

# Fuzija senzora za lokalizaciju mobilnog robota u vanjskom okruženju

---

**Barać, Dario**

**Master's thesis / Diplomski rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:337739>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2024-11-23**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Sveučilišni diplomski studij računarstva

Diplomski rad

**Fuzija senzora za lokalizaciju mobilnog  
robotu u vanjskom okruženju**

Rijeka, ožujak 2023.

Dario Barać  
0069082669

SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Sveučilišni diplomski studij računarstva

Diplomski rad

**Fuzija senzora za lokalizaciju mobilnog  
robotu u vanjskom okruženju**

Mentor: prof.dr.sc. Kristijan Lenac

Rijeka, ožujak 2023.

Dario Barać  
0069082669

**SVEUČILIŠTE U RIJECI**  
**TEHNIČKI FAKULTET**  
**POVJERENSTVO ZA DIPLOMSKE ISPITE**

Rijeka, 15. ožujka 2022.

Zavod: **Zavod za računarstvo**  
Predmet: **Mobilna robotika**  
Grana: **2.09.04 umjetna inteligencija**

## ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Dario Barać (0069082669)**  
Studij: **Diplomski sveučilišni studij računarstva**  
Modul: **Računalni sustavi**

Zadatak: **Fuzija senzora za lokalizaciju mobilnog robota u vanjskom okruženju /  
Sensor fusion localization for outdoor mobile robot**

Opis zadatka:

Implementirati programsku podršku za fuziju senzora na mobilnom robotu temeljenu na Robot Operating System (ROS) sustavu. Predložiti nekoliko različitih konfiguracija dostupnih senzora i parametara fuzije i testirati točnost određivanja položaja na stvarnom robotu u vanjskom okruženju.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

*Dario Barać*

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:



Prof. dr. sc. Kristijan Lenac

Predsjednik povjerenstva za  
diplomski ispit:



Prof. dr. sc. Kristijan Lenac



## Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, ožujak 2023.

-----  
Dario Barać

# Zahvala

Zahvaljujem mentoru prof.dr.sc. Kristijanu Lencu na korisnim raspravama, savjetima i pomoći tijekom pisanja ovog rada. Zahvaljujem kolegi Sandru Šafaru na pomoći sa prikupljanjem podataka senzora na robotu. Zahvaljujem obitelji na podršci tijekom studiranja.

# Sadržaj

Popis slika	ix
Popis tablica	xi
<b>1 Uvod</b>	<b>1</b>
<b>2 Lokalizacija mobilnog robota</b>	<b>3</b>
2.1 Senzori . . . . .	4
2.1.1 Enkoderi kotača . . . . .	5
2.1.2 Inercijalni i orijentacijski senzori . . . . .	5
2.1.3 Senzori za mjerenje udaljenosti . . . . .	7
2.1.4 Prijemnici za aktivne orijentire . . . . .	10
2.2 Postupci za lokalizaciju . . . . .	11
2.2.1 Odometrija . . . . .	11
2.2.2 Satelitska navigacija - GNSS . . . . .	15
<b>3 Fuzija senzora za lokalizaciju</b>	<b>19</b>
3.1 Čestični filter . . . . .	20
3.2 Kalman filter . . . . .	21
3.2.1 Izračun - predikcija . . . . .	22
3.2.2 Izračun - korekcija . . . . .	23

## Sadržaj

3.2.3	Korištenje Kalman filtra u praksi . . . . .	26
3.2.4	Prošireni Kalman filter - EKF . . . . .	27
<b>4</b>	<b>Istovremena lokalizacija i kartiranje</b>	<b>29</b>
4.1	EKF SLAM . . . . .	29
4.2	SLAM zasnovan na grafovima . . . . .	30
4.3	Vizualni RGB-D SLAM . . . . .	31
4.3.1	Zatvaranje petlje . . . . .	32
<b>5</b>	<b>ROS (Robot Operating System)</b>	<b>33</b>
5.1	Osnovni pojmovi i arhitektura . . . . .	33
5.1.1	Čvorovi . . . . .	34
5.1.2	Komunikacija između čvorova . . . . .	34
5.1.3	Okviri . . . . .	36
5.2	ROS 2 . . . . .	36
5.3	Alati . . . . .	37
5.4	Fuzija senzora za lokalizaciju . . . . .	37
5.4.1	Konfiguracija čvorova za estimaciju . . . . .	38
5.4.2	Fuzija GNSS poruka . . . . .	40
5.4.3	Konfiguracija za lokalnu i globalnu procjenu poze . . . . .	42
5.5	SLAM i vizualna odometrija . . . . .	43
<b>6</b>	<b>Pioneer 3-AT mobilni robot i dostupni senzori</b>	<b>44</b>
6.1	u-blox ZED-F9P prijemnik . . . . .	45
6.2	Intel Realsense D455 kamera . . . . .	45

<b>7</b>	<b>Rješenje za lokalizaciju mobilnog robota u vanjskom okruženju</b>	<b>48</b>
7.1	ROS 2 lansirna datoteka za lokalizaciju . . . . .	48
7.1.1	Argumenti lansirne datoteke . . . . .	50
7.1.2	Konfiguracija Realsense D455 kamere . . . . .	50
7.2	Skripta za testiranje lokalizacije . . . . .	51
7.2.1	Definiranje testa za lokalizaciju u YAML datoteci . . . . .	52
7.2.2	Rad skripte za testiranje lokalizacije . . . . .	53
<b>8</b>	<b>Testiranje i usporedba konfiguracija za lokalizaciju</b>	<b>56</b>
8.1	Prikupljanje poruka senzora za lokalizaciju na robotu . . . . .	56
8.1.1	Problemi sa snimanjem i sensorima . . . . .	57
8.2	Metodologija testiranja točnosti lokalizacije . . . . .	59
8.3	Testiranje konfiguracija RGB-D vizualne odometrije . . . . .	61
8.3.1	Prilagodba snopova (Bundle adjustment) . . . . .	61
8.3.2	Metoda registracije okvira i korištenje neholonomskih ograničenja . . . . .	62
8.4	Testiranje konfiguracija EKF fuzije senzora . . . . .	65
8.4.1	Filtriranje vizualne odometrije - utjecaj odabira varijabli iz poruka senzora . . . . .	65
8.4.2	EKF fuzija naredbi kretanja, vizualne odometrije i IMU senzora	67
8.4.3	Testiranje RTAB-Map SLAM rješenja . . . . .	69
8.5	Diskusija . . . . .	72
<b>9</b>	<b>Zaključak</b>	<b>74</b>
	<b>Bibliografija</b>	<b>75</b>
	<b>Sažetak</b>	<b>79</b>

# Popis slika

2.1	Uzorci inkrementalnog i apsolutnog enkodera kotača . . . . .	5
2.2	LIDAR: princip rada i primjer senzora . . . . .	7
2.3	Princip rada ultrazvučnog senzora udaljenosti . . . . .	8
2.4	Slika dubinske kamere . . . . .	9
2.5	3D oblak točaka . . . . .	9
2.6	Epipolarna geometrija . . . . .	11
2.7	Lokalni i globalni koordinatni sustav mobilnog robota . . . . .	13
2.8	GNSS greške tipične za urbana okruženja . . . . .	17
2.9	Korištenje GNSS RTK sustava . . . . .	18
3.1	Rezultat lokalizacije čestičnim filtrom . . . . .	21
3.2	Rezultat koraka korekcije Kalman filtra . . . . .	25
3.3	Glavni koraci iteracije Kalman filter algoritma . . . . .	26
4.1	SLAM zasnovan na grafovima . . . . .	31
5.1	Metode komunikacije ROS čvorova . . . . .	35
5.2	Transformacija GNSS poruka . . . . .	42
6.1	Pioneer 3-AT mobilni robot sa korištenim sensorima . . . . .	45
6.2	Senzori korišteni za lokalizaciju Pioneer 3-AT robota . . . . .	46
6.3	Procjene dubine (udaljenosti) dobivene Realsense D455 kamerom . . . . .	47

## *Popis slika*

8.1	Primjeri grešaka GNSS procjene položaja . . . . .	58
8.2	Prikupljanje podataka za testiranje lokalizacije . . . . .	59
8.3	Putanje vizualne odometrije uz korištenje prilagodbe snopova i bez korištenja . . . . .	62
8.4	Putanje vizualne odometrije za različite konfiguracije pristupa registracije okvira i ograničenja modela kretanja . . . . .	64
8.5	Putanje vizualne odometrije sa i bez EKF filtriranja . . . . .	67
8.6	Putanje konfiguracija EKF fuzije vizualne odometrije, naredbi kretanja i IMU poruka . . . . .	69
8.7	3D karta igrališta dobivena RTAB-Map SLAM rješenjem . . . . .	70
8.8	Putanje konfiguracija EKF fuzije vizualne odometrije, naredbi kretanja i IMU poruka . . . . .	71

# Popis tablica

7.1	Argumenti <i>ekf_localization.launch.py</i> lansirne datoteke . . . . .	50
7.2	ROS parametri korišteni za <i>realsense2_camera_node</i> čvor . . . . .	51
8.1	Referentne poze robota u određenim trenucima snimanja poruka senzora tijekom kretanja po okviru na igralištu . . . . .	60
8.2	Rezultati testiranja RGB-D odometrije sa i bez prilagodbe snopova (eng. <i>Bundle adjustment</i> ) . . . . .	61
8.3	Rezultati testiranja konfiguracija RGB-D odometrije (različite vrijednosti <i>Odom/Strategy</i> i <i>Odom/Holonomic</i> parametara) . . . . .	63
8.4	Varijance varijabli u matricama Kalman filtra $\mathbf{Q}$ i $\mathbf{P}_0$ . . . . .	65
8.5	Usporedba rezultata vizualne odometrije i EKF filtrirane vizualne odometrije . . . . .	66
8.6	Usporedba rezultata EKF fuzije vizualne odometrije, naredbi kretanja i IMU poruka . . . . .	68
8.7	Usporedba rezultata EKF filtrirane vizualne odometrije i EKF fuzije vizualne odometrije sa RTAB-Map detekcijom zatvaranja petlje . . . . .	71



# Poglavlje 1

## Uvod

Mobilni roboti, odnosno roboti koji se kreću u svojoj okolini tijekom rada koriste se u raznim industrijama. Neke od mogućih primjena uključuju prijevoz steriliziranih materijala u bolnicama, prijevoz paleta u skladištima te detekciju i prskanje korova u poljoprivredi. Neovisno o samoj primjeni, mobilni robot mora razumjeti svoju okolinu kako bi uspješno odradio svoje zadaće. Stupanj potrebnog razumijevanja ovisi o tome koliko je okolina u kojoj robot radi prilagođena njemu i njegovim zadacima, te složenosti i predvidljivosti samih zadataka. Vještine potrebne robotu uključuju opažanje okoline (percepcija), izgradnju karte statičnih dijelova okoline i praćenje objekata u pokretu (modeliranje okoline), određivanje vlastitog položaja i orijentacije (lokalizacija), te planiranje puta do cilja (navigacija).

U ovom radu razmatra se problem lokalizacije mobilnog robota. Mobilni roboti mogu koristiti razne senzore i metode obrade mjerenja senzora kako bi procijenili svoj položaj u odabranom referentnom koordinatnom sustavu. U mjerenjima dobivenim od senzora prisutne su greške, a svaka metoda ima određene prednosti i nedostatke. Fuzija senzora za lokalizaciju obuhvaća metode koje kod procjene položaja i orijentacije kombiniraju mjerenja dobivena od više različitih senzora, i pritom uvažavaju različite stupnjeve točnosti i preciznosti mjerenja pojedinih senzora. Takvi postupci mogu postići manju grešku procjene položaja od procjene dobivene pomoću samo jednog senzora [1].

Cilj ovog rada bio je implementirati programsku podršku za fuziju senzora na

## *Poglavlje 1. Uvod*

mobilnom robotu temeljenu na Robot Operating System (ROS) sustavu, predložiti nekoliko različitih konfiguracija dostupnih senzora i parametara fuzije i testirati točnost određivanja položaja na stvarnom robotu u vanjskom okruženju.

U **poglavlju 2** definiran je problem lokalizacije mobilnog robota. Opisano je nekoliko senzora čiji se podaci dobiveni obradom mjerenja mogu koristiti za lokalizaciju, metode za lokalizaciju (odometrija i satelitska navigacija), te nedostaci korištenja samo jedne metode lokalizacije. **Poglavlje 3** opisuje fuziju senzora za lokalizaciju i korake izračuna za fuziju senzora Kalman filtrom. **Poglavlje 4** opisuje prednosti korištenja karte za lokalizaciju i pristupe rješavanju problema istovremenog kartiranja i lokalizacije. U **poglavlju 5** opisan je ROS (*Robot Operating System*) i ROS paketi za lokalizaciju fuzijom senzora i kartiranje. **Poglavlje 6** opisuje Pioneer 3-AT mobilni robot i senzore korištene za testiranje lokalizacije. U **poglavlju 7** opisano je rješenje problema - ROS lansirna datoteka implementirana za pokretanje čvorova potrebnih za lokalizaciju i skripta za testiranje točnosti procjena položaja dobivenih različitim konfiguracijama sustava za lokalizaciju. **Poglavlje 8** sadrži opis prikupljanja podataka senzora za lokalizaciju na robotu i rezultate testiranja točnosti različitih metoda lokalizacije i konfiguracija fuzije senzora Kalman filtrom.

## Poglavlje 2

# Lokalizacija mobilnog robota

Lokalizacija mobilnog robota je proces procjene poze robota, odnosno položaja i orijentacije robota u odabranom referentnom koordinatnom sustavu. Sposobnost lokalizacije neophodna je vještina za rješavanje nekih vrsta zadataka mobilnog robota [2]. Primjer takvog zadatka je autonomna navigacija, kod koje se od robota očekuje da će samostalno isplanirati put i upravljati kretanjem kako bi došao do svojeg cilja. Periodična usporedba trenutne procjene poze sa željenom pozom robotu omogućuje prilagodbu smjera i brzine kretanja kako bi to ostvario [3].

Za lokalizaciju se koriste informacije dobivene obradom mjerenja senzora robota, koje na određeni način opisuju stanje okoline ili stanje samog robota. Ovisno o vrsti korištenih senzora, lokalizacija može biti relativna ili apsolutna. Kod relativne lokalizacije procijenjuje se promjena poze u odnosu na neki trenutak iz prošlosti integracijom izmjerenih brzina ili akceleracija. Apsolutna lokalizacija podrazumijeva procjenu poze u odnosu na odabrani koordinatni sustav. Primjer rješenja je korištenje sustava za satelitsku navigaciju, koja omogućuju procjenu koordinati prijemnika na Zemlji. Alternativno rješenje problema apsolutne lokalizacije je uz pomoć karte prostora u kojem se robot kreće. Neke od mogućih vrsta karata su karte značajki i karte ćelija zauzetosti (eng. *occupancy grid map*) [4]. Karte značajki definirane su skupom značajki (primjerice točaka, linija, kuteva ili ravnina) i njihovim položajima u odabranom koordinatnom sustavu. Karte ćelija zauzetosti prostor dijele na ćelije jednakih veličina, od kojih je svaka zauzeta ili slobodna. Obradom mjerenja senzora robot dobiva detekcije značajki, te smjer i udaljenost do njih. Podudaranjem de-

## Poglavlje 2. Lokalizacija mobilnog robota

tektiranih značajki sa značajkama na karti dobiva se procjena poze robota na karti [4].

U nastavku ovog poglavlja opisano je nekoliko vrsta senzora čiji se podaci dobiveni mjerenjem mogu koristiti za lokalizaciju, te metode lokalizacije, odnosno postupci koji mjerenja senzora pretvaraju u informacije o pozi robota. Pojedine metode imaju uvjete rada koji mogu drastično pogoršati točnost procjene. Robustnost lokalizacije na greške procjena dobivenih pomoću pojedinačnih senzora i na moguće kvarove senzora može se postići udruživanjem (fuzijom) mjerenja više različitih senzora, što je opisano u poglavlju 3.

### 2.1 Senzori

Mobilni roboti obrađuju mjerenja raznih vrsta senzora kako bi dobili informacije potrebne za lokalizaciju. Osnovna podjela korištenih senzora je na propriocepcijske, koji mjere unutarnje stanje samog robota (primjerice akceleraciju, brzinu ili usmjerenje), i percepcijske, čija mjerenja opisuju određeno svojstvo okruženja robota robota, primjerice udaljenosti do objekta, intenzitet svjetla, ili prisutnost poznatog objekta (orijentira) u okolini robota [3]. Senzori se također mogu podijeliti na pasivne i aktivne. Pasivni senzori mjere energiju iz okoline koja ulazi u senzori. Primjeri pasivnih senzora su mikrofoni i senzori temperature. Aktivni senzori emitiraju energiju u okolinu i mjere reakciju okoline [3]. Primjer su ultrazvučni senzori udaljenosti koji emitiraju ultrazvuk i mjere ga. Vrijeme između emitiranja ultrazvučnog vala i detekcije odbijenog vala omogućuje procjenu udaljenosti između senzora i objekta ispred senzora. Neki kriteriji po kojima se performanse senzora mogu razlikovati su [3]:

- Osjetljivost (eng. *sensitivity*) - opisuje kolika je minimalna promjena mjerene veličine koju senzor može detektirati, odnosno opisati
- Preciznost (eng. *precision*) - opisuje koliko su slična različita mjerenja istog stanja
- Točnost (eng. *accuracy*) - opisuje koliko su mjerenja stanja udaljena od stvarnog stanja, odnosno očekivanu količinu greške prisutne u mjerenjima

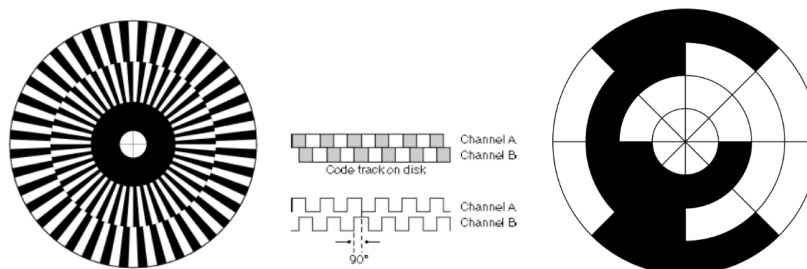
## Poglavlje 2. Lokalizacija mobilnog robota

- Frekvencija uzorkovanja

U nastavku je navedeno nekoliko vrsta senzora koji mogu koristiti prilikom lokalizacije robota.

### 2.1.1 Enkoderi kotača

Enkoderi su elektromehanički uređaji koji mjere broj okretaja kotača. Mogu mjeriti relativni pomak i smjer kotača (inkrementalni enkoderi), ili položaj kotača, odnosno kut u odnosu na određenu os (apsolutni enkoderi). Na robotima se često koriste inkrementalni optički enkoderi, koji mjere rotaciju kotača detektiranjem svjetla kroz prozirne uzorke na rotirajućem disku [4]. Uzorci diska inkrementalnog i apsolutnog enkodera prikazani su na slici 2.1.



Slika 2.1 Uzorci inkrementalnog (lijevo) i apsolutnog (desno) enkodera kotača [4].

### 2.1.2 Inercijalni i orijentacijski senzori

Inercijalni senzori koriste načelo inercije kako bi mjerili akceleraciju ili kutnu brzinu tijela [4]. **Akcelerometri** su elektromehanički senzori koji mjere akceleraciju tijela tako da proizvedu količinu naboja ili promjenu kapaciteta proporcionalnu sili koja djeluje na tijelo po određenoj osi [5]. Korištenjem tri akcelerometara na ortogonalnim osima može se procijeniti akceleracija u trodimenzionalnom prostoru [6].

Dvostrukom (numeričkom) integracijom izmjerenih akceleracija može se procijeniti promjena položaja. Najjednostavniji primjer je procjena položaja u samo jednoj

## Poglavlje 2. Lokalizacija mobilnog robota

dimenziji. Potrebno je imati inicijalne procjene položaja  $\hat{p}_0$  i brzine  $\hat{v}_0$ . Ako akcelerometar periodički (svakih  $\Delta t$  sekundi) daje mjerenje akceleracije  $\hat{a}$ , procjena položaja i brzine u vremenskom koraku  $k$  može se dobiti pomoću novog mjerenja  $\hat{a}_k$  i procjena položaja i brzine iz prethodnog koraka  $k - 1$ :

$$\hat{v}_k = \hat{v}_{k-1} + \Delta t \hat{a}_k \quad (2.1)$$

$$\hat{p}_k = \hat{p}_{k-1} + \Delta t \hat{v}_{k-1} + \frac{1}{2} \Delta t^2 \hat{a}_k \quad (2.2)$$

Svako mjerenje  $\hat{a}$  dobiveno od akcelerometra sadrži određenu grešku pridruženu stvarnoj akceleraciji. Greške mjerenja imaju više komponenti: pristrane (korelirane) greške (eng. *bias error*) koje se zbrajaju stvarnoj vrijednosti, slučajni šum (eng. *noise*) i greške skaliranja stvarnih vrijednosti (eng. *scale factor error*) [7]. Parametri za izračun potreban za uklanjanje dijela greške svakog mjerenja može se odrediti postupkom umjeravanja senzora (eng. *calibration*). Zbog greške prisutne u mjerenjima  $\hat{a}$  se u svakom vremenskom koraku i procjenama trenutnog položaja i brzine zbraja određena greška. Greške procjena s vremenom postaju sve veće zato što se prethodna procjena položaja i brzine koriste u izračunu trenutnih procjena. Zbog toga korištenje samo akcelerometra nije dovoljno za dugoročnu procjenu položaja - potrebno je uključiti i dodatne izvore informacija u procjenu.

**Žiroskopi** su senzori koji ovisno o izvedbi mjere kut ili kutnu brzinu oko jedne ili više osi [8]. Dostupni su mehanički, optički i MEMS žiroskopi. Optički žiroskopi mjere kutnu brzinu istovremenim odašiljanjem dviju laserskih zraka u suprotnim smjerovima po kružnoj putanji i detekcijom zraka u jednoj točki putanje. Ako je jedna zraka detektirana ranije od druge, fiksirani sustav (žiroskop) je rotiran, a razlika u fazi je proporcionalna brzini rotacije [4].

**Kompasi i magnetometri** mjere apsolutnu orijentaciju u odnosu na magnetske polove Zemlje. Ovisno o izvedbi, mjere orijentaciju oko između jedne do tri osi.

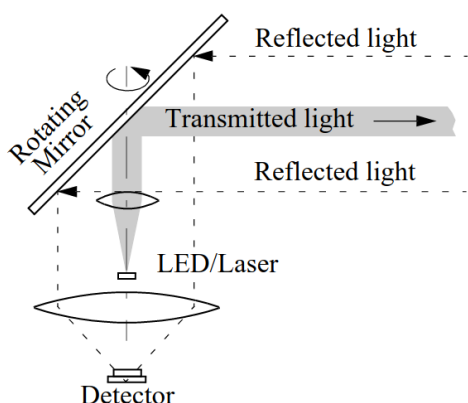
Akcelerometri i žiroskopi se često udružuju zajedno sa magnetometrima u inercijalnim mjernim jedinicama (eng. *inertial measurement unit - IMU*), uređajima koji mjere akceleraciju i promjenu orijentacije u trodimenzionalnom prostoru [8]. Ako je uključen magnetometar, moguće je odrediti i apsolutnu orijentaciju [9].

### 2.1.3 Senzori za mjerenje udaljenosti

Postoje razni percepcijski senzori čija mjerenja opisuju udaljenosti do objekata u okolini. Nakon obrade mjerenja, dobivene informacije su korisne za modeliranje okoline, lokalizaciju i izbjegavanje prepreka, ali obrada može biti računski zahtjevna zato što senzori iz te kategorije, primjerice laserski skeneri udaljenosti ili dubinske kamere, mogu proizvesti velike količine višedimenzionalnih podataka [3].

