše je vykresleno v souřadnicovém systému místnosti - osa  $x^f$  tohoto systému je kolmá na stěnu, kterou při iniciální rotaci mapoval senzor č. 1.

### 5.2.1 Výpočet referenční pozice

K určení referenční pozice bylo využito technik počítačového vidění. V článku [31] je popsána metoda pro sledování pozice zeleného míče ve videu. Dále je v článku poskytnut kód v Pythonu využívající knihovnu OpenCV, který tuto metodu realizuje. Testovací experiment byl proveden tak, aby bylo následně možné pomocí tohoto kódu získat referenční pozici. Celý experiment byl nahrán shora na kameru a na navigační systém byl upevněn zelený kruh. Na obrázku 5.4 vlevo je vidět navigační systém se zelenou značkou a konstrukce pro upevnění kamery. Na obrázku 5.4 vpravo je snímek z nahraného videa během zpracovávání algoritmem z [31]. Žlutá kružnice uzavírá co nejtěsněji detekovanou zelenou oblast a červená linka značí spočítanou trajektorii.

Kód z [31] byl nakonec využit takřka bez úprav. Jediným zásahem bylo připsání několika řádek zajišťujících ukládání pozice do souboru (kód je na přiloženém CD ve složce



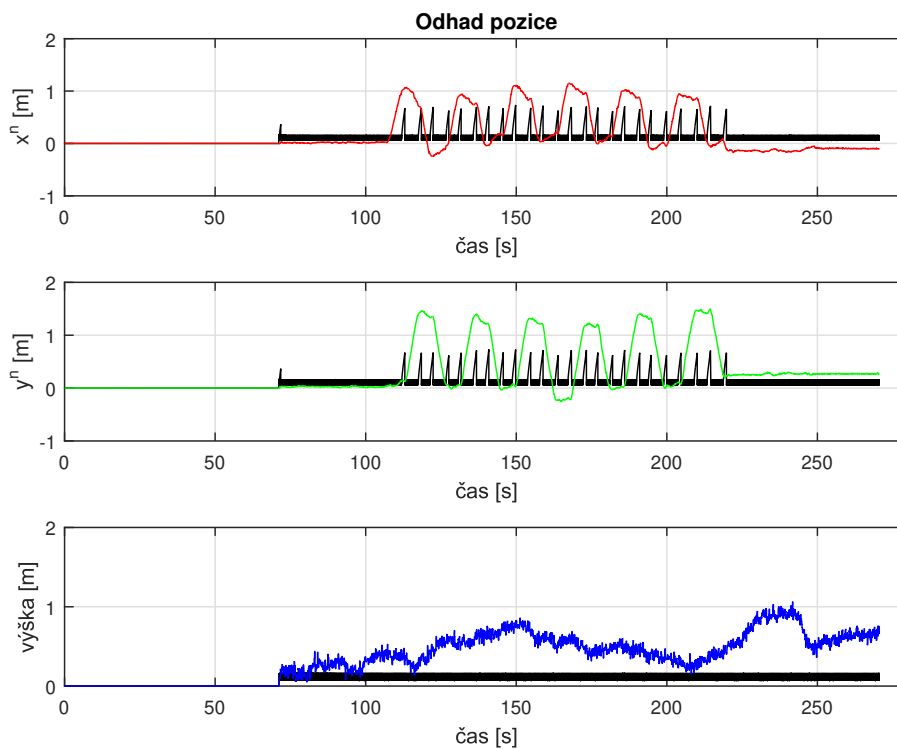
Obrázek 5.4: Vlevo: navigační systém se zelenou značkou a konstrukce pro upevnění kamery; vpravo: snímek z nahraného videa během zpracování algoritmem z [31]

./CV\_Python). Uložené pozice byly poté přepočítány do souřadnicového systému místnosti (definován výše v této kapitole). Kód pro toto přepočítání a pro porovnání referenční a odhadnuté pozice je na příloženém CD ve složce ./Matlab/test7\_evaluation.

## 5.2.2 Analýza výstupů

### Pozice

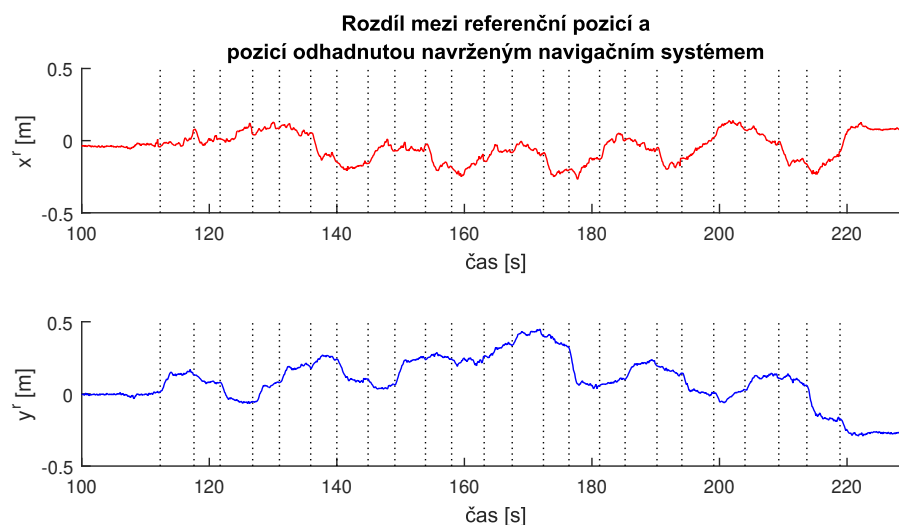
Na obrázku 5.5 je znázorněna odhadnutá pozice v navigačním souřadnicovém systému a její směrodatná odchylka (odmocnina příslušného diagonálního prvku matice  $P_k$ ).



Obrázek 5.5: Pozice: odhad (barevně) a směrodatná odchylka (černě)

Odhad výšky je korigován pouze senzorem atmosferického tlaku, kdy jako referenční tlak je vzat průměr tlaku z fáze sběru dat. Jelikož je takto vzatý referenční tlak konstantní, průběh odhadu výšky kopíruje průběh měřeného tlaku a do odhadu výšky je tak zanášena chyba senzoru atmosferického tlaku. Při experimentu byl navigační systém zvednut z výšky 0 do výšky cca 0,3 m, ve které byl proveden pohyb a následně opět položen do výšky 0. Jak je vidět z obrázku 5.5 dole, odhad tomu příliš neodpovídá, což je způsobeno vnesením chyby z měření tlaku.

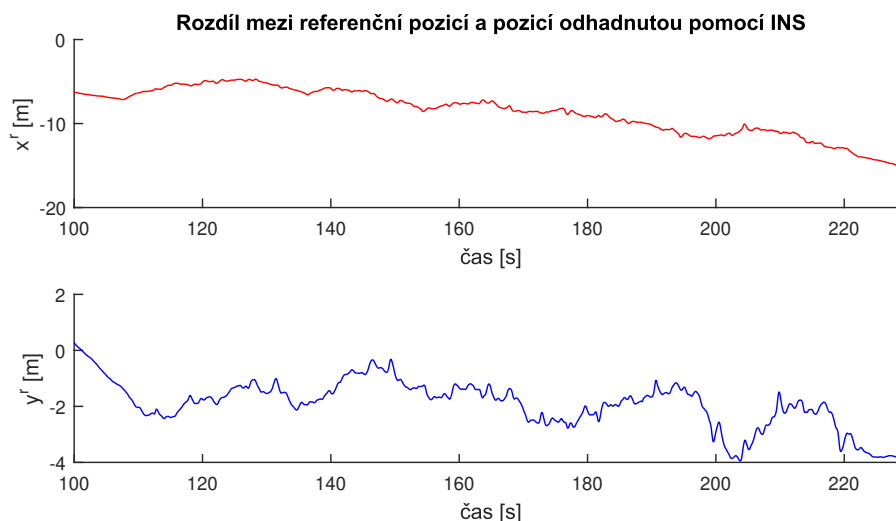
Na obrázku 5.6 je rozdíl referenční pozice získané pomocí metod počítačového vidění a pozice odhadnuté pomocí EKF. Svislými tečkovanými čarami jsou označeny časové okamžiky, ve kterých přibližně došlo k rotaci o 90 stupňů a následně k současné invalidaci parametrů přímek odhadnutých z historií všech sensorů vzdálenosti. Tato současná invalidace všech odhadnutých přímek vznikne kvůli navzorkování rohů místnosti do historií sensorů a má za následek vypnutí korekcí pozice v horizontální rovině na přibližně 1 sekundu. Po tuto dobu je navigační systém odkázán pouze na inerciální senzory. Následně dojde k doplnění historií bodů nově viděnými body a opětovnému zapnutí korekcí. Z obrázku 5.6 je vidět, že rozdíl odhadů od reference často výrazně roste chvíli po rotaci (spoléhání se pouze na inerciální senzory) a následně se výrazný růst zastaví (po obnovení korekcí), což je očekávané chování. Ve chvílích, kdy systém spoléhá pouze na inerciální senzory roste také nejistota odhadů pozice (viz obrázek 5.5).



