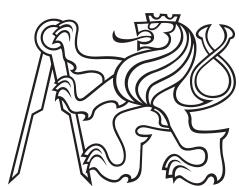


Diplomová práce



**České
vysoké
učení technické
v Praze**

F3

Fakulta elektrotechnická
Katedra kybernetiky

Přesné měření vlastní trajektorie vozidla

David Pekárek

Vedoucí: Ing. Jan Čech, Ph.D.
Obor: Kybernetika a Robotika
Srpen 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Pekárek** Jméno: **David** Osobní číslo: **434691**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Kybernetika a robotika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Přesné měření vlastní trajektorie vozidla

Název diplomové práce anglicky:

Precise Vehicle Ego-Motion Trajectory Measurement

Pokyny pro vypracování:

Odhadování vlastní trajektorie vozidla pouze z inerciální jednotky (IMU) je obecně považováno za nedostatečně nepřesné díky integraci chyb. Nedávný článek [1] navrhuje metodu dosahující slibné výsledky, ve které je trajektorie odhadována z IMU pomocí neuronové sítě. Autoři poskytují kód spolu s výsledky na standardní databázi KITTI [2].

1. Seznamte se s metodami odhadování vlastní trajektorie vozidla.
2. Stáhněte si kód z [1] a opakujte experiment na databázi KITTI a na simulovaných IMU datech.
3. Nasbírejte vlastní databázi s referenční (ground-truth) trajektorií jedoucího vozidla. Data budou obsahovat záznamy palubních senzorů (IMU + kamera) a referenční trajektorii. Přednostně použijte zmenšenou experimentální platformu (instrumentovaný model RC auta vyvinutý na katedře kybernetiky a řízení).
4. Otestujte metodu na vlastní databázi a vyhodnoťte přesnost.
5. (Volitelně) Testujte nějakou metodu vizuální odometrie, např. [3] nebo komerční metodu dodávanou spolu se ZED kamerou [4] a porovnejte výsledky s odhadem z IMU. Uvažujte možnosti fúze vizuálních a inerciálních odhadů [5].

Seznam doporučené literatury:

- [1] M. Brossard, A. Barrau, S. Bonnamel. AI-IMU Dead-Reckoning, In arXiv, 2019. <https://arxiv.org/abs/1904.06064>
- [2] Andreas Geiger and Philip Lenz and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proc. CVPR, 2012.
- [3] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civer, J. J. Berlles. S-PTAM: Stereo Parallel Tracking and Mapping. Robotics and Autonomous Systems. Vol. 93, 2017.
- [4] StereoLabs. ZED Stereo Camera, 2019. <https://www.stereolabs.com/zed/>.
- [5] Guoquan Huang, Visual-Inertial Navigation: A Concise Review. In Proc. ICRA, 2019.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Jan Čech, Ph.D., skupina vizuálního rozpoznávání FEL

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

Datum zadání diplomové práce: **13.02.2020**

Termín odevzdání diplomové práce: **14.08.2020**

Platnost zadání diplomové práce:
do konce zimního semestru 2021/2022

Ing. Jan Čech, Ph.D.
podpis vedoucí(ho) práce

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

prof. Mgr. Petr Páta, Ph.D.
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

Poděkování

Tímto bych rád poděkoval vedoucímu mé diplomové práce Ing. Janu Čechovi, Ph.D., za množství cenných rad, vynaložené úsilí a snahu o předání nepřeberného množství poznatků. Dále bych rád poděkoval za jeho trpělivost a vstřícnost.

Poděkování patří také docentu Ing. Zdeňku Hurákovi, Ph.D. a docentu Ing. Pavlovi Pačesovi, Ph.D. za zapůjčení jejich vybavení a předání důležitých poznatků.

Poslední a nejdůležitější poděkování patří mé rodině, která mi umožnila a následně mě podporovala při mé cestě studiem.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 13. srpna 2020

Abstrakt

Tato diplomová práce se zabývá odhadováním vlastní trajektorie vozidla na základě měření z inerciální jednotky, jež by mohlo být důležitým prvkem autonomního a asistovaného řízení. Výchozím testovaným algoritmem s vynikajícími výsledky byla metoda AI-IMU Dead-Reckoning Brossard et al., 2019. Podstatou metody je filtrování zašuměných dat za pomocí Invariantního Rozšířeného Kalmanova Filtru, který jako zdroj šumových parametrů využívá neuronové sítě.

V rámci práce byly ověřeny publikované výsledky a doplněny nejasnosti v původním článku. Dále byl vygenerován dataset vhodný pro testování inerciálních metod lokalizace. Na syntetickém datasetu byla testována přesnost a robustnost algoritmů v závislosti na šumových intenzitách. Měřící TOMI RC subscale platforma byla upravena pro potřeby tohoto typu měření a současně byl vytvořen dataset obsahující referenční trajektorii, inerciální data a kamerové záznamy. AI-IMU Dead-Reckoning metoda byla laděna a testována na nově vytvořeném datasetu. Na obou datasetech syntetickém i reálném bylo měřeno až 20% zlepšení oproti konvenčním metodám Kalmanovy filtrace.

Klíčová slova: IMU, trasování, lokalizace, strojové učení, neuronové sítě

Vedoucí: Ing. Jan Čech, Ph.D.
Praha, Na Zderaze 269/4, místnost: G-2

Abstract

This diploma thesis deals with the estimation of the trajectory of a vehicle which is based on data captured by inertial measurement units. The IMU based trajectory estimation should be a significant part of autonomous and assisted driving control. We primarily tested algorithm AI-IMU Dead-Reckoning Brossard et al., 2019 which reports impressive results. The core of this method is Invariant Extended Kalman filter which uses Deep Neural Networks as the source of the noise parameters.

Besides testing the algorithm we attempted to clarify the original paper. The synthesized dataset for the inertial algorithm tests was proposed and compared to the algorithms depending on the noise parameters. TOMI RC subscale platform, developed by researchers at CTU in Prague, was adjusted to provide dataset with the ground-truth trajectory, the inertial data and the stereo-camera images. Throughout the testing, it was proved that AI-IMU Dead-Reckoning method proposes the improvement up to 20 % in comparison with the conventional methods based on Kalman filtering.

Keywords: IMU, tracking, localization, machine learning, neural networks

Title translation: Precise Vehicle Ego-Motion Trajectory Measurement

| Obsah | |
|--|-----------|
| 1 Úvod | 1 |
| 2 Metody pro měření polohy | 3 |
| 2.1 Odometrie | 4 |
| 2.2 Inerciální senzory | 5 |
| 2.3 Vizuální odometrie | 6 |
| 2.3.1 Stereo-kamera | 7 |
| 2.3.2 Všesměrová kamera | 7 |
| 2.4 Lidar a Radar | 7 |
| 2.4.1 Lidar | 8 |
| 2.4.2 Radar | 8 |
| 2.5 Globální navigační satelitní systém | 8 |
| 2.5.1 GPS | 9 |
| 2.5.2 Galileo | 9 |
| 2.5.3 GLONASS | 10 |
| 2.5.4 BeiDou | 10 |
| 2.6 Další metody | 10 |
| 3 Algoritmy pro měření polohy | 11 |
| 3.1 GNSS | 11 |
| 3.1.1 Zjištování polohy | 11 |
| 3.1.2 Minimalizace odchylky | 13 |
| 3.1.3 DGPS | 13 |
| 3.2 Inerciální senzory | 14 |
| 3.2.1 Úhlová rychlosť | 15 |
| 3.2.2 Specific force | 15 |
| 3.2.3 Chyby senzorů a jejich korekce | 16 |
| 3.2.4 Kalmanův filtr | 16 |
| 3.2.5 Rozšířený Kalmanův filtr ... | 17 |
| 3.2.6 Invariantní Rozšířený Kalmanův filtr | 18 |
| 4 AI-IMU Dead-Reckoning | 19 |
| 4.1 Princip | 19 |
| 4.2 Prezentované výsledky | 21 |
| 4.3 Doplňující testování | 22 |

| | | | |
|---|-----------|--|----|
| 5 Experiment na simulátoru | 25 | | |
| 5.1 Možnosti generátoru | 26 | 6.3 Parametry filtru a trénování neuronové sítě | 47 |
| 5.2 Dataset..... | 26 | 6.4 Výsledky | 49 |
| 5.2.1 Definování IMU modelu..... | 26 | | |
| 5.2.2 Vytvoření souboru definujícího pohyb..... | 27 | | |
| 5.2.3 Výsledný dataset | 28 | | |
| 5.3 Výsledky | 31 | | |
| 5.3.1 Metoda nefiltrované integrace | 32 | | |
| 5.3.2 Rozšířená Kalmanova filtrace | 33 | | |
| 5.3.3 AI-IMU Dead-Reckoning | 34 | | |
| 5.3.4 Porovnání trajektorií | 35 | | |
| 6 Experiment na TOMI platformě | 37 | | |
| 6.1 Měřící TOMI platforma | 37 | | |
| 6.1.1 Hardware | 38 | | |
| 6.1.2 TOMI vybavení..... | 38 | | |
| 6.1.3 Konfigurace hardwaru a příprava na vytvoření datasetu .. | 40 | | |
| 6.2 Dataset..... | 43 | | |

Obrázky

| | | | |
|--|----|--|----|
| 1.1 TOMI platforma..... | 2 | 5.7 Porovnání výsledků všech úrovní zašumění pro metodu Kalmanovy filtrace | 34 |
| 4.1 Struktura IEKF [1] | 20 | 5.8 Porovnání výsledků všech úrovní zašumění pro AI-IMU Dead-Reckoning | 35 |
| 4.2 Struktura výchozího systému [1] | 21 | 5.9 Porovnání trajektorií vypočítaných nefiltrovanou integrační metodou v závislosti na pamámetru σ | 35 |
| 4.3 Testovaný EKF..... | 22 | 5.10 Porovnání trajektorií vypočítaných EKF metodou v závislosti na pamámetru σ | 36 |
| 4.4 Změny kovariančních šumových matic | 22 | 5.11 Porovnání trajektorií vypočítaných AI-IMU Dead-Reckoning metodou v závislosti na pamámetru σ | 36 |
| 4.5 Průměrný EPE založený na třech základních IMU metodách | 24 | | |
| 4.6 Lokalizace jízdy #33 2011/09/30 | 24 | | |
| 5.1 Vzorový motion file [2] | 27 | 6.1 Schéma zapojení palubních přístrojů na TOMI platformě před dodatečnou úpravou [3] | 39 |
| 5.2 Porovnání originální a generované trajektorie | 29 | 6.2 Fotografie palubních přístrojů .. | 40 |
| 5.3 Porovnání originální a generované úhlové rychlosti na ose z | 30 | 6.3 Schéma zapojení a komunikace DGPS | 42 |
| 5.4 Porovnání originálního a generovaného zrychlení na ose x .. | 30 | 6.4 Umístění přijímačů a GPS antén | 42 |
| 5.5 Porovnání výsledků všech úrovní zašumění pro integrační metodu .. | 33 | 6.5 Referenční trajektorie | 45 |
| 5.6 Přiblížené porovnání výsledků všech úrovní zašumění pro integrační metodu neobsahující σ_0 a σ_6 | 33 | 6.6 Referenční rychlosť | 46 |
| | | 6.7 Referenční úhel | 46 |

| | |
|--|----|
| 6.8 Porovnání hodnot zrychlení Kitti(nahoře) a TOMI(dole) | 47 |
| 6.9 Porovnání hodnot úhlového zrychlení Kitti(nahoře) a TOMI(dole) | 48 |
| 6.10 Porovnání pozice v závislosti na použitém algoritmu | 48 |
| 6.11 EndPoint Error vlastní databáze po 20 sekundách | 50 |
| 6.12 EndPoint Error vlastní databáze po 40 sekundách | 50 |
| 6.13 Cumulative Error vlastní databáze po 20 sekundách | 51 |
| 6.14 Cumulative Error vlastní databáze po 40 sekundách | 51 |

Tabulky

| | |
|--|----|
| 4.1 Porovnání algoritmů | 21 |
| 5.1 Specifikační tabulka generovaných datasetů | 28 |
| 5.2 Specifikační tabulka | 31 |
| 5.3 Porovnání průměrných EPE [m] v závislosti na σ | 34 |
| 6.1 Tabulkové parametry MTi-10 series | 42 |
| 6.2 Specifikace předzpracovaného datasetu | 44 |
| 6.3 Použité parametry Rozšířeného Kalmanova filtru | 48 |

Kapitola 1

Úvod

Systémy autonomního a asistovaného řízení často vyžadují přesný odhad vlastní trajektorie vozidla. Lokalizace založená na inerciálních jednotkách je využívána převážně v leteckém průmyslu, ale postupně proniká i do ostatních oborů napříč odvětvím. Nedávná práce AI-IMU Dead-Reckoning [1] prezentuje velice slibné výsledky. Jedná se o metodu využívající neuronové sítě jako zdroj parametrů při filtrování dat.

Cílem této diplomové práce je seznámit se s metodami odhadování vlastní trajektorie vozidla a opakovat experiment týmu, který vyvinul metodu AI-IMU Dead-Reckoning. Opakování experimentu je rozděleno do dvou částí. První část se venuje přípravě simulátoru, tvorbě datasetu a následnému testování na vlastních simulovaných datech. Testování algoritmů probíhá v závislosti na míře zašumění vstupních dat. Druhá část je věnována přípravě RC platformy a měření datasetu, který by mohl být dále využíván k dalším experimentům. V rámci vytvořeného datasetu je následně testována samotná AI-IMU Dead-Reckoning metoda společně s konvenčními metodami, tj. Kalmanovy filtrace.

Práce je strukturována následovně. V druhé kapitole jsou diskutovány jednotlivé metody lokalizace rozdělené podle druhu měřené veličiny. Třetí kapitola se venuje algoritmu zpracování a principu lokalizace.

Kapitola čtvrtá pojednává o publikaci AI-IMU Dead-Reckonig a zodpovídá nejasnosti, které mohou vznikat po přečtení původní publikace. Hlavní zodpovězenou otázkou je zde porovnání navržené metody a filtrace, na kterou algoritmus navazuje.

1. Úvod

Kapitola pátá představuje syntetický experiment akcelerometrických a gyroskopických dat, ze kterých je vytvořen rozsáhlý dataset. Hlavním přínosem tohoto principu je možnost testování robustnosti algoritmů na základě úrovně zašumění dat, nejen v metodách zde testovaných, ale i algoritmech vyvinutých v budoucnosti.

Šestá kapitola představuje přípravu a modifikace měřící soustavy, kterou je v tomto případě RC platforma. RC TOMI platforma byla sestrojena Toyota týmem v rámci spolupráce katedry kybernetiky a katedry řízení, jehož byl autor této práce členem. V průběhu kapitoly je dále popsán vytvořený dataset, který může být následně využíván pro algoritmy založené na inerciální nebo vizuální odometrii. V samotném závěru této kapitoly jsou algoritmy popsané a testované napříč prací aplikovány na vytvořeném datasetu. Celou práci zakončuje výsledné porovnání dosažených výsledků.



Obrázek 1.1: TOMI platforma

Kapitola 2

Metody pro měření polohy

Mezi historické způsoby měření polohy může být zařazen původní způsob lokalizace lodí, ve kterých měřili rychlosť pohybu v uzlech, čas na hodinách a směr pohybu na základě kompasu. Tímto způsobem definovali úlohu, které dnes říkáme Dead-Reckoning. Analogii s historií můžeme hledat i ve využití Sextantu, s jehož pomocí byla určována absolutní pozice na základě hvězd. Dnes jsou tyto způsoby lokalizace překonány, ale základní princip měření a výpočtu přetrval dodnes.

Postupně se rozvíjející obor autonomního řízení poskytuje prostor a potřebu pro zlepšení dosavadních zdrojů informací dodávaných dopravním prostředkům. Tyto informace jsou získány na základě měření mnoha veličin a následně zpracovány a přepracovány do užitečných informací, rozšiřujících znalost stavu, pozici, okolí vozidla a mnoha dalších využitelných poznatků, které řidič v různé míře bere vědomě i podvědomě v potaz. Následně je možné s touto informací naložit pro asistenci řidiče v ovládání vozidla, či nahrazení řidiče samotného.

V automobilovém průmyslu existuje mnoho rozdílných technik využívaných pro lokalizaci. *Self-localization* metody nelze snadno rozdělit do několika skupin, jelikož je v mnoha pracích zaměňován způsob vyčítání veličin a jejich následné zpracování, bude tato kapitola pojednávat pouze o měřených veličinách. Tyto způsoby budou rozřazeny a následně klasifikovány podle jejich výhod, nevýhod, omezení a spolehlivosti.

- V současné době nejrozšířenějším způsobem zjišťování kinematických informací je odometrie, která využívá znalosti počátečního stavu a změny pohybu na základě vyčítání rotačních enkodérů. Tyto enkodéry poskytují informace o rotaci os, ze kterých lze následně dopočítat rychlosť, pozici a natočení vozidla.
- *Global navigation satellite system* neboli globální navigační satelitní systém je přístup určování polohy na základě triangulace, kde se vozidlo lokalizuje pomocí znalosti vzdálenosti od minimálně čtyř družic.
- Lidarová lokalizace je způsob určování polohy na základě vyzařování mnoha paprsků a snímání doby, za kterou se paprsek odrazí zpět. Podobná radarová lokalizace je způsob založený na stejném principu jako lidarový snímač s rozdílnou vysílanou veličinou, kterou je zde radiová vlna.
- Kamera je speciálním případem senzoru, který vozidlu zajišťuje vizuální informace o okolí. Tyto informace mohou být následně využity pro identifikaci objektů, hledání souvislostí nebo určení vzdáleností objektů v případě použití stereo-kamery.
- *Inertial Measurement Unit* případně i inerciální jednotka je zařízení obsahující akcelerometr, gyroskop případně i magnetometr, které poskytují informace pro určení zrychlení, úhlové rychlosti a orientace vůči magnetickému poli Země.