#### Laserski skener udaljenosti

Laserski skeneri udaljenosti, poznati pod nazivom LIDAR (*LIght Detection And Ranging*), mjere udaljenosti do objekata u okolini odašiljanjem laserske zrake i detektiranjem reflektirane zrake. Brzina elektromagnetskog zračenja, vrijeme odašiljanja i vrijeme detekcije koriste se za procjenu udaljenosti do prepreke u smjeru zrake [4]. Udaljenosti do točaka koje pokrivaju veće područje dobije se rotacijom zrcala koje usmjeruje zraku, odnosno skeniranjem. Ovisno o izvedbi može se razlikovati kutna širina zrake i dimenzionalnost prostora koji skener pokriva (2D ili 3D). Princip rada LIDAR-a i primjer LIDAR senzora prikazani su na slici 2.2.



(a) Princip rada LIDAR senzora [3]

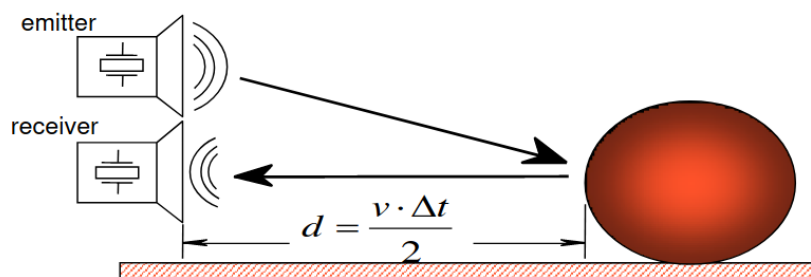


(b) Sick LMS 200 LIDAR senzor [6]

Slika 2.2 LIDAR: princip rada i primjer senzora

### Ultrazvučni senzor udaljenosti

Ultrazvučni senzori udaljenosti, poznati i pod nazivom SONAR (*SOund NAvigation and Ranging*), uključuju ultrazvučni zvučnik i mikrofoni u istom kućištu. Udaljenost do prepreke računa se pomoću vremena emitiranja, vremena detekcije odbijenog vala i brzine zvuka [6] (prikazano na slici 2.3). Za detektiranje udaljenosti do prepreka u različitim smjerovima mogu se koristiti polja ultrazvučnih senzora na jednom robotu, ali kod takvih konfiguracija prisutan je problem interferencije [3].



Slika 2.3 Princip rada ultrazvučnog senzora udaljenosti [6]. Udaljenost od senzora do prepreke  $d$  računa se pomoću brzine zvuka  $v$  i vremena između emitiranja ultrazvučnog signala i detekcije odbijenog signala  $\Delta t$ .

### Dubinske kamere

Dubinske kamere (eng. *depth camera*) omogućuju procjenu udaljenosti do objekata u sceni koju kamera vidi. Dubinske kamere proizvode dubinsku sliku (eng. *depth image*), odnosno dvo-dimenzionalno polje piksela, gdje se svakom pikselu u slici pridružuje vrijednost proporcionalna udaljenosti od kamere do objekta u sceni kojeg taj piksel predstavlja. Primjer dubinske slike prikazan je na slici 2.4.

Model akvizicije slike i intrinzični parametri kamere opisuju smjer zraka koje iz ishodišta koordinatnog sustava kamere prolaze kroz svaki piksel slike. Smjerovi zraka, zajedno sa dubinama pojedinih piksela mogu se iskoristiti za konverziju dubinske slike u 3D oblak točaka (eng. *point cloud*), odnosno skup točaka u 3D prostoru [11]. Uz

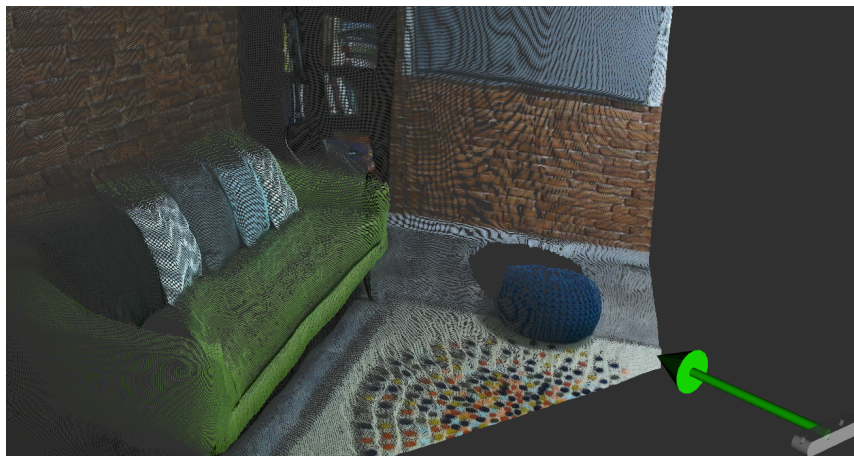


## Poglavlje 2. Lokalizacija mobilnog robota



Slika 2.4 Crno-bijela slika scene (lijevo) i dubinska slika iste scene (desno) [10].

koordinate točaka može se spremati i boja točaka ako dubinska kamera ima RGB senzor. Primjer oblaka točaka prikazan je na slici 2.5.



Slika 2.5 3D oblak RGB točaka dobiven obradom slika dubinske kamere [11]. Na slici je prikazan i 3D model kamere, te jedna od osi koordinatnog okvira kamere (zeleni vektor).

Postoji više vrsta dubinskih kamera. Aktivne **kamere za infracrveno zračenje** emitiraju u okolinu pseudoslučajni, unaprijed poznati uzorak točaka infracrvenog svjetla, koje se zatim detektiraju. Ovisno o veličini i oštrom snimljenih točaka, određuje se udaljenost za pojedine piksele snimljene slike [4].

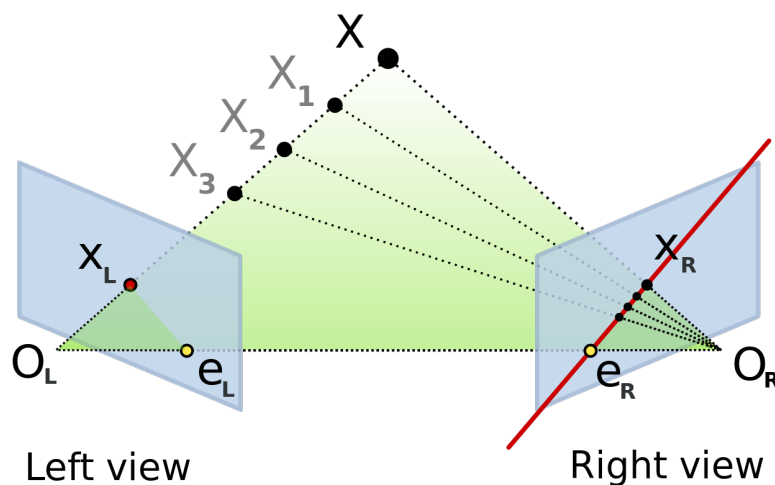
**Kamere strukturiranog svjetla** (eng. *structured light*) u kućištu sadrže projektor koji emitira u okolinu sliku sa određenim uzorcima, primjerice linijama. Projicirani uzorci deformiraju se na različite načine ovisno o objektu na kojem su projicirani. Analizom snimljenih deformacija poznatog uzorka procjenjuje se oblik objekta koji je slikan [4].

**Stereo kamere** iskorištavaju jedan od načina na koji ljudski vid procjenjuje udaljenost do objekata u okolini, odnosno pogled na scenu iz dvije perspektive istovremeno [3]. Dvije kamere između kojih je poznat razmak istovremeno snimaju okolinu i proizvode slike koje se malo razlikuju zbog različitog položaja. Položaji piksela lijeve i desne slike koji predstavljaju istu točku u sceni se razlikuju. Pomak između položaja tih piksela se zove disparitet i inverzno je proporcionalan udaljenosti do točke. Što je veći razmak između dvije kamere (eng. *baseline*), procjena udaljenosti je bolja, ali uz preveliki razmak dio scene neće biti vidljiv u obje slike pa za neke objekte nije moguće izračunati disparitet [3].

Pronalazak odgovarajućih parova piksela (eng. *pixel correspondence*) koji predstavljaju istu točku može biti računski zahtjevan. Epipolarna geometrija opisuje relacije između slika iste scene dobivenih kamerama u različitim položajima. Uz poznavanje intrinzičnih parametara kamera (žarišna duljina, distorzija slike, transformacija koordinati piksela u okvir kamere) i relativene orijentacije i razmaka između kamera (ekstrinzičnih parametara), skup mogućih položaja odgovarajućeg para piksela jedne slike može se ograničiti na jednu liniju na drugoj slici (prikazano na slici 2.6). Ta linija zove se epipolarna linija. Postupkom koji se zove epipolarna rektifikacija slike se mogu transformirati tako da je epipolarna linija uvijek vodoravna, što olakšava traženje parova [3].

#### 2.1.4 Prijemnici za aktivne orijentire

Postupkom triangulacije moguće je uz poznavanje položaja dvije točke trokuta i svih kuteva trokuta odrediti položaj treće točke trokuta. Trilateracija omogućuje izračun položaja ako su poznate udaljenosti do tri točke čiji su položaji poznati. Kod korištenja ovih metoda za lokalizaciju koriste se aktivni orijentiri koji odašiljaju primjerice ultrazvuk, svjetlost ili radio signale. Udaljenosti se mogu procijeniti mjerenjem jačine



Slika 2.6 Epipolarna geometrija opisuje relacije između slika iste scene dobivenih kamerama u različitim položajima [12].

signala i vremena između slanja i detekcije signala. Poznati primjer je GPS (*Global Positioning System*) i ostali sustavi za satelitsku navigaciju - sateliti se koriste kao orijentiri za pozicioniranje prijemnika na Zemlji. Sustavi za pozicioniranje koji koriste aktivne orijentire mogu biti skupi i nefleksibilni jer često zahtjevaju instalaciju u prostoru u kojem se primjenjuju [3].

## 2.2 Postupci za lokalizaciju

### 2.2.1 Odometrija

Odometrija je proces estimacije promjene položaja u odnosu na početnu točku (relativna lokalizacija) obradom mjerenja inercijalnih senzora ili enkodera na kotačima. Mjerenja senzora koriste se za određivanje brzine ili akceleracije, a do procjene položaja se dolazi jednostrukom ili dvostrukom integracijom mjerenja po vremenu, ovisno o tome koristi li se procjena brzine ili akceleracije [3]. Računalo periodički dobiva mjerenja senzora. U slučaju mjerenja brzine, nova procjena položaja dobije se doda-

vanjem umnoška novog mjerenja i vremenskog koraka prethodnoj procjeni položaja.

Osim točnosti samih mjerenja, i trajanje perioda između dva dobivena mjerenja može imati utjecaj na kvalitetu procjene položaja, budući da brzina često nije jednaka izmjerenoj tijekom cijelog perioda [3]. Greške prisutne u mjerenjima propagiraju se u procjenu promjene položaja dobivene odometrijom u svakom koraku, pa procjena položaja s vremenom postaje sve manje točna [2]. Odometrija može biti korisna u slučajevima kada nije potrebno dugoročno pratiti položaj, primjerice za lokalno (kratkoročno) planiranje putanje i izbjegavanje prepreka. U slučaju dugoročnog praćenja položaja inače je potrebno uz odometriju koristiti i metode koje procjenjuju apsolutni položaj kako bi se ispravile akumulirane greške.

## Odometrija kotača

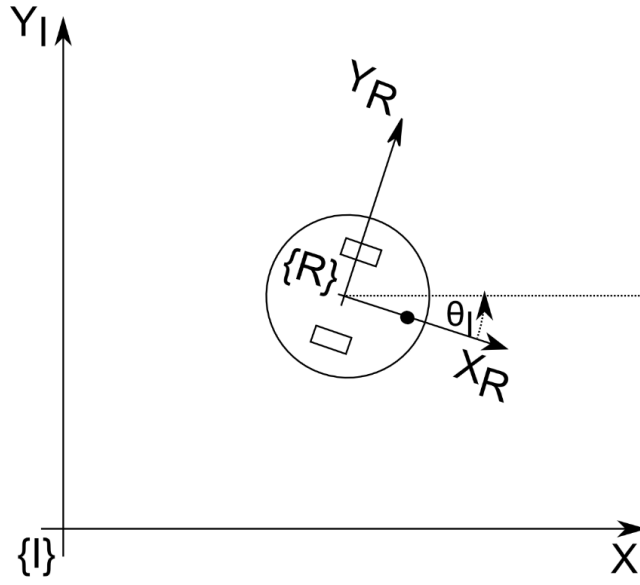
Za određivanje brzine i položaja pomoću mjerenja enkodera kotača potrebno je znati kako brzina rotacije kotača utječe na brzinu samog robota, što je opisano kinematičkim modelom robota, te promjere i razmak između kotača. U nastavku je kao primjer prikazan izračun potreban za odometriju robota sa diferencijalnim pogonom [4].

Kod diferencijalnog pogona koriste se dva pogonska kotača. Kut zakreta robota određen je razlikom brzina lijevog i desnog kotača. Na slici 2.7 prikazani su lokalni koordinatni sustav robota  $R$ , čije je ishodište uvijek u središtu robota i globalni koordinatni sustav  $I$ . Za početak je potrebno odrediti brzine robota u lokalnom koordinatnom sustavu robota  $R$ . Robot se može kretati naprijed i nazad te zakretati se, ali ne može ići po strani. Zato je  $\dot{y}$  komponenta brzine uvijek jednaka 0. Kotač polumjera  $r$  nakon rotacije kutem  $\phi$  prođe put  $\phi r$ . Brzina  $x$  komponente okvira mobilnog robota ovisi o brzini lijevog ( $\dot{\phi}_l$ ) i brzini desnog ( $\dot{\phi}_r$ ) kotača:

$$\dot{x}_R = \frac{r\dot{\phi}_l}{2} + \frac{r\dot{\phi}_r}{2} \quad (2.3)$$

Robot se može zakretati tako da se rotira samo jedan kotač ili rotiranjem kotača u suprotnim smjerovima. Kutna brzina zakreta robota ovisi o polumjeru kotača  $r$ , razmaku između kotača  $d$ , i kutnim brzinama lijevog i desnog kotača  $\dot{\phi}_l$  i  $\dot{\phi}_r$ :

$$\dot{\theta} = \frac{\dot{\phi}_r r}{2} - \frac{\dot{\phi}_l r}{2} \quad (2.4)$$



Slika 2.7 Lokalni koordinatni sustav ( $R$ ) robota s diferencijalnim pogonom i globalni koordinatni sustav  $I$  [4]. Kut  $\theta$  opisuje zakret mobilnog robota.

Za određivanje promjene položaja u globalnom koordinatnom sustavu, potrebno je pretvoriti brzine robota u lokalnom okviru u one u globalnom. Linearne brzine po osima u globalnom okviru ovise o kutu zakreta robota, a kutna brzina ostaje ista:

$$\dot{x}_I = \cos(\theta)\dot{x}_R \quad (2.5)$$

$$\dot{y}_I = \sin(\theta)\dot{x}_R \quad (2.6)$$

$$\dot{\theta}_I = \dot{\theta}_R \quad (2.7)$$

U računalnim implementacijama odometrije brzina kotača  $\dot{\phi}$  se procjenjuje promjenom kuta kotača (mjereno enkoderom) u poznatom vremenskom koraku  $\Delta t$ :

$$\Delta\phi = (\phi_k - \phi_{k-1})/\Delta t \quad (2.8)$$

Ubacivanjem procjene brzine kotača  $\Delta\phi$  u prethodno opisane izraze dobiju se procjene brzina  $\Delta x_I$ ,  $\Delta y_I$  i  $\Delta\theta$ . Množenjem sa vremenskim korakom može se doći do

nove procjene položaja i orijentacije za korak  $k$ :

$$\begin{bmatrix} x_I(k) \\ y_I(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} x_I(k-1) \\ y_I(k-1) \\ \theta(k-1) \end{bmatrix} + \begin{bmatrix} \Delta x_I \\ \Delta y_I \\ \Delta \theta \end{bmatrix} \Delta t \quad (2.9)$$

Iz opisanog izračuna za odometriju vidljivo je da krive procjene polumjera kotača  $r$  i razmaka između kotača  $d$  mogu uvesti greške u svakom novom koraku procjene. Dodatni problemi mogu biti proklizavanje kotača i prepreke na putu koje zaustavljaju robota. Kotači se mogu rotirati i bez pomaka samog robota, što uvodi dodatne greške u procjenu.

## Vizualna odometrija

Vizualna odometrija je inkrementalni proces estimacije kretanja u prostoru korištenjem samo slika dobivenih od kamere [13]. Promjena položaja kamere u periodu između dvije uzastopno dobivene slike procjenjuje se pomoću razlika između dvije uzastopne slike. Kako bi procjena kretanja pomoću promjena u slici bila moguća, u okolini kamere treba biti dovoljno karakterističnih točaka, tekstura ili oblika koji se mogu podudarati na dvije uzastopne slike.

Za procjenu kretanja mogu se koristiti slike jedne kamere (monokularna vizualna odometrija), dvije kamere (stereo odometrija), ili više od dvije kamere. U slučaju monokularne odometrije procijenjene udaljenosti kamere su skalirane - za određivanje stvarnih udaljenosti potrebno je kod procjene uključiti dodatna poznata ograničenja, primjerice veličine objekta u okolini, ili mjerenja drugih senzora kao što su IMU ili GNSS prijemnik. Pristupi za rješavanje vizualne odometrije mogu se podijeliti na tri kategorije. Metode zasnovane na značajkama detektiraju karakteristične točke ili oblike i podudaraju ih na uzastopnim slikama. Pomak kamere procjenjuje se pomoću pomaka između većeg broja parova značajki na uzastopnim slikama. Metode zasnovane na izgledu (globalne metode) koriste vrijednosti piksela cijelih slika ili regija slika. Hibridne metode kombiniraju korištenje značajki i korištenje globalnih metoda [13]. Pregled metoda rješavanja problema vizualne odometrije i vrsta vizualnih značajki koje se mogu koristiti za podudaranje između slika dostupan je u diplomskom radu [14].

## 2.2.2 Satelitska navigacija - GNSS

Globalni sustav za satelitsku navigaciju (eng. *Global Navigation Satellite System - GNSS*) je sustav koji korisniku, odnosno uređaju koji prima signal omogućuje da pomoću signala dobivenih od četiri ili više satelita bilo gdje na Zemlji procijeni svoj položaj [15]. Prvi i najpoznatiji takav sustav je GPS (*Global Positioning System*), koji je 1978. godine razvijen u SAD-u. Osim GPS-a trenutno su aktivni i sljedeći globalni sustavi za satelitsku navigaciju [16]:

- GLONASS, razvijen u Rusiji
- BeiDou, razvijen u Kini
- Galileo, razvijen u državama Europske unije

Većina GNSS prijemnika u modernim uređajima, primjerice mobitelima, može istovremeno koristiti signale od više sustava koji su navedeni, čime se povećava broj i dostupnost satelita među kojima se može odabrati podskup satelita korištenih za izračun i procjenu položaja. Time se povećava robustnost procjene, a odabir satelita posredno utječe i na točnost procjene položaja. Navedeni sustavi razlikuju se po brojevima satelita u konstelaciji, frekvenciji određenih signala i točnosti, ali generalni pristup pozicioniranja je isti.

### Procjena položaja

Procjena položaja pomoću GNSS-a [17] radi se pomoću trilateracije. Sateliti kontinuirano šalju signal prema Zemlji, a prijemnici na Zemlji prate signal nekoliko satelita istovremeno. Udaljenosti do pojedinih satelita se procjenjuju pomoću vremena propagacije signala. Pomoću informacija zapisanih u porukama koje sateliti šalju, prijemnik može odrediti kada je poruka poslana i gdje je satelit bio u trenutku kada je poslana.

Udaljenosti do satelita određuju se množenjem vremena od slanja do prijema signala s brzinom svjetlosti. Zbog toga je točnost određivanja vremena na satelitima i na prijemniku jako važna za određivanja položaja. Sateliti imaju skupe atomske satove visoke točnosti, ali prijemnici, primjerice u mobitelima, imaju značajno jeftinije

## Poglavlje 2. Lokalizacija mobilnog robota

i manje točne satove. Kako bi bilo moguće uz takve greške sata procijeniti položaj prijemnika, kod rješavanja sustava jednadžbi za određivanje položaja se greška sata (kašnjenje) prijemnika ubacuje kao dodatna nepoznata varijabla. Sustav jednadžbi može se riješiti numeričkim postupkom, primjerice Gauss-Newton algoritmom. Rezultat algoritma je procijenjeni položaj i greška sata prijemnika.

### Uzroci grešaka i točnost procjene položaja

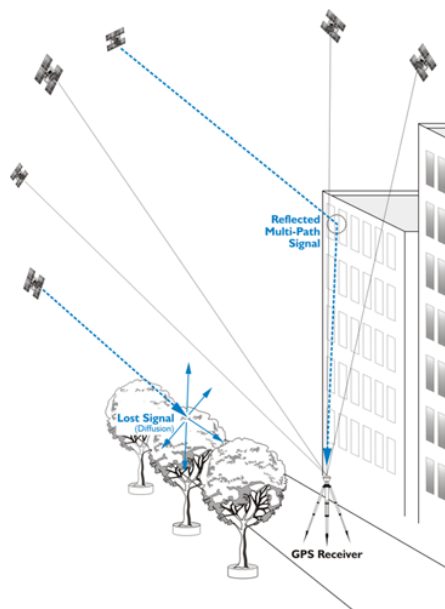
Budući da se udaljenosti od satelita računaju množenjem vremena propagacije signala i brzine elektromagnetskog zračenja, krive pretpostavke o brzini signala mogu dovesti do greške u procjeni položaja. Korištena brzina svjetlosti  $c$  vrijedi u vakuumu, ali signal mora proći kroz ionosferu i troposferu prije nego dođe do površine Zemlje, što usporava signal i dovodi do greške u procjeni [15]. Dio ovih grešaka se ispravlja standardnim modelima kašnjenja signala (Klobucharov model [18]), ali dio greške ostaje u procjeni zbog nesavršenosti modela.

Drugi izvor greške su višestazni putevi signala (eng. *multipath*) - signal se odbija od zgrada i ostalih površina zbog čega više kopija signala dolazi do prijemnika u različitim trenucima, što negativno utječe na izračun udaljenosti [15]. Takve greške su česte u urbanim okruženjima (slika 2.8). Procjena položaja može postati bolja povećanjem broja satelita koji su vidljivi. Kod korištenja GNSS-a na otvorenom prostoru, gdje nema visokih zgrada i površina od kojih bi se signal mogao odbijati, generalno se mogu očekivati dobri rezultati bez puno degradacija točnosti. U odnosu na urbana okruženja, broj satelita vidljivih prijemniku trebao bi biti veći, a greška uzrokovana višestaznim putevima signala manja.

GNSS prijemnici u pametnim telefonima tipično imaju točnost oko 5 metara u dobrim uvjetima [19] (može se očekivati da je stvaran položaj udaljen do 5 metara od procjene). Skuplji prijemnici koji imaju mogućnost korištenja RTK (*Real-time Kinematic*) sustava mogu postići točnost i do  $<2\text{cm}$ .