Obrázek 5.6: Rozdíl mezi referenční pozicí a pozicí odhadnutou navrženým systémem

Na obrázku 5.7 je pro srovnání zobrazeno, jaký by byl rozdíl mezi referenční pozicí a pozicí odhadnutou nad stejnými daty inerciálním navigačním systémem s náklonem a sklonem korigovaným akcelerometrem. Zatímco chyba v pozici odhadnuté pomocí navrženého systému při tomto experimentu nepřesáhla 50 cm, tak chyba v pozici odhadnuté pouze pomocí INS by vzrostla v jedné horizontální ose až k 18 metrům. Navíc by tato chyba nadále rostla i po ukončení pohybu. Kód pro výpočet INS odhadů je na příloženém CD ve složce `./Matlab/test7_evaluation`.

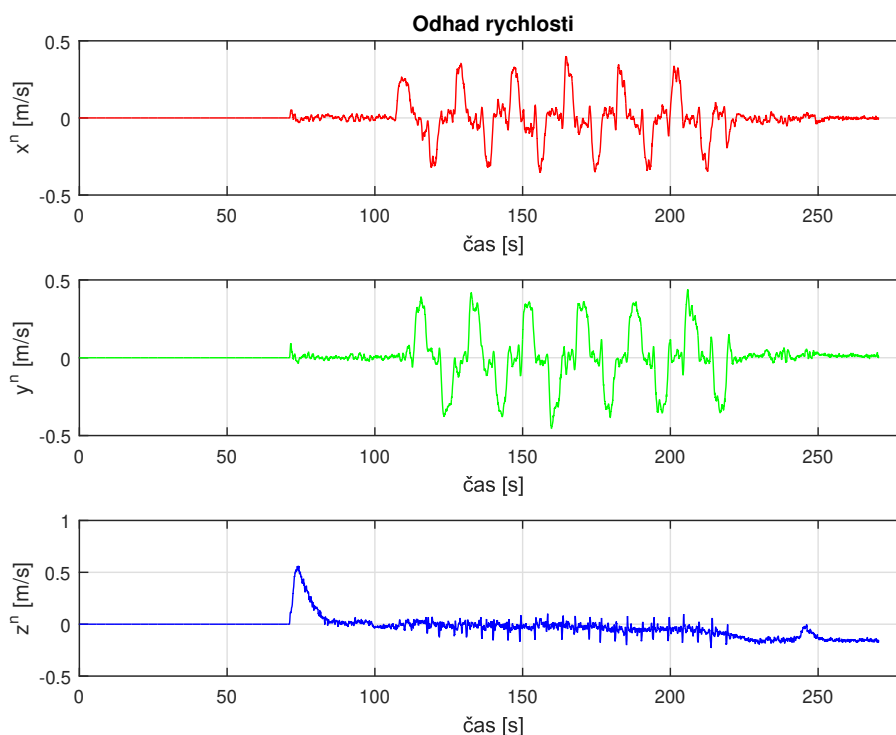




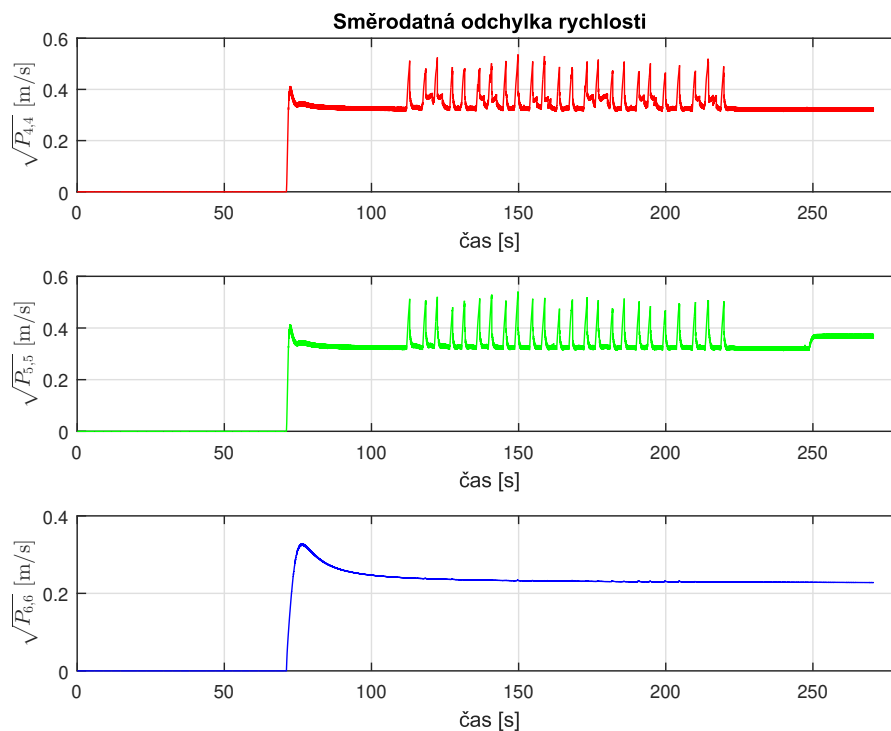
Obrázek 5.7: Rozdíl mezi referenční pozicí a pozicí odhadnutou pomocí INS s náklonem a sklonem korigovaným akcelerometrem

### Rychlost

Na obrázku 5.8 jsou odhady rychlosti v navigačním souřadnicovém systému a na obrázku 5.9 její směrodatné odchylky. Stejně jako u pozice je vidět kolísání směrodatné odchylky ve chvílích spoléhání se na inerciální senzory.