■ 2.1 Odometrie

Odometrie je způsob zjišťování kinematiky vozidla na základě znalosti vlastností Ackermannova podvozku. Jedná se o velice dlouho známé a levné řešení široce rozšířené napříč výzkumem i průmyslem. Tento typ měření probíhá kompletně na palubě vozidla a měří údaje o rychlosti a úhlu natočení kol. Pro vyčítání dat bývá nejčastěji použito enkodérů rychlosti a natočení s vysokým rozlišením. Nejvíce rozšířenými jsou pak enkodéry optické. Z principu měření je nutno podotknout, že se jedná o lokalizaci relativní s nutností znalosti počátečního bodu k určení pozice absolutní.

Typický rotační enkodér obsahuje děrovaný disk, přerušující trasu mezi zdrojem světla a optickým detektorem. Na výstupu je typicky měřena série čtvercových vln, kde jedna úroveň indikuje přerušení a druhá přímé ozáření senzoru. Následný počet náběžných a sestupných pulsů detekuje úhlovou rychlosť, ze které lze následně dopočítat polohu. V případě použití alespoň dvou párů zdroje světla a senzoru lze navíc identifikovat i směr pohybu [4].

Úhel natočení předních kol lze vyčítat při použití speciálního enkodéru. Zařízení obsahuje zakódovaný disk s předraženými řadami lomených oblouků. Tyto řady jsou vybaveny zdroji světla umístěnými tak, aby vyčítaný úhel byl jednoznačně určen.

Mezi nesporné výhody rotačních enkodérů patří jejich nízká cena, minimální výpočetní nároky, vysoká spolehlivost, rozlišení a přesnost. V důsledku těchto faktů, patří tento způsob vyčítání údajů o pohybu k nejpoužívanějším napříč automobilovým průmyslem, ale i robotikou jako takovou. Bohužel tento způsob určování polohy obsahuje ze svého principu mnoho nepřesností a akumulujících se chyb, způsobených rozdílným prokluzem pneumatik v závislosti na povrchu. Prokluz způsobený vysokou změnou akcelerace či prudké změny směru pohybu. Další nepřesnost měření zanáší změny v průměru kola. Nepřesnost mohou také způsobovat rozdílné tlaky v pneumatikách. Tyto rozdíly může zapříčinit původní nahuštění pneumatiky, ale i její teplota. Tento typ chyby se postupně integruje a způsobuje nepřesnosti, které je nutné následně softwarově eliminovat [4] [5].

2.2 Inerciální senzory

Původní inerciální senzory byly vyvinuty a používány pro vojenské účely. Tento typ měření byl a dodnes je využíván v letadlech, raketách, námořních a obrněných vozidlech. První varianty těchto měření nebyly, pro jejich rozměry a hmotnost, použitelné v tradičních vozidlech, tak jak jsou známy dnes. Postupný vývoj mikro-elektronických senzorů a navýšení výpočetní kapacity, umožnilo využití těchto senzorů i pro automobilový průmysl.

Inerciální senzory typicky obsahují senzory lineárního, úhlového zrychlení a magnetického pole. Tyto jednotky následně poskytují dostatečné údaje pro výpočet pohybu a rotace.

- Zrychlení zajišťují v IMU jednotce tři ortogonální akcelerometry, poskytující data o zrychlení v 3D prostoru.
- Informaci o úhlové rychlosti IMU jednotce dávají tři ortogonální gyroskopy.
- Magnetické pole měří tři magnetometry, jejichž princip byl využíván v kompasech. Tento senzor měří sílu a orientaci magnetického pole Hallovým nebo magnetorezistivním jevem.

Následný výpočet rychlosti, polohy a natočení je založen na znalosti počátečních parametrů. Nezbytné počáteční parametry jsou výchozí pozice, počáteční rychlosť a úhel natočení v předem určené souřadné soustavě. Výsledný pohyb je vypočítán integrací akcelerace a příslušné úhlové rychlosti. Aktuální pozice tedy může být specifikována šesti parametry. Lineární akcelerace je reprezentována třemi parametry podle os x, y, z. Rotační komponentou je pak úhlová rychlosť podle osy x, y, z [4].

Celý proces určování polohy za pomoci inerciální jednotky probíhá na palubě vozidla, což je zároveň jeho největší přednost. Další důležitou výhodou tohoto měření je nezávislost měření na typu povrchu či dosahu signálu a je tedy použitelný v hustě osídlených místech, tunelech, lesích a místech, kde ostatní metody selhávají. Jako slabina tohoto typu měření může být zmíněn právě samotný princip určování rychlosti a polohy, který je závislý na znalosti počátečních stavů, obdržených z jiného typu senzoru a integrující se chyba, která v průběhu měření narůstá a zanáší tak vysoké odchylky. Filtrování šumu a určování výsledné pozice bude věnována část 3.2.3 [4].

■ 2.3 Vizuální odometrie

Vizuální odometrie je mladý a stále se rozvíjející obor, který sbírá informace z okolí vozidla za pomocí kamer. Společně s radarovými a lidarovými daty je základním kamenem autonomního řízení. Na rozdíl od dále zmíněných senzorů jako jediný sbírá informace významem srovnatelné s lidským okem. Dodatečná algoritmizace mu pak dává možnost porozumění těmto záznamům a následnému rozhodnutí. V porovnání s ostatními senzory sbírá v obraze daleko více informací, jimiž jsou barvy, textury a různé příznaky využitelné v následné softwarové identifikaci. Kombinace kvalitního záznamu a grafického výpočetního výkonu mu umožňuje identifikovat objekty, povrch vozovky, detektování a sledování pruhů, rozpoznávání značek, světelného značení a mnoha dalších možností.

Principem je nalezení korespondujících bodů a následné určování rotace a translace kamery. Obecně se dají přístupy rozdělit na dvě kategorie podle použitého typu kamery. První typ je založený na kamerách disponujících jednou čočkou, ale tento přístup postrádající hloubkové mapy je méně přesný. Druhá metoda je založená na stereo-kamerách, které využívají podobnosti na základě jednočočkových záznamů a hloubkových map současně. Nejlepších výsledků v této oblasti dosahuje metoda ORB-SLAM2, která je podrobně rozebrána v publikaci ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras [6].

Vizuální sběr informací má mnoho výhod, ale jako většina způsobů měření má i svoje nedostatky. Výpočetní náročnost mnohonásobně převyšuje porovnávané metody. Rozdíly v měrené scéně citelně ovlivňují podmínky počasí a denní doby. Dále to jsou nečistoty na čočce, námraza a zamlžení.

2.3.1 Stereo-kamera

Mezi nejvíce rozšířené senzory lze zařadit stereo-kameru, která obsahuje dvě čočky a tudíž umožňuje mimo detekce objektů určit i jejich vzdálenost. Se správně kalibrovanou stereo-kamerou lze například porovnat dvě stejné scény zaznamenané z různých perspektiv. Jednou z nejpoužívanějších metod určení polohy bodu je princip triangulace. Triangulační metodou lze při znalosti průsečíku paprsků, ohniskové vzdálenosti a relativní pozice kamery, určit polohu bodu z obrazu. Určení vzdálenosti objektu je možné i s kamerou obsahující pouze jeden objektiv, ale výsledná nepřesnost je znatelně vyšší [4] [7].

2.3.2 Všesměrová kamera

Omnidirectional camera neboli všesměrová kamera je zařízení sbírající obraz z 360° záběru. Na trhu je mnoho těchto zařízení lišících se převážně v délce dohledu, rozlišení a obnovovací frekvenci. Daleký dohled je v případě těchto kamer důležitým parametrem, jelikož jej mají oproti kamerám tradičním nižší a při lokalizaci v prostoru a rozpoznávání objektů narážejí na svoje limity. Modelování prostoru je v případě všesměrové kamery více komplexní v porovnání s kamerou perspektivní [4].

2.4 Lidar a Radar

Mezi nejvíce komerčně používané indikátory vzdálenosti v automobilovém průmyslu patří lidarové a radarové zařízení. Toto zařízení poskytuje vozidlu robustní informaci o vzdálenosti předmětů v okolí. Přístroj obsahuje zdroj vysílající pulzní signál do okolí a čidlo snímající jeho odraz.

■ 2.4.1 Lidar

Lidarové měření neboli *light detection and ranging* probíhá na principu vysílání pulsů světelného paprsku. Paprsek je následně vyčítán na čidle a vzdálenost objektu je dopočítána na základě principu TOA *time of arrival*. Lidarové detektory lze rozdělit na dvou-dimenzionální a tří-dimenzionální, jež oba vyhodnocují vzdálenost za pomoci TOA. Většina lidarů sbírá paprsky za pomocí *oscillating mirror* pro dvou-dimenzionální zařízení a *rotating mirror* pro zařízení tří-dimenzionální. Zařízení má široký úhel záběru a vysoký dosah. Existuje názor, že je Lidar vhodným senzorem pro autonomní řízení, nicméně vzhledem k jeho vysoké ceně, nízké životnosti, velikosti a náchylnosti na atmosférické podmínky, panuje i názor opačný. Nadějí v tomto odvětví jsou Solid-State Lidary, které nedisponují mechanickou částí, ale zatím nejsou na trhu [4] [8].

■ 2.4.2 Radar

Radar může být brán jako alternativa k senzoru lidarovému. Radarové měření na rozdíl od toho lidarového, probíhá na principu vysílání rádiových vln. Odražená vlna je následně vyčítána na čidle a vzdálenost je dopočítávána TOA principem. Rádiová a světelná vlna se šíří stejnou rychlostí, ale rádiové vlny mají mnohem nižší frekvenci. Na rozdíl od Lidaru může brát radar v potaz Dopplerův jev. Na základě toho principu je schopen přímo určit rychlosť pohybu objektu. Může fungovat na větší vzdálenosti, je méně náchylný ke změnám počasí a bývá i cenově dostupnější. Nicméně vlnová délka rádiových vln je větší oproti světelným, což má za následek nižší rozlišení. V praktických případech nalezení nebo sledování objektu využití radaru převyšuje to lidarové. V současné době využívají principu radaru, adaptabilní tempomaty a systémy nouzového brzdění [4].

■ 2.5 Globální navigační satelitní systém

GNSS neboli *Global Navigation Satellite System* je způsob určení polohy ze signálů vysílaných družicemi na oběžné dráze Země. Jedná se o množinu satelitů obíhajících Zemi, tak aby bylo co nejvíce míst trvale pokryto signálem. Z poskytnutého signálu jsou GNSS přijímače schopny určit polohu a přesný čas.

GNSS umožňuje přijímači určit *real-time* pozici a přesný čas. Určení vzdálenosti je založeno na principu *Time Of Arrival* dáleznáčeného TOA. Přijímač měří dobu přenosu signálu od okamžiku vyslaní družicí a vyčtení na přijímači. Výpočet vzdálenosti je proveden vynásobením rychlostí šíření signálu. Kvalita GNSS služby bývá hodnocena podle několika kritérií [4].

- Rozdílem mezi naměřenou a reálnou pozicí
- Schopností systému vyhodnotit možnou odchylku
- Schopností systému fungovat bez přerušení
- Dostupností neboli procentuálně vyjádřeného času, po který systém splňoval předchozí kritéria.

Mezi nejpoužívanější a hlavně samostatně fungující služby lze zařadit americký polohovací systém GPS, evropské Galileo, ruský GLONASS a čínský BeiDou. Kombinací služeb těchto poskytovatelů se lze dostat na přesnější výsledky než samostatně.

2.5.1 GPS

Americký družicový systém byl plně zprovozněn v roce 1994. Přesný, souvislý, třídimenzionální systém dodávající informace o poloze, rychlosti a přesném čase je formace minimálně 24 satelitů umístěných na 6 orbitálních rovinách. GPS pro svoji činost využívá CDMA (*code division multiple access*) multiplex, jež je vysílán na dvou frekvencích L1 a L2. Frekvenční pásmo L1 je 1,575.42 MHz a L2 operující na frekvenci 1,227.6 MHz. Důležitou vlastností GPS je neomezený počet pasivních uživatelů [4] [9].

2.5.2 Galileo

Galileo je evropský navigační systém, který byl financován z prostředků Evropské Unie a je odpovědí na americký, ruský, čínský a ostatní systémy. V současné době je Galileo nejpřesnějším volně dostupným systémem. Systém má být tvořen 30 operačními družicemi, které budou obíhat ve výšce okolo 23 000 km nad povrchem země. Mezi další přednosti patří, že se nejedená o vojenský projekt, takže je minimalizováno riziko výpadku v případě vojenského konfliktu. Novinkou je *Search and Rescue* zajíšťující oboustrannou komunikaci a lokalizaci pro případ automobilové či jiné nehody [9].

■ 2.5.3 GLONASS

GLONASS je ruský vojenský družicový systém financovaný ruskou vládou. Umístěn je ve výšce okolo 19 100 km nad zemským povrchem, což mu dává výhodu při pokrytí v oblastech okolo pólů. Sestává z 21 aktivních družic, využívajících modulaci FDMA *Frequency Division Multiple Access* [4].

■ 2.5.4 BeiDou

BeiDou Navigation Satellite System byl vyvinut čínskou lidovou republikou za účelem potřeb ekonomických, sociálních a bezpečnosti. První verze čínského navigačního systému byla dokončena v roce 2000 a měla označení BDS-1. Systém BDS-1 pokrýval pouze území Čínské lidové republiky a její bezprostřední okolí. Druhá verze s označením BDS-2 byla dokončena roku 2012 a pokrývá celý asijsko-pacifický region. Aktuální verze BDS-3 pokrývá celý svět a je tak čínskou alternativou původním velkým hráčům, jako jsou GPS, GLONASS a další [10].

■ 2.6 Další metody

VANET neboli *Vehicular Networks* je komunikační systém zajišťující lokalizaci, zlepšení bezpečnosti a maximalizaci využitelnosti komunikací. VANET je typ mobilní ad-hoc sítě pro sdílení informací o okolí mezi samotnými auty, ale i jejich vzájemné lokalizace. Pro tyto účely bylo rezervováno 5.8/5.9 GHz pásmo, jež je nyní využíváno pro chytré funkce a nástroje. Mezi její aplikace patří řízené skupiny vozidel, upozornění na kolize, sdílení dopravní situace či přítomnost zvířat a překážek na komunikacích [4].

Optická lokalizace je obor, kde je za pomocí externích objektivů sledován cíl, který není vybaven senzorkou nebo zdrojem signálu. Tento způsob bývá často znám pod názvem jeho největších průkopníků společnosti VICON. V tomto případě si objektiv musí vystačit s informací v obraze, kde cíl může ale nemusí být vybaven charakteristickým prvkem. V prvním případě je externím objektivem sledován cíl mající připevněny charakteristické značky, kterými mohou být LED diody, či unikátní kódy. V případě selhání původních navigačních systémů, pokud cíl nechce být lokalizován, nebyl nebo nemohl být vybaven lokalizačním zařízením, si musí systém vystačit s pouhým obrazem.

Kapitola 3

Algoritmy pro měření polohy

3.1 GNSS

V minulé kapitole byla rozebrána většina samostatně fungujících poskytovatelů a jejich technologií. V praxi se využívá, pokud je potřeba dosáhnout co nejpřesnějších výsledků, kombinace družic různých poskytovatelů. Z každé dostupné družice lze dopočítat vzdálenost s určitou přesností. Tyto vzdálenosti se algoritmicky a podle přesnosti nakombinují a následně z nich lze určit pozici.

3.1.1 Zjištování polohy

V oboru globálních navigačních satelitních systémů existují dvě metody zjištění polohy.

- Jednocestné měření je způsob výpočtu času, kde přijímač nekomunikuje s družicí. Přijímač tedy přijme signál z několika družic současně a výslednou vzdálenost dopočítá v měřeném bodě. Tyto družice a přijímače musí mít precizně synchronizovaný čas v jednotkách nanosekund.
- Dvoucestné měření je způsob, kdy určovaný bod přijme signál z družice a následně jej odešle zpět na družici, která dopočítá časový rozdíl mezi

odeslaným a obdrženým signálem. Výhodou toho měření je, že čas odeslání a přijmutí signálu je měřen na stejném zařízení, čímž lokalizace dosahuje daleko lepsí přesnosti. Vzhledem k faktu, že vzdálenost mezi vysílačem a přijímačem je větší než 20 000 km, a tudíž nároky na vyslání signálu zpět na družici jsou vysoké i vzhledem k faktu, že je schopna obsloužit jen omezený počet uživatelů, nedává technologie prostor ke komerčnímu využití a tento způsob se nevyužívá [4].

Jelikož se dvoucestné měření v komerčním využití nepoužívá, budeme se následně věnovat pouze typu jednocestnému. V případě, že by hodiny vysílače a přijímače byly synchronizovány, lze dopočítat vzdálenost ze vzorce (3.1), kde R_i je vzdálenost přijímače a jednoho konkrétního vysílače, c je rychlosť šíření signálu a τ je časový rozdíl mezi vysláním a příjmem signálu.

$$R_i = c\tau \quad (3.1)$$

Jelikož každá hodnota R_i v tří-dimenzionálním prostoru vykreslí sférický povrch, je k určení polohy přijímače nutno vyčítat minimálně ze tří družic. Nicméně v reálných situacích je přesná synchronizace hodin družice a přijímače komplikovaná. Z tohoto důvodu bylo zavedeno měření posunu času ϵ každého satelitu, jehož pomocí lze synchronizovat čas na lokálním přijímači s časem na družici. Měřená veličina nazvaná *pseudorange* ρ je definována vzorcem (3.2), kde c je rychlosť šíření signálu a δt_u je posun hodin na přijímači. Tento vztah může být přepsán na rovnici (3.3), kde x_u, y_u, z_u jsou souřadnice přijímače a x_{sj}, y_{sj}, z_{sj} jsou souřadnice j-tého satelitu [9].

$$\rho_j = R_i + \epsilon = R_i + c\delta t_u = R_i + b_{ut} \quad (3.2)$$

$$\rho_j = \sqrt{(x_{sj} - x_u)^2 + (y_{sj} - y_u)^2 + (z_{sj} - z_u)^2} + b_{ut} \quad (3.3)$$

Z těchto rovnic vyplývá, že pro zjištění posunu času je potřeba signálu z minimálně čtyř družic a s každou další družicí se přesnost zlepšuje [4].