## Poglavlje 2. Lokalizacija mobilnog robota



Slika 2.8 GNSS greške tipične za urbana okruženja [15]

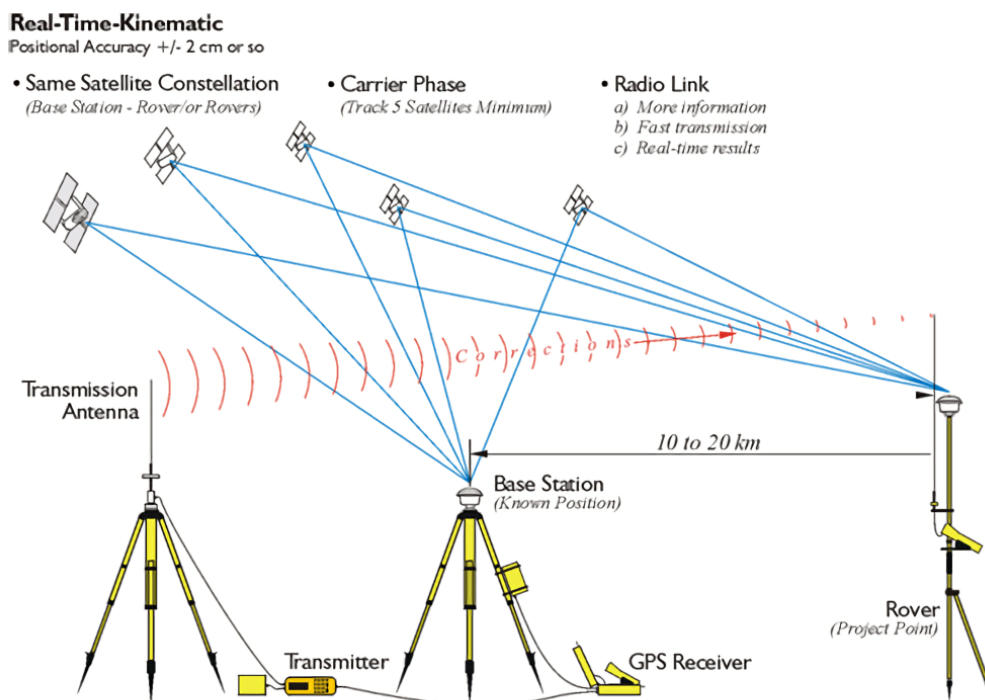
### Tehnike poboljšanja točnosti i preciznosti procjene položaja - DGNSS i RTK

Veliki udio izvora grešaka koje utječu na GNSS pozicioniranje su korelirane, primjerice očekivano je da će greške uzrokovane ionosferskim kašnjenjem biti slične za nekoliko uzastopnih mjerenja ako prijemnik ostane na približno istom položaju [15]. Rezultati GNSS pozicioniranja mogu se poboljšati korištenjem baznih stanica za koje su unaprijed poznati položaji. Takve stanice mogu kontinuirano GNSS-om raditi procjenu vlastitog položaja i uspoređivati procijenjeni položaj sa stvarnim koji je već poznat. Kada bazna stanica izračuna grešku procjene, može odrediti kakve su korekcije potrebne za izračun točnog položaja ako je poznata procjena položaja koja ima korelirane greške u uzastopnim mjerenjima. Takve korekcije bazna stanica kontinuirano šalje prijemnicima koji se nalaze u okolini (slika 2.9). Ovaj pristup se zove diferencijalni GNSS (DGNSS) [15].

Real Time Kinematics (RTK) je DGNSS [15] tehnika koja omogućava točnost do  $<2\text{cm}$ . Udaljenost prijemnika u pokretu do bazne stanice može biti i do 20

## Poglavlje 2. Lokalizacija mobilnog robota

km, a podaci o korekcijama se mogu slati preko mobilnih komunikacijskih mreža ili interneta. U Hrvatskoj je moguće korištenje CROPOS sustava [20], koji čine 33 referentnih GNSS stanica za RTK i slične korekcije. Slični sustavi su dostupni i u drugim državama.



Slika 2.9 Korištenje GNSS RTK sustava [15]

## Poglavlje 3

# Fuzija senzora za lokalizaciju

Udruživanjem više senzora i izvora podataka tijekom lokalizacije, odnosno fuzijom senzora, moguće je dobiti točniju procjenu položaja i orijentacije robota [1]. Jednostavan način udruživanja je usrednjavanje procjena tražene veličine dobivenih od svakog senzora, ali metode koje kod procjene uzimaju u obzir i očekivanu grešku svakog korištenog senzora mogu postići bolje rezultate. Robustnost lokalizacije robota dodatno raste ako korištena metoda za fuziju senzora uz najbolju procjenu poze kao rezultat daje i neizvjesnost procjene. Time se robotu omogućuje da odustane od akcija potencijalno opasnih za njega ili okolinu, ako nije siguran u trenutnu procjenu poze.

Neizvjesnost procjene poze robota može se modelirati razdiobama vjerojatnosti [3]. U slučaju lokalizacije na karti ćelija može se definirati diskretna razdioba tako da se svakoj ćeliji karte odredi vjerojatnost da odgovara položaju robota. Procjena se može predstaviti i skupom čestica, odnosno uzoraka iz razdiobe mogućih poza, ili višedimenzionalnom normalnom razdiobom, kod koje vektor očekivanih vrijednosti predstavlja najvjerojatniju pozu.

Od mjerenja proprioceptijskih senzora odometrijom se dobivaju procjene poze koje s vremenom sadrže sve veću grešku (opisano u 2.1.2). Mjerenja perцепcijskih senzora (detekcije značajki na karti) mogu ukloniti dio neizvjesnosti procjene, ali nisu nužno dostupna u svakom koraku procjene. Fuzija proprioceptijskih i perцепcijskih senzora može se izvesti korištenjem Bayes filtra, odnosno rekurzivnom primjenom

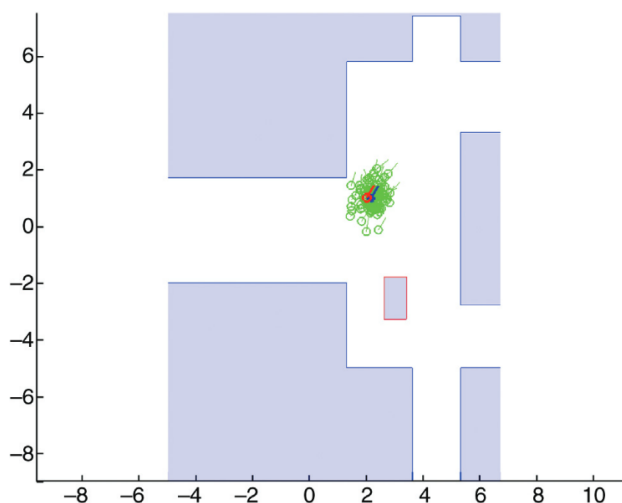
Bayes-ove formule za izračun najvjerojatnije trenutne poze za svako novo mjerenje:

- kod novih detekcija značajki vjerojatnosti trenutnog položaja se računaju pomoću prethodne procjene položaja i uvjetnih vjerojatnosti detektiranja značajki ovisno o položaju
- kod novih odometrijskih mjerenja vjerojatnosti trenutnog položaja se računaju pomoću prethodne procjene položaja i uvjetnih vjerojatnosti dobivanja zadnjeg mjerenja ovisno o zadnjem i novom položaju
- uzima se pretpostavka da sljedeći položaj ne ovisi o prethodnim položajima, ako je poznat trenutni položaj (Markovljevo svojstvo)

Ovaj pristup zove se **Markovljeva lokalizacija** [4] i često nije primjenjiv u praksi jer je računski jako zahtjevan - u svakom koraku izračuna razmatraju se svi mogući položaji robota. Čestični filter i Kalman filter uvode ograničenja koja olakšavaju izračun i fuziju senzora za lokalizaciju na stvarnim robotima.

### 3.1 Čestični filter

Čestični filter smanjuje računске zahtjeve za Markovljevu lokalizaciju tako da u svakom vremenskom koraku moguće poze ograniči na odabran broj točaka u prostoru (čestica) [4]. Na početku lokalizacije bira se slučajni uzorak čestica na cijeloj karti ili oko inicijalne poze ako je poznata. Kod kretanja robota, pomak robota izračunat odometrijom se primjenjuje na sve čestice, ali pomaku svake čestice se dodaje i slučajno uzorkovani šum koji predstavlja grešku odometrije. Prilikom detekcije orijentira (značajki) na karti, računa se vjerojatnost poze svake čestice uzimajući u obzir detektirane orijentire. Dio čestica koji ima nisku vjerojatnost se odbacuje, a čestice koje imaju najvišu vjerojatnost se dupliciraju. Na taj način se uzastopnim detekcijama orijentira čestice skupljaju oko najvjerojatnijeg položaja robota (slika 3.1).



Slika 3.1 Rezultat lokalizacije čestičnim filtrom na 2D karti [2]. Prikazan je skup čestica mogućih poza (zeleno), stvarna poza (crveno) i procjena poze dobivena usrednjavanjem skupa čestica (plavo).

## 3.2 Kalman filtar

Kalman filtar je matematički postupak za procjenu stanja dinamičkih sustava [21]. Primjenjuje se kada je potrebno uzastopno određivati stanje sustava koji se mijenja tijekom vremena, koristeći mjerenja koja sadrže grešku. Algoritam za procjenu trenutnog stanja koristi procjenu prethodnog stanja i mjerenja trenutnog stanja. Budući da nije potrebno pamtititi više prethodnih stanja, postupak nije računski zahtjevan i prigodan je za korištenje u ugradbenim sustavima.

Kalman filtar pretpostavlja da je dinamički sustav linearan, te da se šum prisutan u promatranom dinamičkom sustavu i u mjerenjima senzora može predstaviti slučajnom varijablom sa funkcijom gustoće vjerojatnosti normalne razdiobe. Ako su pretpostavke točne, procjene stanja dobivene filtrom su optimalne, odnosno imaju najmanju moguću varijancu [21].

Algoritam se sastoji od dva glavna koraka koji se ponavljaju. U prvom koraku predviđa se novo stanje pomoću trenutne procjene stanja i modela promatranog dinamičkog sustava. Stanje sustava je predstavljeno vektorom normalno raspodijeljenih

slučajnih varijabli. Drugi korak je korekcija - nova mjerenja, koja sadrže određenu količinu greške, koriste se za ispravljanje predviđenog stanja. Novo stanje se računa kao težinski prosjek procijenjenog i izmjerenog stanja, tako da se težine dodijeljuju ovisno o omjeru neizvjesnosti (varijance) procjene i mjerenja.

### 3.2.1 Izračun - predikcija

Stanja sustava modelira se kao normalno raspodijeljen slučajni vektor. Procijenjeno stanje u trenutku  $k$  opisano je vektorom očekivanih vrijednosti  $\hat{\mathbf{x}}_k$  i matricom kovarijanci greške procjene  $\mathbf{P}_k$ , (eng. *state estimate error covariance*).

Za korištenje Kalman filtra potrebno je matematički modelirati dinamički sustav koji se promatra. Linearni dinamički sustavi sa diskretnim vremenskim koracima mogu se opisati u matričnom obliku. Transformacija očekivanog stanja procjene iz prethodnog koraka u trenutni definira se sa:

$$\hat{\mathbf{x}}_k = \mathbf{F}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k \quad (3.1)$$

gdje je  $\mathbf{F}$  predikcijska matrica koja opisuje promjenu stanja iz prethodnog vremenskog koraka u trenutni,  $\mathbf{u}$  upravljački vektor (eng. *control vector*) koji opisuje utjecaj okoline na sustav, a  $\mathbf{B}$  upravljačka matrica koja opisuje dodatnu promjenu promatranog stanja uzrokovanu vanjskim utjecajem.

Transformacija kovarijance procjene stanja definira se sa:

$$\mathbf{P}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q} \quad (3.2)$$

$\mathbf{Q}$  je matrica šuma procesa (eng. *process noise covariance*), koja opisuje dodatnu neizvjesnost koju se uvodi u svakom koraku predikcije, što je korisno za modeliranje stvarnih sustava kod kojih ne možemo deterministički odrediti sljedeće stanje pomoću prethodnog. Korak korekcije koristi mjerenja stanja kako bi se spriječio neograničen rast kovarijanci, odnosno neizvjesnosti procjene stanja sustava.

Jedan primjer korištenja Kalman filtra je praćenje položaja ( $p$ ) i brzine ( $v$ ) vozila u jednoj dimenziji [22]. Ako vozilo ne mijenja brzinu između koraka predikcije, novi

### Poglavlje 3. Fuzija senzora za lokalizaciju

položaj i brzina u koraku  $k$  može se dobiti sa:

$$p_k = p_{k-1} + \Delta t v_{k-1} \quad (3.3)$$

$$v_k = v_{k-1} \quad (3.4)$$

Ako model uključuje naredbu za željenu akceleraciju ( $a$ ) koju vozač može u svakom trenutku dati vozilu, izrazi se mijenjaju:

$$p_k = p_{k-1} + \Delta t v_{k-1} + \frac{1}{2} \Delta t^2 a \quad (3.5)$$

$$v_k = v_{k-1} + \Delta t a \quad (3.6)$$

U svakom trenutku postoji odstupanje ubrzanja stvarnog sustava u odnosu na zadano ubrzanje zbog nepredvidljivih utjecaja okoline (primjerice proklizavanje), koji u svakom vremenskom koraku uvode određenu grešku u procjenu. Rast greške procjene u svakom koraku može se modelirati zbrajanjem normalno raspodijeljenih slučajnih varijabli  $X_p \sim \mathcal{N}(0, \sigma_p^2)$  i  $X_v \sim \mathcal{N}(0, \sigma_v^2)$  svakoj novoj procjeni položaja i brzine.

Opisani model može se definirati i u matričnom obliku koji je prikladniji za izračun na računalima:

$$\hat{\mathbf{x}}_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a = \mathbf{F} \hat{\mathbf{x}}_{k-1} + \mathbf{B} u_k \quad (3.7)$$

$$\mathbf{P}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{P}_{k-1} \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} + \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} = \mathbf{F} \mathbf{P}_{k-1} \mathbf{F}^T + \mathbf{Q} \quad (3.8)$$

#### 3.2.2 Izračun - korekcija

U svakom koraku predikcije povećava se neizvjesnost procjene stanja dodavanjem matrice  $\mathbf{Q}$  matrici kovarijanci procjene stanja  $\mathbf{P}$ . Ukoliko se u svakom koraku radi samo predikcija, kovarijance neograničeno rastu i procijenjena stanja postaje beskorisna. Zato je u petlji filtra potreban i korak korekcije, koji koristi mjerenja stanja kako bi spriječio neograničen rast kovarijanci, odnosno neizvjesnosti procjene stanja sustava.

### Poglavlje 3. Fuzija senzora za lokalizaciju

Prvo je potrebno od senzora dobiti mjerenja koja opisuju stanje sustava. Sensori nisu savršeni - pretpostavlja se da su u svakom vektoru mjerenja senzora ( $\mathbf{y}_k$ ) prisutne normalno raspodijeljene greške s očekivanim vrijednostima 0 i opisane matricom kovarijance  $\mathbf{R}$ , koja se koristi za modeliranje točnosti i preciznosti senzora.

Mjerenja senzora potrebno je transformirati ako mjerene veličine ne odgovaraju veličinama koje filter procjenjuje. Primjer takve transformacije je skaliranje u slučaju da je mjerna jedinica različita u mjerenjima senzora i u vektoru stanja filtra, ili rotacija ako su određene varijable stanja i mjerenja (primjerice komponente brzine) izražene u različitim koordinatnim okvirima. Formulacija Kalman filtra vektor mjerenja dobivenih u trenutnom koraku ( $\mathbf{y}_k$ ) opisuje kao normalno raspodijeljen slučajni vektor dobiven transformacijom trenutnog stvarnog stanja i dodavanjem šuma. Transformaciju stanja sustava u domenu mjerenja i obrnuto opisuje matrica  $\mathbf{H}$ . Matrica  $\mathbf{H}$  može se iskoristiti za izračun razdiobe mjerenja očekivane za trenutnu procjenu stanja:

$$\mu_{expected} = \mathbf{H}\hat{\mathbf{x}}_k \quad (3.9)$$

$$\Sigma_{expected} = \mathbf{H}\mathbf{P}_k\mathbf{H}^T \quad (3.10)$$

što omogućuje usporedbu trenutne procjene stanja sa stvarnim mjerenjima.

Cilj koraka korekcije je iskoristiti nova mjerenja za poboljšanje trenutne procjene stanja, odnosno za smanjivanje kovarijanci greške stanja u matrici  $\mathbf{P}_k$ . Korekcija procjene stanja može se opisati kao težinski prosjek procijenjenog i izmjenjenog stanja, sa težinama dodijeljenim ovisno o omjeru varijanci procjene i mjerenja, tako da se minimizira varijanca procjene stanja dobivene ovom operacijom [21]. U slučaju jednodimenzionalnog vektora stanja, za danu predikciju stanja  $\mathcal{N}(\mu_{pr}, \sigma_{pr}^2)$  i mjerenje  $\mathcal{N}(\mu_{mj}, \sigma_{mj}^2)$ , korekcijom se dobije razdioba  $\mathcal{N}(\mu_{kor}, \sigma_{kor}^2)$  za koju vrijedi [22]:

$$\mu_{kor} = \mu_{pr} + k(\mu_{mj} - \mu_{pr}) \quad (3.11)$$

$$\sigma_{kor}^2 = (1 - k)\sigma_{pr}^2 \quad (3.12)$$

$$k = \frac{\sigma_{pr}^2}{\sigma_{pr}^2 + \sigma_{mj}^2} \quad (3.13)$$

Vidljivo je da očekivana vrijednost nove razdiobe između očekivanih vrijednosti  $\mu_{pr}$  i  $\mu_{mj}$ , te da je bliže očekivanoj vrijednosti razdiobe koja ima manju varijancu. Vrijed-



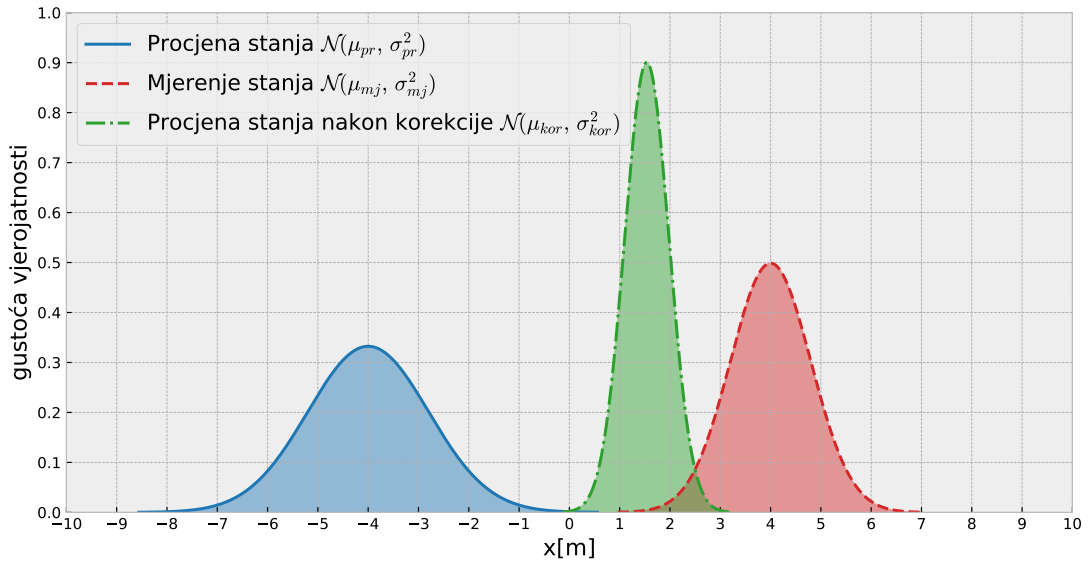
### Poglavlje 3. Fuzija senzora za lokalizaciju

nost  $k$ , koja određuje koliko će korekcija procjenu stanja približiti mjerenju, uvijek je između 0 i 1, a dobivena varijanca  $\sigma_{kor}^2$  uvijek je manja od  $\sigma_{pr}^2$ .

Rezultat množenja funkcija gostoće razdiobe procjene i razdiobe mjerenja proporcionalan je razdiobi koja se dobije korekcijom [23] [24]:

$$\mathcal{N}(\mu_{pr}, \sigma_{pr}^2) \cdot \mathcal{N}(\mu_{mj}, \sigma_{mj}^2) \propto \mathcal{N}(\mu_{kor}, \sigma_{kor}^2) \quad (3.14)$$

Vizualni primjer rezultata koraka korekcije prikazan je na slici 3.2.



Slika 3.2 Za danu procjenu stanja varijable  $x$   $\mathcal{N}(\mu_{pr}, \sigma_{pr}^2)$  (plavo) i mjerenje stanja  $\mathcal{N}(\mu_{mj}, \sigma_{mj}^2)$  (crveno), rezultat koraka korekcije Kalman filtra je optimalna procjena stanja  $\mathcal{N}(\mu_{kor}, \sigma_{kor}^2)$  (zeleno), koja je između originalne procjene i mjerenja. Korekcijom se smanjuje varijanca procjene stanja.

Prebacivanjem prethodno opisanih izraza u matrični oblik i ubacivanjem razdiobe procjene stanja i razdiobe izmjerenog stanja dobiju se izrazi potrebni za korak korekcije za višedimenzionalne vektore stanja. Konstantu  $k$  zamjenjuje matrica Kalmanovog pojačanja  $\mathbf{K}$  (eng. *Kalman gain*):

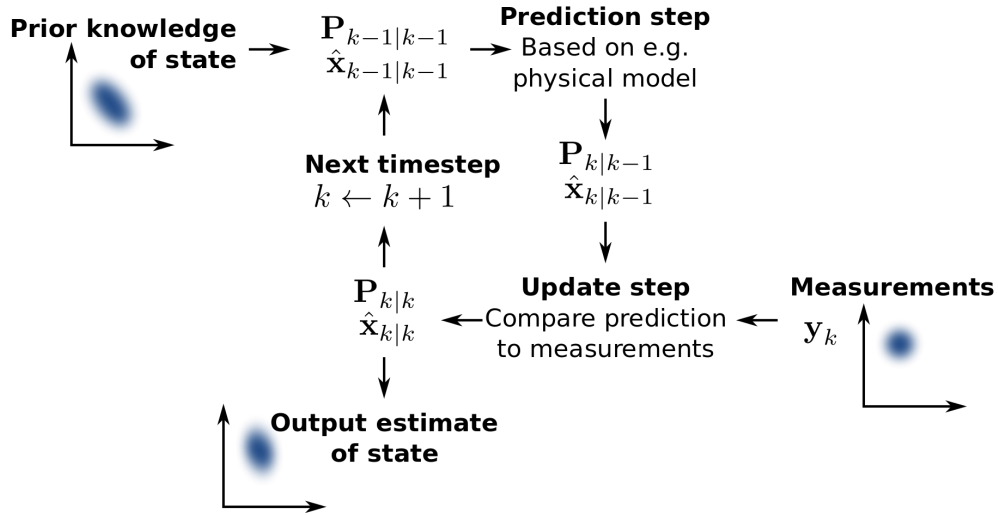
$$\mathbf{K} = \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.15)$$

Pomoću matrice  $\mathbf{K}$  računa se optimalna procjena stanja  $\hat{\mathbf{x}}'_k$  za dobivena mjerenja i pripadajuća matrica kovarijance stanja  $\mathbf{P}'_k$ :

$$\hat{\mathbf{x}}'_k = \hat{\mathbf{x}}_k + \mathbf{K}(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k) \quad (3.16)$$

$$\mathbf{P}'_k = \mathbf{P}_k - \mathbf{KHP}_k \quad (3.17)$$

Ako je neizvjenost mjerenja manja od neizvjenosti originalne procjene, procjena stanja  $\hat{\mathbf{x}}'_k$  je nakon korekcije približena izmjerenom stanju, a vrijednosti kovarijanci u  $\mathbf{P}'_k$  su smanjene u odnosu na  $\mathbf{P}_k$ . Izračunom  $\hat{\mathbf{x}}'_k$  i  $\mathbf{P}'_k$  završava vremenski korak  $k$ . U koraku  $k + 1$   $\hat{\mathbf{x}}'_k$  se koristi u izračunu za predikciju  $\hat{\mathbf{x}}_{k+1}$  i  $\mathbf{P}_{k+1}$ . Predikcija stanja, uzimanje novih mjerenja i korekcija procjene mogu se ponavljati u petlji sve dok je potrebno dobivati procjenu stanja (prikazano na slici 3.3).



Slika 3.3 Glavni koraci Kalman filter algoritma [25]. U koraku predikcije predviđa se sljedeće stanje pomoću procjene trenutnog stanja i modela dinamičkog sustava. Nakon uzimanja novih mjerenja, optimalna procjena trenutnog stanja dobije se korekcijom predikcije pomoću novih mjerenja (*update step* na slici).