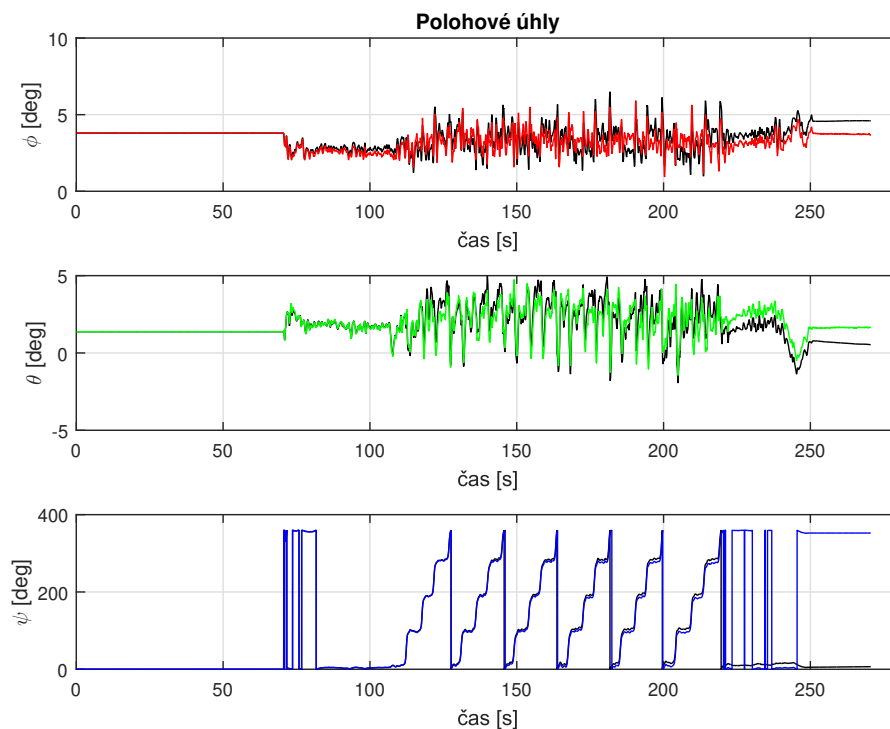
Obrázek 5.8: Odhad rychlosti



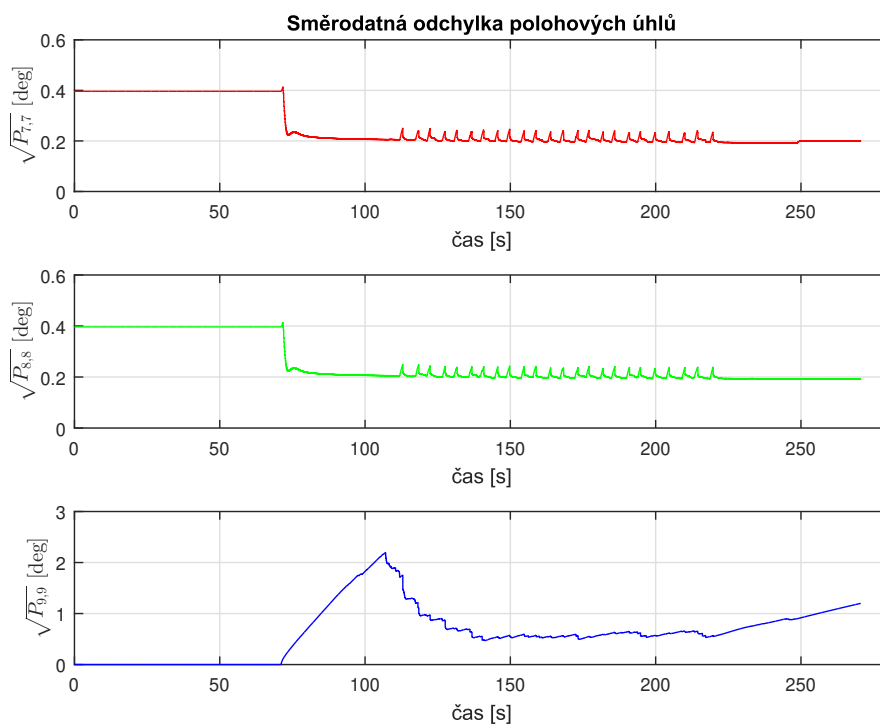
Obrázek 5.9: Směrodatná odchylka rychlosti

### Polohové úhly

Odhady polohových úhlů jsou na obrázku 5.10 a jejich směrodatné odchylky na obrázku 5.11. V rámci experimentu byla snaha o to, aby iniciální poloha navigačního systému byla



Obrázek 5.10: Odhad polohových úhlů (barevně) a polohové úhly získané integrací úhlových rychlostí kompenzovaných o iniciální odhad biasu senzoru úhlových rychlostí (černě)



Obrázek 5.11: Směrodatná odchylka polohových úhlů

shodná s konečnou polohou. V ideálním případě by tudíž měly být úhly  $\phi$ ,  $\theta$  a  $\psi$  na začátku experimentu a na jeho konci totožné. Pro náklon a sklon toto v podstatě platí. Rozdíl úhlu  $\phi$  ze začátku experimentu a z jeho konce je přibližně  $0,05^\circ$  a rozdíl úhlu  $\theta$   $-0,27^\circ$ . Rozdíl úhlu  $\psi$  je  $-7,6^\circ$ . Tato chyba je zčásti dána chybou v odhadu a zčásti ne dokonale přesným položením navigovaného objektu do původní polohy.

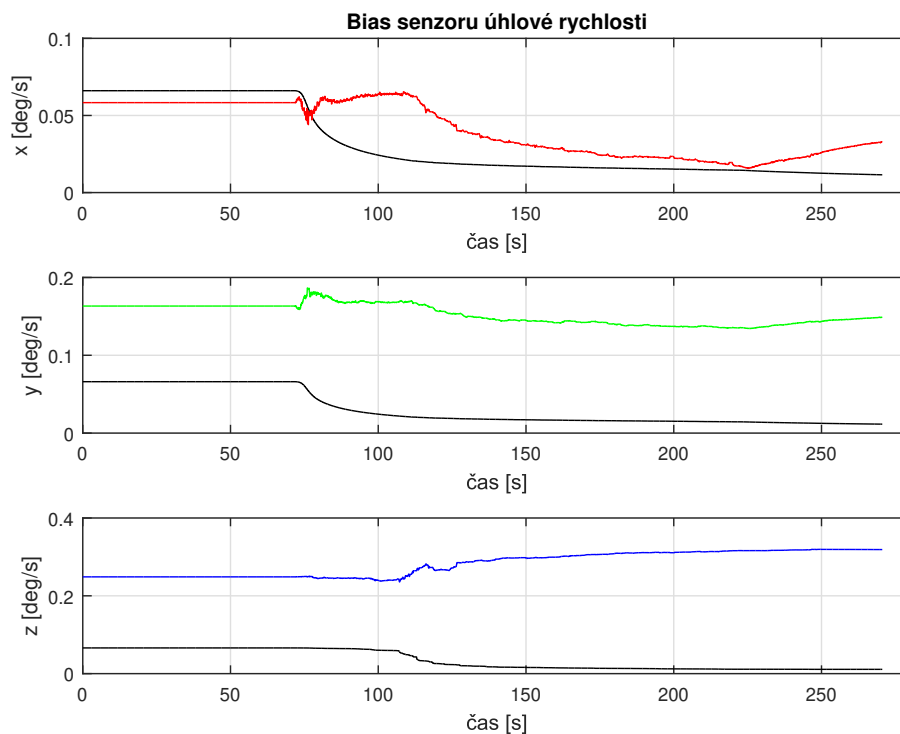
Na obrázku 5.12 je snímek iniciální polohy a finální polohy získaný z natočeného videa (kamera byla mezi snímky v absolutním klidu). Na první pohled není vidět rozdíl. Modře je na obrázku znázorněna spojnice mezi dvěma šrouby držícími konstrukci pro nesení navigačního systému. Vypočtením úhlu mezi těmito spojnicemi bylo zjištěno, že rozdíl úhlu  $\psi$  daný nepřesným položením navigovaného objektu do původní polohy je cca  $-2^\circ$ . Rozdíl mezi iniciálním a finálním odhadem  $\psi$  je tedy přibližně  $-5,6^\circ$ .



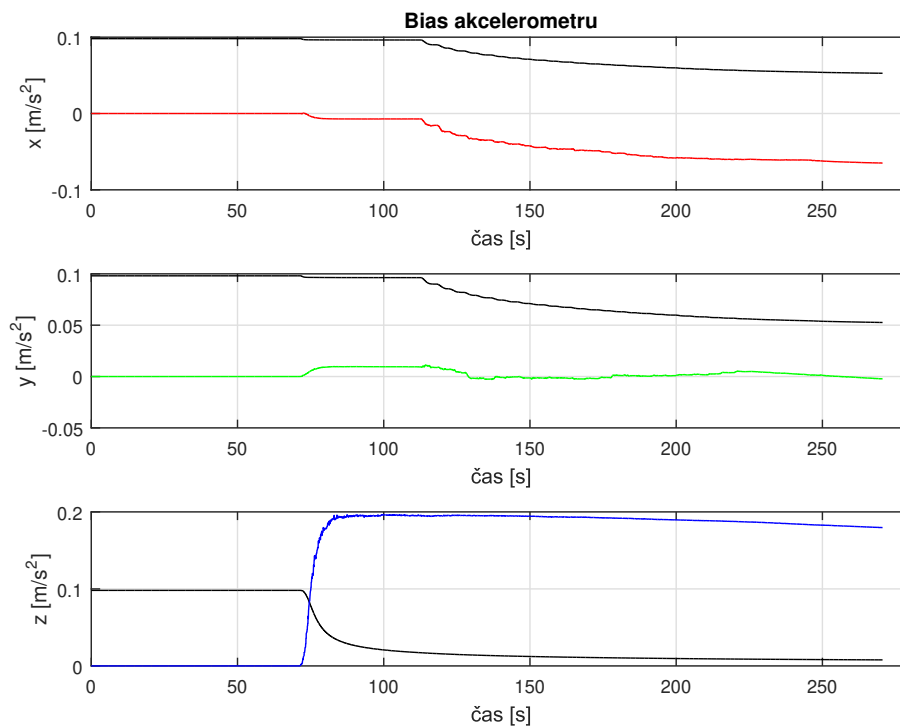
Obrázek 5.12: Vlevo: iniciální poloha, vpravo: finální poloha

## Biasy inerciálních senzorů

Odhady biasů inerciálních senzorů jsou na obrázcích 5.13 a 5.14 společně s jejich směrodatnými odchylkami.