■ 3.1.2 Minimalizace odchylky

Výpočet vzdálenosti je použitelný pouze v případě, kdy byl posun časových stupnic určen s přesností v jednotkách nanosekund. Posun stupnic o $1\mu s$ znamená poziční posun o 300 m. Při měření vzniká několik systematických chyb [9].

- Odchylka 10ns atomových hodin satelitu způsobí rozdíl tří metrů.
- Pozice oběžné dráhy satelitu je známa na 1 – 5m.
- Rychlosť světla se mění při průchodu ionosférou a troposférou a není tedy konstantní.
- Přijímač je schopen detektovat pouze příchozí čas a je závislý na přesnosti synchronizace.
- Přijímač vyčítá i odražené signály.
- Určení časového posunu je nejpřesnější, pokud jsou satelity blízko u sebe, což je způsobeno *Dilution Of Precision*.

Původní navigační systémy nebyly vyvíjeny a určeny pro komerční využití a na přesnost jednosměrné metody nebyl kladen takový důraz. Pro zlepšení přesnosti existuje mnoho způsobů, kterými lze téměř eliminovat počet zdrojů chyb měření. Mezi nejznámější patří.

- Numerické metody pro Dilution Of Precision
- DGPS neboli Diferenciální GPS pro zlepšení přesnosti polohy
- SBAS neboli *Satellite Based Augmentation System* pro zlepšení přesnosti a spolehlivosti polohy
- A-GPS neboli *Assisted-GPS* pro zlepšení citlivosti v uzavřených prostorách
- HSGPS neboli *High Sensitivity-GPS* pro zlepšení rozlišení

■ 3.1.3 DGPS

Princip diferenciální GPS je velice účinný. Měřící systém obsahuje navíc oproti klasickému měření jednu další referenční stanici(měřící bod), který je

umístěn na předem lokalizovaném místě. Základní stanice určí na základě dlouhodobého měření svoji pozici. Na základě znalosti své polohy určí odchylku vzdálenosti a *pseudorange* každého satelitu. Tato odchylka je stejná pro všechny GNSS přijímače v blízkém okolí, avšak je použitelná v okolí až 200 km. Korekce může být aplikována v reálném čase i dodatečně. Důležité však je, aby odchylka byla aplikována na *pseudorange* měření a nikoliv na výslednou pozici. V případě přímé korekce na výslednou polohu může vést i ke zvětšení odchylky pozice. Proces korekce může být popsán následujícími třemi kroky.

1. Určení korekce na základní stanici pro každou družici na základě její známé a aktuálně vyčtené pozice
2. Přenos korekčních hodnot *pseudorange* na měřený bod
3. Úprava cílových *pseudorange* a následné dopočítání polohy

Tímto způsobem mohou být odstraněny všechny chyby, které nejsou způsobeny rušením na přijímači nebo odrazem signálu. Přenos korekčních hodnot mezi referenční stanicí a měřeným bodem může být proveden přes různá rozhraní, ale zpoždění výpočtu je pak závislé na rychlosti a zpoždění komunikačního rozhraní.

■ 3.2 Inerciální senzory

Inerciální senzory jsou často využívány pro měření polohy a orientace. Integrace úhlové rychlosti měřené na gyroskopu poskytuje hodnotu změny úhlu pro daný časový úsek. Při odečtení gravitačního zrychlení lze dvojitou integrací akcelerometru získat polohu senzoru, což je podmíněno znalostí výchozí orientace senzoru. Proces integrace měření z inerciální jednotky bývá nazýván *Dead-Reckoning*. V případě precizní znalosti výchozí pozice senzoru a ideálního vyčtení hodnot nezatížených šumem, může proces vést k precizním výsledkům určení polohy.

V reálném měření je však senzor zatížen nezanedbatelným šumem a posunem *bias*. Z tohoto důvodu jsou následné výpočty zatíženy navýšující se chybou *integration drift*. Inerciálními senzory lze obecně měřit přesnou změnu pozice s vysokou frekvencí vzorkování, ale přesnost se zhoršuje při delších časech měření. Z tohoto důvodu bývá vhodné jejich výsledky kombinovat s další metodou jiného typu, kde se výsledky nezhoršují s narůstajícím časem

měření. Při měření akcelerací, úhlových rychlostí a magnetického pole Země rozlišujeme několik vztazných soustav [11].

1. *The body frame* b popisuje pohyb a rotaci měřeného tělesa.
2. *The navigation frame* n popisuje pohyb a rotaci vzhledem k mapě.
3. *The inertial frame* i popisuje pohyb a rotaci inerciální jednotky.
4. *The earth frame* e popisuje pohyb a rotaci vzhledem k magnetickému poli Země.

■ 3.2.1 Úhlová rychlosť

Gyroskop měří úhlovou rychlosť *body framu* s ohledem na *inercial frame* vyjádřený v *body framu*. Tato úhlová rychlosť může být vyjádřena vzorcem (3.4), kde R^{bn} je rotační matice z *navigation frame* do *body frame*, ω_{ie} je *earth rate* s ohledem na *inertial frame*. $\omega_{ie} = 7.28 \cdot 10^{-5} \text{rads}^{-1}$. V případě, kdy *navigation frame* není definován s ohledem na *earth frame*, *transport rate* ω_{en} je nenulový. Úhlová rychlosť, která je potřeba pro určení orientace *body frame* s ohledem na *navigation frame* se značí ω_{nb} [11].

$$\omega_{ib}^b = R^{bn}(\omega_{ie}^n + \omega_{en}^n) + \omega_{nb}^b \quad (3.4)$$

■ 3.2.2 Specific force

Akcelerometr měří *specific force* f v *body frame* b. Tento jev může být popsán vztahem (3.5), kde g značí gravitační vektor, a_{ii}^n značí lineární zrychlení v *navigation frame*. Výpočet a_{ii}^n popisuje vzorec (3.6), kde R^{ne} je rotační matice z *earth frame* do *navigation frame* a je rotace z *inertial frame* do *earth frame*. Proměnná a_{ii}^i značí zrychlení v *inertial frame*.

$$f^b = R^{bn}(a_{ii}^n - g^n) \quad (3.5)$$

$$a_{ii}^n = R^{ne}R^{ei}(a_{ii}^i) \quad (3.6)$$

Pro navigační účely a určení pozice může být lineární akcelerace vyjádřena vzorcem (3.7), kde p^n , v^n a a^n jsou pozice, rychlosť a akcelerace v *navigation frame* [11].

$$\frac{d}{dt}p^n|_n = v^n \quad \frac{d}{dt}v^n|_n = a_{ii}^n \quad (3.7)$$

■ 3.2.3 Chyby senzorů a jejich korekce

Jak již bylo zmíněno gyroskopická a akcelerační měření jsou zatíženy chybami, jimiž jsou převážně časově proměnlivý *bias* a přítomnost šumu měření. Pro kalibraci gyroskopu stačí zařízení nechat v klidu a měřit vyčítané hodnoty. V případě akcelerometru to tak snadné není, jelikož do os měření vstupují hodnoty gravitačního zrychlení, které se v případě i minimálního vychýlení osy z oproti ose gravitačního pole Země, projevuje ve více osách současně, a není tedy možné gravitační zrychlení odečíst pouze z osy z. Této problematice se věnuje publikace Using Inertial Sensors for Position and Orientation Estimation [11].

■ 3.2.4 Kalmanův filtr

Kalmanův filtr je algoritmus vyvinutý matematikem Rudolfem Kálmánem. Tento algoritmus je schopen z dat, zatížených nepřesnostmi měření a šumem, odhadnout neznámé hodnoty proměnných. K tomuto výpočtu využívá model systému, naměřená data a údaje o předchozím stavu systému. Jeho přínos je široce využíván napříč zpracováním signálu, navigací, robotikou a mnoha dalšími obory [12].

Algoritmus ve formulaci stavového prostoru lineárních dynamických systémů poskytuje rekurzivní řešení pro lineární optimální filtrační problém. Každý další krok je určen z kroku předchozího a nových naměřených hodnot. Je důležité podotknout, že je Kalmanův filtr výpočetně efektivní v porovnání s konkurenčními metodami. V případě, že se jedná o lineární a diskrétní dynamický systém, je možno princip algoritmu rozdělit na dvě části *Process equation* a *Measurement equation* [12].

Process equation je popsán rovnicemi (3.8) a (3.9), kde $F_{k+1,k}$ je matici přenosu, x_k je stav k-tého vzorku a w_k je šum s nulovou střední hodnotou a kovarianční maticí $E[w_k w_k^T]$ [12].

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k} \mathbf{x}_k + \mathbf{w}_k \quad (3.8)$$

$$E[\mathbf{w}_n \mathbf{w}_k^T] = \begin{cases} \mathbf{Q}_k & \text{pro } n = k \\ \mathbf{0} & \text{pro } n \neq k \end{cases} \quad (3.9)$$

Measurement equation je popsán rovnicemi (3.10) a (3.11), kde y_k je pozorovatelný výstup, H_k je matici měření a v_k je šum měření s nulovou střední hodnotou a kovarianční maticí $E[v_k v_k^T]$ [12].

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (3.10)$$

$$E[\mathbf{v}_n \mathbf{v}_k^T] = \begin{cases} \mathbf{R}_k & \text{pro } n = k \\ \mathbf{0} & \text{pro } n \neq k \end{cases} \quad (3.11)$$

3.2.5 Rozšířený Kalmanův filtr

V předchozí kapitole byl jako výchozí předpoklad lineární diskrétní dynamický systém. V případě, že je model systému nelineární, může být Kalmanův filtr rozšířen o linearizační proces. Tomuto filtru se říká Rozšířený Kalmanův filtr dále nazývaný jenom EKF. Základní myšlenkou filtru je tedy linearizovat stavový model respektive rovnice (3.12) a (3.13), kde w_k a v_k jsou nezávislé šumové procesy s kovariančními maticemi R_k a Q_k . Funkce $f(k, x_k)$ a $h(k, x_k)$ značí nelineární matice přenosu a měření [12].

$$\mathbf{x}_{k+1} = \mathbf{f}_k \mathbf{x}_k + \mathbf{w}_k \quad (3.12)$$

$$\mathbf{y}_k = \mathbf{h}_k \mathbf{x}_k + \mathbf{v}_k \quad (3.13)$$

K linearizaci je využita Taylorova approximace prvního řádu. Po použití approximace dostaneme rovnice (3.14) a (3.15), kde je nutno zadefinovat dvě nové množiny (3.16) a (3.17), kde $\bar{\mathbf{y}}_k$ může být považován za vektor pozorování v čase n a d_k za plně pozorovatelné vstupy [12].

$$\mathbf{x}_{k+1} \approx \mathbf{F}_{k+1} \mathbf{x}_k + \mathbf{w}_k + \mathbf{d}_k \quad (3.14)$$

$$\bar{\mathbf{y}}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (3.15)$$

$$\bar{\mathbf{y}}_k = \mathbf{y}_k - \{\mathbf{h}(\mathbf{x}, \hat{\mathbf{x}}_k^-) - \mathbf{H}_k \hat{\mathbf{x}}_k^-\} \quad (3.16)$$

$$\mathbf{d}_k = \mathbf{f}(\mathbf{x}, \hat{\mathbf{x}}_k) - \mathbf{F}_{k+1,k} \hat{\mathbf{x}}_k \quad (3.17)$$

■ 3.2.6 Invariantní Rozšířený Kalmanův filtr

Invariant Extended Kalman filter je metoda založená na EKF využívající alternativních souřadnic získaných z externích zdrojů, čímž řeší nekonzistentnost EKF v důležitých aplikacích SLAM úloh. Přínosem IEKF je konvergence, stabilita a robustnost, ve kterých standardní EKF selhává, a jeho využití je, pro lokalizaci a navigaci napříč automobilovým průmyslem [13].

Kapitola 4

AI-IMU Dead-Reckoning

Výchozím bodem této práce je publikace s názvem "AI-IMU Dead-Reckoning". Článek byl publikován skupinou vědců na univerzitě MINES ParisTech v roce 2019. Vědci Martin Brossard, Axel Baray a Silviere Bonnabel v něm publikují novou lokalizační metodu. Publikace vychází z potřeby vyplnit nedostatky lokalizace v oblastech, kde standardní metody selhávají. Současně je algoritmus založený na měření inerciálních senzorů, které jsou levnější v porovnání s konkurenčními způsoby měření. Všechny výsledky jsou testovány na profesionální Kitti databázi, jež byla naměřena v centru města Karlsruhe v roce 2011 skupinou vědců z Karlsruhe Institute of Technology ve spolupráci s Toyota Technological Institute at Chicago. Na základě publikovaných výsledků lze soudit, že metoda může dosahovat podobné přesnosti pouze z inerciálních měření jako algoritmy založené na Vizuální odometrii, Radaru nebo Lidaru [1].

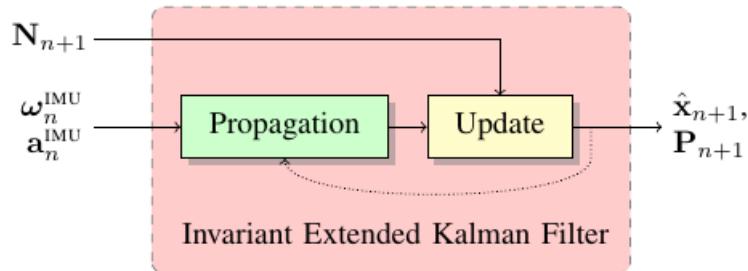
Měření je založeno na získávání dat z inerciální jednotky. Klíčovým prvkem zpracování inerciálních dat, je Invariantní Rozšířený Kalmanův filtr v kombinaci s *Deep Neural Network*, která se trénuje na šumové charakteristiky měření a následně dynamicky mění parametry filtru, pro jeho přesnější výsledky [1] [14].

4.1 Princip

Jak již bylo zmíněno, princip lokalizace vychází z filtrování pomocí Kalmanova filtru, který byl navržen pouze pro lineární stavový prostor. Z tohoto

důvodu je vhodnější *Extended Kalman Filter* nadále popisovaný EKF, který využívá navíc Taylorovu approximaci, čímž uzpůsobuje původní filtr na nelineární systémy. Tento algoritmus předpokládá předchozí znalost či manuální ladění kovariančních matic procesního šumu a šumu měření, které jsou pevně specifikovány pro celý systém [15].

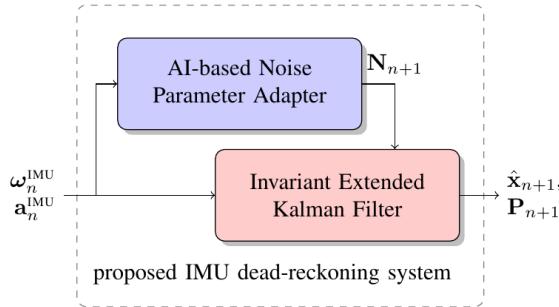
Nástavbou EKF je *Invariant Extended Kalman Filter* dále nazývaný jen IEKF. IEKF vyplňuje nekonzistentnost EKF za pomocí vnější dodatečné znalosti systému. Schéma principu je k vidění na obrázku 4.1. Vstupy algoritmu jsou šumové kovarianční matice $N_{n+1} \in \mathbf{R}^{2 \times 2}$, měřené úhlové rychlosti a lineární zrychlení ω_n^{IMU} a a_n^{IMU} náležící inerciální vztažné soustavě. Výstupem systému jsou předpokládané stavy \hat{x}_{n+1} a předpokládaná kovariance P_{n+1} . Množina stavů \hat{x}_{n+1} obsahuje v tomto případě rotaci, rychlosť, pozici, bias gyroskopu, bias akcelerometru a rotaci s translací mezi vztažnou soustavou inerciální jednotky a auta [1].



Obrázek 4.1: Struktura IEKF [1]

Systém lze rozdělit do dvou základních bloků. Schéma propojení a přenášení data jsou k vidění na obrázku 4.2.

1. *AI-based Noise Parameter Adapter* v reálném čase určuje nejvhodnější šumové kovariační matice N_n . Tento proces založený na základě *Deep Neural Network Learning*, přijímá měřená a neupravená inerciální data, z nichž odhadne již zmíněné šumové kovarianční matice, bez nutnosti dodatečné znalosti předchozích stavů.
2. IEKF integruje data inerciálního měření, která v souvislosti s dynamickým modelem a šumovými kovariančními maticemi, převede na odhad pozice \hat{x}_{n+1} a předpoklad kovariance P_{n+1} .

**Obrázek 4.2:** Struktura výchozího systému [1]

4.2 Prezentované výsledky

Vývojáři prezentují výsledky vyhodnocení měření z datasetu KITTI [16]. Publikované výsledky jsou následně porovnávány s nejpřesnějšími dostupnými a autorům známými metodami, kterými jsou IMLS, ORB-SLAM2. Dále poměřovaná IMU integrace se do této skupiny zařadit nedá, ale patrně je zde uváděna pro porovnání výsledků ze stejného zdroje dat [6].

- IMLS lokalizace je založena na vyčítání dat z Lidaru.
- ORB-SLAM2 metoda lokalizuje na základě vyčítání informací z kamer.
- IMU je nefitrovaná integrace z inerciálních dat.
- AI-IMU Dead-reckoning je metoda založená na IEKF v kombinaci hlu-bokým strojovým učením lokalizující na základě inerciálních měření.