### 3.2.3 Korištenje Kalman filtra u praksi

Kod primjene Kalman filtra za estimaciju stanja, dinamički sustav čije se stanje estimira treba biti linearan. Potrebno je konfigurirati matrice kovarijance šuma procesa

$\mathbf{Q}$  i šuma senzora  $\mathbf{R}$ , što može biti zahtjevno [1]. Ovisno o vrijednostima, filter će više vjerovati predikciji stanja ili mjerenjima. Karakteristike šuma i očekivana točnost može biti navedena u specifikacijama senzora. Potrebno je zadati i inicijalnu procjenu stanja i kovarijance stanja,  $\hat{\mathbf{x}}_0$  i  $\mathbf{P}_0$ . Filter može konvergirati na stvarnu vrijednost i slučaju da je inicijalna procjena značajno udaljena od stvarnog stanja. Brzina konvergencije ovisi o vrijednostima  $\mathbf{P}_0$  - ako su vrijednosti jako male, filteru je potrebno više vremena da dođe blizu stvarnog stanja.

### 3.2.4 Prošireni Kalman filter - EKF

Brojni dinamički sustavi koji su korisni u praksi, primjerice razni modeli kretanja mobilnih robota, nisu linearni. Kalman filter je potrebno prilagoditi kako bi se mogao koristiti za estimaciju stanja takvih sustava. Jedno rješenje za to je prošireni Kalman filter (eng. *Extended Kalman Filter - EKF*). EKF podrazumijeva da dinamički model i model senzora nisu nužno linearni [21]. Promjena stanja između vremenskih koraka i model mjerenja opisani su sa (moguće nelinearnim) funkcijama  $f$  i  $h$ :

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_{k-1} \quad (3.18)$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (3.19)$$

gdje je  $\mathbf{w}_{k-1}$  normalno raspodijeljeni šum procesa, a  $\mathbf{v}_k$  šum senzora. Predikcija očekivanih vrijednosti stanja  $\hat{\mathbf{x}}$  u koraku  $k$  računa se pomoću funkcije  $f$ :

$$\hat{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \quad (3.20)$$

Kalman filter stanje u svakom trenutku modelira kao vektor normalno raspodijeljenih slučajnih varijabli. Za transformaciju matrice kovarijance stanja iz prethodnog koraka u novu i dobivanje kovarijance mjerenja koristi se linearizacija funkcija, odnosno Jacobijeva matrica tih funkcija u okolini prethodnog stanja [21]:

$$\mathbf{F}_k := \mathbf{J}_{f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)} \quad (3.21)$$

$$\mathbf{H}_k := \mathbf{J}_{h(\hat{\mathbf{x}}_k)} \quad (3.22)$$

### *Poglavlje 3. Fuzija senzora za lokalizaciju*

Ostali koraci algoritma su jednaki standardnom Kalman filtru.

U slučajevima kada su promatrani dinamički sustav i model mjerenja jako nelinearni, *Unscented* varijanta Kalman filtra (UKF) može postići točniju procjenu stanja, ali je računski zahtjevnija [21]. U ovom radu nije korišten UKF.

## Poglavlje 4

# Istovremena lokalizacija i kartiranje

Za dugoročnu lokalizaciju fuzijom senzora potrebno je među izvorima za fuziju imati mjerenja koja se mogu iskoristiti za procjenu apsolutnog položaja jer greška procjene odometrijom brzo raste. Praktično rješenje za apsolutnu lokalizaciju u vanjskom okruženju je korištenje satelitske navigacije, ali točnost procjene ovisi o faktorima na koje se ne može uvijek utjecati, primjerice broj i raspored satelita vidljivih u radnom okruženju robota. Alternativno rješenje za apsolutnu lokalizaciju je korištenje karte prostora i detektiranje značajki čiji je položaj na karti poznat.

Za izgradnju karte često se koriste LIDAR senzori ili kamere. Ako karta prostora nije dostupna u trenutku kada robot kreće sa radom, može se stvoriti istovremenom lokalizacijom i kartiranjem (eng. *Simultaneous Localization And Mapping - SLAM*). SLAM proširuje problem lokalizacije tako da je uz pozu robota potrebno istovremeno procijeniti i položaj značajki detektiranih u prostoru i na taj način stvoriti kartu [2]. U nastavku su ukratko opisani neki pristupi rješenju SLAM problema.

### 4.1 EKF SLAM

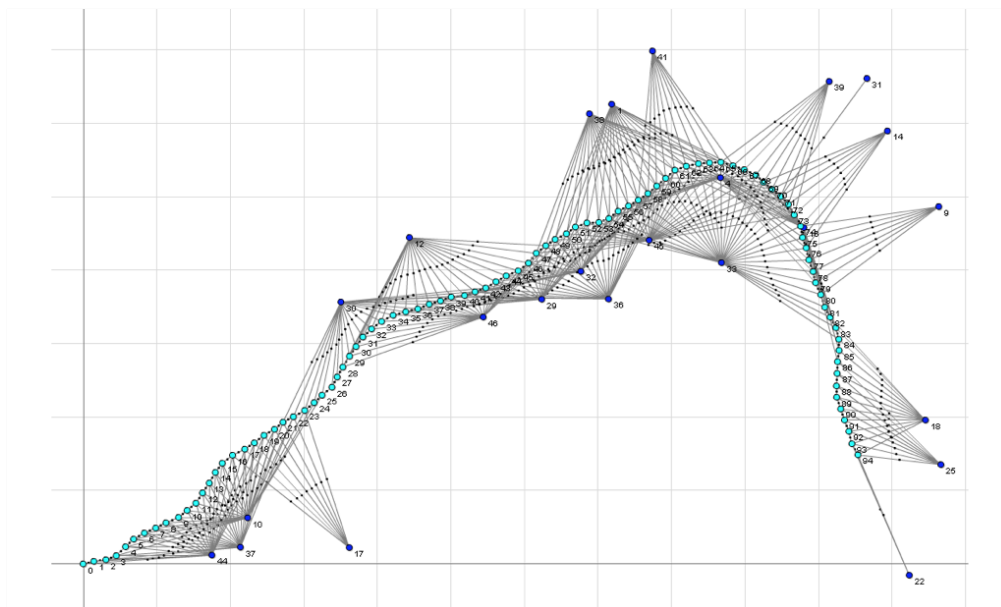
EKF SLAM koristi prošireni Kalman filtar za lokalizaciju robota, ali proširuje problem estimacije stanja tako da vektor varijabli stanja koje filtar kontinuirano procjenjuje uz stanje robota sadrži i položaj značajki [4]. U koraku predikcije procjenjuje se samo novo stanje robota. Nakon mjerenja, u koraku korekcije u vektor stanja dodaju

se značajke koje robot prvi put vidi i ažurira se procjena položaja postojećih značajki ako su ponovno detektirane. Uz uvjet da je pomoću senzora moguće procijeniti smjer i udaljenost do značajke, položaj značajke može se inicijalizirati tako da se položaju robota zbroji pomak određen tim mjerenjima. Primjer primjene je izgradnja karte tijekom kretanja kroz šumu. Značajke u ovom slučaju mogu biti debela stabala koja se detektiraju traženjem kružnih lukova određenih veličina u skenovima dobivenim LIDAR sensorom.

## 4.2 SLAM zasnovan na grafovima

Putanja robota i detektirane značajke od kojih se karta sastoji mogu se zajedno predstaviti grafom čiji su čvorovi poze robota u različitim vremenskim koracima ili značajki u prostoru [26]. Bridovi takvog grafa predstavljaju relativne transformacije odnosno pomak između poza robota u uzastopnim vremenskim koracima te između pojedinih značajki i svih poza (trenutaka) u kojima su detektirane (slika 4.1). Transformacije zapisane u bridovima mogu se inicijalno odrediti pomoću odometrije za putanju robota i modela mjerenja perцепcijskih senzora za izračun položaja značajki. Pretpostavlja se da svaka transformacija inicijalizirana na taj način sadrži određeni šum (grešku), pa je svakom bridu pridružena varijanca koja modelira neizvjesnost.

Pristup zasnovan na grafovima SLAM formulira kao problem procjene najveće izglednosti (eng. *Maximum Likelihood Estimation - MLE*) [4]. Cilj je optimizacijom odabrati položaje značajki na karti i poze robota u svakom trenutku tako da se uz ograničenja definirana bridovima maksimizira združenu vjerojatnost svih opažanja robota i detekcija značajki tijekom kretanja robota [4]. Optimizacijski problem može se riješiti iterativnim numeričkim metodama, primjerice stohastičkim gradijentnim spustom.



Slika 4.1 SLAM zasnovan na grafovima [26]. Apscisa i ordinata na slici predstavljaju osi koordinatnog okvira 2D karte značajki. Poze robota (cijan) i detektirane značajke (plavo) su čvorovi grafa. Bridovi su relativne transformacije (pomak) između čvorova, inicijalizirane pomoću opažanja (mjerjenja senzora). Putanja i karta (položaji značajki) optimiziraju se maksimizacijom združenih vjerojatnosti svih opažanja.

### 4.3 Vizualni RGB-D SLAM

Korištenjem kamera robot dobiva velike količine podataka korisnih za modeliranje okoline, lokalizaciju i donošenje odluka. Robustnost postupaka za lokalizaciju koji koriste kamere dodatno se povećava ako slike sadrže i informacije o dubini piksela, odnosno udaljenosti. Dodatna prednost dubinskih (RGB-D) kamera je da omogućuju izradu detaljnih 3D karata okoline kada se koriste kao senzor za SLAM. Trodimenzionalni prostor može se modelirati na računalu kao oblak točaka (eng. *point cloud*), odnosno skup točaka i njihovih koordinata u 3D prostoru.

Vizualni RGB-D SLAM [4] algoritmi inkrementalno grade kartu prostora. Slike dobivene od kamere pretvaraju se u oblak točaka vidljivog prostora u blizini robota. Globalna karta je inicijalno jednaka prvom oblaku koji je dobiven na taj

način. Karta postaje veća tako da se novi oblaci točaka dobiveni tijekom kretanja spajaju sa globalnom kartom. Jedan način spajanja je podudaranjem točaka prisutnih u oblakcima točaka dobivenih od uzastopnih slika pomoću ICP (*Iterative Closest Point*) algoritma [4]. Zbog šuma senzora, nestatičnih objekata u okolini i drugih razloga spajanje oblaka u svakoj iteraciji uvodi dodatnu grešku koja se akumulira. S vremenom karta postaje iskrivljena. Jedan način ispravljanja karte je ponovnim posjećivanjem dijela karte koje je robot prethodno posjetio, odnosno zatvaranjem petlje.

### 4.3.1 Zatvaranje petlje

Vizualni RGB-D SLAM algoritmi za globalnu optimizaciju karte, odnosno za spajanje manjih oblaka točaka mogu koristiti prethodno opisani pristup zasnovan na grafovima, tako da se relativne transformacije računaju između oblaka točaka umjesto pojedinačnih značajki. Povratak robota na prethodno posjećenu lokaciju - zatvaranje petlje putanje (eng. *loop closure*), omogućuje dodavanje novog brida u graf [4]. Dodani brid predstavlja ograničenje da trenutna poza robota mora biti blizu određenoj pozi iz prošlosti, čime se može ukloniti veliki dio grešaka putanje skupljenih tijekom kretanja između tih poza.

Jedan način detektiranja zatvaranja petlje je korištenjem BoVW (*Bag of Visual Words*) metode [27]. BoVW omogućuje pretvaranje skupa lokalnih značajki slike, primjerice SIFT [28], u vektor fiksne dimenzionalnosti koji opisuje prisutnost i broj karakterističnih značajki - "vizualnih riječi". Na taj način mogu se dobiti globalni deskriptori slike, koji omogućuju procjenu sličnosti slika izračunom udaljenosti deskriptora. Ako se periodično tijekom kretanja spremaju BoVW deskriptori slika kamere, zatvaranje petlje može se detektirati izračunom sličnosti trenutne slike i prethodnih. Primjeri SLAM algoritama koji koriste BoVW za zatvaranje petlje su ORB-SLAM2 [29] i RTAB-Map [30]. Detaljniji pregled i opis rješenja za vizualni RGB-D SLAM dostupan je u diplomskom radu [14].



# Poglavlje 5

## ROS (Robot Operating System)

Robot Operating System (ROS) [31] je radni okvir i skup knjižnica otvorenog koda koji omogućuje stvaranje robotskih aplikacija. Koristi se na sveučilištima za edukaciju i istraživanja te u raznim industrijama<sup>1</sup>. ROS robotskim aplikacijama pruža prilagođene usluge koje tipično nudi opći operacijski sustav: apstrakcija sklopovlja, korištenje upravljačkih programa, komunikacija između procesa na jednom ili više računala i upravljanje paketima. Kao podloga za ROS tipično se koristi Ubuntu distribucija Linux operacijskog sustava zbog najbolje podrške i integracije, ali ROS je dostupan i na ostalim distribucijama Linux-a te na Windows i MacOS operacijskim sustavima.

ROS paketi sadrže implementacije raznih algoritama koji se primjenjuju u robotici, primjerice postupci za mapiranje, lokalizaciju ili planiranje putanje. U sklopu ROS okruženja često se koriste razni alati koji olakšavaju razvoj i primjenu robota, uključujući simulator robota Gazebo [32] i program za vizualizaciju stanja robota i podataka RViz [33].

### 5.1 Osnovni pojmovi i arhitektura

ROS je zamišljen kao skup programa, knjižnica i alata koji pokriva sve osnovne potrebe za implementaciju programske podrške robota, a jezgra sustava je generalna

---

<sup>1</sup><https://github.com/vmayoral/ros-robotics-companies>

## Poglavlje 5. ROS (Robot Operating System)

kako bi bila prilagodiva i korisna za razne primjene. Osnova ROS-a je graf čvorova - programa koji komuniciraju i surađuju razmjenom poruka koje se šalju na teme te pozivima servisa i akcija [31]. U nastavku su objašnjeni ovi pojmovi.

### 5.1.1 Čvorovi

ROS čvorovi (eng. *node*) su programi koji rješavaju specifične zadatke potrebne za rad robota, kao što su planiranje putanje i detekcija prepreka, ili nude opće usluge potrebne drugim čvorovima, primjerice pretvorba točaka iz jednog koordinatnog sustava u drugi. Ponašanje čvorova često se može konfigurirati postavljanjem vrijednosti parametara koje čvor nudi. ROS čvorovi mogu se programirati u raznim jezicima, ali najbolje su podržani C++ i Python jezici. Robotske aplikacije često zahtjevaju korištenje više čvorova. Lansirne datoteke (eng. *launch files*) omogućuju konfiguraciju i pokretanje više čvorova jednim pozivom naredbe.

### 5.1.2 Komunikacija između čvorova

Čvorovi mogu komunicirati na više načina. Pregled mogućih metoda komunikacije je na slici 5.1. Glavni način komunikacije je preko tema (eng. *topic*). Čvor na odabranoj temi (ili više njih) može objavljevati poruke ili pretplatiti se na njih, nakon čega će dobivati sve poruke koje su poslone na tu temu. Svaka tema ima definiranu strukturu poruke i tipove podataka koji se šalju na nju. Kao primjer, tema */cmd\_vel* se često koristi za upravljanje kretanjem robota, tako da se na temu pošalje poruka sa željenom linearnom i kutnom brzinom. Tip poruke za ovu temu je *geometry\_msgs/Twist*, a definicija formata je prikazana u isječku koda 5.1.

Pretplata na teme i razmjena poruka nije idealno sučelje za neke usluge koje čvorovi mogu nuditi, primjerice jednokratni zahtjevi. U tim slučajevima mogu se koristiti servisi i akcije. Servisi se koriste kao sučelje za kratkotrajne usluge koje čvor nudi. Svaki servis ima definiran tip podataka za poziv i odgovor. Za razliku od tema gdje se neograničen broj čvorova može pretplatiti i primiti poruke, čvor koji pozove servis dobije direktan odgovor.

Akcije su slične servisima, ali koriste se za dugotrajne radnje kao što je navigacija

## Poglavlje 5. ROS (Robot Operating System)

```
Vector3 linear
```

```
float64 x
```

```
float64 y
```

```
float64 z
```

```
Vector3 angular
```

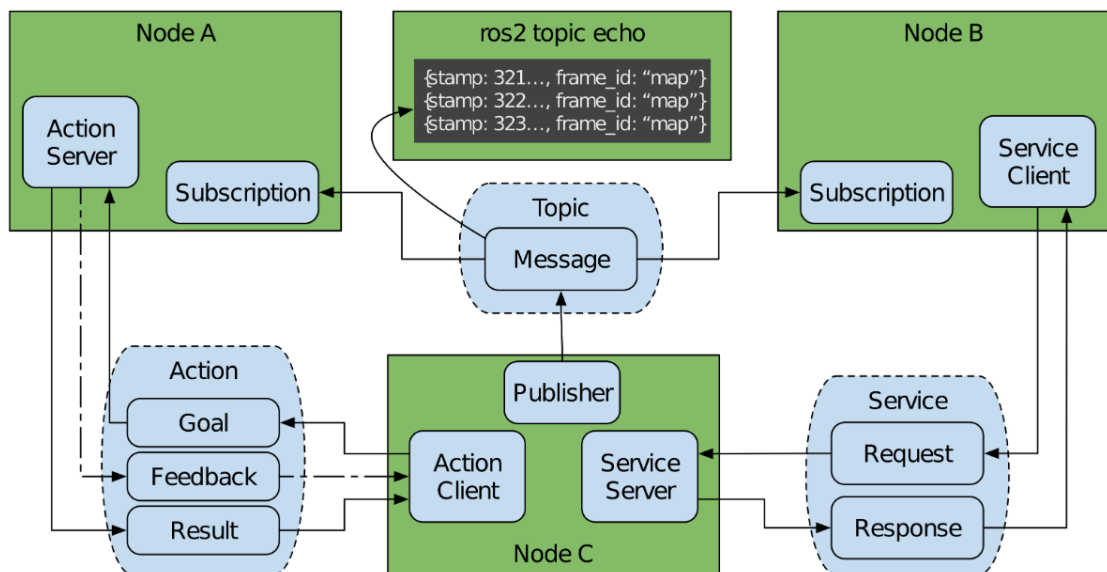
```
float64 x
```

```
float64 y
```

```
float64 z
```

Isječak koda 5.1 Primjer definicije ROS poruke (*geometry\_msgs/Twist*)

do određene točke. Pozvane akcije mogu objavama na temu slati povratne informacije čvoru koji ih je pozvao. Akcije mogu biti otkazane.



Slika 5.1 Metode komunikacije ROS čvorova - teme, servisi i akcije [34]

### 5.1.3 Okviri

Praćenje raznih koordinatnih sustava kroz vrijeme i transformacija točaka iz jednog sustava u drugi čest je zadatak tijekom razvoja programske podrške za robote. Robotske aplikacije koje koriste ROS za rad sa koordinatnim sustavima koriste *tf2* (*transform library 2*) paket i knjižnice [35]. ROS okviri (eng. *frame*) su imenovani koordinatni sustavi organizirani u obliku stabla - svaki okvir može imati samo jednog roditelja i više potomaka. Korištenje *tf2* funkcija čvorovima omogućava transformaciju točaka između dva okvira i objavljivanje transformacija između okvira.

Transformacije mogu biti statičke ili dinamičke. Primjer statičke transformacije je između okvira koji definira bazu robota (*base\_link*) i okvira u čijem je središtu GPS modul fiksiran na robotu (*gps*). Neki primjeri dinamičkih transformacija su između baze robota i centra mape (transformacija se mijenja kada robot promjeni položaj) ili transformacija između okvira zglobova na robotskoj ruci.

## 5.2 ROS 2

Razvoj ROS-a je započeo 2007. godine s ciljem nastanka platforme koja sadrži sve osnovne komponente potrebne za istraživanje i razvoj novih algoritama u robotici. Godine razvoja robota u ROS okruženju pokazale su da dizajn prve verzije nije optimalan jer ne uzima u obzir široki raspon mogućih potreba robota i uvjeta u kojima mogu raditi [34]. Neki od nedostataka prve verzije ROS-a su manjak sigurnosnih mehanizama, slaba mogućnost prilagodbe očekivane pouzdanosti komunikacije i centralizirano rješenje za koordinaciju čvorova.

U odnosu na ROS 1, ROS 2 ima bolju podršku za ugradbene sustave, sustave u stvarnom vremenu i oblikovan je s ciljem lakšeg i pouzdanijeg korištenja u industrijskim okruženjima [34], ali ima manje dostupnih paketa. Nova verzija ROS-a ima ugrađene mehanizme za sigurnost (autentikacija, enkripcija i kontrola pristupa) i mogućnost konfiguracije očekivane pouzdanosti komunikacije (*Quality of Service - QoS*). Dodana je i mogućnost upravljanjem životnim ciklusom čvorova - ponašanje čvora može se mijenjati promjenom stanja u kojem se nalazi. Algoritmi i čvorovi se razvijaju pomoću klijentskih knjižnica, koje omotavaju jezgru sustava implementi-

## Poglavlje 5. ROS (Robot Operating System)

ranu u C jeziku (*rcl* knjižnica). Službeno su podržane klijentske knjižnice za C++ i Python jezike, ali dostupne su i za još neke jezike. Za razliku od prve verzije ROS-a, koja implementira vlastiti protokol za komunikaciju, ROS 2 koristi postojeći standard DDS (*Data Distribution Service*) [36]. Sučelje dostupnih usluga za komunikaciju definira *rmw* (*ROS middleware*). Dostupno je više *rmw* implementacija i može se konfigurirati koju će ROS 2 čvorovi koristiti.

### 5.3 Alati

Postoje razni alati koji olakšavaju razvoj i korištenje ROS aplikacija:

- **Gazebo** [32] je 3D simulator koji omogućava simuliranje fizike, robota i njihovih senzora u raznim okruženjima.
- **RViz** [33] je alat koji omogućuje vizualizaciju raznih mjerenja robota, primjerice oblaka točaka, slika s kamere, modela okoline koje koristi robot (2D ili 3D karta) i ostalih poruka.
- **ros2cli** je glavni alat za pokretanje, upravljanje i nadgledanje robotskih aplikacija. Između ostalog sadrži podnaredbe za pokretanje čvorova i lansiranih datoteka, pregled i snimanje tema i poruka koje se objavljuju u stvarnom vremenu, te pozivanje servisa.
- **ros2 bag** je *ros2cli* podnaredba koja omogućuje snimanje svih poruka koje se šalju u određenom periodu i reprodukciju slanja svih poruka iz snimljene datoteke, što je korisno za testiranje čvorova. Primjer korištenja je snimanje poruka svih senzora tijekom kretanja stvarnog robota i korištenje za testiranje algoritama za lokalizaciju.
- **colcon** je alat za upravljanje ROS paketima

### 5.4 Fuzija senzora za lokalizaciju

ROS paket *robot\_localization* [1] omogućuje fuziju senzora za estimaciju stanja, odnosno položaja i orijentacije za mobilne robote. Za fuziju senzora može se koris-

## Poglavlje 5. ROS (Robot Operating System)

titi EKF ili UKF (*Unscented Kalman Filter*). Moguća je fuzija proizvoljnog broja senzora i izvora podataka. Čvorovi za estimaciju podatke sa senzora dobivaju pretplatom na teme na kojima dolaze poruke sa mjerenjima senzora. Za svaki senzor, odnosno izvor mjerenja stanja moguće je odabrati za koje varijable stanja će se koristiti mjerenja iz poruka. Kao primjer, moguće je koristiti više izvora IMU poruka, tako da se od jednog koristi samo mjerenja orijentacije, a od drugog samo akceleracije.