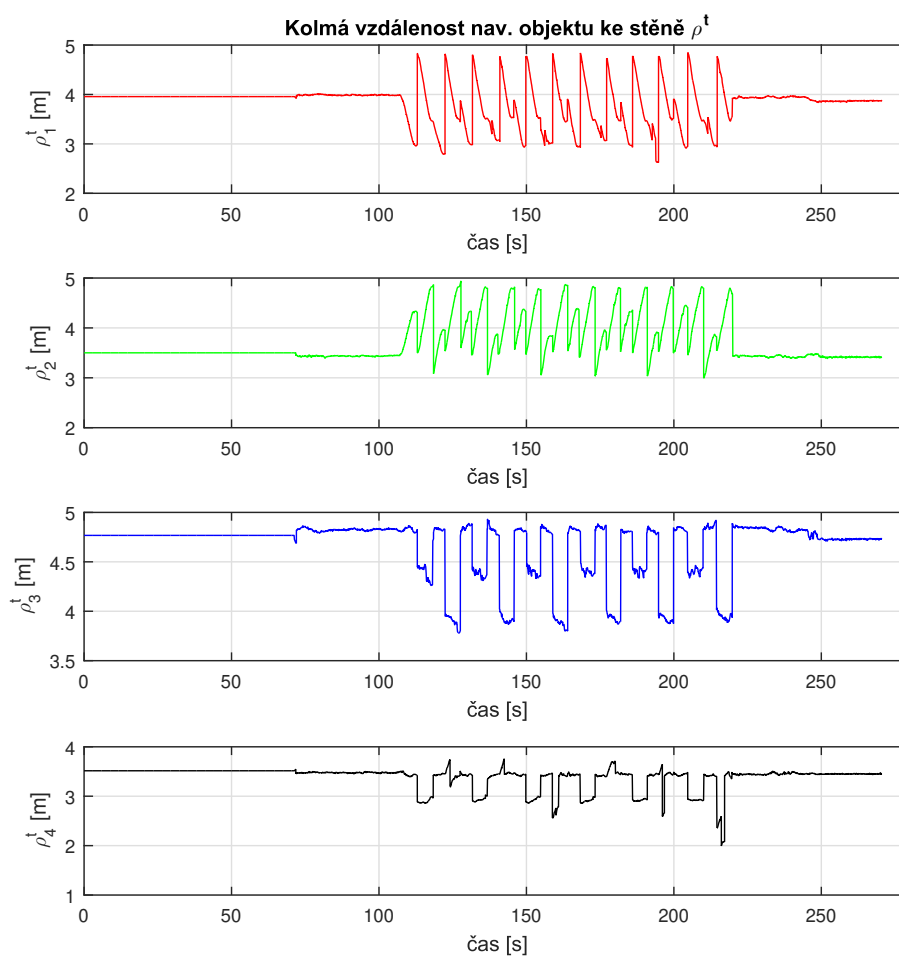
Obrázek 5.13: Bias senzoru úhlové rychlosti: odhad (barevně) a směrodatná odchylka (černě)



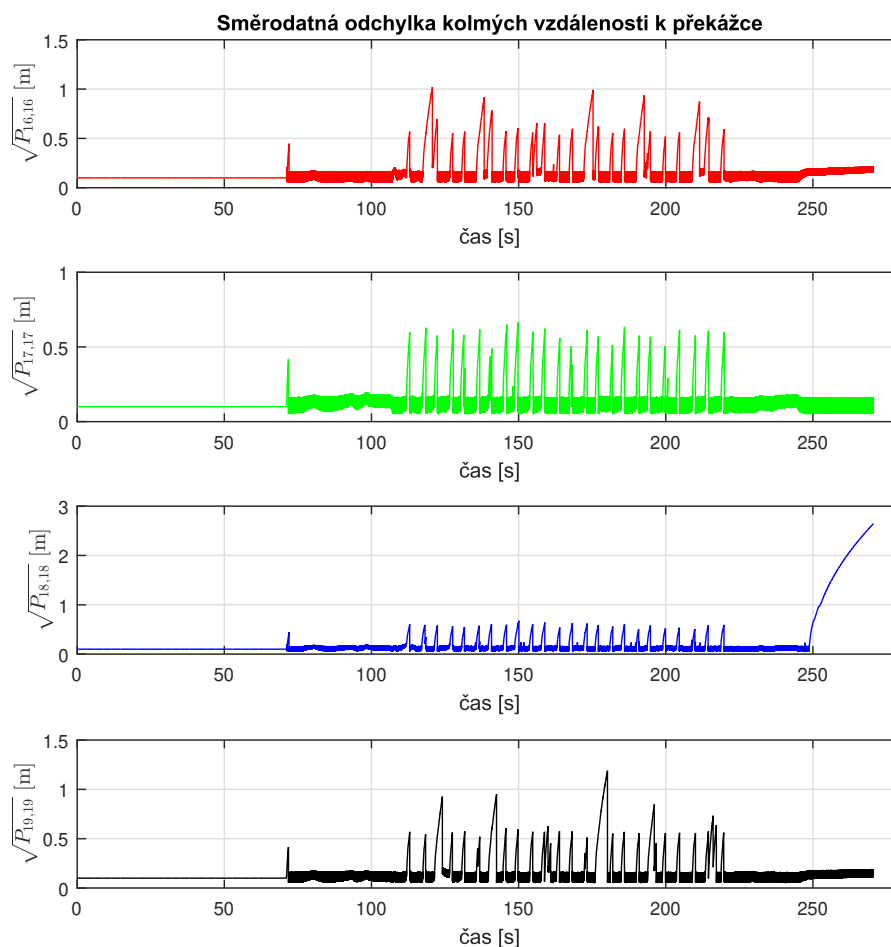
Obrázek 5.14: Bias akcelerometru: odhad (barevně) a směrodatná odchylka (černě)

### Kolmé vzdálenosti k překážce

Odhady kolmých vzdáleností k překážce jsou na obrázku 5.15 a jejich směrodatné odchylky jsou na obrázku 5.16. Směrodatná odchylka roste ve chvílích, kdy je z vektoru měření odstraněna kompenzovaná vzdálenost pro daný senzor  $i$ . U směrodatné odchylky kolmé vzdálenosti k překážce č. 3 (modrá linka) je vidět růst na konci experimentu. To je způsobeno tím, že při pokládání navigačního systému do původní polohy navzorkoval distanční senzor č. 3 kus podlahy namísto stěny. Proto došlo k invalidaci jeho historie „viděných“ bodů a odstranění kompenzované vzdálenosti č. 3 z vektoru měření. To má za následek růst nejistoty v odhadu  $\rho_3^t$ .

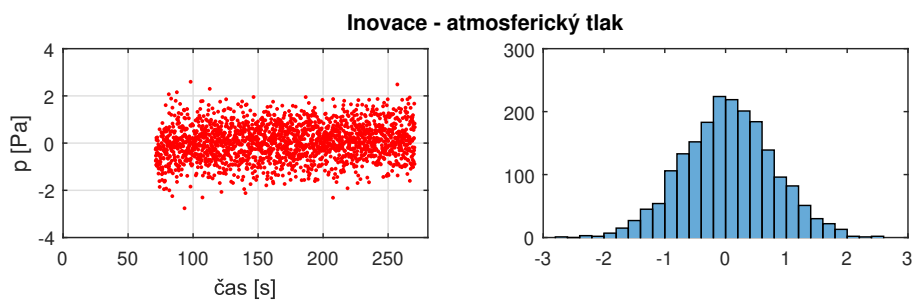


Obrázek 5.15: Odhad  $\rho^t$

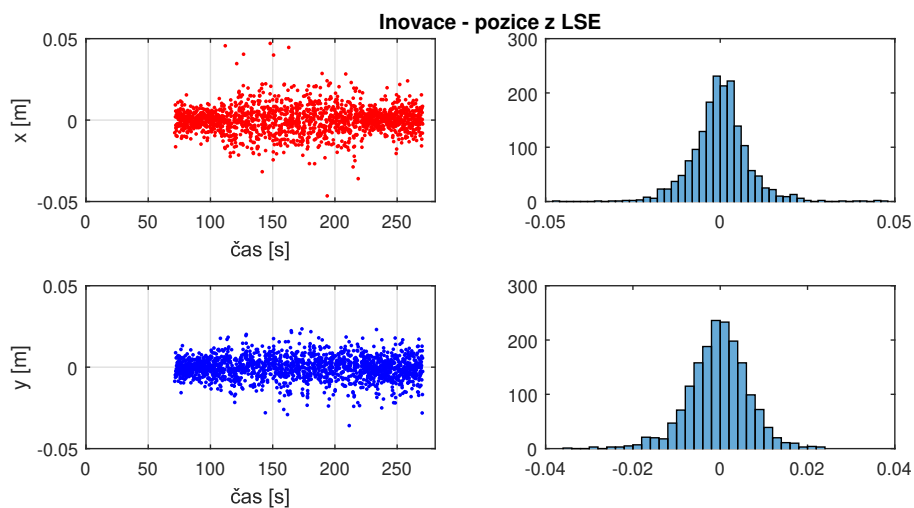
Obrázek 5.16: Směrodatná odchylka  $\rho^t$ 