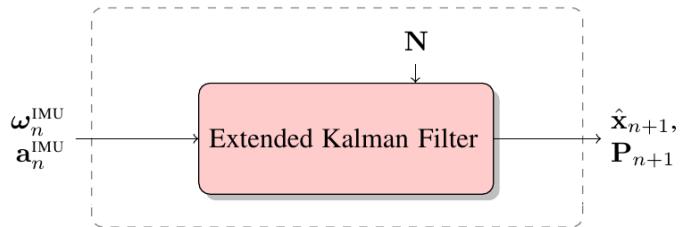
Z prezentovaných výsledků v práci [1], je možné pozorovat, že metoda na Kitti databázi dosahuje vynikajících výsledků viditelných v tabulce 4.1. Hodnota $t_{rel}^{avg} [\%]$ značí průměrný translační error v procentech z uražené vzdálenosti a hodnota $r_{rel}^{avg} [deg/m]$ značí průměrný rotační error ve stupních na metr. Kompletní rozbor metodiky testování lze dohledat v původní publikaci [1].

| Metoda | $t_{rel}^{avg} [\%]$ | $r_{rel}^{avg} [deg/m]$ |
|-----------------------|----------------------|-------------------------|
| IMU | 171 | 0.31 |
| IMLS | 0.98 | 0.12 |
| ORB-SLAM2 | 1.17 | 0.27 |
| AI-IMU Dead-Reckoning | 1.10 | 0.23 |

Tabulka 4.1: Porovnání algoritmů

4.3 Doplňující testování

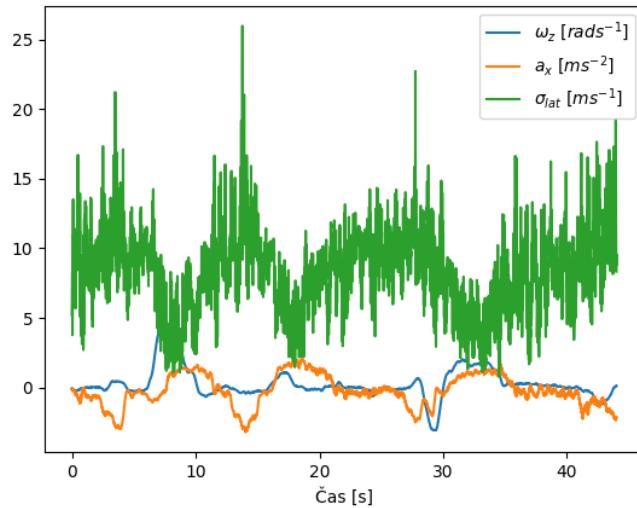
V původní publikaci nicméně chybělo komplexnější porovnání přínosu hluboké neuronové sítě, jehož doplnění se věnuje následující kapitola. Oproti původní publikaci byla přidána varianta lokalizace založená na EKF, jenž na rozdíl od publikované varianty nemá dynamicky se měnící šumové kovarianční matice popsány v kapitole 4.1. Vynecháním bloku s neuronovou sítí a její nahrazení koeficientem bylo dosaženo funkcionality popsáné v kapitole 3.2.5 a vizualizované na obrázku 4.3.



Obrázek 4.3: Testovaný EKF

Výrazná změna těchto hodnot je patrná hlavně v případě zatáčení a změny akcelerace. Tento trend je dobře viditelný v obrázku 4.4. Zde σ_{lat} a σ_{up} jsou laterální a vertikální parametry šumu kovarianční matice měření.

$$N_i = \text{diag}(\sigma_{lat}, \sigma_{up}) \quad (4.1)$$



Obrázek 4.4: Změny kovariančních šumových matic

Šumové kovarianční matice N_i byly nahrazeny průměrnou odhadovanou hodnotou napříč celým datasetem, což simuluje případ, kdy vývojáři naladí filtr manuálně. Strukturu a průměrné hodnoty ukazují rovnice (4.1), (4.2) a (4.3).

$$\sigma_{lat} = 1.02 \quad (4.2)$$

$$\sigma_{up} = 8.37 \quad (4.3)$$

Jako testovací metrika je zde využit *EndPoint Error* dále značen jako EPE. EPE je definován jako euklidovská vzdálenost mezi referenčním a odhadovaným bodem na mapě, popsáný vzorcem (4.4), kde EPE_i je vzdálenost referenčního a odhadovaného bodu i-tého vzorku. x_i, y_i, z_i jsou souřadnice referenčního bodu a $\hat{x}_i, \hat{y}_i, \hat{z}_i$ jsou souřadnice odhadované polohy. CumE je definován jako kumulativní součet EPE přes celou trajektorie vypočítaný ze vzorce (4.5).

$$EPE_i = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \quad (4.4)$$

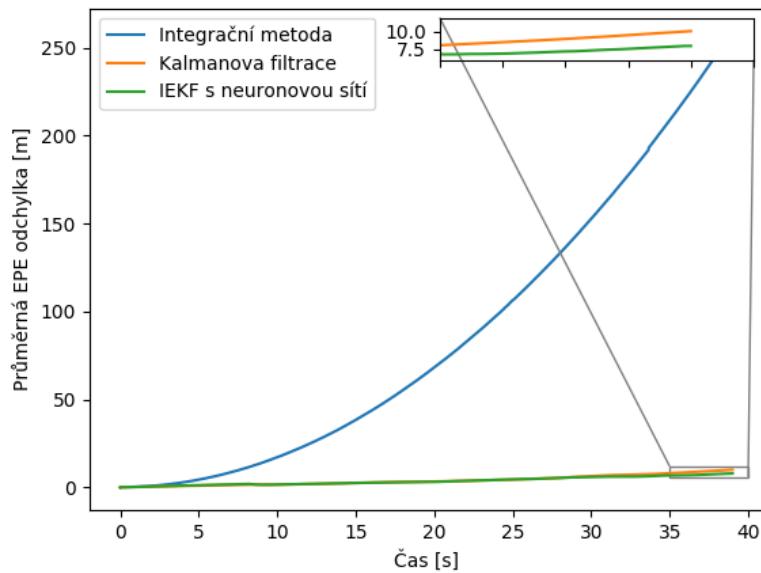
$$CumE_n = \sum_{i=0}^n EPE_i \quad (4.5)$$

Jelikož KITTI dataset obsahuje nespočet různých měření a trajektorií o různých délkách, všechny datasety jsou od počátku v pohybu nebo s okamžitým rozjezdem a IMU jednotka i v klidu neustále vyčítá šum, bylo zvoleno porovnávání jednotlivých jízd v závislosti na čase. Maximální měřený čas byl zvolen 40 sekund, protože 29 ze 42 datasetů obsahuje záznam minimálně této délky, a je tedy možné výsledky hodnotit. Měřenou a vizualizou hodnotou je průměrný EPE z jízd datasetu splňující předchozí podmínky. Výsledné porovnání metod založených na čisté integraci, EKF a AI-IMU Dead-Reckoning lze spatřit na obrázku 4.5.

Podle předpokladu se u Integrační metody akumuluje chyba, jak bylo popsáno v kapitole 3.2 a výsledky vedou k vysoké odchylce polohy v tomto případě přesahující 250 metrů po 40 sekundách jízdy. Zajímavých výsledků dosahují metody založené na Kalmanově filtraci, kde standardní EKF dosahuje průměrné odchylky 10 m. AI-IMU Dead-Reckoning s průměrnou odchylkou 8 metrů je nejlepší testovaným algoritmem a potvrzuje svůj přínos oproti standardním metodám.

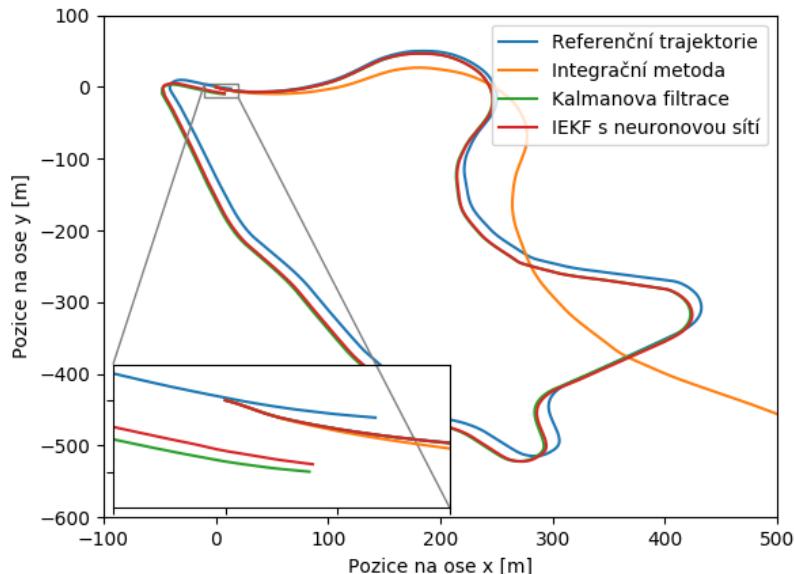
Následující experiment ukazuje, že zlepšení AI-IMU Dead-Reckoning metody oproti EKF algoritmu není velké. Závislosti zašumění datasetu a přínosům plynoucím z využití metody se budou věnovat kapitoly 5 a 6. Tuto závislost potvrdil i sám autor v komentářích u publikovaného projektu. Dalším přínosem toho principu je strojové učení šumových parametrů, které se mohou lišit při aplikaci na jiné vozidlo nebo při využití odlišných senzorů, čímž

4. AI-IMU Dead-Reckoning



Obrázek 4.5: Průměrný EPE založený na třech základních IMU metodách

zde odpadá nutnost jejich manuálního ladění. Pro znázornění lokalizačního zlepšení byl vygenerován obrázek 4.6. Na obrázku může být vyzdvihnuto zlepšení AI-IMU Dead-Reckoning algoritmu oproti EKF, ale i nepoužitelnosti čisté integrace pro precizní lokalizaci.



Obrázek 4.6: Lokalizace jízdy #33 2011/09/30

Kapitola 5

Experiment na simulátoru

V první fázi experimentu nebyla měřící TOMI platforma dokončena a nebylo známo, jak se bude algoritmus chovat na jiné databázi, než prezentované Kitti. K ověření bylo využito projektu GNSS-INS-SIM společnosti ACEINNA, Inc [2]. GNSS-INS-SIM je GNSS a INS simulační projekt, který na základě referenční trajektorie, generuje výstup GNSS měření, akcelerometrická, gyroskopická a magnetometrická data.

Uživatel toho generátoru může specifikovat různé varianty vstupu, kterými jsou například body trajektorie nebo rychlosti v různých osách. Mezi další vlastnosti generátoru patří možnost zvolit požadovanou frekvenci výstupních dat, variantu několika základních akcelerometrů nebo specifikaci vlastních šumových charakterů. Tato vlastnost umožňuje porovnávat funkčnost algoritmů, na základě různých úrovní zašumění včetně varianty absolutně bez šumu. Tímto lze testovat robustnost, ale i odhalovat chyby v algoritmech. V době tvorby této práce byl generátor stále v testovací verzi a mnoho funkcí, slibovaných autory, muselo být implementováno pro potřeby této práce. Seznam úprav je následující [2].

- import vlastních dat z referenčních bodů
- specifikace limitu akcelerometru
- specifikace limitu gyroskopu
- specifikace limitu magnetometru
- možnost formátování výstupních dat

5.1 Možnosti generátoru

Generátor poskytuje několik vzorových implementací. Jedná se především o způsoby využití knihovny, ale i různé způsoby zpracování a analýzy dat.

- demo na generování a ukládání dat na předdefinovaných trajektoriích
- demo analýzy gyroskopických a akcelerometrických dat na základě Allanova variance [17]
- demo popisující import vlastních dat

5.2 Dataset

Pro generování vlastního datasetu bylo využito trajektorií z původní KITTI databáze. Tento způsob byl zvolen i kvůli možnosti porovnání rozdílných dat a původních publikovaných výsledků.

5.2.1 Definování IMU modelu

V původní konfiguraci jsou přednastaveny tři IMU modely značené jako low-accuracy, mid-accuracy, high-accuracy, ale je možno definovat i vlastní model. Parametry vlastního modelu jsou:

- gyro bias
- gyro angle random walk
- gyro bias instability
- gyro bias instability correlation
- accelerometer bias
- accelerometer velocity random walk
- accelerometer bias instability

- accelerometer bias instability correlation
- magnetometer noise std

■ 5.2.2 Vytvoření souboru definujícího pohyb

Specifikaci pohybu zajišťuje soubor *motion file*. Obsah souboru lze rozdělit na dvě části. V první části jsou počáteční souřadnice, rychlosť a rotace. V té druhé lze naleznout posloupnost jednotlivých změn pohybu a doby, po kterou jsou tyto změny vykonávány. Vzorový příklad souboru definujícího pohyb, který nejdříve zůstane stát v počáteční pozici po dobu 200 sekund a následně se nakloní o 45 stupňů a rozjede na rychlosť 10 ms^{-1} . Tento pohyb bude vykonáván 250 sekund [1].

| ini lat (deg) | ini lon (deg) | ini alt (m) | ini vx_body (m/s) | ini vy_body (m/s) | ini vz_body (m/s) | ini yaw (deg) | ini pitch (deg) | ini roll (deg) |
|------------------|------------------|-------------------|-------------------------|-------------------------|-------------------------|------------------|-------------------------|-------------------|
| 32 | 120 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| command type | yaw (deg) | pitch (deg) | roll (deg) | vx_body (m/s) | vy_body (m/s) | vz_body (m/s) | command duration (s) | GPS visibility |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 1 |
| 5 | 0 | 45 | 0 | 10 | 0 | 0 | 250 | 1 |

Obrázek 5.1: Vzorový motion file [2]

Důležitým parametrem je přepínač *command type*, který nabývá hodnot v rozmezí 1-5, jejichž specifikaci lze nalézt v následujícím výčtu:

1. definuje úhlovou rychlosť a zrychlení
2. definuje úhel a rychlosť, které je třeba dosáhnout
3. definuje změnu polohy a změnu rychlosti
4. definuje absolutní pozici a změnu rychlosti
5. definuje změnu pozice a absolutní rychlosť

5.2.3 Výsledný dataset

Pro generování datasetu je nutné definovat trajektorie a příslušná natočení. K tomuto účelu bylo využito původní Kitti databáze, jejíž trajektorie, úhly natočení a časové značky sloužily jako ideální zdroj pohybu. Výsledný dataset dále nazýváme Synthetic Kitti.

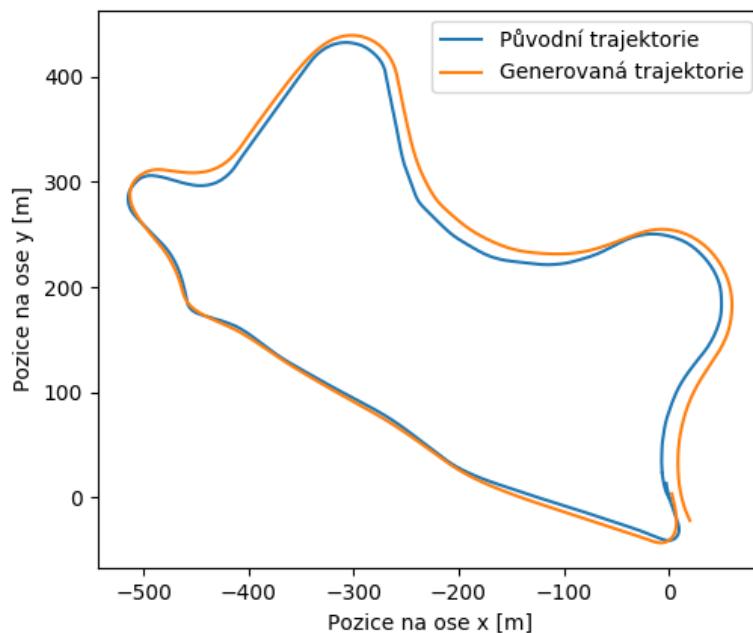
Dále bylo nutné nastavit šumové parametry. Tomuto účelu posloužily hodnoty dodávané výrobci IMU jednotek. Mezi levnějšími jednotkami tuto úlohu zastupuje AHRS380SA. Střední třídu reprezentuje IMU381 a vysoce precizní IMU jednotky zastupuje HG9900. Hodnota kitti-gen reprezentuje generovaný dataset s obdobnými parametry jako originální Kitti, kitti-orig reprezentuje uváděné hodnoty Kitti databáze a worse reprezentuje dataset horší kvality. Generátor umožňuje měnit 8 parametrů, ale hodnoty gyro bias a accelerometer bias byly ponechány nulové a v tabulce nejsou uvedeny. Hodnoty gyro bias instability correlation a accelerometer bias instability correlation byly nechány na původní hodnotě 100, a nejsou tedy v tabulce zaznamenány. Hodnoty gyro a accelerometer bias instability se nepodařilo pro Kitti dataset dohledat a jsou v tabulce označeny písmenem X.

| accuracy | gyro angle random walk (deg/h) | gyro bias instability (deg/h) | accelerometer velocity random walk (ms ⁻²) | accelerometer bias instability (ms ⁻²) |
|------------|---|-------------------------------------|---|---|
| high | 5.8 | 4.8 | 4.1 | 3.6 |
| | 5.8 10 ⁻⁷ | 4.8 10 ⁻⁷ | 4.1 10 ⁻⁷ | 3.6 10 ⁻⁶ |
| | 5.8 | 4.8 | 4.1 | 3.6 |
| mid | 7.3 | 1.7 | 5.0 | 5.0 |
| | 7.3 10 ⁻⁵ | 1.7 10 ⁻⁵ | 5.0 10 ⁻⁴ | 5.0 10 ⁻⁵ |
| | 7.3 | 1.7 | 5.0 | 5.0 |
| low | 2.2 | 4.8 | 8.3 | 2.0 |
| | 2.2 10 ⁻⁴ | 4.8 10 ⁻⁵ | 8.3 10 ⁻⁴ | 2.0 10 ⁻⁴ |
| | 2.2 | 4.8 | 8.3 | 2.0 |
| kitti-gen | 2.2 | 4.8 | 2.0 | 2.0 |
| | 2.2 10 ⁻³ | 4.8 10 ⁻⁵ | 2.0 10 ⁻² | 2.0 10 ⁻⁴ |
| | 2.2 | 4.8 | 2.0 | 2.0 |
| kitti-orig | 8.7 | X | 3.0 | X |
| | 8.7 10 ⁻² | X | 3.0 10 ⁻² | X |
| | 8.7 | X | 3.0 | X |
| worse | 2.2 | 4.8 | 2.2 | 2.0 |
| | 2.2 10 ⁻³ | 4.8 10 ⁻⁵ | 2.2 10 ⁻¹ | 2.0 10 ⁻⁴ |
| | 2.2 | 4.8 | 2.2 | 2.0 |

Tabulka 5.1: Specifikační tabulka generovaných datasetů

Z důvodu neúplnosti a občasných chyb v původním datasetu, bylo pro vygenerování souboru definujícího pohyb použito přepínače na hodnotu dvě, a tím definice pohybu na základě úhlu natočení a absolutní rychlosti. Tato volba zanesla do výsledné trajektorie odchylky, jak v původním měření rychlosti, tak v zaokrouhllování čísel. Porovnání trajektorie původní a generované je vidět na obrázku 5.2. Nicméně generování identické trajektorie nebylo cílem, a tudíž je tento dataset považován za výchozí a tato kapitola na něm bude postavena.