EKF i UKF implementacije u *ekf\_localization\_node* i *ukf\_localization\_node* čvorovima za predikciju stanja koriste sve-smjerni trodimenzionalni kinematički model model koji ima šest stupnjeva slobode [1]. Model je generalan i ne uključuje kinematička ograničenja koja mogu biti poznata, ali može se primjeniti za razne robote, uključujući i dronove. Vektor stanja ima 15 varijabli i definiran je kao:

$$\left[ X \ Y \ Z \ roll \ pitch \ yaw \ \dot{X} \ \dot{Y} \ \dot{Z} \ \ddot{roll} \ \ddot{pitch} \ \ddot{yaw} \ \ddot{X} \ \ddot{Y} \ \ddot{Z} \right] \quad (5.1)$$

Prati se 3D položaj i orijentacija, linearna brzina i akceleracija, te kutne brzine. Za matricu  $\mathbf{H}$  (koja je u formulaciji EKF-a definirana kao Jacobijeva matrica modela mjerenja  $h$ ) svakog senzora koristi se prilagođena matrica identiteta, budući da su varijable iz vektora stanja prisutne u podržanim tipovima poruka senzora. Vrijednosti na dijagonali  $\mathbf{H}$  matrice su 0 ili 1 ovisno o tome žele li se koristiti mjerenja senzora za ispravljanje procjene varijable koja je na tom položaju. Kovarijance mjerenja senzora (matrica  $\mathbf{R}$ ) uzimaju se direktno iz poruka koje dolaze sa senzora, nije ih moguće konfigurirati. Matrice inicijalne kovarijance estimacije stanja  $\mathbf{P}_0$  i šuma procesa  $\mathbf{Q}$  mogu se prilagoditi u konfiguracijskoj datoteci čvorova za estimaciju.

Više-senzorska fuzija je implementirana na način da EKF i UKF čvorovi odrade korake Kalman filtra (predikcija i korekcija) za svako novo mjerenje, neovisno o izvoru, a najnovije estimirano stanje objavljuju na */odometry/filtered* temu fiksnom frekvencijom koju zadaje korisnik. Rad filtra kreće u trenutku kada dođe prva poruka senzora nakon što se pokrene čvor.

### 5.4.1 Konfiguracija čvorova za estimaciju

Paket *robot\_localization* ima razne parametre za konfiguraciju fuzije [37]. Čvorovi *ekf\_localization\_node* i *ukf\_localization\_node* dijele skoro sve parametre. Konfigu-

## Poglavlje 5. ROS (Robot Operating System)

racija se može podijeliti na općenitu i na konfiguraciju pojedinih senzora. U nastavku je opisano nekoliko bitnijih parametara za lokalizaciju. Općenita konfiguracija između ostalog obuhvaća:

- frekvenciju objave estimiranog stanja (*frequency* parametar)
- vrijeme nakon kojeg filter odradi samo predikciju bez korekcije ako senzor prestane objavljivati poruke (*sensor\_timeout*)
- odabir 2D ili 3D lokalizacije, korisno ako je poznato da se robot kreće samo u ravnini (*two\_d\_mode*)
- *use\_control* parametar, koji omogućuje korištenje naredbi danih za kretanje robota (*/cmd\_vel* poruke) u koraku predikcije
- inicijalnu matricu kovarijanci greške procjene  $\mathbf{P}_0$
- matricu kovarijanci šuma procesa  $\mathbf{Q}$
- odabir koordinatnih okvira, opisano u nastavku

EKF i UKF čvorovi mogu se koristiti za lokalizaciju u okviru karte ili okviru za odometriju. U slučaju korištenja metoda pozicioniranja kod kojih uzastopna stanja nisu nužno blizu (izmjereni položaj za dva uzastopna mjerenja može biti udaljen), dokumentacija preporučuje lokalizaciju na karti. Ako se za fuziju koriste samo metode kod kojih se očekuje da će putanja biti kontinuirana (primjerice odometrija kotača, vizualna odometrija i IMU), preporučuje se lokalizacija u okviru za odometriju, što je u skladu sa generalnim preporukama za korištenje tih okvira [38]. Za odometriju u */odom* okviru se očekuje da je kontinuirana, odnosno da nema skokova položaja.

Okviri unutar kojih se lokalizira su zadani na standardna imena (*map* i *odom*), ali moguće ih je i prilagoditi. Rad čvorova se malo razlikuje ovisno o tome lokalizira li se robot na mapi ili ne, pa je potrebno postaviti *world\_frame* parametar kako bi čvor ispravno radio. Zadana vrijednost je *odom*.

Izvori informacija za fuziju mogu biti razne vrste ROS poruka:

- *geometry\_msgs/PoseWithCovarianceStamped*, koja sadrži položaj, orijentaciju i kovarijance mjerenja
- *geometry\_msgs/TwistWithCovarianceStamped*, koja sadrži linearne i kutne br-

## Poglavlje 5. ROS (Robot Operating System)

zine te kovarijance mjerenja

- *nav\_msgs/Odometry*, koja udružuje *Pose* i *Twist* poruke (položaj, orijentacija, brzine i kovarijance)
- *sensor\_msgs/Imu*, koja sadrži orijentaciju, kutne brzine i linearnu akceleraciju, zajedno sa kovarijancama

Mogu se koristiti i procjene položaja dobivene GNSS-om iz *sensor\_msgs/NavSatFix* poruka, uz uvjet da se koristi *navsat\_transform\_node* čvor koji prevodi poruke u *nav\_msgs/Odometry* sa položajem u koordinatnom okviru u kojem se radi procjena stanja.

Izvori informacija za fuziju se dodaju parametrima oblika *odomN*, *imuN*, i tako dalje, ovisno o vrsti poruke izvora. *N* označava broj senzora pojedinih vrsta senzora (ako se koriste dva izvora odometrije u konfiguraciji bi bili prisutni *odom0* i *odom1*). Vrijednost parametra je tema s koje dolaze poruke senzora. Za svaki senzor može se konfigurirati:

- Odabir svih varijabli iz poruka senzora koje će se koristiti za fuziju. Odabir se radi postavljanjem *true* ili *false* vrijednosti u vektoru iste dimenzionalnosti kao vektor stanja filtra. Ovaj parametar (*sensorN\_config*) je obavezan.
- maksimalan broj poruka ovog senzora koje će biti spremljene u periodu između koraka filtriranja (*sensorN\_queue\_size*)
- *sensorN\_differential* parametar, koji omogućuje korištenje senzora koji mjere apsolutni položaj ili orijentaciju za mjerenje brzine, na način da se brzina procjeni oduzimanjem dva uzastopna mjerenja
- *sensorN\_relative* parametar - ako se koristi, sva mjerenja senzora se transformiraju tako da se od njih oduzme prvo dobiveno mjerenje senzora, što osigurava da početni položaj i orijentacija budu jednaki  $(0, 0, 0)$

### 5.4.2 Fuzija GNSS poruka

Paket *robot\_localization* omogućuje fuziju položaja dobivenog GNSS-om, ali čvorovi za lokalizaciju ne podržavaju izravno korištenje *sensor\_msgs/NavSatFix* poruka.



## Poglavlje 5. ROS (Robot Operating System)

Potrebno je pokrenuti *navsat\_transform\_node* čvor, koji pretvara GNSS poruke u poruke sa odometrijom koje sadrže samo dobiveni položaj transformiran u okvir u kojemu se radi lokalizacija. Čvor počinje objavljivati poruke s odometrijom na temu */odometry/gps* kada dobije sve informacije potrebne za definirati transformaciju.

Kako bi se geografska širina i dužina u WGS84 koordinatnom sustavu dobivena GNSS-om mogla koristiti za lokalizaciju, potrebno ih je pretvoriti u položaj u *odom* ili *map* okviru. Svaki dobiveni položaj se prvo transformira iz WGS84 u UTM koordinatni sustav. Sljedeći korak je definiranje transformacije iz UTM u *map* okvir, odnosno definiranje translacije i rotacije *map* okvira u odnosu na UTM, u kojem *x* os gleda prema istoku, a *y* os prema sjeveru. Informacije potrebne za definiranje transformacije mogu se odrediti pomoću mjerenja senzora, ili odabrati ručno.

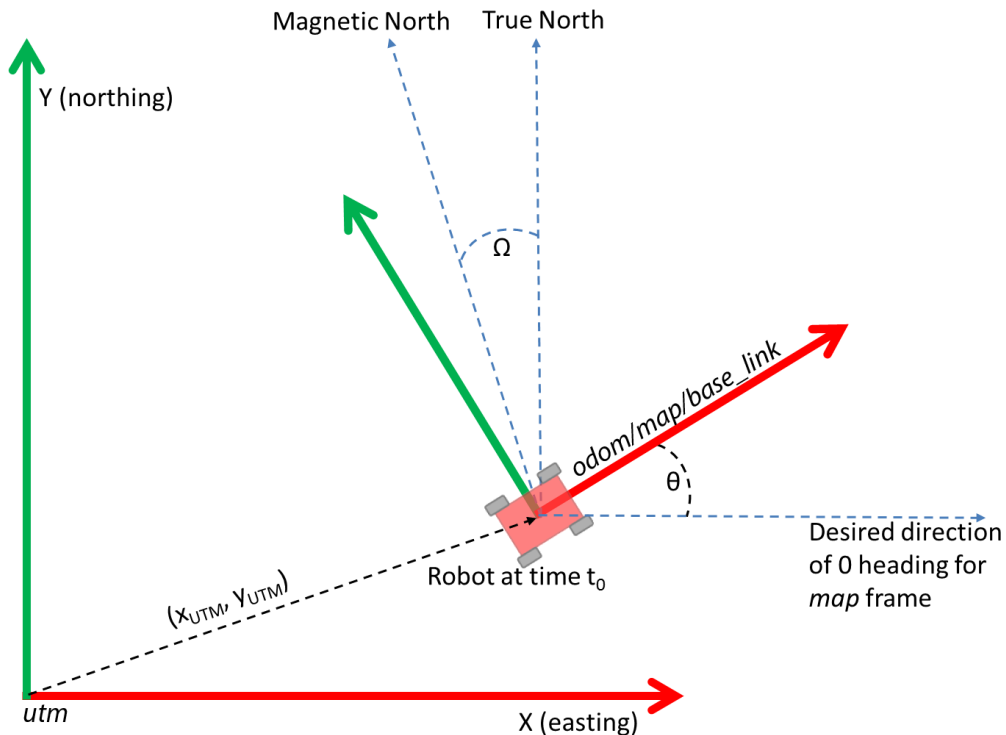
U slučaju korištenja senzora parametar parametar *wait\_for\_datum* treba biti postavljen na *false* i čvor prima poruke sa sljedećih tema:

- */gps/fix* - GNSS položaj
- */odometry/filtered* - položaj u */map* okviru
- */imu/data* - čita se apsolutna orijentacija u odnosu na Zemlju - očekuje se da vrijednost 0 označava da robot gleda prema istoku

U trenutku kada čvor dobije sve navedeno, najnovije poruke sa svake od tema se koriste za definiranje transformacije: translacija se definira uzimajući u obzir da bi položaj u zadnjoj GNSS poruci trebao imati položaj na karti jednak zadnjem dobivenom, a orijentacija dobivena iz IMU poruke se koristi za rotaciju karte u odnosu na UTM (slika 5.2). Ova transformacija se može definirati i ručno pozivom servisa *SetDatum* ili parametrom *datum* u konfiguraciji, koji sadrži geografsku širinu i duljinu ishodišta mape i zakret oko *Z* osi u odnosu na UTM koordinatni sustav. Nakon što je transformacija definirana, koristi se za svaki novi dobiveni položaj prije objave na temi s odometrijom.

### 5.4.3 Konfiguracija za lokalnu i globalnu procjenu poze

U ROS okruženju se za odometriju unutar *odom* okvira očekuje da je putanja kontinuirana (položaj nema diskretnih skokova) i da precizno opisuje kretanje robota u kraćim periodima, budući da se koristi za kratkoročno planiranje putanje i izbjegavanje prepreka. Najtočnija procjena položaja se tipično očekuje od poruka odometrije čiji je okvir *map*, gdje estimirani položaj može imati diskretne skokove. Kod korištenja GNSS poruka prilikom fuzije senzora za lokalizaciju se preporučuje korištenje dva odvojena čvora, jedan za lokalnu odometriju koja koristi samo senzore čija je procjena stanja kontinuirana, te jednog čvora za globalnu lokalizaciju, koji koristi sve dostupne senzore, uključujući i GNSS. U tom slučaju čvorovi kojima je potrebna procjena poze robota biraju lokalnu ili globalnu procjenu ovisno o svojim potrebama.



Slika 5.2 Transformacija GNSS poruka pomoću *navsat\_fix* čvora [37]. Kako bi se GNSS poruke mogle koristiti za lokalizaciju, potrebno je definirati translaciju i rotaciju *map* okvira u odnosu na UTM okvir ručno u konfiguraciji ili automatski porukama odometrije za translaciju i magnetometra za rotaciju.

## 5.5 SLAM i vizualna odometrija

RTAB-Map (*Real-Time Appearance-Based Mapping*) [30] je rješenje za vizualnu odometriju i 3D SLAM u stvarnom vremenu. Koristi se pristup zasnovan na grafovima i BoVW za vizualnu detekciju zatvaranja petlje. Karta se gradi u stvarnom vremenu. Za percepciju okoline mogu se koristiti stereo kamere, RGB-D kamere ili LIDAR. Kako bi kartiranje u stvarnom vremenu bilo moguće i za velika područja, koristi se pristup za upravljanje memorijom koji ograničava broj lokacija koje se razmatraju za moguće zatvaranje petlje i optimizaciju grafa. Moguće je korištenje IMU i GPS mjerenja za uvođenje dodatnih ograničenja u graf tijekom optimizacije. Generirana karta može se izvesti u oblicima prikladnim za 2D ili 3D navigaciju robota.

RTAB-Map je dostupan u ROS 2 okruženju u obliku *rtabmap* i *rtabmap\_ros* paketa koji sadrže više čvorova. Čvorovi za odometriju *stereo\_odometry*, *rgbd\_odometry* i *icp\_odometry* mogu se koristiti tijekom kartiranja ili samostalno. Istovremena lokalizacija i kartiranje implementirana je u *rtabmap* čvoru. Karta i putanja kretanja može se vizualizirati pomoću *rtabmapviz* čvora.

## Poglavlje 6

# Pioneer 3-AT mobilni robot i dostupni senzori

Cilj ovog rada bio je implementirati programsku podršku za lokalizaciju fuzijom senzora na mobilnom robotu sa ROS sustavom, predložiti nekoliko različitih konfiguracija senzora i parametara fuzije, te testirati točnost određivanja položaja na stvarnom robotu u vanjskom okruženju. Implementacija rješenja za fuziju obuhvaća izradu ROS 2 lansirne datoteke koja pokreće sve potrebne čvorove za obradu poruka senzora i lokalizaciju fuzijom senzora, te konfiguraciju čvorova i parametara fuzije.

Mogući pristupi za rješenje problema lokalizacije ovise o vrsti robota, dostupnim sensorima i o okruženju robota. U ovom radu rješava se problem lokalizacije u vanjskom okruženju. Za testiranje rješenja korišten je Pioneer 3-AT mobilni robot [39] (slika 6.1) sa pogonom na četiri kotača. Robot se može zakretati na mjestu rotacijom lijevih i desnih kotača u suprotnim smjerovima (eng. *skid-steer*) [40]. Robotom upravlja NVIDIA Jetson AGX Xavier ugradbeno računalo sa Ubuntu 18.04 Linux operacijskim sustavom i ROS 2 verzijom Galactic. ROS čvor RosAria upravlja kretanjem robota slanjem naredbi mikrokontrolerima na robotu. Robot ima ugrađene optičke kvadraturne enkodere kotača koji se mogu koristiti za odometriju. Uz odometriju kotača za lokalizaciju su korišteni i ZED-F9P GNSS prijemnik, Realsense D455 dubinska kamera, te IMU ugrađen u kućište kamere. Konfiguracija sklopovlja i operacijskog sustava robota nije bila dio ovog diplomskog rada.



Slika 6.1 Pioneer 3-AT mobilni robot sa korištenim sensorima

## 6.1 u-blox ZED-F9P prijemnik

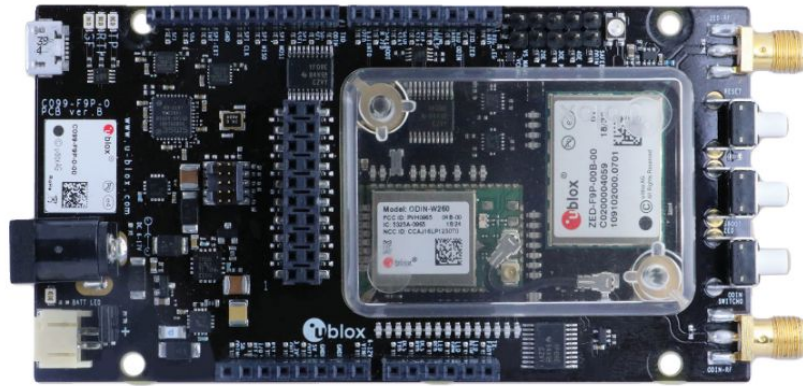
u-blox ZED-F9P [41] modul je višepojasni GNSS prijemnik koji može koristiti RTK korekcije. Moguće je istovremeno korištenje četiri GNSS sustava - GPS, GLONASS, Galileo i BeiDou. Prijemnik omogućuje dobivanje procjene položaja do 20 puta u sekundi. Na robotu je korištena C099-F9P [42] aplikacijska pločica sa ZED-F9P modulom. Čvor `nmea_serial_driver` iz `nmea_naval_driver` ROS paketa pretvara poruke prijemnika u ROS `sensor_msgs/NavSatFix` poruke. U ovom radu nisu korištene RTK korekcije za procjenu položaja. Pločica sa prijemnikom je prikazana na slici 6.2a.

## 6.2 Intel Realsense D455 kamera

Intel Realsense D455 je RGB dubinska kamera. Kamera je prikazana na slici 6.2b. Za određivanje dubine (udaljenosti) koriste se dva senzora za IR (infracrvene) slike i

## Poglavlje 6. Pioneer 3-AT mobilni robot i dostupni senzori

IR projektor uzorka točaka. Korištenje projektora je opcionalno, ali može poboljšati rezultate ako u okolini nema previše infracrvenog zračenja iz drugih izvora (primjerice Sunce). Slike kamere i procjene udaljenosti dobivene kamerom prikazane su na slici 6.3. U kameri je ugrađen Bosch BMI055 IMU, koji sadrži troosni akcelerometar i troosni žiroskop.



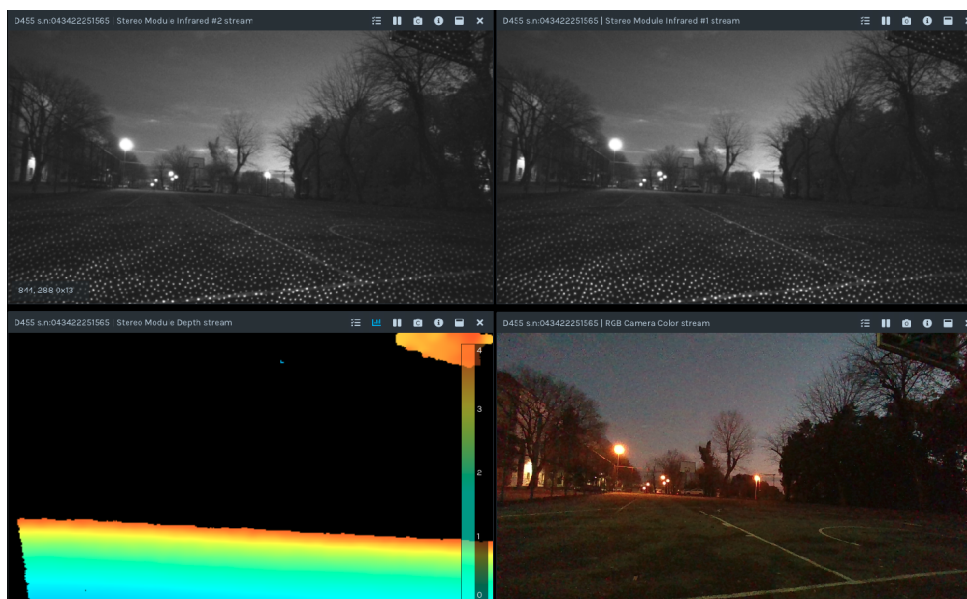
(a) u-blox C099-F9P aplikacijska pločica sa ZED-F9P GNSS prijemnikom [42]



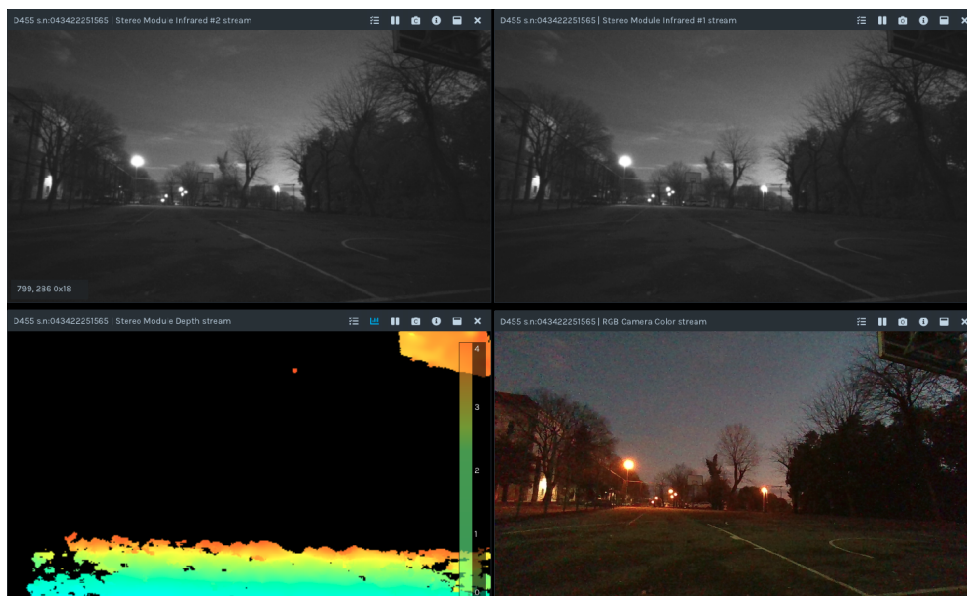
(b) Intel Realsense D455 kamera [43]

Slika 6.2 Senzori korišteni za lokalizaciju Pioneer 3-AT robota

Poglavlje 6. Pioneer 3-AT mobilni robot i dostupni senzori



(a) Procjena dubine uz korištenje IR projektor. Na slikama stereo IR kamere vidljiv je uzorak IR točaka projiciran u okolinu.



(b) Procjena dubine bez korištenja IR projektor.

Slika 6.3 Slike stereo IR kamere (slike u gornjim redovima), RGB kamere i procjena dubine (udaljenosti) između 0 i 4m (lijeva slika u donjim redovima) dobivene Intel Realsense D455 kamerom uz korištenje IR projektor (a) i bez (b).

# Poglavlje 7

## Rješenje za lokalizaciju mobilnog robota u vanjskom okruženju

U prethodnom poglavlju ukratko je opisan mobilni robot i senzori dostupni za testiranje lokalizacije. Kao rješenje za lokalizaciju fuzijom senzora implementiran je ROS 2 paket *sensor\_fusion\_localization*, koji sadrži lansirnu datoteku sa čvorovima potrebnim za fuziju senzora, konfiguracijske datoteke za čvorove i za vizualizaciju lokalizacije u RViz programu, te skriptu za testiranje točnosti procjene položaja i usporedbu raznih konfiguracija rješenja za lokalizaciju. Za usporedbu točnosti računa se greška procjene, odnosno udaljenost procijenjenog položaja od stvarnog položaja robota. Uz usporedbu izračunatih točnosti za pojedine konfiguracije skripta omogućuje i vizualnu usporedbu procijenjenih putanji robota dobivenih različitim konfiguracijama. U nastavku je opisano rješenje.

### 7.1 ROS 2 lansirna datoteka za lokalizaciju

Lansirna datoteka (*ekf\_localization.launch.py*) implementirana je u Python jeziku i omogućuje lokalizaciju sa nekoliko različitih metoda, ovisno o korištenim parametrima. Omogućeno je korištenje slika dubinske kamere Realsense D455 za vizualnu odometriju i SLAM pomoću čvorova RTAB-Map paketa. Procjena poze dobivena kamerom može se iskoristiti kao izvor informacija za fuziju senzora Kalman filtrom



## Poglavlje 7. Rješenje za lokalizaciju mobilnog robota u vanjskom okruženju

pomoću čvora *robot\_localization* paketa. Dodatni izvori informacija za filter mogu biti odometrija kotača ili GNSS.