## Inovace

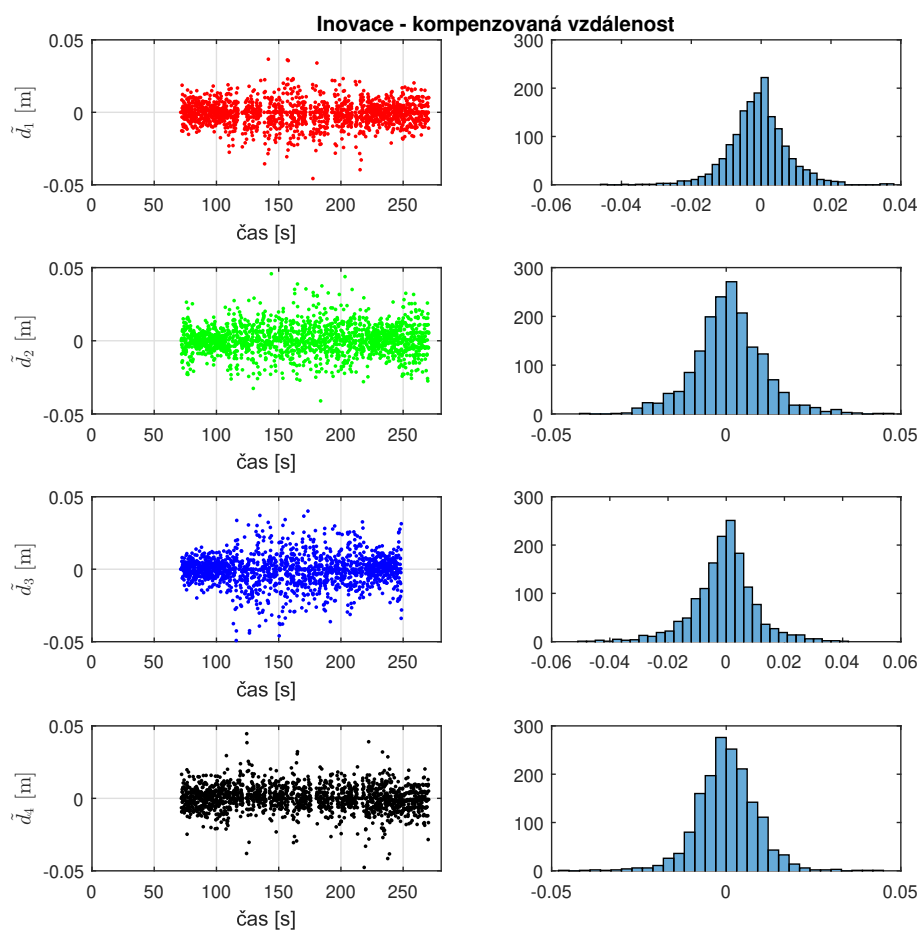
V této části jsou na obrázcích 5.18, 5.19 a 5.17 znázorněny inovace, tedy rozdíl mezi vektorem měření a očekávaným vektorem měření získaným z odhadnutých stavů, a jejich histogramy. V ideálním případě by měly mít inovace charakter bílého šumu. Bělost inovací značí správné naladění Kalmanova filtru. EKF se nepodařilo naladit zcela optimálně tak, aby inovace byly zcela bílé ve všech testovacích scénářích. Při experimentech byla tato bělost porušována zejména několik milisekund před vypnutím korekcí v daném kanálu, nebo po jejich zapnutí. To je způsobeno někdy ne zcela včasnou invalidací parametrů



Obrázek 5.17: Inovace - tlak



Obrázek 5.18: Inovace - pozice z LSE



Obrázek 5.19: Inovace - kompenzované vzdálenosti

odhadnutých průmek, a nebo naopak jejich brzkou opětovnou validací. Přesná délka trvání těchto neběstí je závislá na vzdálenosti k překážce a na rychlosti pohybu navigovaného objektu kolem místa způsobujícího vypnutí korekcí v daném kanálu.

## 6 Závěr

---

Cílem diplomové práce bylo vytvoření navigačního systému pro *indoor* aplikace, včetně experimentálního ověření jeho funkčnosti a přesnosti. Pro dosažení cíle byla práce rozdělena do několika částí.

První z nich byla integrace použitých senzorů do jednoho uceleného systému. Zadaná inerciální měřicí jednotka DMU11 nemohla být použita, protože původně plánovaná druhá verze systému MiniINS, který ji bude obsahovat, nakonec nebyla k dispozici. Proto byla použita verze první, jejíž součástí je inerciální měřicí jednotka MPU-9150.

Druhou částí byl rozbor navigační úlohy a návrh navigačního algoritmu na základě tohoto rozboru. Navržený algoritmus je založený na rozšířeném Kalmanově filtru, je nezávislý na budově a ke své činnosti potřebuje pouze data z inerciálních senzorů, senzoru barometrického tlaku a čtyř senzorů vzdálenosti. Vedle modelu rozšířeného Kalmanova filtru byly navrženy postupy pro nakládání s měřenými vzdálenostmi, které ve výsledku umožňují použití těchto změřených vzdáleností v datové aktualizaci Kalmanova filtru ke korekci horizontální pozice a úhlu vybočení. Návrh a ladění celého navigačního algoritmu bylo nejnáročnější částí celé práce.

Třetí částí byla implementace navrženého algoritmu do výpočetní jednotky navigačního systému. Jako výpočetní jednotka byl použit mikrokontrolér v jednotce MiniINS, která je součástí realizovaného navigačního systému. Většina původních kódů MiniINS byla využita, pouze navigační algoritmus byl nahrazen nově navrženým. Dále byl dopsán ovladač pro senzor vzdáleností a upraveny zprávy posílané přes sběrnici CAN.

Poslední částí bylo testování navigačního systému. Pro potřeby prvotního testování byla vytvořena speciální testovací platforma. Tato platforma umožňuje pohyb navigačního systému v jedné ose a rotaci kolem vertikální osy a s jejím využitím lze určit referenční trajektorii a orientačně i úhel vybočení. Data naměřená na této platformě byla cenná při návrhu navigačního algoritmu. Závěrečné experimentální ověření navrženého navigačního systému proběhlo mimo tuto platformu v neupraveném prostředí. Chyba v odhadu horizontální pozice při tomto experimentu nepřesáhla 50 cm. Pro porovnání, chyba v horizontální pozici odhadnuté nad stejnými daty inerciálním navigačním systémem s náklonem a sklonem korigovaným akcelerometrem by narostla v jedné horizontální ose až k 18 metrům.

Oproti navigačním systémům pracujícím s informacemi o vzdálenostech v celém zorném poli je navržený algoritmus méně výpočetně náročný, na druhou stranu je také méně robustní. Chvilé, kdy se navigační jednotka spoléhá pouze na inerciální senzory, jsou rizikové a na stabilitu systému má v takovýchto okamžicích velký vliv použitá inerciální měřicí jednotka. Ke zrobustnění algoritmu by mohlo přispět uchovávání dlouhodobé mapy okolí. Ve chvílích, kdy by se navigovaný objekt dlouhodobě pohyboval v jedné místnosti, by tato mapa mohla pomoci zkrátit dobu, po kterou se navigační systém musí spoléhat pouze na inerciální senzory.