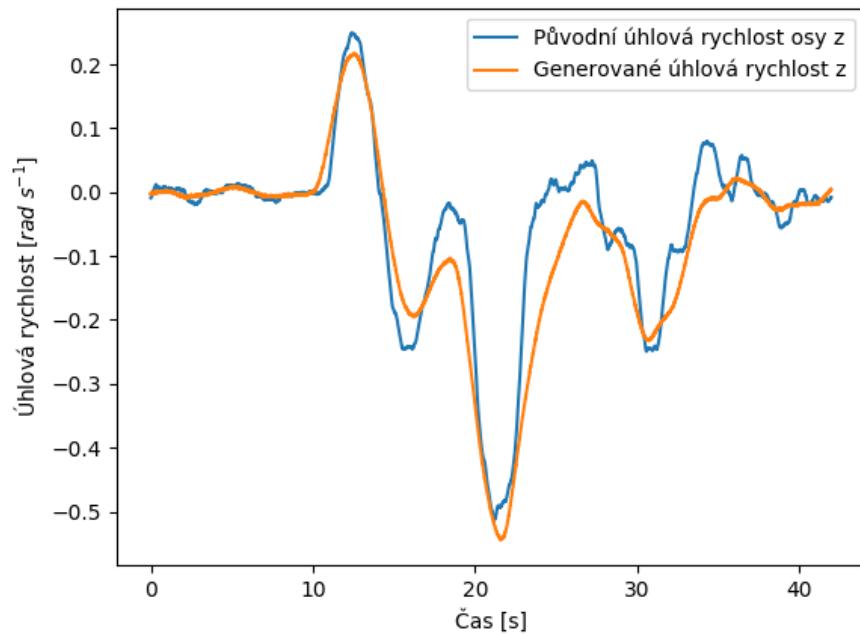
Dataset jako takový obsahuje 39 různých jízd, které dohromady čítají 1,5 hodiny. Všech předešlých 39 záznamů je generováno v šesti variantách zašumění, a tudíž celý dataset obsahuje přes 9 hodin záznamu. Odkaz na dataset je dostupný v příloze této práce.



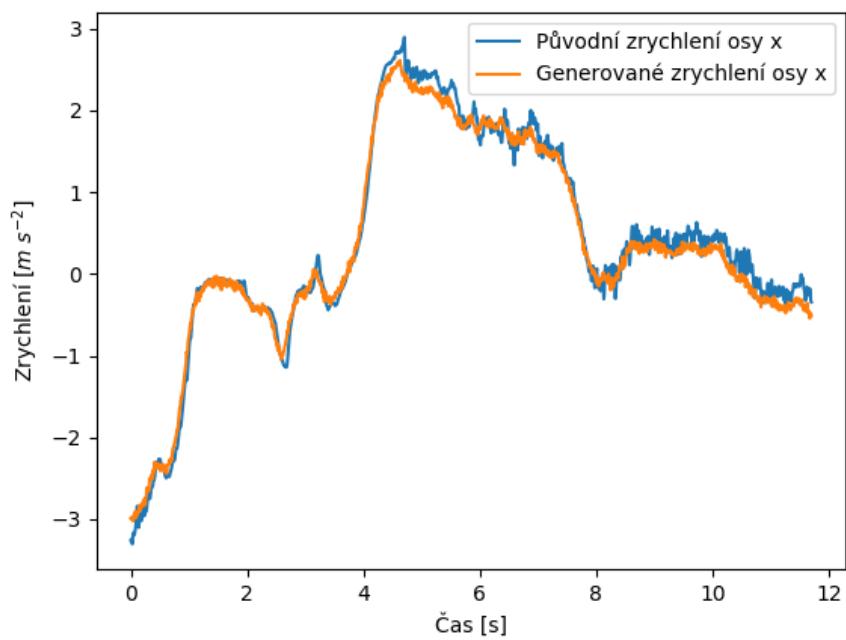
Obrázek 5.2: Porovnání originální a generované trajektorie

Pro porovnání původní a nově generované úhlové rychlost a lineárního zrychlení slouží obrázky 5.3 a 5.4. Na těchto obrázcích je možné pozorovat, že původní data obsahují rychlejší přechody a občasné záhvěvy, což je způsobeno převážně mechanickým rušením na reálném vozidle. Konkrétně se pak jedná o datasety kitti-original a kitti-generated.

5. Experiment na simulátoru



Obrázek 5.3: Porovnání originální a generované úhlové rychlosti na ose z



Obrázek 5.4: Porovnání originálního a generovaného zrychlení na ose x

5.3 Výsledky

Pro určení polohy a porovnání výsledků jsou využity 3 metody. Algoritmy jsou porovnávány se šesti úrovněmi zašumění datasetu. Použity byly šumové charakteristiky σ_{1-6} specifikované v tabulce 5.1 pod názvem *accelerometer velocity random walk*, jejichž přiřazení znakům v grafech specifikuje tabulka 5.2. Porovnávanými algoritmy jsou.

- Nefiltrovaná integrace na základě vzorců (5.1), (5.2) a (5.3).
- Rozšířená Kalmanova filtrace zastupující nejpoužívanější způsob určování vektoru rychlosti a polohy napříč různými obory, kde je využita implementace [1], jejíž princip je naznačen na obrázku 4.3.
- AI-IMU Dead-Reckoning metoda využívající výsledku práce [1].

Hodnoty σ_1 , σ_2 , σ_3 byly získány jako hodnoty uváděné vývojářem simulátoru pro IMU jednotky zastupující třídu vysoké, střední a nízké preciznosti měření a zastupují konkrétní přístroje HG9900, IMU381 a AHRS380SA. Hodnota σ_4 je úroveň šumu blížící se původní Kittti databázi a je uváděná v publikaci [1]. Zhoršenou úroveň zašumění reprezentuje hodnota šumu σ_6 .

| | | |
|--------------|------------|---------------------------------------|
| AHRS380SA | σ_1 | $4.1 \times 10^{-7} \text{ m s}^{-2}$ |
| IMU381 | σ_2 | $5.0 \times 10^{-4} \text{ m s}^{-2}$ |
| HG9900 | σ_3 | $8.3 \times 10^{-4} \text{ m s}^{-2}$ |
| Kitti gen. | σ_4 | $2.0 \times 10^{-2} \text{ m s}^{-2}$ |
| Kitti orig. | σ_5 | $3.0 \times 10^{-2} \text{ m s}^{-2}$ |
| Navýšený šum | σ_6 | $2.2 \times 10^{-1} \text{ m s}^{-2}$ |

Tabulka 5.2: Specifikační tabulka

Inerciální jednotka měří relativní veličiny, a je tedy nutno dodat algoritmu počáteční podmínky, kterými jsou R_0 neboli vzájemné natočení akcelerometru a mapy. Dále to je počáteční rychlosť v_0 a pozice na mapě p_0 . Nezbytnou znalostí pro správný výpočet je gravitační zrychlení, které bylo ponecháno předdefinované výrobcem s hodnotou $9,78 \text{ m s}^{-2}$. Jelikož se jedná o simulátor, mohla být vztažná soustava akcelerometru a vozidla považována za jednu a tu samou.

Pro porovnání výsledků metod byl zvolen graf typu boxplot. Zvolen byl pro jeho názorné zobrazení statistických hodnot. Čtverec vymezuje horní a dolní

kvartil s barevnou čárou uprostřed vyznačující medián. Čáry mimo čtverec vymezují $Q_1 - 1.5(Q_3 - Q_1)$ pro dolní čáru a $Q_3 + 1.5(Q_3 - Q_1)$ pro čáru horní, kde Q_i je kvartil. Všechny hodnoty vybočující z vymezené oblasti jsou značeny kolečkem.

5.3.1 Metoda nefiltrované integrace

V této kapitole byla pozornost zaměřena na zkoumání chování algoritmu integrace v závislosti na měnícím se šumu. Bias byl v tomto porovnání vynechán, jelikož může být eliminován kalibrací senzorů, která je popsána a odkázána v kapitole 3.2.

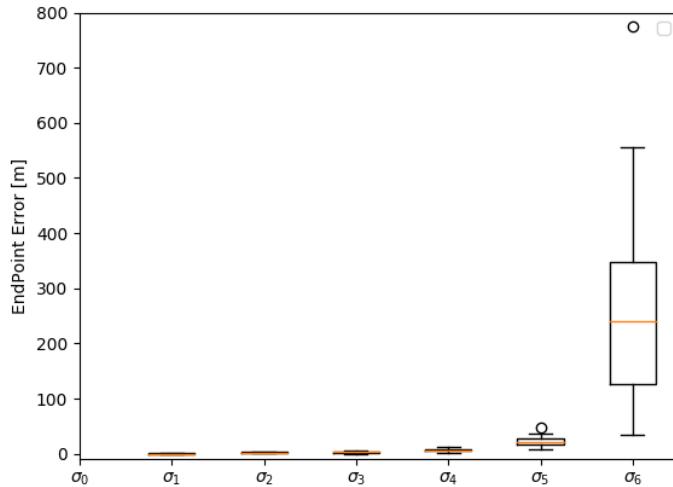
Pure IMU Integration neboli nefiltrovaná integrace inerciální jednotky je způsob výpočtu založený na vztažích popisujících kinematiku vozidla (5.1), (5.2) a (5.3). Kde ω_n je úhlová rychlosť získaná v čase n, jež je měřena na gyroskopu. Lineární úhlové zrychlení a_n je veličina měřená na akcelerometru. Hodnota dt značí rozdíl mezi časovými známkami a indikuje tak interval jednotlivých měření. Třídimenzionální rotační matice značená R_n^{IMU} , která určuje vzájemnou rotaci *world frame a IMU frame*. Zápis $\exp((\omega_n dt)_\times)$ značí vektorový součin *Skew-symmetric matrix* neboli Antisymetrických matic. Tato rotace se dá vyjádřit i obdobně jako maticový součin rotačních matic $R_x R_y R_z$. Rychlosť a pozice vzorku n jsou značeny v_n, p_n [1].

$$R_{n+1}^{IMU} = R_n^{IMU} \exp((\omega_n dt)_\times) \quad (5.1)$$

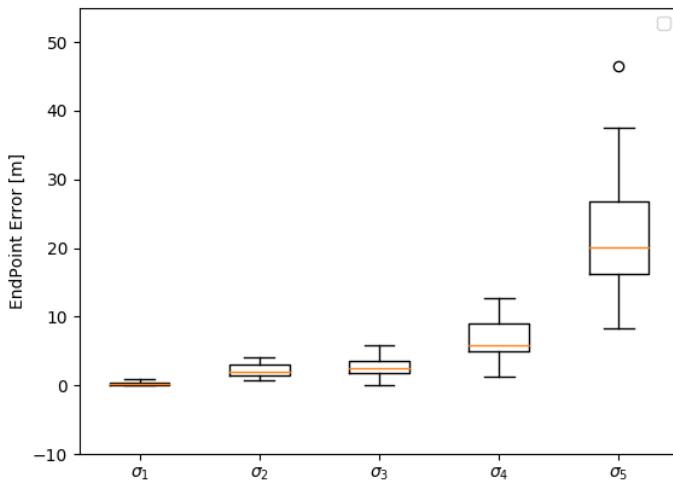
$$v_{n+1}^{IMU} = v_n^{IMU} + (R_n^{IMU} a_n + g) dt \quad (5.2)$$

$$p_{n+1}^{IMU} = p_n^{IMU} + v_n^{IMU} dt \quad (5.3)$$

Algoritmus obsahuje tři integrace, které zatěžují výsledek akumulující se chybou. Tato chyba není nijak eliminována a tudíž se výsledná trajektorie vychyluje již po několika sekundách jízdy. Pro znázornění tohoto trendu byly vytvořeny graf 5.5 a jeho přiblížená část na grafu 5.6. Tyto grafy zachycují EPE po 40 sekundách jízdy v závislosti na šesti šumových parametrech σ , jejichž hodnoty jsou v tabulce 5.2. Zde je nutné se pozastavit nad úrovní šumu Kittí databáze σ_5 u které se předpokládá vysoká preciznost. Tuto vysokou hodnotu σ oproti hodnotám dodávaným výrobci inerciálních jednotek lze přisoudit externím zdrojům vibrací, kterými mohou být motor, hřídele, pneumatiky, povrch cesty a mnohá další.



Obrázek 5.5: Porovnání výsledků všech úrovní zašumění pro integrační metodu



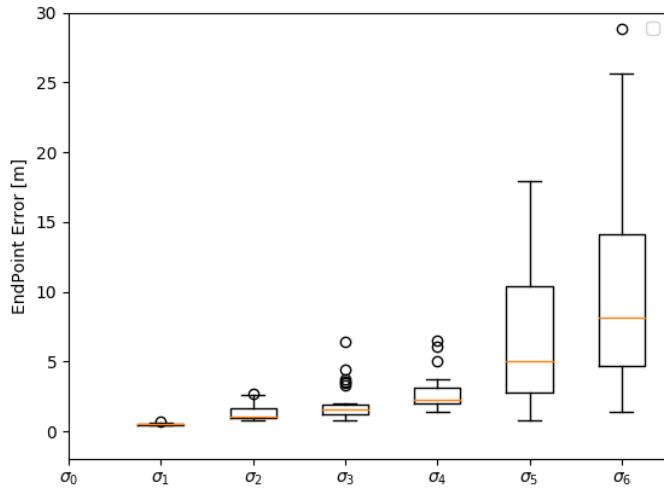
Obrázek 5.6: Přiblížené porovnání výsledků všech úrovní zašumění pro integrační metodu neobsahující σ_0 a σ_6

5.3.2 Rozšířená Kalmanova filtrace

Metoda lokalizace založená na Rozšířené Kalmanově filtraci inerciálních dat, je jeden z nejrozšířenějších způsobů v tomto oboru a jeho implementaci a obdobám se věnuje kapitola 3.2.5 či podrobněji publikace [11]. V této části bude pozornost zaměřena na výsledky dosažené na identickém datasetu a podmírkách jako v kapitole 5.3.1, z jejichž rovnic a principu EKF vychází,

ale dává je do kontextu s šumem měřené veličiny a šumem senzoru podle rovnic v kapitole 3.2.5.

Na grafu 5.7 typu boxplot s vyznačenou hodnotou EPE (4.4) může být pozorováno, že zlepšení oproti metodě 5.3.1 je v pěti ze šesti parametrů σ z tabulky 5.2. Zhoršení nastalo v případě nejnižšího zašumění σ_1 a může být přisuzováno nevhodně zvoleným parametrym filtru.