Čvorovi koje lansirna datoteka pokreće su:

- **realsense2\_camera\_node** iz *realsense2\_camera* paketa. Ovaj čvor omogućuje korištenje poruka Realsense D455 senzora: RGB slika, dubinska slika i IMU (žiroskop i akcelerometar).
- **rgbd\_odometry** iz *rtabmap\_ros* paketa. Ovaj čvor koristi RGB i dublinske slike Realsense kamere za vizualnu odometriju i objavljuje poruke na temu */rgbd\_odom*.
- **rtabmap** iz *rtabmap\_ros* paketa. Ovaj čvor se koristi za istovremeno kartiranje i lokalizaciju (RGB-D SLAM) ili lokalizaciju na postojećoj karti. U slučaju detekcije zatvaranja petlje poza se objavljuje na temu */localization\_pose*.
- **rtabmapviz** iz *rtabmap\_ros* paketa. Ovaj čvor se koristi vizualizaciju stvorene karte i putanje procijenjene pomoću RGB-D SLAM rješenja dobivenog *rtabmap* čvorom.
- **gps\_duplicate\_checker** iz *sensor\_fusion\_localization* paketa. Ovaj čvor je implementiran kao dio ovog rada jer je tijekom testiranja uočeno da se GNSS poruke objavljene na */fix* temi u nekim slučajevima počnu ponavljati (svake sekunde dolazi ista poruka). Ponavljanje poruka dovodi do greške EKF lokalizacije. Čvor za svaku */fix* poruku provjeri ako je jednaka prethodnoj i objavljuje na */fix/checked* temu samo ako nije duplicirana.
- **navsat\_transform\_node** iz *robot\_localization* paketa. Služi za transformaciju GNSS poruka (*sensor\_msgs/NavSatFix*) u oblik koji se može iskoristiti kao izvor za EKF fuziju.
- **ekf\_node** iz *robot\_localization* paketa za lokalizaciju sa proširenim Kalman Filterom.
- **rviz2** iz *rviz2* paketa. Pokreće se sa postojećom konfiguracijom za vizualizaciju lokalizacije robota. Prikazane su poruke tema RGB-D odometrije, odometrije kotača, GNSS poruke i EKF fuzija. Prikazan je tok slika RGB kamere.

### 7.1.1 Argumenti lansirne datoteke

Lansirna datoteka ima nekoliko argumenata koji određuju koji će se čvorovi pokrenuti. Opisani su u tablici 7.1. Argumenti određuju koje će sve metode za lokalizaciju biti pokrenute (SLAM, vizualna odometrija, GNSS, EKF lokalizacija). Argument *use\_sim\_time* određuje vrijednost istoimenog parametra svih čvorova. Postavljanje na *true* omogućuje korištenje paketa za lokalizaciju simuliranih robota ili uz reproducirane *rosvbag* poruke senzora.

Tablica 7.1 Argumenti *ekf\_localization.launch.py* lansirne datoteke

Argument	Opis	Zadana vrijednost
<code>realsense</code>	Pokrenuti <code>realsense2_camera_node</code> čvor?	false
<code>rgbd_odom</code>	Pokrenuti <code>rgbd_odometry</code> čvor?	true
<code>rtabmap</code>	Pokrenuti <code>rtabmap</code> čvor?	false
<code>rtabmapviz</code>	Pokrenuti <code>rtabmapviz</code> čvor?	false
<code>rviz</code>	Pokrenuti <code>rviz2</code> čvor?	false
<code>ekf</code>	Pokrenuti <code>ekf_node</code> čvor?	true
<code>gps_checker</code>	Pokrenuti <code>gps_duplicate_checker</code> čvor?	true
<code>navsat_transform</code>	Pokrenuti <code>navsat_transform_node</code> čvor?	true
<code>config_file</code>	Put do konfiguracijske datoteke	zadana konf. dat.
<code>use_sim_time</code>	Vrijednost <code>use_sim_time</code> za čvorove	false

### 7.1.2 Konfiguracija Realsense D455 kamere

Parametri korišteni za *realsense2\_camera\_node* čvor prikazani su u tablici 7.2. Uključeni su tokovi RGB slika, dubinskih slika, te mjerenja žiroskopa i akcelerometra. Korištene su zadane FPS vrijednosti za žiroskop i akcelerometar. Uključena je sinkronizacija dubinskih i RGB slika za jednostavnije korištenje sa RTAB-Map čvorovima. Rezolucija i FPS slika (480x270x15) odabrani su jer su pokazali dovoljno dobre rezultate lokalizacije i kartiranja sa RTAB-Map čvorovima uz pravovremenu obradu slika koje dolaze (testirano na laptopu, AMD Ryzen 5 4500U, 8GB RAM). Veća rezolucija i FPS može postići bolje rezultate uz uvjet da računalo stigne pravovremeno obraditi slike koje dolaze.

Tablica 7.2 ROS parametri korišteni za *realsense2\_camera\_node* čvor

Parametar	Vrijednost
enable_depth	True
enable_color	True
enable_gyro	True
enable_accel	True
accel_fps	63
gyro_fps	200
unite_imu_method	2 (interpolacija)
rgb_camera.profile	480x270x15
depth_module.profile	480x270x15
depth_module.emitter_enabled	1
align_depth.enable	true
enable_sync	true
rgb_camera.enable_auto_exposure	true
depth_module.enable_auto_exposure	true

## 7.2 Skripta za testiranje lokalizacije

Konfiguracija lokalizacije, fuzije senzora i kovarijanci filtra može biti zahtjevna [1]. Razni faktori utječu na performanse lokalizacije. Optimalna konfiguracija senzora mijenja se ovisno o vrsti robota i pogona, radu u zatvorenom ili otvorenom prostoru, vremenskim uvjetima u kojima robot radi, vizualnoj raznolikosti okoline robota, i vrsti tla. Zbog toga je bitno evaluirati i po potrebi prilagoditi konfiguraciju za svaku novu primjenu robota ili kod promjene nekog od navedenih faktora. Kako bi bilo moguće testirati veći broj konfiguracija na što jednostavniji način, razvijena je skripta za testiranje i evaluaciju lokalizacije koja koristi snimljene poruke senzora robota.

Skripta je razvijena sa nekoliko ciljeva:

- automatizirati korake za testiranje više konfiguracija za lokalizaciju robota, tako da je dovoljno pokrenuti jednu naredbu i na kraju dobiti usporedbu rezultata svih konfiguracija: vizualno (putanje) i izračunom točnosti
- omogućiti jednostavno reproduciranje rezultata testova i ponavljanje testiranja na novim porukama senzora
- omogućiti zapis svih konfiguracija i varijanti parametara koje je potrebno testi-

rati u jednu datoteku, ali bez duplikacije parametara koji su jednaki u različitim konfiguracijama, s ciljem lakše organizacije i provedbe testiranja

- omogućiti prilagodbu metode za evaluaciju lokalizacije - kriteriji po kojima se lokalizacija optimizira mogu se razlikovati ovisno o primjeni robota

### 7.2.1 Definiranje testa za lokalizaciju u YAML datoteci

Konfiguracijske datoteke koje sadrže parametre za ROS 2 čvorove mogu se pisati u YAML jeziku. YAML datoteke sa parametrima mogu se jednostavno parsirati i prilagoditi u Python jeziku pomoću *pyaml* paket. Parsiranje konfiguracijske datoteke kao rezultat daje Python riječnik koji preslikava imena svih prisutnih parametara na njihove vrijednosti, koje mogu biti brojevi, stringovi, liste ili također riječnici. Nakon izmjene vrijednosti nekih parametara moguće je spremiti riječnik konfiguracije u novu YAML konfiguracijsku datoteku.

Skripta omogućuje konfiguraciju čvorova koje paket koristi za lokalizaciju (*ekf\_node*, *navsat\_transform\_node*, *rgbd\_odometry* i *rtabmap*). Kao argument prima YAML datoteku koja opisuje konfiguracije koje će se testirati. Opis svake konfiguracije obuhvaća odabir čvorova za lokalizaciju (argumenti za lansirnu datoteku) i parametre za čvorove. Svaka konfiguracija lokalizacije ima sljedeće YAML ključeve:

- *description* - opis kombinacije senzora, opcionalno i služi samo za dokumentaciju
- *node\_params* - svi željeni parametri čvorova za lokalizaciju
- *localization\_launch\_args* - argumenti koji se koriste kod pokretanja lansirne datoteke za lokalizaciju - služe za odabir čvorova, odnosno metode lokalizacije
- *pose\_estimate\_topic* - ROS tema koja se koristi kao procjena poze (tema koja se koristi za lokalizaciju robota)

### Testiranje različitih kombinacija vrijednosti parametara

Odabir parametara čvorova utječe na performanse lokalizacije. Neki primjeri su odabir varijabli iz poruka mjerenja senzora za EKF fuziju, ili vrijednosti matrice

kovarijance šuma procesa  $\mathbf{Q}$ . U skripti za testiranje implementirana je mogućnost testiranja više vrijednosti za odabrane parametre, tako da se u konfiguraciji umjesto vrijednosti parametra postavi YAML ključ “*try*” koji kao vrijednost ima listu svih vrijednosti koje trebaju biti testirane za taj parametar. Ako u konfiguraciji postoji više parametara koji koriste “*try*”, skripta će testirati sve moguće kombinacije svih danih vrijednosti parametara (Kartezijev umnožak po mogućim vrijednostima svakog od parametra). Primjer konfiguracije za testiranje više vrijednosti parametara je u isječku koda 7.2.

### Parametri zajednički svim konfiguracijama lokalizacije koje testiraju

Vrijednosti nekih parametara mogu biti poznati i određeni prije testiranja različitih konfiguracija lokalizacije, primjerice korišteni *tf2* okviri ili parametri za pretvorbu GNSS poruka u poruke s odometrijom. Datoteka sa parametrima zajedničkim svim konfiguracijama za lokalizaciju koje je potrebno testirati sprema se u odvojenu datoteku koja je još jedan argument skripte za testiranje.

### 7.2.2 Rad skripte za testiranje lokalizacije

Skripta za testiranje kao argument prima put do YAML datoteke sa informacijama o *rosbag*-u sa snimljenim porukama senzora robota tijekom kretanja, put do osnovne konfiguracijske datoteke zajedničke svim konfiguracijama koje će se testirati, put do datoteke koja opisuje konfiguracije za testiranje i put do direktorija gdje se spremaju rezultati. Skripta iterira po konfiguracijama za lokalizaciju koje su definirane u datoteci. Za svaku konfiguraciju generira jednu ili više datoteka sa parametrima (ovisno o tome postoje li parametri kojima je postavljeno više vrijednosti za testiranje) tako da se trenutna konfiguracija spoji sa općom (zajedničkom) konfiguracijom.

Za svaku generiranu konfiguraciju radi se testiranje lokalizacije na *rosbag*-u sa podacima senzora:

- pokreće se lansirna datoteka (*ekf\_localization.launch.py*) sa čvorovima za lokalizaciju
- pokreće se snimanje *rosbag*-a sa porukama procjene poze

## Poglavlje 7. Rješenje za lokalizaciju mobilnog robota u vanjskom okruženju

- pokreće se reprodukcija *roslabag*-a sa porukama senzora tijekom kretanja robota
- nakon reprodukcije sprema se slika sa usporedbama putanja svih konfiguracija za lokalizaciju i poziva se evaluacijska funkcija koja ocjenjuje lokalizaciju - rezultati se spremaju u .csv datoteku

### **Evaluacija lokalizacije**

Kvaliteta lokalizacije, odnosno točnost procjene položaja i orijentacije može se evaluirati vizualno usporedbom slika putanja svake testirane konfiguracije ili usporedbom ocjena evaluacijske funkcije. Za evaluaciju je potrebno znati stvarne poze robota u barem jednom odabranom trenutku. Skripta učitava stvarne poze iz datoteke sa informacijama o *roslabag*-u sa porukama senzora (korisnik ih treba upisati u datoteku). Implementirana je evaluacijska funkcija koja ispisuje grešku (apsolutnu udaljenost) procijenjene i stvarne poze (2D položaja i orijentacije) u zadnjem trenutku za koji je poza poznata, te sumu grešaka procjene u svim trenucima u kojima je poznata poza. Uz to se ispisuju i procijenjene kovarijance poze pročitane iz poruka procjene. Evaluacija je detaljnije opisana u sljedećem poglavlju.

```
wheel_odom:
  description: "EKF fused localization with wheel odometry"
  localization_launch_args: ["rgbd_odom:=false", "rtabmap:=false",
↪ "ekf:=true"]
  pose_estimate_topic: "/odometry/filtered"
  node_params:
    ekf_node:
      ros__parameters:
        odom0: /wheel/odometry
        odom0_config: [false, false, false,
                       false, false, false,
                       true, true, false,
                       false, false, true,
                       false, false, false]
        odom0_queue_size: 10
        odom0_nodelay: false
        odom0_differential: false
        odom0_relative:
          try: [true, false]
        process_noise_covariance:
          try: [
↪           [0.05, 0.05, 0.06, 0.03, 0.03, 0.06, 0.025, 0.025, 0.04, 0.01,
↪           0.01, 0.02, 0.01, 0.01, 0.015],
↪           [0.01, 0.08, 0.04, 0.05, 0.07, 0.02, 0.022, 0.021, 0.03, 0.02,
↪           0.02, 0.01, 0.02, 0.05, 0.025]
          ]
```

Isječak koda 7.2 Primjer opisa konfiguracije za lokalizaciju koja definira više mogućih vrijednosti parametara ključnom riječi *try*. Skripta će generirati četiri konfiguracijskih datoteka za testiranje, od kojih će svaka imati različitu kombinaciju vrijednosti za parametre *odom0\_relative* i *process\_noise\_covariance*.

## Poglavlje 8

# Testiranje i usporedba konfiguracija za lokalizaciju

U prethodnom poglavlju opisano je rješenje za lokalizaciju mobilnog robota u vanjskom okruženju i skripta za testiranje točnosti procjena dobivenih raznim konfiguracijama sustava za lokalizaciju. Rješenje i skripta iskoristene su za usporedbu nekoliko različitih konfiguracija lokalizacije i fuzije senzora. Za procjenu stanja robota iskoristene su poruke senzora snimljene tijekom kretanja robota. U ovom poglavlju je opisano prikupljanje podataka za testiranje lokalizacije robota, problemi koji su se pojavili tijekom snimanja, te izračun točnosti i usporedba nekoliko konfiguracija rješenja za lokalizaciju.

### 8.1 Prikupljanje poruka senzora za lokalizaciju na robotu

Kako bi bilo moguće usporediti točnost lokalizacije, snimljeni su podaci senzora na robotu Pioneer 3-AT tijekom kretanja. Snimanje je odrađeno na košarkaškom igralištu u blizini Tehničkog fakulteta u Rijeci<sup>1</sup>. Naredbe za kretanje robota dane su laptopom preko Wi-Fi veze. Za snimanje poruka senzora korišten je alat *ros2 bag*.

---

<sup>1</sup>Prikupljanje podataka senzora na robotu odrađeno je u suradnji sa kolegom Sandrom Šafarom i mentorom.



## Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

Poruke senzora snimljene su tijekom kretanja po unaprijed planiranim putanjama. Robot se kretao po dijelovima okvira igrališta kako bi bilo moguće približno odrediti stvarne poze (eng. *ground truth*) u određenim trenutcima i testirati točnost usporedbom stvarnih i procijenjenih poza. Snimljene su poruke sljedećih tema:

- `/tf_static` - statičke transformacije okvira na robotu
- `/camera/aligned_depth_to_color/image_raw` - dubinska slika Realsense D455 kamere (udaljenosti piksela)
- `/camera/color/image_raw` - RGB slika Realsense D455 kamere
- `/camera/color/camera_info` - intrinzični parametri kamere
- `/camera/imu` - IMU poruke Realsense D455 kamere (žiroskop i akcelerometar)
- `/cmd_vel` - naredbe za kretanje robota
- `/RosAria/pose` - poruke odometrije kotača

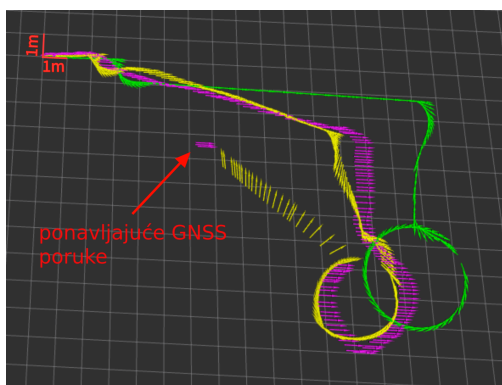
### 8.1.1 Problemi sa snimanjem i sensorima

Snimanje podataka senzora sa robotom odrađeno je u nekoliko navrata. Realsense kamera je dodana na zadnjem snimanju. Zbog problema koji su se pojavili tijekom snimanja veliki dio snimki (*rosbag*-ova) nije bilo moguće iskoristiti za testiranje točnosti. U nekoliko navrata došlo je do nasumičnog gašenja robota. Zbog toga bi snimanje završilo prije dolaska robota na poznati položaj. Osim toga, bilo je i problema sa GNSS porukama i odometrijom kotača:

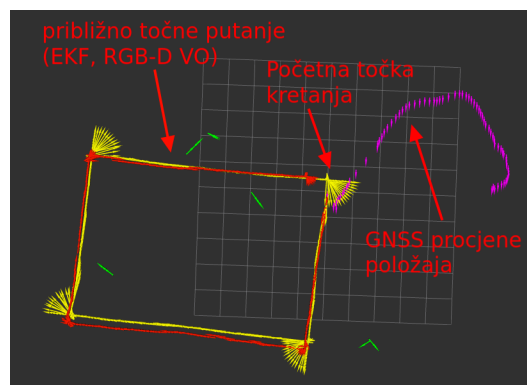
1. Tijekom nekih od pokušaja snimanja, u određenom trenutku GNSS poruke počnu se ponavljati - dolaze redovito (1 Hz), ali budu identične. Zbog toga EKF konvergira u neispravan položaj iz ponavljajućih poruka (slika 8.1a). Ovaj problem je djelomično otklonjen implementacijom čvora koji prosljeđuje GNSS poruke na ROS temu koju filter koristi samo ako se ne ponavljaju.
2. Tijekom nekih pokušaja snimanja GNSS poruke su imale status `NO_FIX` - prijemnik nije vidio dovoljno satelita da bi mogao procijeniti položaj. Zbog toga je `navsat_transform_node` čvor ignorirao poruke i nije bilo moguće koristiti GNSS kao izvor za fuziju.

## Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

- U slučajevima kada su GNSS poruke dolazile (bez greške ponavljanja), putanje su često bile jako različite od stvarnih (slika 8.1b). Mogući razlog su višestazni putevi signala zbog zgrada i stabala u blizini igrališta ili nedovoljan broj vidljivih satelita. Prilikom ovakvih grešaka su i varijance položaja prisutne u GNSS porukama bile veće (između  $50m^2$  i  $60m^2$ ) u odnosu na snimke sa boljom GNSS procjenom putanje (varijance između  $0.75m^2$  i  $10m^2$ ).
- Greške sa snimanjem poruka senzora. U nekim slučajevima je bilo problema sa Wi-Fi vezom između laptopa i robota. Ako je naredba za snimanje poruka (*ros2 bag record*) pokrenuta na laptopu, svaka poruka senzora se preke Wi-Fi veze šalje na laptop. Zbog pucanja veze je dio poruka bio odbačen pa je bilo razdoblja u kojima nije bilo dovoljno poruka za lokalizaciju. Problem nije u potpunosti nestao prebacivanjem snimanja poruka na robota - poruke odometrije kotača nisu dolazile redovito i bilo ih je manje nego što je očekivano. Mogući razlog za to je da računalo nije stiglo obraditi i pohraniti sve poruke na vrijeme jer se snima veliki broj poruka.



(a) Neispravna EKF konvergencija uzrokovana ponavljajućim GNSS porukama



(b) Neispravna putanja dobivena GNSS procjenom položaja

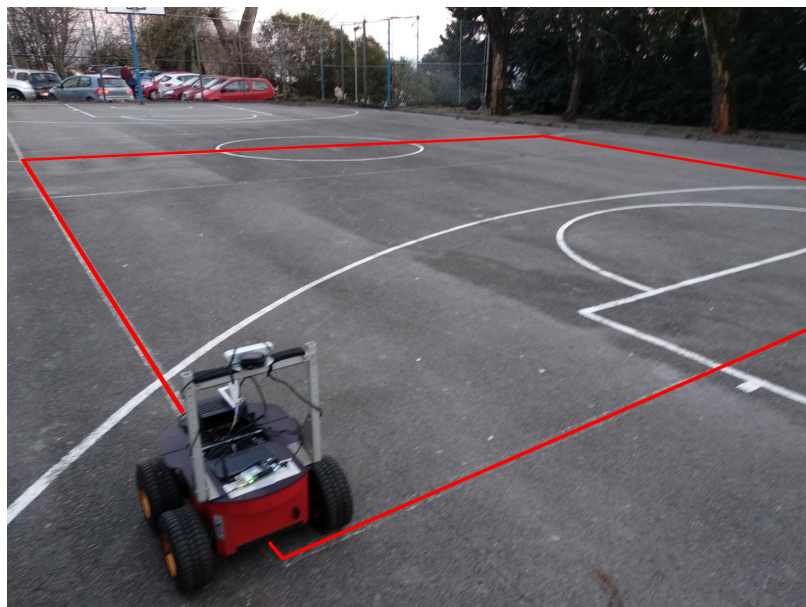
Slika 8.1 Primjeri grešaka GNSS procjene položaja. Prikazane su poruke odometrije kotača (zeleno), GNSS (ljubičasto), RGB-D odometrija (crveno) i EKF fuzija (žuto). Dimenzije ćelija ucrtanog koordinatnog okvira su  $1m \times 1m$ .

Zbog navedenih grešaka veliki dio snimljenih *rosbag*-ova nije bilo moguće iskoristiti za testiranje točnosti lokalizacije pomoću GNSS-a i odometrije kotača. Vizualizacijom putanje odometrije kotača vidljivo je da robot precijenjuje prijedene

udaljenosti i kuteve kod skretanja. Taj problem bi se mogao riješiti umjeravanjem parametara za odometriju. Testiranje točnosti bilo je moguće za konfiguracije koje koriste poruke Realsense kamere i IMU. Uspoređene su u nastavku ovog poglavlja.

## 8.2 Metodologija testiranja točnosti lokalizacije

Konfiguracije rješenja za lokalizaciju testirane su na *rosvbag*-u snimljenom tijekom kretanja robota po okviru ucrtanom na igralištu. Robot je na početku u jednom od kuteva okvira i kreće se po crti dok ne dođe opet na početni položaj (jednom prođe po cijelom okviru). Putanja je prikazana na slici 8.2. Izmjerene su dimenzije okvira po kojem se robot kretao ( $9.25\text{m} \times 9.25\text{m}$ ). Kutevi okvira iskorišteni su kao referentne točke u kojima je poznat položaj  $(x, y)$  i orijentacija  $(\phi)$  robota. Trenutci u kojima je robot bio u referentnim pozama određeni su pregledom snimke RGB kamere na robotu. Referentne poze zapisane su u tablici 8.1.



Slika 8.2 Prikupljanje podataka za testiranje lokalizacije - poruke senzora snimljene su tijekom kretanja robota po okviru igrališta ( $9.25\text{m} \times 9.25\text{m}$ ).

Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

Tablica 8.1 Referentne poze robota u određenim trenutcima snimanja poruka senzora tijekom kretanja po okviru na igralištu

ID	Vrijeme(s)	x(m)	y(m)	$\phi$ (rad)
0	0	0	0	0
1	60	9.25	0	-1.57
2	90	9.25	-9.25	-3.14
3	121	0	-9.25	-4.71
4	149	0	0	0

Točnost procjena dobivenih testiranim konfiguracijama rješenja za lokalizaciju opisana je sa nekoliko metrika. Za položaj se računa greška (Euklidska udaljenost) procijenjenog položaja  $(\hat{x}, \hat{y})$  i stvarnog položaja  $(x, y)$  u trenutku povratka u početni položaj (zatvaranje petlje)  $err_{p_4}$  i suma grešaka u svim referentnim trenutcima  $\sum err_p$ :

$$err_{p_i} = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (8.1)$$

$$\sum err_p = \sum_{i=0}^4 err_{p_i} \quad (8.2)$$

Za orijentaciju se računa greška procjene (apsolutna razlika) u trenutku povratka u početni položaj  $err_{\phi_4}$  i suma svih grešaka  $\sum err_{\phi}$ :

$$err_{\phi_i} = |\phi_i - \hat{\phi}_i| \quad (8.3)$$

$$\sum err_{\phi} = \sum_{i=0}^4 err_{\phi_i} \quad (8.4)$$

Uz opisane greške su za svaku konfiguraciju navedene i varijance procjene stanja na kraju kretanja po okviru (trenutak zatvaranja petlje)  $\sigma_{x_4}^2$ ,  $\sigma_{y_4}^2$  i  $\sigma_{\phi_4}^2$ . Varijance su pročitane iz poruka procjene stanja i mogu dati uvid u stanje procjene kod korištenja Kalman filtra.

## 8.3 Testiranje konfiguracija RGB-D vizualne odometrije

RTAB-Map čvorovi za vizualnu odometriju (*rgbd\_odometry*) i SLAM (*rtabmap*) imaju razne parametare za konfiguraciju lokalizacije<sup>2</sup>. Prije testiranja točnosti za konfiguracije EKF fuzije senzora za lokalizaciju testirano je nekoliko konfiguracija RTAB-Map vizualne odometrije.

### 8.3.1 Prilagodba snopova (Bundle adjustment)

Sa zadanom konfiguracijom *rgbd\_odometry* čvor koristi F2M (*Frame-to-Map*) pristup za registraciju okvira (pomak se određuje registracijom trenutnog okvira na lokalnoj karti stvorenog od nekoliko zadnjih okvira) i postupak prilagodbe snopova (eng. *Bundle Adjustment*) [44] za lokalnu optimizaciju putanje odometrije [30]. Testirano je kako korištenje prilagodbe snopova utječe na točnost procjene poze. Konfiguracije i rezultati su prikazani u tablici 8.3.

Tablica 8.2 Rezultati testiranja RGB-D odometrije sa i bez prilagodbe snopova (eng. *Bundle adjustment*)

(a) Testirane konfiguracije, definirane u datoteci *rgbd\_odom\_ba.yaml*

Konfiguracija	OdomF2M/BundleAdjustment
<code>rgbdo_ba_0</code>	0 (ne koristi)
<code>rgbdo_ba_1</code>	1 (koristi)

(b) Rezultati evaluacije

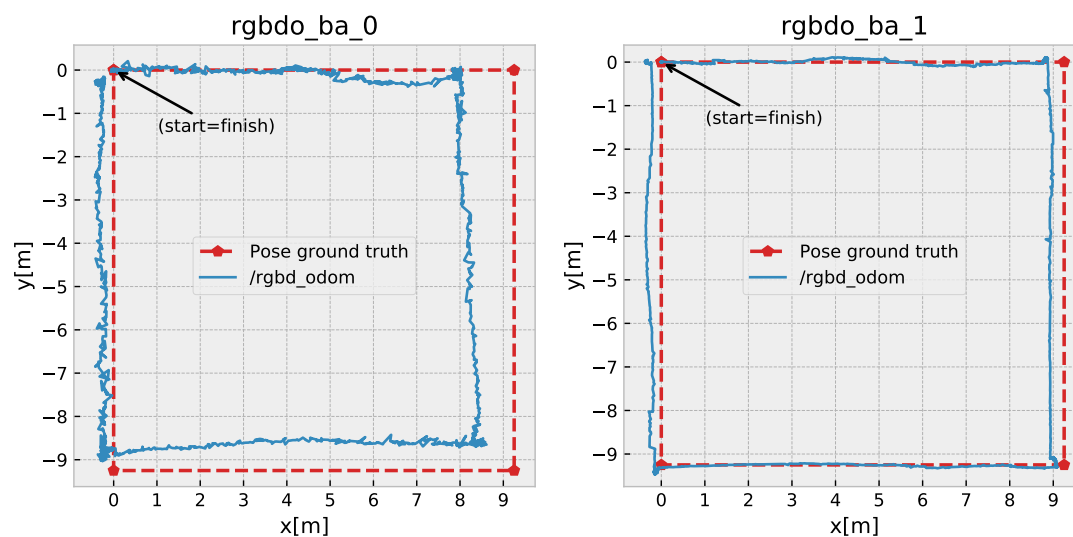
Konfiguracija	$err_{p_4}$ (m)	$\sum err_p$	$\hat{\sigma}_{x_4}^2$	$\hat{\sigma}_{y_4}^2$	$err_{\phi_4}$ (rad)	$\sum err_{\phi}$	$\hat{\sigma}_{\phi_4}^2$
<code>rgbdo_ba_0</code>	0.383	3.197	0.043	0.043	<b>0.102</b>	<b>0.191</b>	0.013
<code>rgbdo_ba_1</code>	<b>0.352</b>	<b>1.267</b>	0.026	0.026	0.122	0.198	0.016

Po metrikama korištenim za evaluaciju prilagodba snopova ima manju grešku položaja, ali veću grešku orijentacije. Putanje odometrije prikazane su na slici 8.3.

<sup>2</sup>[https://docs.ros.org/en/kinetic/api/rtabmap/html/classrtabmap\\_1\\_1Parameters.html](https://docs.ros.org/en/kinetic/api/rtabmap/html/classrtabmap_1_1Parameters.html)

## Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

Iz slike je vidljivo da je putanja odometrije puno bliža stvarnoj putanji u slučaju kada se koristi prilagodba snopova (konfiguracija *rgbdo\_ba\_1*). Zbog toga je prilagodba snopova korištena u svim konfiguracijama u nastavku testiranja.



Slika 8.3 Putanje vizualne odometrije (*/rgbdo\_odom*) uz korištenje prilagodbe snopova (desno) i bez korištenja (lijevo). Referentna putanja prikazana je crvenom bojom (*Pose ground truth*).

### 8.3.2 Metoda registracije okvira i korištenje neholonomskih ograničenja

Registracija okvira, odnosno podudaranje značajki trenutnog okvira sa sličnim značajkama na prethodnim okvirima (slikama kamere) omogućuje procjenu pomaka kamere. Čvor *rgbdo\_odometry* nudi odabir između dva standardna pristupa za registraciju okvira:

- **Frame-To-Map (F2M)**: značajke detektirane na trenutnom okviru podudaraju se sa značajkama lokalne karte stvorene od nekoliko prethodnih okvira [30]

Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

- **Frame-To-Frame (F2F)**: značajke detektirane na trenutnom okviru podudaraju se sa značajkama prethodnog okvira [30]

Pristup rješavanja registracije okvira može se odabrati postavljanjem parametra *Odom/Strategy*.

Roboti mogu biti holonomski (uvijek se mogu kretati u bilo kojem smjeru) ili neholonomski (smjer kretanja im je ograničen) [4]. U ovom radu je korišten neholonomski mobilni robot Pioneer 3-AT. Robot se može kretati po  $x$  osi (lokalnog koordinatnog sustava) i zakretati (mijenjati orijentaciju oko  $z$  osi). Čvor *rgbd\_odometry* može kod procjene kretanja uzimati u obzir neholomska ograničenja robota postavljanjem parametra *Odom/Holonomic* na vrijednost *false*. Testirane su četiri kombinacije parametara *Odom/Strategy* i *Odom/Holonomic*. Konfiguracije i rezultati testiranja su prikazani u tablici 8.3. Putanje su prikazane na slici 8.4.

Tablica 8.3 Rezultati testiranja konfiguracija RGB-D odometrije (različite vrijednosti *Odom/Strategy* i *Odom/Holonomic* parametara)

(a) Testirane konfiguracije, definirane u datoteci *rgbd\_odom.yaml*

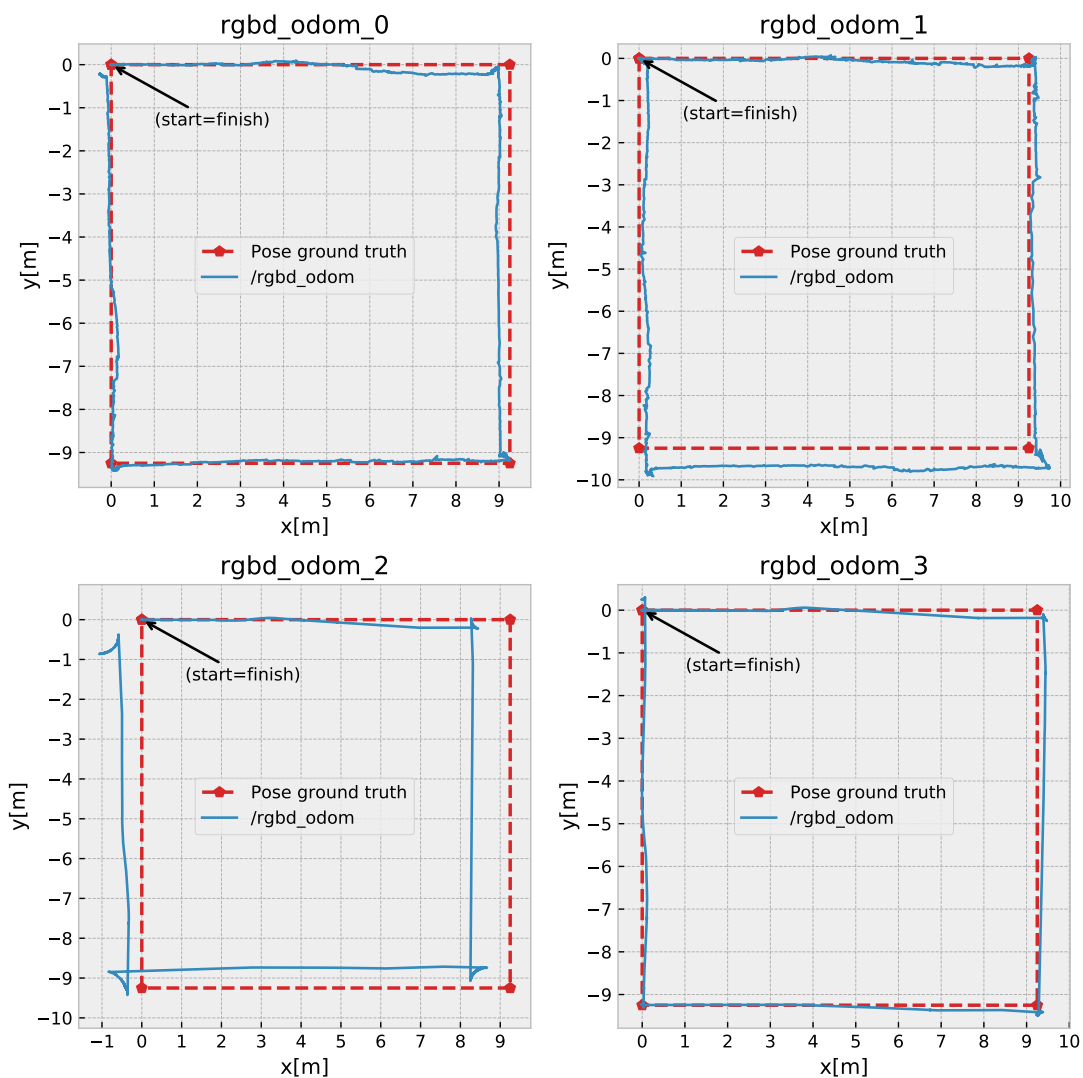
Konfiguracija	Odom/Strategy	Odom/Holonomic
<i>rgbd_odom_0</i>	0 (F2M)	true
<i>rgbd_odom_1</i>	1 (F2F)	true
<i>rgbd_odom_2</i>	0 (F2M)	false
<i>rgbd_odom_3</i>	1 (F2F)	false

(b) Rezultati evaluacije

Konfiguracija	$err_{p_4}$ (m)	$\sum err_p$	$\hat{\sigma}_{x_4}^2$	$\hat{\sigma}_{y_4}^2$	$err_{\phi_4}$ (rad)	$\sum err_\phi$	$\hat{\sigma}_{\phi_4}^2$
<i>rgbd_odom_0</i>	0.340	0.852	0.020	0.020	0.135	0.239	0.019
<i>rgbd_odom_1</i>	<b>0.082</b>	1.440	0.027	0.027	0.104	0.198	0.018
<i>rgbd_odom_2</i>	1.327	3.649	7.695	7.695	0.140	0.220	0.018
<i>rgbd_odom_3</i>	0.250	<b>0.657</b>	0.013	0.013	<b>0.100</b>	<b>0.173</b>	0.012

Greška položaja kod zatvaranja petlje je najmanja za konfiguraciju *rgbd\_odom\_1* (F2F i holonomski model kretanja), ali konfiguracija *rgbd\_odom\_3* (F2F i neholonomski model kretanja) ima najmanju ukupnu grešku položaja i manju grešku orijentacije. Pregledom putanja na slici vidljivo da je putanje sa konfiguracijom

Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju



Slika 8.4 Putanje vizualne odometrije (*/rgbd\_odom*) za različite konfiguracije pristupa registracije okvira (F2M/F2F) i ograničenja modela kretanja. Referentna putanja prikazana je crvenom bojom (*Pose ground truth*). Konfiguracije su opisane u tablici 8.3.

*rgbd\_odom\_3* najbliža stvarnoj putanji robota. Ta konfiguracija je korištena u nastavku testiranja kao izvor za EKF fuziju. Pokretanjem ovog testa više puta uočeno je



da putanje dobivene *rgbd\_odometry* čvorom nisu uvijek iste<sup>3</sup>, ali poredak konfiguracija po točnosti generalno bude isti. Za EKF testove u nastavu su snimljene poruke odometrije sa *rgbd\_odom\_3* konfiguracijom i dodane u *rosbag* sa porukama senzora s ciljem poštene usporedbe EKF konfiguracija (jednaka točnost izvora vizualne odometrije za sve EKF konfiguracije).

## 8.4 Testiranje konfiguracija EKF fuzije senzora

Testirano je nekoliko konfiguracija *ekf\_node* čvora za fuziju senzora. Zbog problema prisutnih u snimljenim *rosbag*-ovima nije bilo moguće u fuziju uključiti mjerenja GNSS-a i odometrije kotača.

### 8.4.1 Filtriranje vizualne odometrije - utjecaj odabira varijabli iz poruka senzora

Za početak su uspoređeni rezultati same vizualne odometrije (*rgbd\_odom\_3* konfiguracija) i EKF fuzije koja od senzora koristi samo poruke vizualne odometrije. Za matrice kovarijance greške procjene  $\mathbf{P}_0$  i šuma procesa  $\mathbf{Q}$  korištene su inicijalno zadane konfiguracije definirane u *ekf\_node* čvoru koje imaju vrijednosti samo na dijagonali. Vrijednosti su prikazane u tablici 8.4.

Tablica 8.4 Varijance varijabli u matricama Kalman filtra  $\mathbf{Q}$  i  $\mathbf{P}_0$ .

Matrica	$x$	$y$	$\phi$	$\dot{x}$	$\dot{y}$	$\dot{\phi}$	$\ddot{x}$	$\ddot{y}$
$\mathbf{Q}$	0.05	0.05	0.06	0.025	0.025	0.02	0.01	0.01
$\mathbf{P}_0$	1e-9	1e-9	1e-9	1e-9	1e-9	1e-9	1e-9	1e-9

Testirane su tri konfiguracije za odabir varijabli koje filter uzima iz poruka vizualne odometrije. Konfiguracije i rezultati su opisane u tablici 8.5. Putanje su prikazane na slici 8.5.

<sup>3</sup><http://official-rtab-map-forum.206.s1.nabble.com/Repeatability-over-same-data-td9130.html#a9201>

Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

Tablica 8.5 Usporedba rezultata vizualne odometrije i EKF filtrirane vizualne odometrije

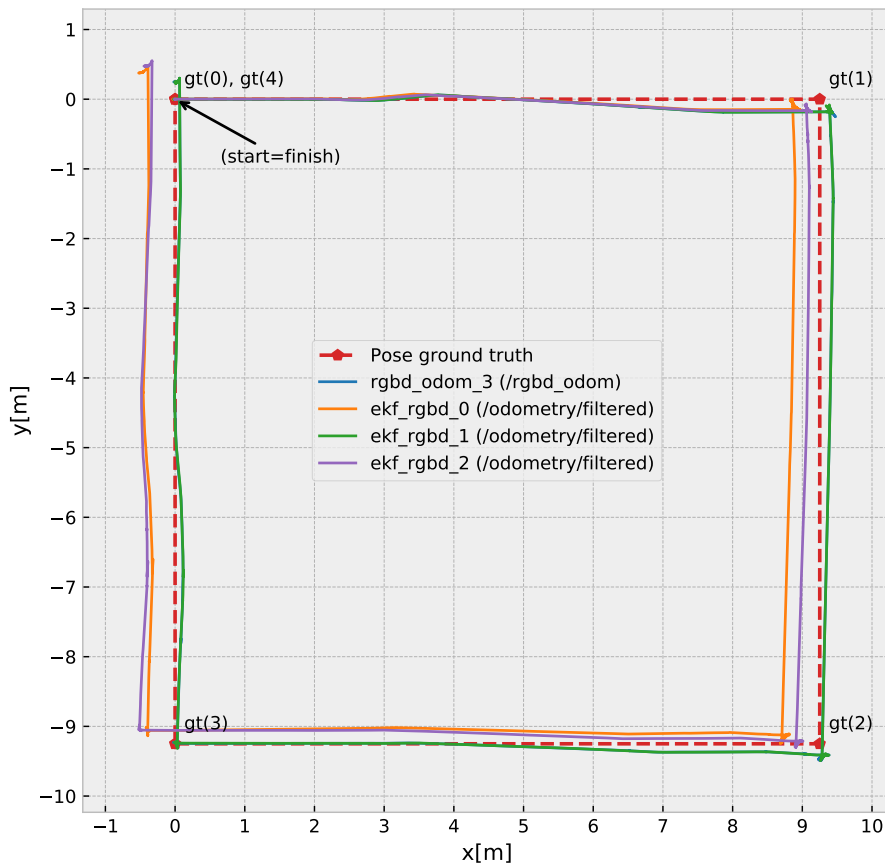
(a) Testirane konfiguracije, definirane u datoteci *ekf\_rgbd.yaml*

Konfiguracija	Varijable poruka korištene za EKF
rgb_d_odom_0	-
ekf_rgb_d_0	samo linearne i kutne brzine
ekf_rgb_d_1	položaj i orijentacija, linearne i kutne brzine
ekf_rgb_d_2	linearne i kutne brzine dobivene oduzimanjem uzastopnih poruka poze

(b) Rezultati evaluacije

Konfiguracija	$err_{p_4}$ (m)	$\sum err_p$	$\hat{\sigma}_{x_4}^2$	$\hat{\sigma}_{y_4}^2$	$err_{\phi_4}$ (rad)	$\sum err_\phi$	$\hat{\sigma}_{\phi_4}^2$
rgb_d_odom_3	0.249	<b>0.651</b>	0.021	0.021	0.099	0.174	0.011
ekf_rgb_d_0	0.638	1.993	56.688	52.254	0.100	0.162	1.346
ekf_rgb_d_1	<b>0.248</b>	0.658	0.001	0.001	0.094	0.171	0.001
ekf_rgb_d_2	0.646	1.786	58.035	52.292	<b>0.081</b>	<b>0.160</b>	1.339

U slučaju kada EKF direktno koristi poze iz poruka vizualne odometrije putanja je jednaka putanji vizualne odometrije (*ekf\_rgb\_d\_1*). To znači da su vrijednosti varijance u porukama dovoljno niske da EKF u potpunosti vjeruje tim mjerenjima. Korištenje EKF-a nije poboljšalo rezultate jer je većina šuma odometrije uklonjena korištenjem lokalne optimizacije putanje prilagodbom snopova i korištenjem nehologomskih ograničenja. Generalno se ne preporučuje direktno korištenje poza iz poruka odometrije jer nije moguće izbjeći greške poze odometrije koje s vremenom postaju sve veće. Vidljivo je da su varijance procjene stanja poze u trenutku zatvaranja petlje puno veće uz korištenje brzina iz poruka odometrije u odnosu na slučaj gdje EKF direktno koristi mjerenja poze iz poruka. Konfiguracije koje koriste brzine iz poruka bolje modeliraju neizvjesnost koja je neizbježni rezultat dugoročnog korištenja odometrije. To olakšava poboljšanje procjene u slučaju da EKF dobije poruku koja senzora koji direktno određuje položaj, ali ne dolazi često (primjerice GNSS) - korak korekcije EKF-a procjenu stanja približi mjerenju položaja proporcionalno omjeru varijance dobivenog mjerenja i varijance trenutne procjene.



Slika 8.5 Putanje vizualne odometrije bez i sa EKF filtriranjem uz različite konfiguracije odabira varijabli za filtriranje. Referentna putanja prikazana je crvenom bojom (*Pose ground truth*). Konfiguracije su opisane u tablici 8.5. Putanje *rgbd\_odom\_3* i *ekf\_rgbd\_1* su jednake.

### 8.4.2 EKF fuzija naredbi kretanja, vizualne odometrije i IMU senzora

Testirana je lokalizacija pomoću EKF fuzije naredbi kretanja robota (*/cmd\_vel* tema) i poruka IMU senzora te tri konfiguracije fuzije naredbi kretanja, vizualne odometrije i IMU poruka. Konfiguracije se razlikuju po varijablama koje se uzimaju iz IMU poruka (samo akceleracija, samo kutne brzine ili oboje). Varijable za fuziju vizualne odometrije odabrane su jednako kao u konfiguraciji *ekf\_rgbd\_2*. Naredbe

Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

kretanja koriste se za procjenu akceleracije u koraku predikcije EKF-a. Korištenje naredbi kretanja uključeno je postavljanjem *use\_control* parametra na vrijednost *true*. Konfiguracije i rezultati prikazani su u tablici 8.6. Putanje su prikazane na slici 8.6.