Dalším námětem pro případné vylepšování systému je přidání senzoru vzdálenosti, který by směřoval kolmo dolů. Tento senzor by umožnil odhadování referenčního tlaku pomocí Kalmanova filtru a přispěl by k lepším odhadům vertikální pozice a rychlosti.

## Seznam použité literatury

---

- [1] SOTÁK, M., SOPATA, M., BRÉDA, R., ROHÁČ, J. a VÁCL, L. *Integrácia navigačných systémov*. Košice: Róbert Bréda „Vydané nákladom vlastným“, 2006. ISBN 80-969619-9-3.
- [2] CAI, G., CHEN, B. M. a LEE, T. H. *Unmanned Rotorcraft Systems*, kapitola Coordinate Systems and Transformations. Springer-Verlag London, 2011. ISBN 978-0-85729-634-4.
- [3] PAČLÍK, M. *Optimalizace softwaru pro vyhodnocení navigačních dat*. Praha, 2016. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická.
- [4] *IMU Errors and Their Effects* [online]. NovAtel, rev. A. Dostupné z: <<https://www.novatel.com/assets/Documents/Bulletins/APN064.pdf>>.
- [5] GREWAL, M. S., WEILL, L. R. a ANDREWS, A. P. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Inc., 2007, 2. edice. ISBN 978-0-470-09971-1.
- [6] ŠIPOŠ, M., PAČES, P., ROHÁČ, J. a NOVÁČEK, P. Analyses of triaxial accelerometer calibration algorithms. *IEEE SENSOR JOURNAL*. 2012, **12**(5), 1157–1165.
- [7] ROHÁČ, J., ŠIPOŠ, M. a ŠIMÁNEK, J. Calibration of low-cost triaxial inertial sensors. *IEEE Instrumentation & Measurement Magazine*. 2015, **18**(6), 32–38.
- [8] ŠIPOŠ, M. a ROHÁČ, J. Calibration of tri-axial angular rate sensors. In: *10. mezinárodní vědecká konference „Měření, diagnostika, spolehlivost palubních soustav letadel“*, Brno, 2010.
- [9] BROŽ, P. *Navigační systém pro bezpilotní prostředky a ultralehká letadla*. Praha, 2016. Diplomová práce. České vysoké učení technické v Praze, Fakulta elektrotechnická.
- [10] *Ultrasound* [online]. SensorWiki.org. Poslední změna 14. 4. 2016 [cit. 1. 5. 2018]. Dostupné z: <<http://sensorwiki.org/doku.php/sensors/ultrasound>>.
- [11] RUFFO, M., CASTRO, M. D., MOLINARI, L., LOSITO, R., MASI, A., KOVERMANN, J. a RODRIGUES, L. New infrared time-of-flight measurement sensor for robotic platforms. In: *20th IMEKO TC4 International Symposium and 18th International Workshop on ADC Modelling and Testing Research on Electric and Electronic Measurement for the Economic Upturn*, Benevento, Itálie, 2014.
- [12] HALEVY, A. Non-contact distance measurement technologies. *The Optimet Blog* [online]. 15. 4. 2015 [cit. 1. 5. 2018]. Dostupné z: <<http://www.optimet.com/blog/non-contact-distance-measurement-technologies/>>.

- [13] TITTERTON, D. a WESTON, J. *Strapdown Inertial Navigation Technology*. The Institution of Engineering and Technology, 2004, 2. edice. ISBN 978-0-86341-358-2.
- [14] GROVES, P. D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008. ISBN 978-1-58053-255-6.
- [15] GREWAL, M. S. a ANDREWS, A. P. *Kalman filtering: theory and practice using MATLAB*. Wiley, 2008, 3. edice. ISBN 978-0-470-37781-9.
- [16] ALARIFI, A., AL-SALMAN, A., ALSALEH, M., ALNAFESSAH, A., AL-HADHRAMI, S., AL-AMMAR, M. A. a AL-KHALIFA, H. S. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors* [online]. 2016, **16**(5). Dostupné z: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4883398/>>.
- [17] SOLA, J. *Simultaneous localization and mapping with the extended Kalman filter* [online]. verze z 5. října 2014. Dostupné z: <[http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs\\_SLAM/SLAM2D/SLAM%20course.pdf](http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs_SLAM/SLAM2D/SLAM%20course.pdf)>.
- [18] YAP, T. N., JR. a SHELTON, C. R. SLAM in large indoor environments with low-cost, noisy, and sparse sonars. In: *IEEE International Conference on Robotics and Automation*, Kobe International Conference Center, Kobe, Japonsko, květen 2009.
- [19] BEEVERS, K. R. a HUANG, W. H. SLAM with sparse sensing. In: *IEEE International Conference on Robotics and Automation*, Orlando, Florida, květen 2006.
- [20] LIU, S., ATIA, M. M., KARAMAT, T., GIVIGI, S. a NOURELDIN, A. A dual-rate multi-filter algorithm for LiDAR-aided indoor navigation systems. In: *Position, Location and Navigation Symposium - PLANS 2014, 2014 IEEE/ION*, Monterey, USA, květen 2014.
- [21] WERNER, T. *Optimalizace*. Elektronická skriptá, verze z 23. října 2015. České vysoké učení technické v Praze, Fakulta elektrotechnická. Dostupné z: <[https://cw.fel.cvut.cz/old/\\_media/courses/a4b33opt/opt.pdf](https://cw.fel.cvut.cz/old/_media/courses/a4b33opt/opt.pdf)>.
- [22] POLÁK, K. Sběrnice CAN. *Elektrorevue* [online]. 16. 6. 2003 [cit. 3. 5. 2018]. Dostupné z: <<http://www.elektrorevue.cz/clanky/03021/index.html>>.
- [23] NOVÁK, J. CAN bus a jeho aplikace ve vozidlech. In: *Přednáška k předmětu „Komunikační rozhraní počítačů“ (A4M38KRP)*, Katedra měření, Fakulta elektrotechnická, ČVUT v Praze.
- [24] *MPU-9150 Product Specification* [online]. InvenSense, rev. 4.3. Dostupné z: <<https://www.invensense.com/wp-content/uploads/2015/02/MPU-9150-Datasheet.pdf>>.
- [25] *DMU11 Technical Datasheet* [online]. Silicon Sensing, rev. 4. Dostupné z: <<https://www.siliconsensing.com/media/30933/dmu11-00-0100-132-rev-4.pdf>>.
- [26] *Cortex-M4* [online]. ARM. [cit. 6. 5. 2018]. Dostupné z: <<https://developer.arm.com/products/processors/cortex-m/cortex-m4>>.

- [27] *STM32F405xx/STM32F407xx Datasheet* [online]. STMicroelectronics, rev. 8, září 2016. Dostupné z: <<http://www.st.com/resource/en/datasheet/stm32f405rg.pdf>>.
- [28] *MS5611-01BA03 Barometric Pressure Sensor, with stainless steel cap* [online]. TE Connectivity, červen 2017. Dostupné z: <[https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5611-01BA03%7FB3%7Fpdf%7FEnglish%7FENG\\_DS\\_MS5611-01BA03\\_B3.pdf%7FCAT-BLPS0036](https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5611-01BA03%7FB3%7Fpdf%7FEnglish%7FENG_DS_MS5611-01BA03_B3.pdf%7FCAT-BLPS0036)>.
- [29] *TeraRanger Tower Specification Sheet* [online]. Terabee. Dostupné z: <<https://www.terabee.com/wp-content/uploads/2017/09/Towerspecificationsheet-1.pdf>>.
- [30] *TeraRanger Tower User Manual* [online]. Terabee, verze 1.0.0, květen 2016. Dostupné z: <[https://www.terabee.com/wp-content/uploads/2017/09/TeraRangerTowerManualVersion1\\_0.pdf](https://www.terabee.com/wp-content/uploads/2017/09/TeraRangerTowerManualVersion1_0.pdf)>.
- [31] ROSEBROCK, A. Ball Tracking with OpenCV. *PyImageSearch* [online]. 14. 9. 2015 [cit. 8. 12. 2018]. Dostupné z: <<https://www.pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/>>.