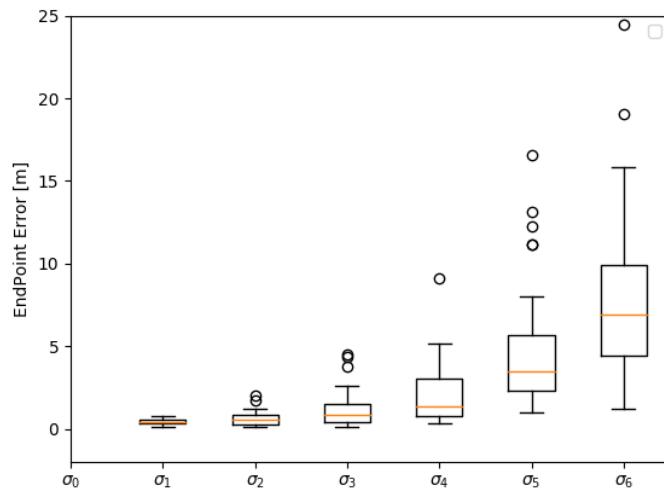
Obrázek 5.7: Porovnání výsledků všech úrovní zašumění pro metodu Kalmanovy filtrace

5.3.3 AI-IMU Dead-Reckoning

AI-IMU Dead-Reckoning metoda, která je podrobně rozebrána v kapitole 4 a publikaci [1], dosahuje podle očekávání nejlepších výsledků. Z grafu 5.8 je možno dedukovat, že přínos metody se zvyšuje s rostoucím zašuměním. Tabulka 5.3 zachycuje porovnání jednotlivých metod a zašumění podle průměrných hodnot EPE, kde AI-IMU Dead-Reckoning dosahuje lepších výsledků až o 20 %.

| Algoritmus | σ_1 | σ_2 | σ_3 | σ_4 | σ_5 | σ_6 |
|-----------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Integrace | 0.32 | 2.11 | 2.62 | 6.5 | 21.23 | 259 |
| EKF | 1.01 | 1.36 | 1.33 | 1.95 | 7.32 | 10.09 |
| AI-IMU Dead-Reckoning | 0.45 | 1.03 | 0.63 | 1.29 | 5.82 | 7.89 |

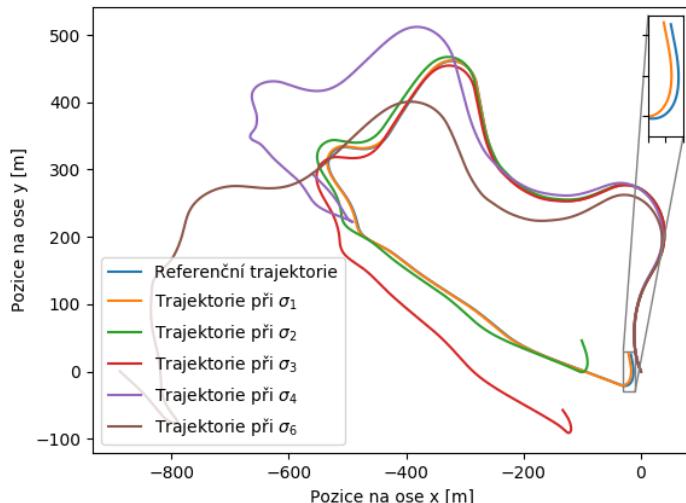
Tabulka 5.3: Porovnání průměrných EPE [m] v závislosti na σ



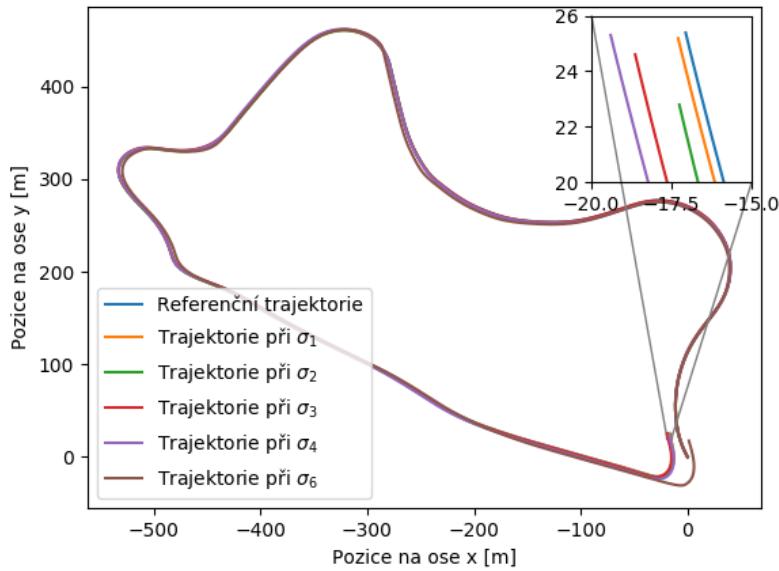
Obrázek 5.8: Porovnání výsledků všech úrovní zašumění pro AI-IMU Dead-Reckoning

■ 5.3.4 Porovnání trajektorií

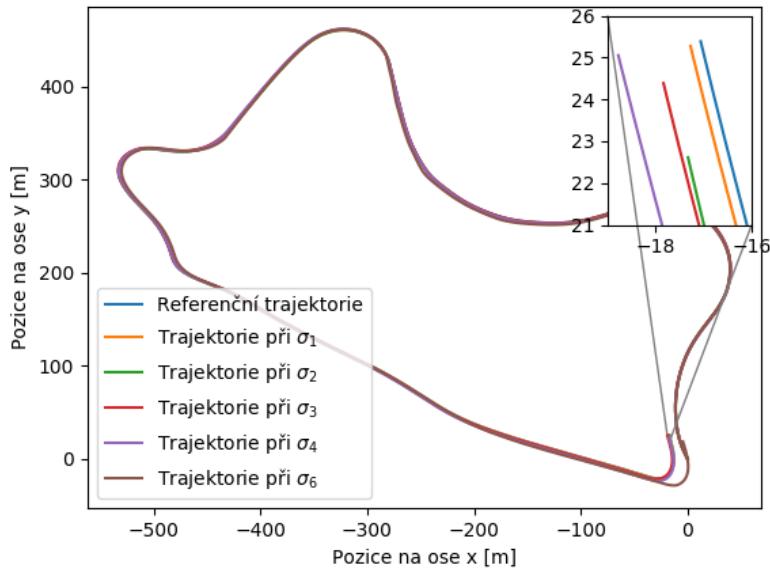
Tato kapitola byla vytvořena pro znázornění změn trajektorie v závislosti na šumovém parametru σ a použité metodě. Hodnoty dosažené na Kittí datasetu nemohly být znázorněny ve stejném grafu, jelikož se jejich referenční trajektorie mírně liší. Přiblížené okno znázorňuje konec trajektorie.



Obrázek 5.9: Porovnání trajektorií vypočítaných nefiltrovanou integrační metodou v závislosti na pamametru σ



Obrázek 5.10: Porovnání trajektorií vypočítaných EKF metodou v závislosti na pamametru σ



Obrázek 5.11: Porovnání trajektorií vypočítaných AI-IMU Dead-Reckoning metodou v závislosti na pamametru σ

Kapitola 6

Experiment na TOMI platformě

Hardwarový experiment byl proveden na TOMI platformě, která byla vyvinuta týmem vývojářů na Českém vysokém učením technickém v Praze, jehož byl autor této práce členem [3]. Experiment spočívá v implementaci algoritmu prezentovaných v práci [1] a jejich *offline* aplikaci na data získaná ze zmíněné platformy. V případě úspěchu by tato implementace mohla být využita jako zdroj informací o pohybu platformy, implementována jako zdroj kinematických dat v reálném čase a hlavně jako dílčí prvek řídících algoritmů ve chvíli, kdy ostatní zdroje selhávají.

Samotný model 1:5 byl vybrán, jako největší možná a zároveň dostupná alternativa skutečného vozidla. Po přihlédnutí k menším variantám, které ne-disponují takto rozsáhlým množstvím prostoru na měřící a výpočetní přístroje, jejich menší nosnost a jízdní vlastnosti rozdílnější od skutečných vozidel, bylo Losi Desert Buggy XL-E 4WD zvoleno jako vhodné pro účely vývojářů z Toyota týmu. Vozidlo bylo následně osazeno mnoha senzory potřebnými pro tento experiment, ale i mnoho dalších nebo spolehlivějších bylo nutno přidat a zároveň softwarově zakomponovat [3].

6.1 Měřící TOMI platforma

Experimentální platforma založená na Ackermanově podvozku, je původně zmenšený RC model auta v poměru 1:5. V originálním konfiguraci RC model obsahoval.

- plně pojízdný a tlumený podvozek
- motor
- řídící jednotku motoru
- ovladač a přijímač
- baterie

Žádný z těchto prvků neobsahoval jakákoli komunikační rozhraní a senzory potřebné k následnému asistovanému či autonomnímu řízení.

6.1.1 **Hardware**

TOMI platforma byla vybavena senzory nezbytnými pro další rozvoj asistovaného i autonomního řízení. Tyto obory vyžadují data o poloze, otáčkách motoru, otáčkách kol, senzory sbírající informace o prostředí ve kterém se pohybují a mnoho dalších informací. Dále je nezbytné, aby vše bylo doplněno výkonnou výpočetní jednotkou, která data vyčítá a následně zpracovává jako užitečnou informaci pro řídící algoritmy.

6.1.2 **TOMI vybavení**

TOMI platforma byla v základní verzi vybavena třemi výpočetními a komunikačními jednotkami:

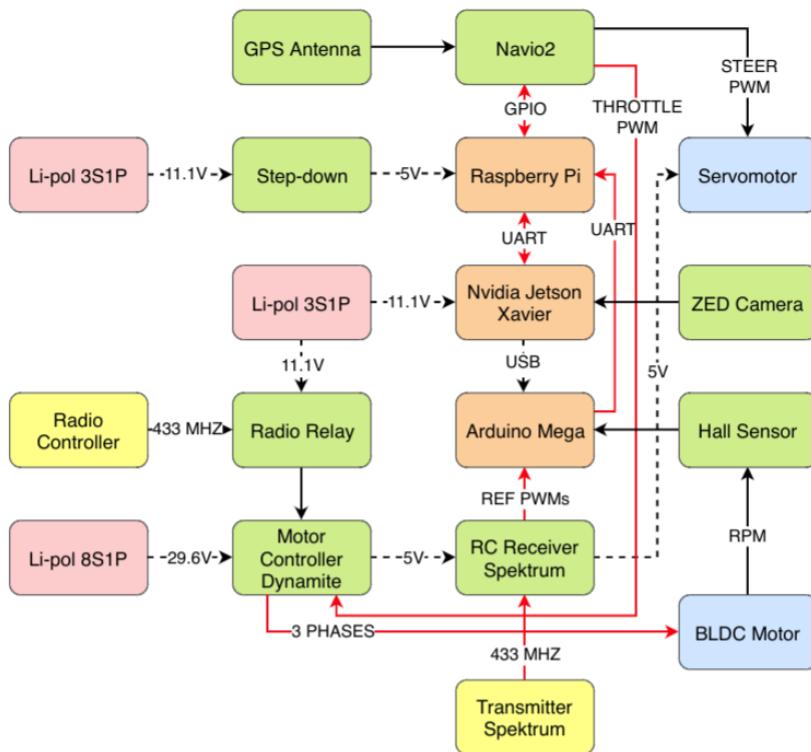
- CPU a GPU jednotkou Nvidia Jetson Xavier
- Raspberry Pi
- Arduino Mega

Nvidia Jetson Xavier byl s jeho 8 jádry a vysokým grafickým výkonem zvolen centrálním uzlem celého systému. Raspberry Pi a Arduino Mega zasílají centrálnímu uzlu data ze senzorů, vyčtené PWM komunikace o řízení motoru a servo motoru. Vyčtená data jsou následně ukládána na SSD disk, který je centrálním úložištěm dat na palubě vozidla. Tyto výpočetní jednotky samy

o sobě obsahují pouze několik základních snímačů s nízkou přesností, a proto byly dovybaveny mnoha senzory a komunikačními rozhraními z nichž je několik důležitých vypsáno.

- Shield Navio 2 rozšiřující Raspberry Pi
- Stereo ZED kamerou
- Hallový senzory
- Radio komunikátorem

Schematické zapojení celého systému je k vidění na obrázku 6.2.



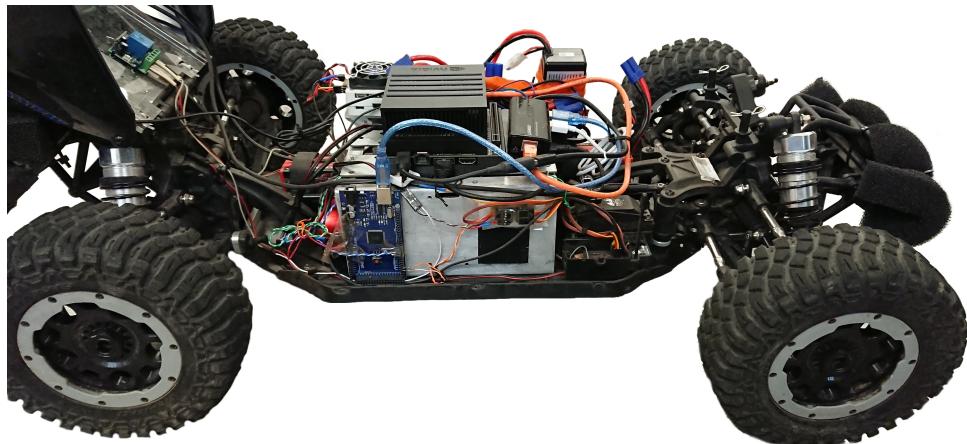
Obrázek 6.1: Schéma zapojení palubních přístrojů na TOMI platformě před dodatečnou úpravou [3]

Shield Navio 2 na svých periferiích obsahuje GNSS přijímač NEO-M8N schopný přijímat informace od poskytovatelů GPS, GLONASS, Beidou a Galileo, duální IMU jednotku obsahující akcelerometry, gyroskopy a magnetometry, PPM/SBUS vstupy schopné vyčítat až 14 PWM kanálů, barometr a mnoho komunikačních rozhraní. Podstatnými senzory pro tento experiment jsou IMU jednotky MPU9250, LSM9DS1 a GNSS přijímač.

6. Experiment na TOMI platformě

ZED kamera je moderní stereo snímač poskytující široký zorný úhel, 15/100 FPS v závislosti na rozlišení v rozsahu 2K-WVGA, *depth range* v rozsahu 0.5-20 m a elektronicky synchronizovaný *rolling shutter* [18].

Hallový senzory v jejich použití vytvářejí rotační enkodér, schopný díky polaritě magnetů zjistit změnu orientace otáčení, rychlosť, ale i měřit ujetou trajektorii. Radio komunikátor slouží jako manuální bezpečnostní prvek pro případ selhání elektronické komunikace nebo zamrznutí systému.



Obrázek 6.2: Fotografie palubních přístrojů

Všechny tyto komponenty bylo následně nezbytné softwarově propojit. K tomuto účelu byl pro svoji jednoduchost, širokou uživatelskou komunitu a dostupnost vzorových implementací využit Python 3.7. Python 3.7 obsahuje širokou škálu knihoven pro komunikaci I2C, UART, PWM, LAN, Wifi, ale i spolehlivou funkci vícevláknových a procesorových aplikací, jichž bylo využito pro nepřerušovaný chod systému.

6.1.3 Konfigurace hardwaru a příprava na vytvoření datasetu

Jak již bylo naznačeno v kapitole 5, spuštění algoritmu potřebuje přesnou referenční trajektorii s dostatečnou vzorkovací frekvencí a přesná inerciální data. V kapitole 6.1.1 bylo naznačeno, že referenční trajektorie mohla být získána z GNSS modulu zahrnutého v shieldu Navio2. Toto řešení bylo v původní variantě použito, ale absence měření *heading angle* vozidla stojícího na místě a horizontální poziční přesnost 2.5 metru, vedla k využití přesnějšího zařízení. Podobný případ nastal u inerciálních jednotek MPU9250 a LSM9DS1, které v prostředí se silným elektromagnetickým rušením motoru a mechanickými vibracemi narážely na svoje limity [19].

Pro nasbírání vlastní databáze bylo nutné vybrat zařízení, které bude schopné měřit přesnou trajektorii s vysokým rozlišením a vysokou obnovovací frekvencí. Po zvážení dostupných technologií v kapitole 2.5 a možnostech v rámci ČVUT, byly z katedry řídící techniky na fakultě elektrotechnické, zapůjčeny jednotky SparkFun GPS-RTK-SMA osazeny čipem ZED-F9P. Tato jednotka je schopna sbírat data s 10 Hz frekvencí a rozlišením 7 cm. Výsledná přesnost záleží na mnoha faktorech, jakými jsou pozice, viditelnost satelitů a doba vlastní lokalizace stacionární GPS.

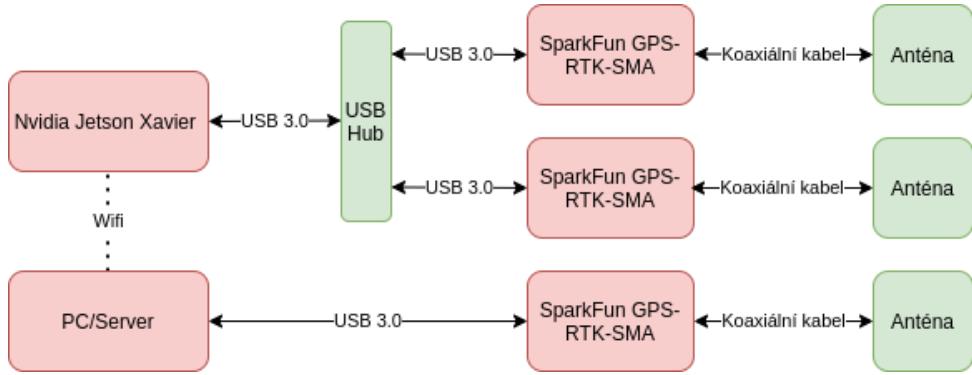
Jeden tento modul byl umístěn na kapotu auta a druhý na křídlo. Tímto způsobem vznikl 90 cm rozestup, ze kterého bylo možné i ve statické pozici měřit úhel natočení. GNSS přijímače jsou citlivé na elektromagnetické rušení a bylo nutno je dodatečně elektromagneticky odclonit. Schéma zapojení a finální umístění je k vidění na obrázcích 6.3 a 6.4.

Třetí GNSS jednotka je oddělena od auta a využívá se jako základní stanice pro diferenciální uspořádání popsané v kapitole 3.1.3. Knihovny a vzorová řešení byly opět v Pythonu 3.7, což usnadňovalo celkovou synchronizaci. Princip a základní implementace DGPS byly využity od týmu vývojářů vedených doc. Ing. Zdeňkem Hurákem, Ph.D [20]. Jejich původní implementace byla upravena pro potřeby tohoto projektu a změny byly následující.

- Komunikaci mezi palubními přijímači místo původního LTE zajišťuje vícevláknový Python skript na jednom zařízení.
- Komunikaci mezi základní stanicí a autem zajišťuje Wifi modul.
- Ukládání dat lze zapnout automaticky oproti původnímu ručnímu zadávání v každém bodě a probíhá na SSD disk připojený k Nvidia Jetson Xavieru.
- V ukládání dat lze nově vybrat variantu .ubx formátu nebo .csv, což zjednoduší práci v případě, že není dostupná knihovna vycítající .ubx zprávy.
- Nahrávání konfiguračních souborů, na které GNSS modul nemá trvalou paměť, zajišťuje skript spouštějící nahrávání.