Tablica 8.6 Usporedba rezultata EKF fuzije vizualne odometrije, naredbi kretanja i IMU poruka

(a) Testirane konfiguracije, definirane u datoteci *ekf\_vo\_imu.yaml*

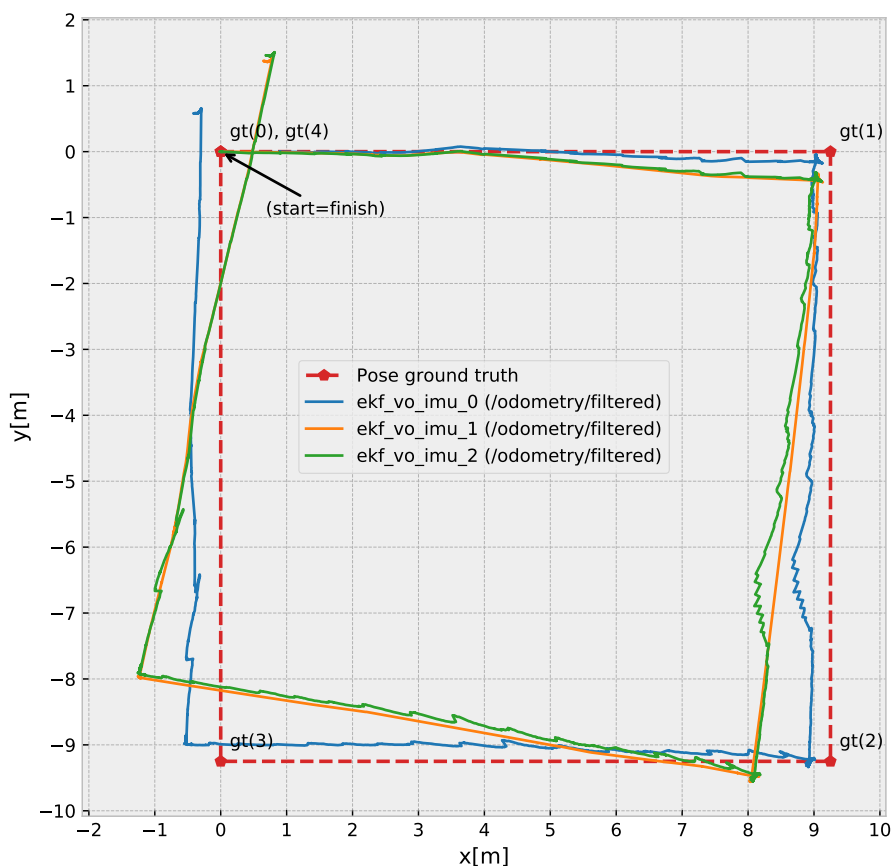
Konfiguracija	Izvori za fuziju
<i>ekf_imu_0</i>	naredbe kretanja, IMU akceleracija i kutne brzine
<i>ekf_vo_imu_0</i>	vizualna odometrija, naredbe kretanja, IMU akceleracija
<i>ekf_vo_imu_1</i>	vizualna odometrija, naredbe kretanja, IMU kutne brzine
<i>ekf_vo_imu_2</i>	vizualna odometrija, naredbe kretanja, IMU akceleracija i kutne brzine

(b) Rezultati evaluacije

Konfiguracija	$err_{p_4}$ (m)	$\sum err_p$	$\hat{\sigma}_{x_4}^2$	$\hat{\sigma}_{y_4}^2$	$err_{\phi_4}$ (rad)	$\sum err_\phi$	$\hat{\sigma}_{\phi_4}^2$
<i>ekf_imu_0</i>	1083.7	2279	1.96e8	1.41e9	0.247	0.782	2849
<i>ekf_vo_imu_0</i>	<b>0.704</b>	<b>1.943</b>	174.33	123.14	<b>0.078</b>	<b>0.163</b>	1.663
<i>ekf_vo_imu_1</i>	1.526	4.969	60.350	32.830	0.155	0.526	1.268
<i>ekf_vo_imu_2</i>	1.618	5.110	196.79	78.054	0.155	0.526	1.639

Konfiguracija *ekf\_imu\_0* ima jako veliku grešku (1083.7m u trenutku zatvaranja petlje). Korištenje IMU poruka uvodi grešku i kod fuzije sa vizualnom odometrijom. Pregledom IMU poruka vidljivo je da je akceleracija na *y* osi uvijek ima vrijednosti između -9 i -10. U dokumentaciji *robot\_localization* paketa piše da je za uklanjanje akceleracije uzrokovane gravitacijom iz IMU poruka potrebno da IMU poruke sadrže informacije o apsolutnoj orijentaciji, što nije dostupno u IMU porukama Realsense kamere. Također, u *rosbag*-u sa porukama na */tf\_static* temi nije snimljena transformacija između okvira *base\_link* (središte robota) i okvira *camera\_imu\_optical\_frame* koji je naveden u IMU porukama. Moguće je da su to uzroci problema sa fuzijom.

Fuzija vizualne odometrije, naredbi kretanja i IMU poruka postiže lošije rezultate ( $\sum err_p$  između 1.943m i 5.110m) od same vizualne odometrije ( $\sum err_p=0.651m$ ) i EKF filtrirane vizualne odometrije ( $\sum err_p=1.786m$ ). Konfiguracije koje koriste



Slika 8.6 Putanje konfiguracija EKF fuzije vizualne odometrije, naredbi kretanja i IMU poruka. Referentna putanja prikazana je crvenom bojom (*Pose ground truth*). Konfiguracije su opisane u tablici 8.6.

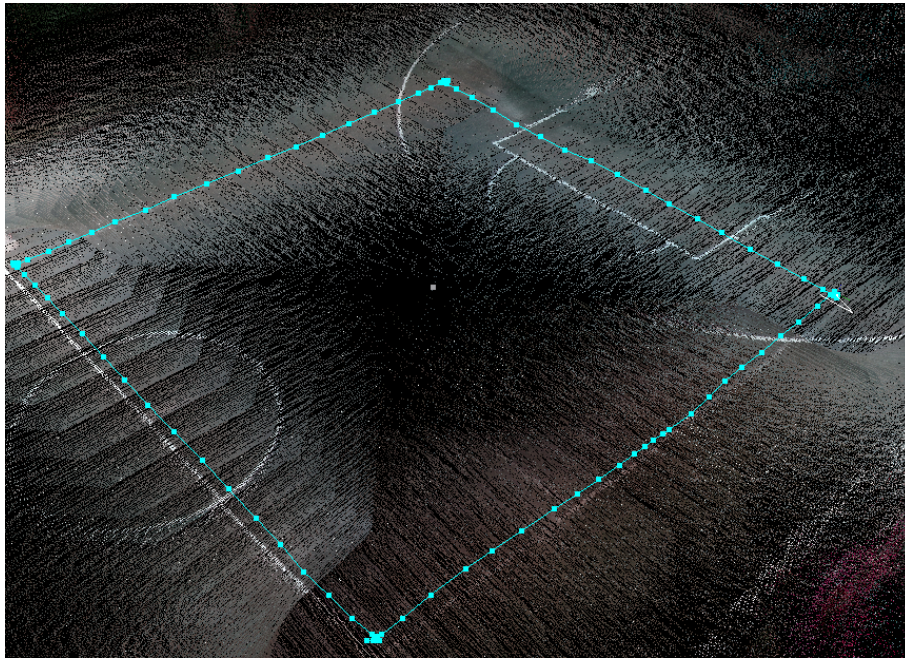
mjerenja žiroskopa (kutne brzine) na svakom skretanju procijene veći kut od stvarnog ( $90^\circ$ ). Bolji rezultati bi se mogli postići umjeravanjem IMU senzora.

### 8.4.3 Testiranje RTAB-Map SLAM rješenja

Testirana je RTAB-Map SLAM implementacija iz čvora *rtabmap*. Karta i putanja su vizualizirane čvorom *rtabmapviz*. Dobivena je 3D karta igrališta i procijena putanje robota koja je jako slična stvarnoj putanji (slika 8.7). Kod povratka robota u početnu točku *rtabmap* detektira zatvaranje petlje - trenutna procjena stanja, koja

## Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

je prije detekcije bila u okolini početne točke, skače u početnu točku. Nakon detekcije i uvođenja ograničenja optimizira se 3D karta i ostatak procijenjene putanje iz prošlosti.



Slika 8.7 3D karta igrališta i procijenjena putanja robota dobivene RTAB-Map SLAM rješenjem.

### **EKF fuzija vizualne odometrije i RTAB-Map detekcije zatvaranja petlje**

Čvor *rtabmap* u trenutku detekcije zatvaranja petlje procijenjenu pozu objavljuje na temu */localization\_pose*. Tema se može iskoristiti kao izvor za EKF. Uspoređena je točnost EKF filtrirane vizualne odometrije i EKF fuzije vizualne odometrije i detekcije zatvaranja petlje. Rezultati su prikazani u tablici 8.7. Putanje su prikazane na slici 8.8.

Poruke detekcije zatvaranja petlje imaju jako niske kovarijance procjene. Iz rezultata se može vidjeti da konfiguracija *ekf\_rtabmap* iskorištava detekciju zatvaranja petlje na kraju putanje - greška položaja je smanjena u odnosu na EKF koji koristi

Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

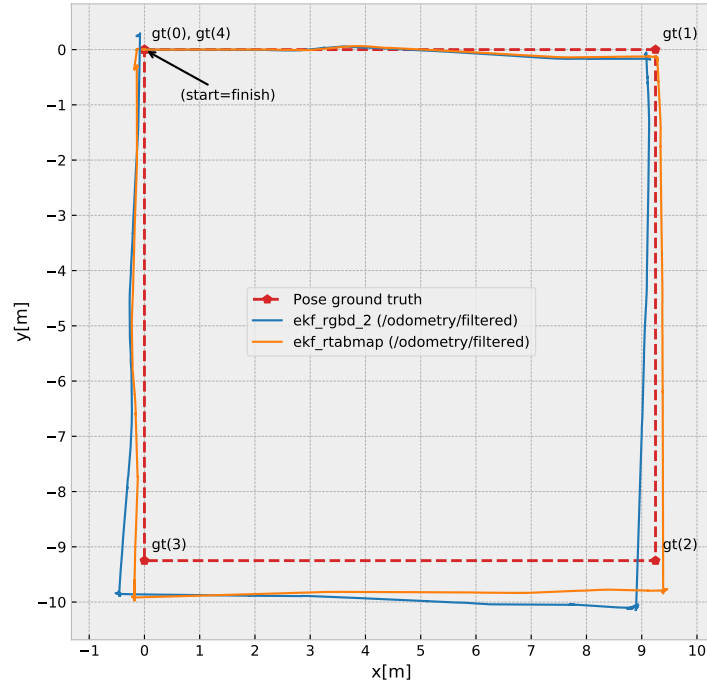
Tablica 8.7 Usporedba rezultata EKF filtrirane vizualne odometrije i EKF fuzije vizualne odometrije sa RTAB-Map detekcijom zatvaranja petlje

(a) Testirane konfiguracije, definirane u datoteci *ekf\_rtabmap.yaml*

Konfiguracija	Izvori za EKF fuziju
ekf_rgbd_2	vizualna odometrija
ekf_rtabmap	vizualna odometrija, RTAB-Map detekcije zatvaranja petlje

(b) Rezultati evaluacije

Konfiguracija	$err_{p_4}$ (m)	$\sum err_p$	$\hat{\sigma}_{x_4}^2$	$\hat{\sigma}_{y_4}^2$	$err_{\phi_4}$ (rad)	$\sum err_{\phi}$	$\hat{\sigma}_{\phi_4}^2$
ekf_rgbd_2	0.269	2.242	65.307	53.183	<b>0.068</b>	<b>0.179</b>	1.421
ekf_rtabmap	<b>0.128</b>	<b>1.524</b>	0.010	0.010	0.087	0.201	0.006



Slika 8.8 Putanje konfiguracija EKF fuzije vizualne odometrije, naredbi kretanja i IMU poruka. Referentna putanja prikazana je crvenom bojom (*Pose ground truth*). Konfiguracije su opisane u tablici 8.7.

samo vizualnu odometriju i kovarijance procjene stanja su blizu nuli. Slično kao i u prethodnim testovima, korištenje EKF-a nije poboljšalo rezultate u odnosu na samostalno korištenje vizualne odometrije i SLAM rješenja.

## 8.5 Diskusija

Poruke ROS tema dubinske kamere, GNSS prijemnika, IMU senzora i odometrije kotača snimljene su tijekom kretanja robota po planiranim putanjama s ciljem testiranja točnosti raznih konfiguracija lokalizacije u vanjskom okruženju. Pregledom nekih od snimljenih putanji uočeno je da su prijeđene udaljenosti i rotacije dobivene odometrijom kotača veće od stvarnih. Greška odometrije kotača mogla bi se smanjiti umjeravanjem. GNSS prijemnik je također imao problema - duplicirane poruke, nedovoljan broj praćenih satelita (status poruke *NO\_FIX*) i putanje s velikim razlikama u odnosu na stvarnu putanju. Moguće objašnjenje za dio grešaka procjene su višestazni putevi signala zbog stabala i zgrada u blizini igrališta gdje je odrađeno testiranje. Točnost procjene položaja pomoću GNSS-a mogla bi se poboljšati korištenjem RTK korekcija.

Zbog problema prisutnih u snimljenim *rosvag*-ovima testiranje točnosti bilo je moguće samo za konfiguracije koje koriste kameru i IMU senzor. Točnost je testirana usporedbom procijenjenih i referentnih poza tijekom kretanja po okviru na igralištu. Uspoređeno je nekoliko konfiguracija *rgbd\_odometry* čvora za vizualnu odometriju. Najmanje greške procjene postignute su sa parametrima *Odom/Strategy=1* i *Odom/Holonomic=false*. Konfiguracije vizualne odometrije testirane su na slikama razlučivosti ( $480 \times 270$ ) uz 15 okvira u sekundi. U budućem radu bi se moglo testirati kako promjena razlučivosti i broja okvira u sekundi utječe na točnost procjene poze. Također bi se mogao testirati utjecaj preostalih RTAB-Map parametara RGB-D odometrije, primjerice odabir vizualnih značajki koje se koriste.

Testirana je RTAB-Map implementacija istovremenog kartiranja i lokalizacije zajedno sa RGB-D vizualnom odometrijom. Dobivena je 3D karta koja se može iskoristiti za lokalizaciju robota. RTAB-Map je uspješno detektirao zatvaranje petlje u trenutku povratka robota u početni položaj i daje dobre rezultate lokalizacije.



## Poglavlje 8. Testiranje i usporedba konfiguracija za lokalizaciju

ROS čvor *rtabmap* ima mogućnost korištenja GNSS poruka tijekom izgradnje karte i lokalizacije. To bi se moglo testirati u budućem radu.

Testirano je filtriranje vizualne odometrije proširenim Kalman filtrom (EKF) i EKF fuzija naredbi za kretanje robota, vizualne odometrije i IMU poruka. Korištenje Kalman filtra na ovom skupu podataka senzora robota nije poboljšalo rezultate u odnosu na vizualnu odometriju. Veliki dio slučajnog šuma već je uklonjen konfiguracijom vizualne odometrije pa EKF filtriranje nije bilo potrebno. Putanja korištena za testiranje lokalizacije robota je kratka (37 metara). Korištenje Kalman filtra moglo bi biti korisno za uklanjanje grešaka odometrije skupljenih tijekom dugoročne lokalizacije, uz uvjet da su među izvorima za fuziju dostupne procjene apsolutnog položaja (primjerice GNSS) ili poze (detekcija zatvaranja petlje). U ovom radu točnost procjene se računa samo na četiri referentne točke putanje. Usporedba točnosti procjene različitih konfiguracija lokalizacije na složenijoj i nekoliko puta dužoj putanji od one korištene u ovom radu mogla bi dati bolji uvid u moguće prednosti korištenja Kalman filtra. Za dobivanje referentne putanje (eng. *ground truth*) mogao bi se koristiti GNSS prijemnik sa RTK korekcijama ili sustavi za praćenje objekata za koje je poznato da imaju visoku točnost.

# Poglavlje 9

## Zaključak

U ovom radu razmatran je problem lokalizacije, odnosno problem procjene položaja i orijentacije mobilnog robota u odabranom referentnom koordinatnom sustavu. Opisani su senzori korisni za lokalizaciju, lokalizacija odometrijom i satelitskom navigacijom, te mane korištenja samo jedne od tih metoda. Bolja točnost lokalizacije može se postići fuzijom više senzora za lokalizaciju. Opisan je izračun za fuziju senzora Kalman filtrom i pristupi za istovremeno kartiranje i lokalizaciju.

Implementirano je rješenje za lokalizaciju fuzijom senzora na mobilnom robotu sa ROS sustavom i skripta za testiranje točnosti procjena položaja i orijentacije robota dobivenih različitim konfiguracijama sustava za lokalizaciju. Rješenje je testirano sa Pioneer 3-AT mobilnim robotom. Snimljene su poruke senzora tijekom kretanja robota. Zbog problema sa snimanjem bilo je moguće testirati točnost samo za konfiguracije lokalizacije koje koriste dubinsku kameru i IMU senzor. Testirano je nekoliko konfiguracija RTAB-Map vizualne odometrije, istovremeno kartiranje i lokalizacija (SLAM) i nekoliko konfiguracija EKF fuzije senzora. Na putanji robota korištenoj za testiranje točnosti fuzija senzora Kalman filtrom nije dala bolje rezultate od korištenja RTAB-Map rješenja za vizualnu odometriju i SLAM, ali moguće je da bi EKF poboljšao rezultate dugoročne lokalizacije ako je dostupana procjena položaja GNSS-om. Predloženo je testiranje točnosti na dužoj i složenijoj putanji robota za bolji uvid u moguće prednosti korištenja Kalman filtra.

# Bibliografija

- [1] T. Moore and D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.
- [2] S. Huang and G. Dissanayake, *Robot Localization: An Introduction*, August 2016.
- [3] D. S. Roland Siegwart, Illah Reza Nourbakhsh, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2011.
- [4] N. Correll, *Introduction to Autonomous Robots*. Magellan Scientific, 2016.
- [5] OMEGA - How to Measure Acceleration? , s Interneta, <https://www.omega.com/en-us/resources/accelerometers> , kolovoz 2018.
- [6] Autonomous Mobile Robots - Perception: Sensors. , s Interneta, [https://ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/asl-dam/documents/lectures/autonomous\\_mobile\\_robots/spring-2015/AMR\\_2015\\_Lecture\\_4\\_Perception\\_I\\_Sensors.pdf](https://ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/asl-dam/documents/lectures/autonomous_mobile_robots/spring-2015/AMR_2015_Lecture_4_Perception_I_Sensors.pdf) , 2015.
- [7] ANALOG DEVICES - What are the major error sources for inertial sensors? , s Interneta, [https://www-dev.cldnet.analog.com/en/education/education-library/faqs/faq\\_what\\_are\\_the\\_major\\_error\\_sources\\_for\\_inertial.html](https://www-dev.cldnet.analog.com/en/education/education-library/faqs/faq_what_are_the_major_error_sources_for_inertial.html) , 2021.
- [8] Vectornav - What is an Inertial Measurement Unit? , s Interneta, <https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu> .
- [9] CEVA - What is an IMU Sensor? , s Interneta, <https://www.ceva-dsp.com/ourblog/what-is-an-imu-sensor/> , studeni 2018.
- [10] TUM - RGB-D Vision. , s Interneta, [https://cvg.cit.tum.de/research/rgb-d\\_sensors\\_kinect](https://cvg.cit.tum.de/research/rgb-d_sensors_kinect) , 2014.

## Bibliografija

- [11] STEREO LABS - Adding Depth Perception in ROS. , s Interneta, <https://www.stereolabs.com/docs/ros/depth-sensing/> , 2021.
- [12] Arne Nordmann, “Epipolar geometry,” [https://commons.wikimedia.org/wiki/File:Epipolar\\_geometry.svg](https://commons.wikimedia.org/wiki/File:Epipolar_geometry.svg), 2007.
- [13] D. Scaramuzza and F. Fraundorfer, “Visual Odometry [Tutorial],” *IEEE Robotics & Automation Magazine*, vol. 18, pp. 80–92, 2011.
- [14] M. D. Prpić, “Kartiranje vanjskih prostora u stvarnom vremenu s primjenama u poljoprivredi,” Diplomski rad, Sveučilište u Rijeci, Tehnički fakultet, 2022, dostupno na: <https://urn.nsk.hr/urn:nbn:hr:190:291052>.
- [15] Jan Van Sickle, “GEOG 862 GPS and GNSS for Geospatial Professionals,” <https://www.e-education.psu.edu/geog862/home.html>, 2019.
- [16] GPS.GOV - Other Global Navigation Satellite Systems (GNSS). , s Interneta, <https://www.gps.gov/systems/gnss/> 2021.
- [17] B. Ciechanowski, “Global Positioning System,” <https://ciechanow.ski/gps/>, 2022.
- [18] “Klobuchar Ionospheric Model,” [https://gssc.esa.int/navipedia/index.php/Klobuchar\\_Ionospheric\\_Model](https://gssc.esa.int/navipedia/index.php/Klobuchar_Ionospheric_Model), 2011.
- [19] GPS.gov - GPS Accuracy. , s Interneta, <https://www.gps.gov/systems/gps/performance/accuracy/> , 2021.
- [20] CROPOS. , s Interneta, <https://www.cropos.hr/> , 2022.
- [21] Y. Pei, S. Biswas, D. Fussell, and K. Pingali, “An Elementary Introduction to Kalman Filtering,” *Communications of the ACM*, vol. 62, 10 2017.
- [22] Bzarg - How a Kalman filter works, in pictures. , s Interneta, <https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/> , 2015.
- [23] N. Freier, “Kalman filter: Multiplying normal distributions,” [https://www.norbert-freier.de/dateien/kalman\\_filter\\_multiplying\\_normal\\_distributions\\_norbert\\_freier\\_2013.pdf](https://www.norbert-freier.de/dateien/kalman_filter_multiplying_normal_distributions_norbert_freier_2013.pdf), 2013.
- [24] R. Faragher, “Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Lecture Notes],” *IEEE Signal Processing Magazine*, vol. 29, pp. 128–132, 09 2012.

## Bibliografija

- [25] Petteri Aimonen, “Epipolar geometry,” [https://commons.wikimedia.org/wiki/File:Basic\\_concept\\_of\\_Kalman\\_filtering.svg](https://commons.wikimedia.org/wiki/File:Basic_concept_of_Kalman_filtering.svg), 2011.
- [26] Frank Dellaert, “What are Factor Graphs?” <https://gtsam.org/2020/06/01/factor-graphs.html>, 2020.
- [27] J. Sivic and A. Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos,” vol. 2, 11 2003, pp. 1470 – 1477 vol.2.
- [28] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–, 11 2004.
- [29] R. Mur-Artal and J. Tardos, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras,” *IEEE Transactions on Robotics*, vol. PP, 10 2016.
- [30] M. Labbé and F. Michaud, “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *Journal of Field Robotics*, vol. 36, 10 2018.
- [31] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, “ROS: an open-source Robot Operating System,” vol. 3, 01 2009.
- [32] Gazebo Sim. , s Interneta, <https://gazebosim.org/home> , 2022.
- [33] rviz - ROS Wiki. , s Interneta, <http://wiki.ros.org/rviz> , 2018.
- [34] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, 05 2022.
- [35] T. Foote, “Tf: The transform library,” 04 2013, pp. 1–6.
- [36] DDS Foundation - What is DDS? , s Interneta, <https://www.dds-foundation.org/what-is-dds-3/> , 2020.
- [37] robot\_localization wiki. , s Interneta, [https://docs.ros.org/en/melodic/api/robot\\_localization/html/index.html](https://docs.ros.org/en/melodic/api/robot_localization/html/index.html) , 2016.
- [38] W. Meeussen, “REP 105 - Coordinate Frames for Mobile Platforms,” <https://www.ros.org/reps/rep-0105.html>, 2010.
- [39] Pioneer 3-AT. , s Interneta, <https://www.generationrobots.com/media/Pioneer3AT-P3AT-RevA-datasheet.pdf> , 2011.

## Bibliografija

- [40] MIT CSAIL - Robo-Rats Locomotion: Skid-steer Drive. , s Interneta, <https://groups.csail.mit.edu/drl/courses/cs54-2001s/skidsteer.html> , 2004.
- [41] u-blox ZED-F9P GNSS module. , s Interneta, <https://www.u-blox.com/en/product/zed-f9p-module> , 2021.
- [42] u-blox C099-F9P application board. , s Interneta, <https://www.u-blox.com/en/product/c099-f9p-application-board> , 2021.
- [43] Intel® RealSense™ Depth Camera D455. , s Interneta, <https://www.intelrealsense.com/depth-camera-d455/> , 2020.
- [44] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment - a modern synthesis,” in *Workshop on Vision Algorithms*, 1999.

# Sažetak

U ovom radu razmatran je problem lokalizacije, odnosno problem procjene položaja i orijentacije mobilnog robota. Opisano je nekoliko senzora i metoda za lokalizaciju u vanjskom okruženju te fuzija senzora Kalman filtrom. Implementirano je rješenje za lokalizaciju ROS mobilnog robota pomoću postojećih ROS paketa. Lokalizacija je testirana sa Pioneer 3-AT mobilnim robotom, u-blox ZED-F9P GNSS prijemnikom i Intel Realsense D455 kamerom. Testirano je i uspoređeno nekoliko rješenja za lokalizaciju: RTAB-Map SLAM, vizualna odometrija te EKF fuzija vizualne odometrije i IMU senzora.

***Ključne riječi*** — mobilna robotika, lokalizacija, fuzija senzora, Kalman filtar, SLAM

## Abstract

This thesis considers the problem of localizing a mobile robot, i.e. the problem of estimating a robots position and orientation. An overview of several sensors and methods for outdoor localization is provided, along with a description of Kalman filter sensor fusion. Localization was tested with a Pioneer 3-AT mobile robot, u-blox ZED-F9P GNSS receiver and Intel Realsense D455 depth camera. Several localization approaches were tested and compared: RTAB-Map SLAM, visual odometry and EKF fusion of visual odometry and IMU data.

***Keywords*** — mobile robotics, localization, sensor fusion, Kalman filter, SLAM