## Seznam obrázků

---

2.1	Orientace souřadnicového systému navigovaného tělesa vůči tečnému, inspirováno obálkou [1] . . . . .	3
2.2	Orientace tečného souřadnicového systému vůči navigačnímu, inspirováno obálkou [1] . . . . .	4
2.3	Princip kapacitního akcelerometru, převzato z [3] . . . . .	5
2.4	Princip vibračních senzorů úhlové rychlosti, podle [3] . . . . .	5
2.5	Kompenzovatelné chyby senzorů: (a) offset, (b) <i>scale</i> faktor, (c) nelinearita, (d) asymetrie, (e) hluchá oblast, (f) kvantizace, převzato z [5] . . . . .	6
2.6	Druhy INS . . . . .	11
2.7	Fúze INS a dalších systémů, převzato z [3] . . . . .	11
2.8	Klasifikace <i>indoor</i> navigačních systémů . . . . .	14
2.9	Body ve 2D a jejich kolmé projekce do přímky procházející počátkem, převzato z [21] . . . . .	16
2.10	Body ve 2D a jejich kolmé projekce do přímky, převzato z [21] . . . . .	17
3.1	MiniINS v1 použitá v realizovaném navigačním systému . . . . .	19
3.2	MiniINS na kalibrační náklonné plošině . . . . .	20
3.3	Odchytky velikosti změřené (modře)/kompenzované (červeně) specifické síly od velikosti vektoru tíhového zrychlení . . . . .	21
3.4	Úhlové rychlosti naměřené při rotacích kolem jednotlivých os . . . . .	22
3.5	Terabee TeraRanger Tower použitá v realizovaném navigačním systému, převzato z [29] . . . . .	23
3.6	Návrh propojení senzorů do jednoho systému . . . . .	25
3.7	Fotografie vytvořeného navigačního systému . . . . .	25
4.1	Definice proměnných . . . . .	29
4.2	Geometrie navigační úlohy . . . . .	30
4.3	Frekvenční odezva dolní propusti . . . . .	31
4.4	Ilustrační příklad bodů „viděných“ senzory při pohybu v jedné ose . . . . .	34
4.5	Obrázek k algoritmu 1 . . . . .	36
4.6	Příklad možného stavu historie bodů . . . . .	37
4.7	Vývoj SSE při přechodech přes roh místnosti . . . . .	38
4.8	Vývoj vzdálenosti právě „viděného“ bodu od proložené přímky při přechodech přes roh místnosti . . . . .	39
4.9	Vývoj rozdílů mezi právě změřenou a zpožděnou vzdáleností při přechodech přes roh místnosti . . . . .	40
4.10	Vývoj diference rozdílů mezi právě změřenou a zpožděnou vzdáleností při přechodech přes roh místnosti . . . . .	41

4.11	Fáze navigačního algoritmu . . . . .	43
4.12	Průběh iniciálního mapování <sup>(1)</sup> podle rovnice (4.1); <sup>(2)</sup> podle rovnice (4.11); <sup>(3)</sup> podle algoritmu 3) . . . . .	44
4.13	Příklad možných poloh před iniciálním mapováním (vlevo - dobře, vpravo - špatně) . . . . .	44
4.14	Tok dat v algoritmu . . . . .	46
4.15	Metody korekce s minimální časovou náročností při použití ARM DSP knihovny, převzato z [3] . . . . .	48
5.1	Vytvořený navigační systém upevněný na testovací platformě . . . . .	51
5.2	Záběry některých testovacích scénářů využívajících testovací platformu v idealizovaném prostředí . . . . .	51
5.3	Testovací scénář č. 7 (překážky černě, skutečná trajektorie červeně, odhad- nutá trajektorie modře) . . . . .	52
5.4	Vlevo: navigační systém se zelenou značkou a konstrukce pro upevnění ka- mery; vpravo: snímek z nahraného videa během zpracovávání algoritmem z [31] . . . . .	53
5.5	Pozice: odhad (barevně) a směrodatná odchylka (černě) . . . . .	53
5.6	Rozdíl mezi referenční pozicí a pozicí odhadnutou navrženým systémem . . . . .	54
5.7	Rozdíl mezi referenční pozicí a pozicí odhadnutou pomocí INS s náklonem a sklonem korigovaným akcelerometrem . . . . .	55
5.8	Odhad rychlosti . . . . .	55
5.9	Směrodatná odchylka rychlosti . . . . .	56
5.10	Odhad polohových úhlů (barevně) a polohové úhly získané integrací úhlo- vých rychlostí kompenzovaných o iniciální odhad biasu senzoru úhlových rychlostí (černě) . . . . .	56
5.11	Směrodatná odchylka polohových úhlů . . . . .	57
5.12	Vlevo: iniciální poloha, vpravo: finální poloha . . . . .	57
5.13	Bias senzoru úhlové rychlosti: odhad (barevně) a směrodatná odchylka (černě) . . . . .	58
5.14	Bias akcelerometru: odhad (barevně) a směrodatná odchylka (černě) . . . . .	58
5.15	Odhad $\rho^t$ . . . . .	59
5.16	Směrodatná odchylka $\rho^t$ . . . . .	60
5.17	Inovace - tlak . . . . .	60
5.18	Inovace - pozice z LSE . . . . .	61
5.19	Inovace - kompenzované vzdálenosti . . . . .	61

## Seznam tabulek

---

3.1	Základní parametry senzoru úhlové rychlosti v MPU-9150 [24] . . . . .	20
3.2	Základní parametry akcelerometru v MPU-9150 [24] . . . . .	20
3.3	Odhadnuté parametry chybového modelu akcelerometru . . . . .	21
3.4	Odhadnuté parametry chybového modelu senzoru úhlové rychlosti . . . . .	21
3.5	Základní parametry senzoru barometrického tlaku MS5611, podle [28] . . . . .	23
3.6	Základní parametry zařízení TeraRanger Tower, podle [30] . . . . .	24
3.7	TeraRanger Tower - závislost frekvence na vzdálenosti (přesný mód při vypnutém <i>crosstalk avoidance</i> ), podle [11] . . . . .	24
3.8	Seznam zpráv s odhady navigačních proměnných posílaných po sběrnici CAN . . . . .	27
A.1	Úplný seznam zpráv posílaných po sběrnici CAN . . . . .	76

## Seznam algoritmů

---

1	Výpočet $\alpha_i^n, \rho_i^n$ z obecné rovnice přímky . . . . .	36
2	Validace parametrů $\alpha_i^n, \rho_i^n$ . . . . .	38
3	Aktualizace iniciální historie bodů . . . . .	41
4	Aktualizace historie bodů . . . . .	42
5	Konstrukce vektoru měření . . . . .	45



## Seznam zkratek

---

ASCII	American Standard Code for Information Interchange
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
CSMA/CR	Carrier Sense Multiple Access with Collision Resolution
DSP	Digital Signal Processing
EKF	Extended Kalman Filter
ENU	East-North-Up
FPU	Floating-Point Unit
GLONASS	Globalnaja navigacionnaja sputnikovaja sistěma
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
MAC	Multiply and Accumulate
MEMS	Micro-Electro-Mechanical System
NED	North-East-Down
RISC	Reduced Instruction Set Computing
RSS	Received Signal Strength
SIMD	Single Instruction, Multiple Data
SLAM	Simultaneous Localization and Mapping
SPI	Serial Peripheral Interface
SSE	Error Sum of Squares
SVD	Singular-Value Decomposition
TDoA	Time Difference of Arrival
ToF	Time of Flight

TTL	Transistor-Transistor-Logic
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus

# Seznam symbolů

---

$\ \cdot\ $	Euklidovská norma
$\hat{\{}}$	odhad z EKF
$\{ \}^b$	proměnná vyjádřena v souřadnicovém systému navigovaného tělesa ( <i>body coordination system</i> )
$\{ \}^t$	proměnná vyjádřena v neseném tečném souřadnicovém systému
$\{ \}^n$	proměnná vyjádřena v navigačním souřadnicovém systému
$\phi$	náklon ( <i>roll</i> )
$\theta$	sklon ( <i>pitch</i> )
$\psi$	kurz ( <i>yaw</i> )
$\alpha_i^r$	úhel který svírá $i$ -tý směr souřadnicového systému $r$ s kolmicí na $i$ -tou překážku
$\rho_i^r$	kolmá vzdálenost počátku souřadnicového systému $r$ od překážky č. $i$
$\omega$	úhlová rychlost
$C_a^b$	transformační matice ze souřadnicového systému $a$ do systému $b$
$d_i$	vzdálenost změřená senzorem č. $i$
$\tilde{d}_i$	vzdálenost změřená senzorem č. $i$ kompenzovaná o náklon a sklon
$f$	funkce určující dynamický systém, nebo specifická síla
$F$	Jakobián funkce $f$
$h$	funkce určující vztah vektoru měření se stavovým vektorem
$H$	Jakobián funkce $h$
$K$	matice Kalmanova zesílení
$p$	vektor pozice
$\tilde{p}$	vektor pozice odhadnuté pomocí metody nejmenších čtverců
$P$	kovarianční matice odhadu stavového vektoru
$P^-$	kovarianční matice odhadu stavového vektoru po predikci
$P^+$	kovarianční matice odhadu stavového vektoru po korekci
$Q$	kovarianční matice procesního šumu
$R$	kovarianční matice šumu měření
$\mathbb{R}$	množina reálných čísel