Precizní IMU jednotky byly zapůjčeny od doc. Ing. Pavla Pačese, Ph.D. celkem tři, ale kvůli původním problémům s knihovnami byla vybrána vývojová deska "Xsens MTi-3-DK Development Kit" osazená modulem Xsens MTi 10-series IMU Module. Vývojová deska má několik možností komunikace se zařízením ukládajícím data, kterými jsou I2C, SPI, UART a USB. Zvolena

6. Experiment na TOMI platformě



Obrázek 6.3: Schéma zapojení a komunikace DGPS



Obrázek 6.4: Umístění přijímačů a GPS antén

byla možnost propojení přes USB. Problém nastal s Python knihovnami, které neměly splnitelné závislosti. Tento problém vznik kombinací linuxu a ARM procesoru, což byla kombinace, na kterou nebyly knihovny uzpůsobeny. Tento problém byl následně vyřešen využitím starších knihoven pro starší verzi Xsense běžící na Pythonu 2.7. Základní parametry IMU jednotky jsou v tabulce 6.1 nebo v manuálu [21]. V nastavení použitém pro naše měření snímá na frekvenci 100 Hz.

| MTi-10 | akcelerometr | gyroskop |
|--------|------------------------|-------------------------|
| rozsah | 200 ms^{-2} | 450 degs^{-1} |
| bias | 0.05 ms^{-2} | 0.2 degs^{-1} |

Tabulka 6.1: Tabulkové parametry MTi-10 series

Synchronizace mezi oběma dvěma GNSS zařízeními a IMU jednotkou byla plánována přes hlavní uzel multiprocessingu se sdílenými proměnnými. Využitím Pythonu 2.7 vznikl problém, jelikož autorovi práce nebyl multiprocessing mezi různými verzemi znám, a neporařilo se najít způsob, který by tento problém řešil. Synchronizace musela být založena na bash skriptech spoléhajících se na systémový čas Jetsonu.

6.2 Dataset

Měřící soustava byla popsána v části 6.1. Tato část se bude zabývat strukturou datasetu, softwarovým zpracováním dat a následně dosaženým výsledkům v oblasti vytvořeného datasetu.

Dataset obsahuje celkem 19 jízd v rozmezí 30 sekund až 15 minut, jež byly měřeny v různých podmínkách, kterými byl nový asfalt, škrabanka a polní cesta. Většina jízd je o poznání kratší oproti Kitti databázi, jelikož RC model auta nemůže být provozován na standardních komunikacích a měření probíhalo převážně v zákazech vjezdu nebo pěších zónách. Celková délka záznamu je 45 minut a jeho velikost přesahuje 45 GB. Tento objem dat je způsoben obrázky ze stereo-kamery. Záznam včetně obrázků nemůže být kvůli své velikosti přiložen přímo k této práci, ale odkaz na datové úložiště bude v příloze.

Složková struktura datasetu je následující. Každá jízda má vlastní složku a je pojmenována po datu měření a předvedeném manévrů. Složka příslušné jízdy obsahuje:

- podsložku "images" se záznamy ze stereokamery ve formátu JPEG v rozlišení 2560×720 při 30 FPS
- záznam ze zadní GPS po korekci DGPS
- záznam z přední GPS po korekci DGPS
- záznam z IMU jednotky
- záznam dat o řízení vozidla
- soubor obsahující záznamy pro srovnání systémového času a času GPS
- soubor obsahující záznamy pro srovnání systémového času a času stereo-kamery

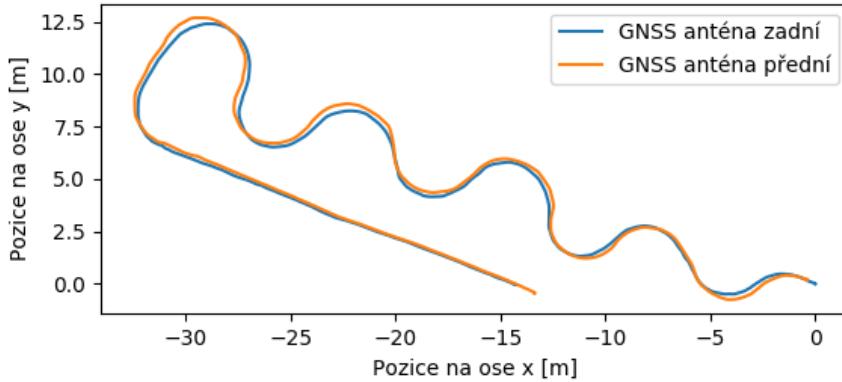
Algoritmus v publikaci [1] předpokládá a načítá data pouze ve specifikovaném formátu. Načítání datasetů probíhá ze zvolené složky obsahující jednotlivé jízdy v souboru typu Pickle. Pickle je způsob pro serializaci a následnou de-serializaci objektů v jazyce Python. V praxi to pak znamená, že objekty a data jsou serializována a uložena na disk, ze kterého mohou být následně načtena a používána. Jedná se o specifický formát využívaný v Pythonu, a proto není doporučené tento formát používat napříč programovacími jazyky. Z tohoto důvodu byl měřený dataset uložen převážně v souborech typu .csv, a tak bude i nahrán na datové úložiště. Pro využití v této práci byl vytvořen skript, který je schopný původní dataset transformovat do podoby spustitelné v originální implementaci. Celý pickle soubor spustitelný v AI-IMU Dead-Reckoning implementaci musí obsahovat pouze jeden Python slovník, jehož specifikaci obsahuje tabulka 6.2. Zde je nutno upozornit, že hodnoty využívá i funkce na trénování datasetu, a proto musí obsahovat i referenční hodnoty. Samotný algoritmus referenční hodnoty využívá pouze jako zdroje počátečních podmínek p_o , v_0 , ang_0 .

| | |
|---------------|---------------------------|
| <i>ang_gt</i> | úhel natočení [rad] |
| <i>sm_gt</i> | start stop indikátor |
| <i>v_gt</i> | referenční rychlosť [m/s] |
| <i>p_gt</i> | referenční poloha [m] |
| <i>name</i> | název datasetu |
| <i>u</i> | data inerciální jednotky |
| <i>t_0</i> | počáteční čas [s] |
| <i>t</i> | časové značky [s] |

Tabulka 6.2: Specifikace předzpracovaného datasetu

Při předzpracování datasetu byly vypuštěny údaje nepotřebné k této úloze, kterými jsou snímky a data pořízená ZED kamerou a údaje o vnitřních stavech vozidla. Softwarové zpracování datasetu probíhalo následovně.

1. zarovnání IMU a GPS dat podle časových značek
2. bikubická interpolace 10 Hz GNSS záznamu, aby odpovídalo 100 Hz IMU datům
3. přepočet souřadnic mezi LOTLAN soustavou a UTM soustavou
4. výpočet referenčních rychlostí z referenční polohy
5. výpočet referenčního úhlu natočení z funkce atan2 a dvou bodů GPS
6. rotace IMU dat a tím zarovnání vztažné soustavy inerciální jednotky a vztažné soustavy auta.

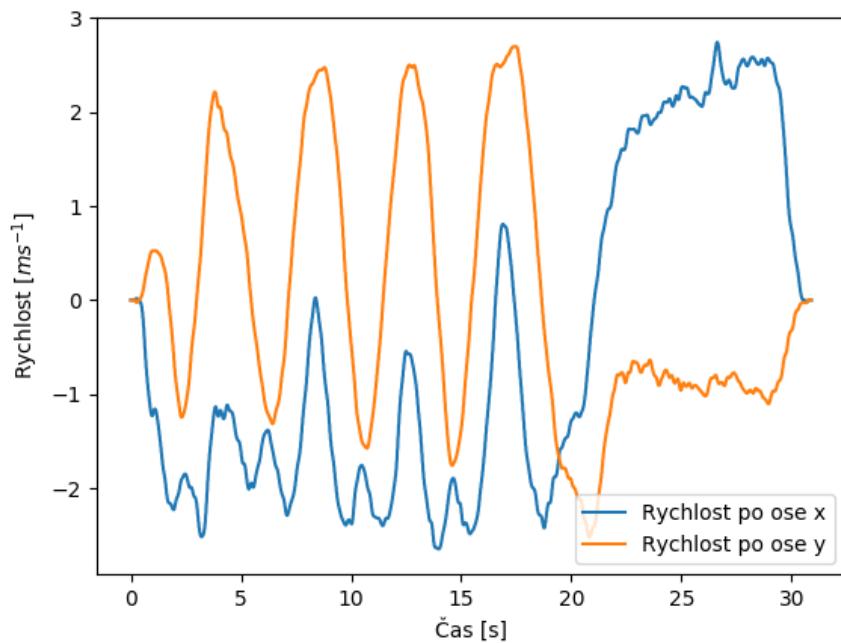


Obrázek 6.5: Referenční trajektorie

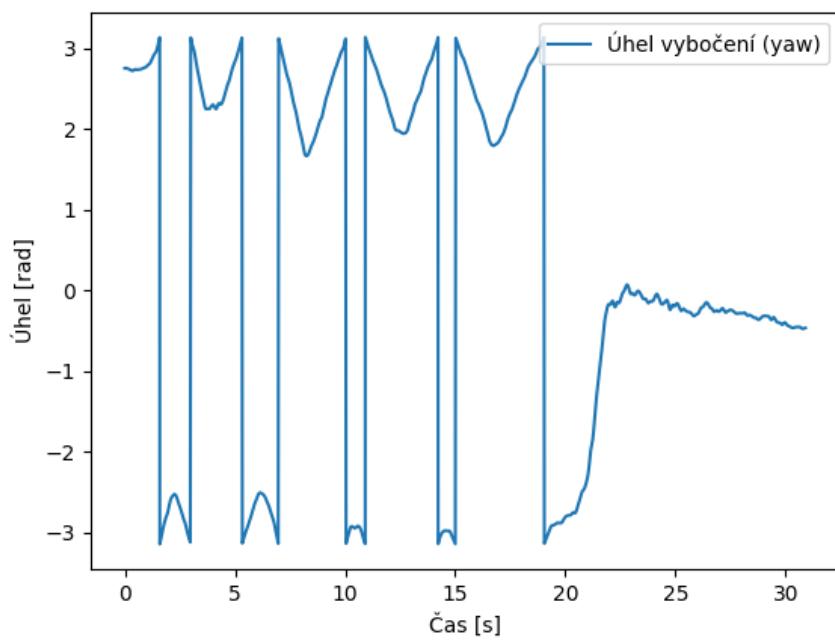
Měřená a přepočítaná trajektorie obou bodů v UTM souřadnicích je k vidění na obrázku 6.5. Pro samotný algoritmus byla výsledná trajektorie počítána jako prostřední bod mezi přední a zadní anténou, což také odpovídá umístění IMU jednotky. Referenční rychlosť z jízdy na obrázku 6.5 je k vidění na obrázku 6.6 a byla vypočítána ze vzorce (6.1), kde p jsou souřadnice bodů x , y , z a $\Delta t = t_{n+1} - t_n$.

$$v_{n+1} = \frac{(p_{n+1} - p_n)}{\Delta t} \quad (6.1)$$

Referenční úhel byl vypočítán na základě vztahů vycházejících z goniometrické funkce tangens. Pro tuto aplikaci byla využita její inverzní funkce arcus tangens v python implementována pod názvem atan2. Tato funkce současně vrací hodnoty v rozmezí od $-\pi$ do π radiánů, což odpovídá požadavku na vstup AI-IMU Dead-Reckoning algoritmu. Vzorová sekvence rychlostí a úhlů, pro trajektorii na obrázku 6.5 je na obrázku 6.6 a 6.7.



Obrázek 6.6: Referenční rychlosť

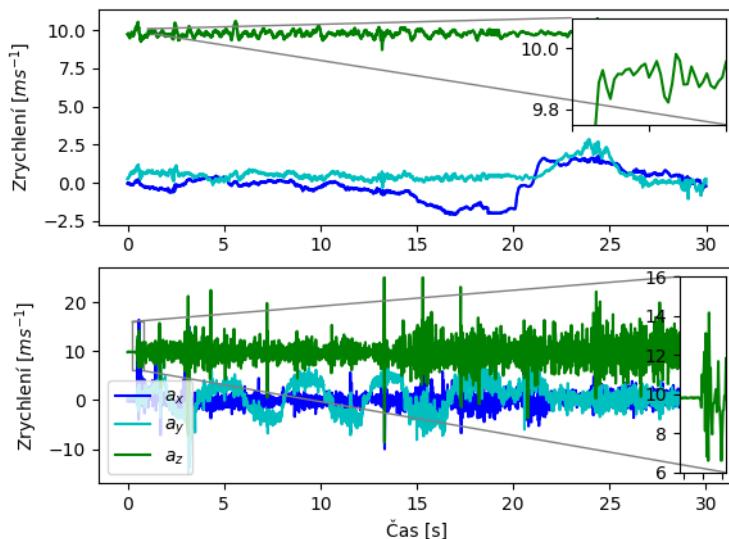


Obrázek 6.7: Referenční úhel

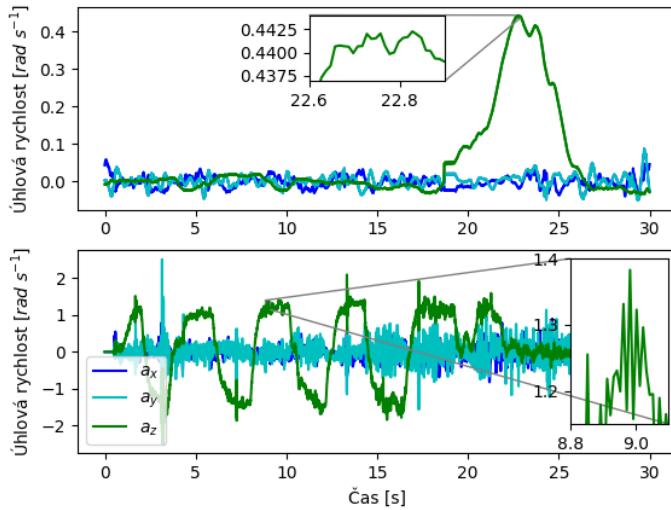
6.3 Parametry filtru a trénování neuronové sítě

Dataset byl vytvořen podle specifikací vývojářů, ale trénovaná síť pochopitelně nemohla na základě původně natrénované sítě fungovat správně. Po krátké korespondenci s původním vývojářem a přečtení doporučení, která autor publikace [1] sdílel v komentářích u projektu na serveru Github.com, započalo ladění úvodních parametrů Kalmanova filtru. Jak již bylo uvedeno v kapitole 4, neuronová síť se je schopna přeucit šumovým parametry nově měřeného zařízení, ale výsledky trénování jsou vysoce závislé na nastavení počátečních parametrů Kalmanova filtru. Tento fakt vylučuje původní předpoklad přesunutí měřícího zařízení pouze s přeucením neuronové sítě. Protože je zlepšení patrné v případě Kitti i generované databáze v předchozím testování, byly parametry postupně laděny podle instrukcí vývojáře. Formát vstupních dat pro trénování neuronové sítě byl stejný, jako datasetu pro spuštění AI-IMU Dead-Reckoning metody popsaného tabulkou 6.2.

Zlepšení na úroveň prezentovanou na Kitti databázi v podobě výsledků samotného Kalmanova filtru, ale i natrénované neuronové sítě se nepodařilo před dokončením této práce dosáhnout. Neúspěch může být hledán ve špatně laděných parametrech Kalmanova filtru, ale i mechanických vibracích na palubě RC modelu. Tyto vibrace mnohonásobně převyšují porovnávanou Kitti databázi a mohou narážet na limity zvoleného akcelerometru nebo jeho nastavení. Porovnání akcelerací a úhlových rychlostí Kitti a TOMI jízdy znázorňují obrázky 6.8 a 6.9.



Obrázek 6.8: Porovnání hodnot zrychlení Kitti(nahoře) a TOMI(dole)

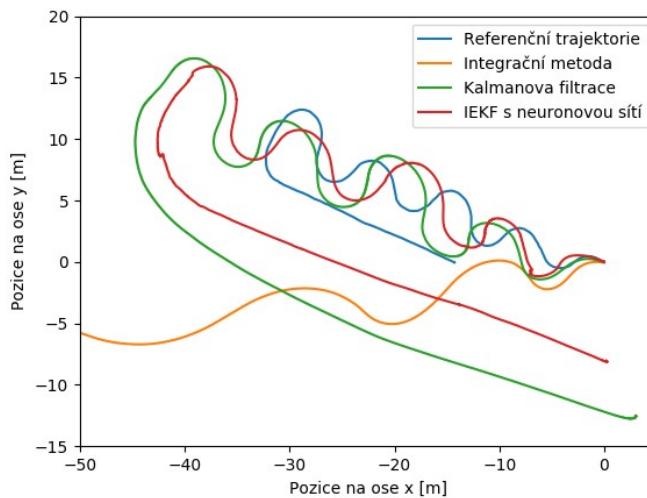


Obrázek 6.9: Porovnání hodnot úhlového zrychlení Kittti(nahoře) a TOMI(dole)

Porovnání nejlépe fungujících parametrů EKF zobrazuje tabulka 6.3. Vzorová jízda a její vypočítané průběhy mohou být viděny na obrázku 6.10.

| typ | parametry kitti | parametry tomi |
|-------------------------------|--------------------|--------------------|
| kovariance gyroskopu | 2×10^{-4} | 8×10^{-4} |
| kovariance akcelerometru | 1×10^{-3} | 1×10^{-1} |
| bias kovariance gyroskopu | 1×10^{-8} | 6×10^{-8} |
| bias kovariance akcelerometru | 1×10^{-6} | 1×10^{-6} |