$\mathbf{v}$	vektor rychlosti
$\mathbf{x}$	stavový vektor
$\hat{\mathbf{x}}^-$	odhad stavového vektoru po predikci
$\hat{\mathbf{x}}^+$	odhad stavového vektoru po korekci
$\mathbf{y}$	vektor měření
$\mathbb{Z}$	množina celých čísel

## Přílohy

### A Kompletní přehled zpráv posílaných po sběrnici CAN

ID	Rozsah indexů v datovém poli	Datový typ	Popis
0x132	0-3	float	změřený tlak vzduchu [mbar]
	4-7	float	tlak vzduchu po filtraci <i>lowpass</i> filtrem [mbar]
0x12A	0-3	float	kompensovaná <sup>(1)</sup> vzdálenost - senzor č. 1 [mm]
	4-7	float	kompensovaná <sup>(1)</sup> vzdálenost - senzor č. 2 [mm]
0x12B	0-3	float	kompensovaná <sup>(1)</sup> vzdálenost - senzor č. 3 [mm]
	4-7	float	kompensovaná <sup>(1)</sup> vzdálenost - senzor č. 4 [mm]
0x100	0-3	float	vzdálenost 1 <sup>(2)</sup> [mm]
	4-7	float	vzdálenost 2 <sup>(2)</sup> [mm]
0x101	0-3	float	vzdálenost 3 <sup>(2)</sup> [mm]
	4-7	float	vzdálenost 4 <sup>(2)</sup> [mm]
0x102	0-3	float	vzdálenost 5 <sup>(2)</sup> [mm]
	4-7	float	vzdálenost 6 <sup>(2)</sup> [mm]
0x103	0-3	float	vzdálenost 7 <sup>(2)</sup> [mm]
	4-7	float	vzdálenost 8 <sup>(2)</sup> [mm]
0x140	0-3	float	kompensovaná <sup>(1)</sup> úhlová rychlost - osa $x^b$ [deg/s]
	4-7	float	kompensovaná <sup>(1)</sup> úhlová rychlost - osa $y^b$ [deg/s]
0x141	0-3	float	kompensovaná <sup>(1)</sup> úhlová rychlost - osa $z^b$ [deg/s]
	4-7	float	kompensovaná <sup>(1)</sup> specifická síla - osa $x^b$ [g]
0x142	0-3	float	kompensovaná <sup>(1)</sup> specifická síla - osa $y^b$ [g]
	4-7	float	kompensovaná <sup>(1)</sup> specifická síla - osa $z^b$ [g]
0x12C	0-3	float	úhlová rychlost - osa $x^{imu}$ <sup>(3)</sup> [deg/s]
	4-7	float	úhlová rychlost - osa $y^{imu}$ <sup>(3)</sup> [deg/s]
0x12D	0-3	float	úhlová rychlost - osa $z^{imu}$ <sup>(3)</sup> [deg/s]
	4-7	float	specifická síla - osa $x^{imu}$ <sup>(3)</sup> [g]
0x12E	0-3	float	specifická síla - osa $y^{imu}$ <sup>(3)</sup> [g]
	4-7	float	specifická síla - osa $z^{imu}$ <sup>(3)</sup> [g]

*Pokračování tabulky na další stránce →*

→ Pokračování tabulky z předchozí strany

ID	Rozsah indexů v datovém poli	Datový typ	Popis
0x600	0-3	float	pozice - osa $x^n$ [m]
	4-7	float	pozice - osa $y^n$ [m]
0x601	0-3	float	pozice - osa $z^n$ [m]
	4-7	float	rychlost - osa $x^n$ [m/s]
0x602	0-3	float	rychlost - osa $y^n$ [m/s]
	4-7	float	rychlost - osa $z^n$ [m/s]
0x603	0-3	float	Eulerův úhel - $\phi$ [rad]
	4-7	float	Eulerův úhel - $\theta$ [rad]
0x604	0-3	float	Eulerův úhel - $\psi$ [rad]
	4-7	float	bias senzoru úhlové rychlosti - osa $x^b$ [rad/s]
0x605	0-3	float	bias senzoru úhlové rychlosti - osa $y^b$ [rad/s]
	4-7	float	bias senzoru úhlové rychlosti - osa $z^b$ [rad/s]
0x606	0-3	float	bias akcelerometru - osa $x^b$ [m/s <sup>2</sup> ]
	4-7	float	bias akcelerometru - osa $y^b$ [m/s <sup>2</sup> ]
0x607	0-3	float	bias akcelerometru - osa $z^b$ [m/s <sup>2</sup> ]
	4-7	float	nula
0x608	0-3	float	kolmá vzdálenost k překážce č. 1 [m]
	4-7	float	kolmá vzdálenost k překážce č. 2 [m]
0x609	0-3	float	kolmá vzdálenost k překážce č. 3 [m]
	4-7	float	kolmá vzdálenost k překážce č. 4 [m]

(1) slovem kompenzovaný se rozumí po kalibraci, filtraci a/nebo prohození os

(2) očíslování senzorů používané zařízením Teraranger Tower

(3) souřadnicový systém IMU

Tabulka A.1: Úplný seznam zpráv posílaných po sběrnici CAN

## B Hodnoty kovariančních matic EKF

$$\mathbf{Q} = \text{diag} \begin{bmatrix} 0,5^2 \\ 0,5^2 \\ 0,5^2 \\ 0,1^2 \\ 0,1^2 \\ 0,1^2 \\ 0,0175^2 \\ 0,0175^2 \\ 0,0175^2 \\ (1,7453 \cdot 10^{-7})^2 \\ (1,7453 \cdot 10^{-7})^2 \\ (1,7453 \cdot 10^{-7})^2 \\ (9,80665 \cdot 10^{-5})^2 \\ (9,80665 \cdot 10^{-5})^2 \\ (9,80665 \cdot 10^{-5})^2 \\ 0,5^2 \\ 0,5^2 \\ 0,5^2 \\ 0,5^2 \end{bmatrix}, \quad \mathbf{P}_0 = \text{diag} \begin{bmatrix} \mathbf{0}^{6 \times 1} \\ 0,0524^2 \\ 0,0524^2 \\ 0 \\ 0,0087^2 \\ 0,0087^2 \\ 0,0087^2 \\ 0,0087^2 \\ 0,0981^2 \\ 0,0981^2 \\ 0,0981^2 \\ 0,0981^2 \\ 0,1^2 \\ 0,1^2 \\ 0,1^2 \\ 0,1^2 \end{bmatrix}, \quad \mathbf{R} = \text{diag} \begin{bmatrix} 0,02^2 \\ 0,02^2 \\ 0,01^2 \\ 0,01^2 \\ 0,01^2 \\ 0,01^2 \\ 0,5^2 \end{bmatrix}$$

## C Obsah příloženého CD

```
root
├── DP_Martin_Paclík_2019.pdf ..... Tento text ve formátu PDF.
├── CV_Python/ ..... Kód pro získání referenční pozice k testu č. 7 (viz 5.2.1).
├── EKF_C/ ..... Navigační algoritmus v jazyce C (viz 4.9).
├── Matlab/ ..... Matlab kódy - algoritmus, kalibrace, vyhodnocení (viz 4.9).
└── Microcontroller/ ..... Kódy mikrokontroléru (viz 4.9).
```

Na CD jsou přiloženy všechny kódy, které byly při práci na DP využity. V obsahu CD výše je u každého názvu přiložené složky popis toho, co obsahuje. U tohoto popisu je vždy v závorce uvedeno, ve které kapitole lze nalézt informace o tom, která část vznikla nově při práci na této DP a která část byla převzata a kde.