Tabulka 6.3: Použité parametry Rozšířeného Kalmanova filtru



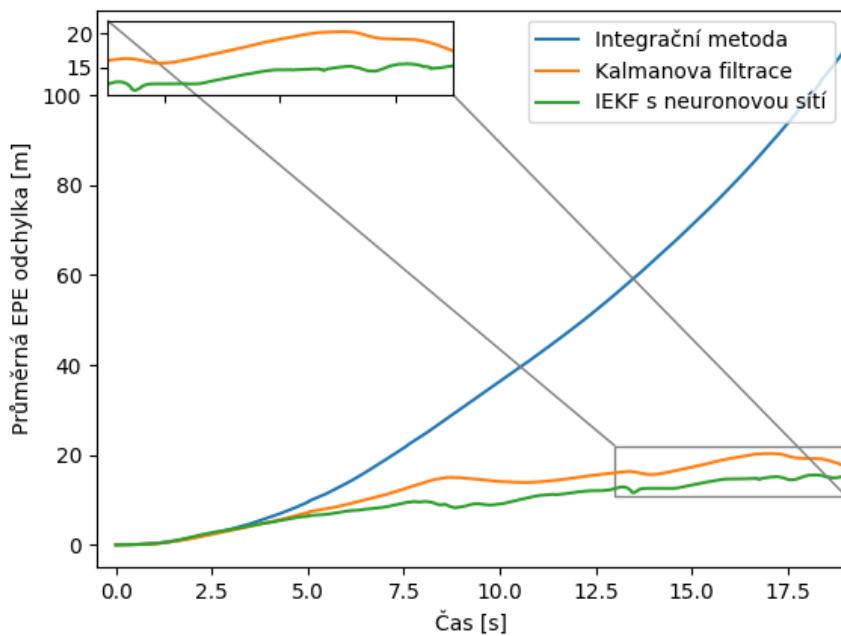
Obrázek 6.10: Porovnání pozice v závislosti na použitém algoritmu

6.4 Výsledky

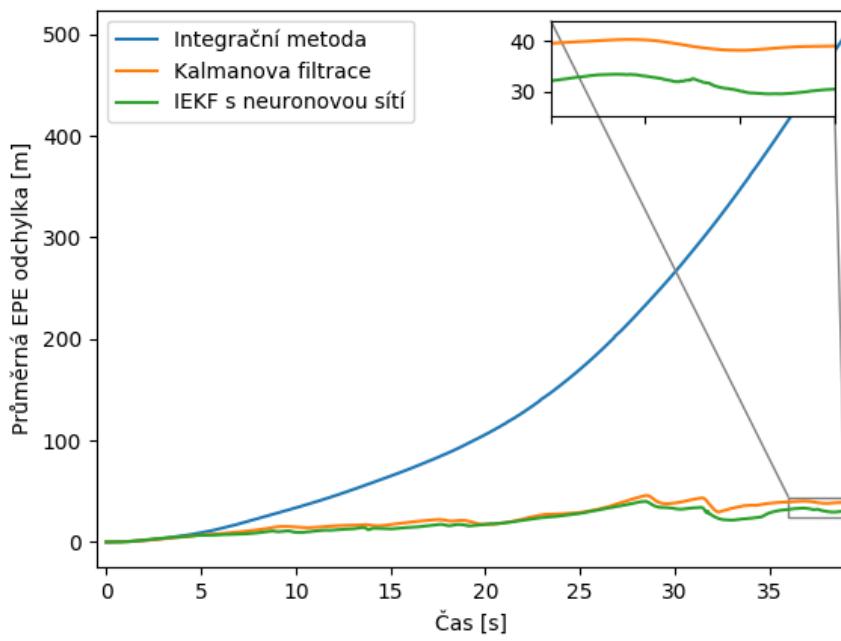
Testování na měřeném datasetu probíhalo opět na všechny tři typy algoritmů jako v kapitolách 4 a 5. Na rozdíl od kapitoly 5 byl k dispozici pouze jeden typ zašumění. Měřeným parametrem byl EndPoint Error a Kumulativní Error popsaný vzorcí (4.4) a (4.5).

Jelikož původní využité podmínce, v podobě minimální délky jízdy 40 sekund, odpovídalo pouze 9 jízd z původních 19 byla přidána i varianta s limitem 20 sekund a její EPE a CumE je k vidění na obrázcích 6.11 a 6.13. Tímto způsobem vznikla dvě porovnání. První porovnání je 20 sekund dlouhé, ale počítá průměrné hodnoty 15 z 19 vytvořených jízd. Druhé porovnání naznamenává průměrné hodnoty po 40 sekundách jízdy, ale vypovídá pouze o 9 jízdách, a tedy necelé polovině datasetu. K vidění je na obrázcích 6.12 a 6.14. Jelikož je dataset více zašuměný a metoda založená na samotné integraci není nijak filtrovaná, chyba se akumuluje a výsledná trajektorie není použitelná pro další zpracování. Zajímavé je porovnání Rozšířeného Kalmanova filtru a AI-IMU Dead-Reckoning metody na obrázcích 6.11 a 6.12. Obě metody si, navzdory problémům při učení neuronové sítě a ladění parametrů Kalmanova filtru vedou lépe než první metoda. AI-IMU Dead-Reckoning si opět drží zlepšení okolo 20 %, které bylo k vidění na obrázku 4.5 při srovnávání těchto metod na původní Kittti databázi.

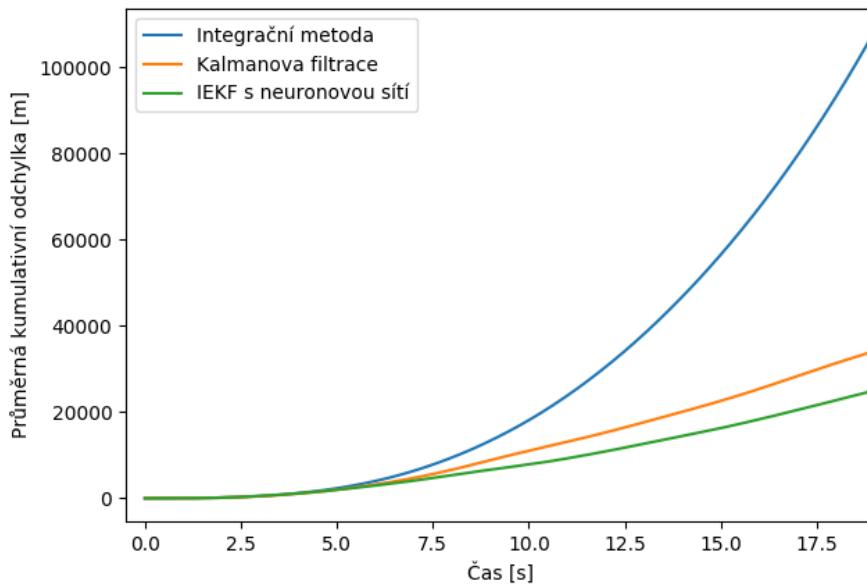
Měření parametru EPE vypovídá o vzdálenosti referenčních bodů a bodů odhadované trajektorie, ale nereflektuje trajektorii předešlou. Tímto ponechává mezeru v hodnocení a může se tedy stát, že ač nebyla odhadovaná trajektorie správná, vozidlo skončí poblíž cílového bodu a vznikne tak mylný předpoklad, že algoritmus funguje správně. Mezeru vyplňuje hodnocení CumE, které jednotlivé hodnoty EPE kumulativně sčítá a konečná hodnota je ohodnocením kvality lokalizace. Hodnoty měřené CumE jsou k vidění na obrázcích 6.13 a 6.14. Tyto hodnoty původní trend EPE potvrzují a dokazují zároveň přínos využití neuronových sítí v lokalizaci.



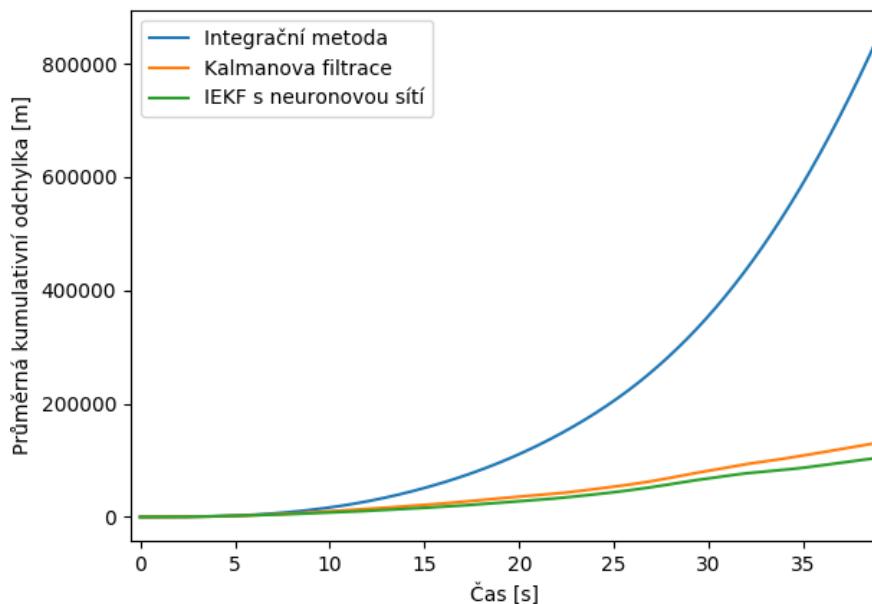
Obrázek 6.11: EndPoint Error vlastní databáze po 20 sekundách



Obrázek 6.12: EndPoint Error vlastní databáze po 40 sekundách



Obrázek 6.13: Cumulative Error vlastní databáze po 20 sekundách



Obrázek 6.14: Cumulative Error vlastní databáze po 40 sekundách

Kapitola 7

Závěr

V samém počátku práce jsme se seznámili s motivací pro vypracování této diplomové práce, která je založena převážně na potřebách v autonomním a asistovaném řízení.

V druhé kapitole proběhlo seznámení se systémy měření trajektorie. Rozebrán byl hlavně princip měřených veličin a jejich nedostatky. Zde je možné pochopit, že neexistuje způsob, který by svými vlastnostmi předčil všechny ostatní. Tento trend nahrává metodám založeným na fúzi jednotlivých metod, které byly záměrně vynechány, jelikož se v dalších kapitolách porovnávala metoda založená na měření pouze jedné veličiny a fúzi jednotlivých metod lze klasifikovat jako rozsáhlejší kategorie měření.

Kapitola tří se věnovala algoritmům zpracování a principu vyhodnocení nejrozšířenějších metod s přihlédnutím ke kapitole druhé. Většinu prostoru zde dostaly algoritmy používané v experimentech pro jejich pochopení a uvedení čtenáře do souvislostí. Dozvědět se dalo o základním ale i pokročilém zpracování dat. Principy družicové lokalizace zde byly představeny až po diferenciální GPS lokalizaci, která se využívala v hardwarových experimentech. V kapitole se lze dozvědět také informace o inerciálních senzorech a zpracování jejich dat, jelikož praktická část této práce je na těchto datech založena.

Čtvrtá kapitola pojednává o publikaci [1] týmu z francouzské univerzity věnující se úloze Dead-Reckoningu, který je úlohou *on board* lokalizace. Rozebrán zde byl základní princip včetně odkazů k pokročilé implementaci v rámci publikace samotné. Publikace jako taková vykazovala několik nejasností, které mohly být úmyslem, ale také pouhou náhodou při její tvorbě. Hlavní nejas-

ností v části s publikací výsledků, bylo vynechání kvantitativních výsledků Rozšířeného Kalmanova filtru, ze kterého publikace vycházela a navazovala na něj. Toto porovnání bylo v rámci této kapitoly doplněno a zároveň bylo přidáno několik porovnávacích metod.

Kapitola pátá se zabývala generátorem akcelerometrických a gyroskopických dat. Tento generátor nebyl v rámci této práce implementován, ale převzat od společnosti Acceina Inc., která jej v testovací verzi poskytla k dispozici pod licencí typu open-source. Bohužel generátor nebyl plně dodělán a muselo být většina potřebných funkcí, jenž jsou vypsány v této kapitole, dodělána autorem této práce. Celkem bylo takto vygenerováno 215 jízd, na něž se následně implementovaly algoritmy popsané v předchozích kapitolách. Odkaz na tyto datasety bude přiložen k této práci a datasety mohou být využity pro další experimenty v rámci ČVUT. V závěru kapitoly je porovnání jednotlivých algoritmů a jejich přesnosti v závislosti na šumových parametrech, které jsou tam také popsány. K diskuzi zůstávají možnosti generátoru, a tedy absence náhodných jevů pozorovaných v reálném měření. Jelikož se však jedná o softwarový generátor a reálná aplikace je porovnávána v následující části, může být experiment považován za úspěšný.

Šestá kapitola pojednává o autorově práci na TOMI platformě v rámci Toyota týmu vzniklého spolupráci katedry kybernetiky a katedry řízení. Na základech platformy vytvořené tímto týmem, byla postavena tato kapitola. V počátku byly uvedeny specifikace TOMI platformy a její měřící a výpočetní výbava. Později jsou specifikovány původní nedostatky pro měření této práce a jednotlivé postupy vedoucí k úpravě platformy. V momentě, kdy byla platforma vybavena adekvátními měřícími přístroji, musela být softwarově propojena a časově synchronizována. Část kapitoly je věnován specifikaci, popisu a zhodnocení vytvořeného datasetu. V samotném závěru je popsána příprava metod pro aplikaci na měřený TOMI dataset, porovnání a zhodnocení výsledků. I přes snahu autora a dodržení postupů doporučených původním vývojářem metody, se nepodařilo dosáhnout stejných výsledků, které byly publikovány v práci [1]. Tento neúspěch může být přisuzován nevhodné zvoleným zjišťováním parametrů filtru. Další problém může být hledán v samotné podstatě RC modelu, který je na rozdíl od reálného vozidla menší a lehčí. Tyto vlastnosti v kombinaci s faktom, že se nejedná o reálné a dobře odtlumené vozidlo, vedou k vibracím s mnohonásobně větší amplitudou měřené akcelerace a úhlové rychlosti. Tyto hodnoty nemusí inerciální jednotka správně měřit a samotný filtr s tímto faktom již nic nezmůže.

Mezi dosaženými výsledky lze vyzdvihnout úpravu generátoru inerciálních dat, vygenerování syntetického datasetu a otestování vlivu úrovní šumu na několika vybraných metodách. Úpravu TOMI subscale platformy pro potřeby vytvoření vlastního datasetu za využití diferenciální GPS metody a precizní

inerciální jednotky. Vytvoření reálně měřeného datasetu s referenční trajektorií, inerciálními daty a záznamy ze stereo-kamery pro vývoj a testování algoritmů v rámci práce, ale i budoucí testování na ČVUT. Předzpracování vlastního datasetu pro testované algoritmy a ladění parametrů filtru, včetně přeúčování neuronových sítí, které se sice nepodařilo dotáhnout k uspokojivým výsledkům, ale může být přínosem a podkladem k dalšímu zkoumání. A v neposlední řadě se podařilo potvrdit, že ač výsledky lokalizace na vlastně měřeném datasetu nejsou uspokojivé, zlepšení metody AI-IMU Dead-Reckoning oproti Rozšířenému Kalmanovu filtrování je až 20 %.

Možnosti pro nalezení funkční konfigurace vidím ve zkoumání zdroje vibrací a dalším zkoumání nastavení akcelerometrů, kterými jsou rozsah, rozlišení, obnovovací frekvence a zesílení. Dále je možné pokračovat v ladění filtru a trénování neuronových sítí, které mohou při úspěšnějším nastavení dávat adekvátní výsledky. Další možností je přesunutí měřící soustavy do reálného vozidla a zopakování experimentu přímo na něm. Zde by mohl být problém s vibracemi menší a algoritmus by mohl dosahovat výsledků srovnatelných s těmi publikovanými na Kitti databázi.

Příloha A

Odkaz na vytvořené datasety

<https://drive.google.com/drive/folders/1Jxso8SIsUmPuX507Ap5CMapyhdEUiUgG?usp=sharing>

Příloha B

Literatura

- [1] Martin Brossard, Axel Barrau, and Silvere Bonnabel. AI-IMU dead-reckoning. *IEEE Transactions on Intelligent Vehicles*, pages 1–1, 2020.
- [2] Open-source Inertial Navigation, GPS/INS, AHRS and Simulation Software. <https://www.aceinna.com/>.
- [3] David Pekárek Martin Vlašimský Jan Čech Tomáš Haniš Michal Bahník, Dominik Filyo and Martin Hromčík. Visually assisted anti-lock braking system. *IEEE Intelligent Vehicles Symposium(IV)*, 2020, To Appear.
- [4] Seyed A. REZA Zekavat and R. Michael Buehrer, editors. *Handbook of Position Location*. Wiley, mar 2018.
- [5] Martin BROSSARD and Silvere BONNABEL. Learning wheel odometry and IMU errors for localization. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, may 2019.
- [6] Raul Mur-Artal and Juan D. Tardos. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, oct 2017.
- [7] Taihú Pire, Thomas Fischer, Gastón Castro, Pablo De Cristóforis, Javier Civera, and Julio Jacobo Berlles. S-PTAM: Stereo parallel tracking and mapping. *Robotics and Autonomous Systems*, 93:27–42, jul 2017.
- [8] Jean-Emmanuel Deschaud. IMLS-SLAM: Scan-to-model matching based on 3d data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2018.

- [9] Essentials of satellite navigation - compendium. https://www.u-blox.com/sites/default/files/products/documents/GPS-Compendium_Book_%28GPS-X-02007%29.pdf.
- [10] Beidou navigation satellite system. <http://en.beidou.gov.cn/SYSTEMS/System/>.
- [11] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. Using inertial sensors for position and orientation estimation. *Foundations and Trends® in Signal Processing*, 11(1-2):1–153, 2017.
- [12] Simon Haykin, editor. *Kalman Filtering and Neural Networks*. John Wiley & Sons, Inc., oct 2001.
- [13] Axel Barrau and Silvère Bonnabel. Invariant kalman filtering. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):237–257, may 2018.
- [14] Deep learning for inertial navigation. <https://towardsdatascience.com/deep-learning-for-inertial-navigation-441b7628a326>.
- [15] Axel Barrau and Silvere Bonnabel. The invariant extended kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812, apr 2017.
- [16] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, aug 2013.
- [17] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149, jan 2008.
- [18] ZED Stereo Camera / Stereolabs. <https://www.stereolabs.com/zed/>.
- [19] NEO-M8-N. https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_%28UBX-15031086%29.pdf.
- [20] ACC - Experiments with commercially available Adaptive Cruise Control systems. <https://gitlab.fel.cvut.cz/aa4cc/acc/gps-recording-system>.
- [21] MTi_10-series_usermanual. <https://www.xsens.com/products/mti-10-series>.