Navigacija mobilnog robota hodača primjenom globalnog sustava satelitske navigacije (GNSS)

Miklaužić, Filip

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:295118

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-16**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Filip Miklaužić

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Doc. dr. sc. Marko Švaco, mag. ing. mech

Student:

Filip Miklaužić
0035219360

Zagreb, 2024

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem mentoru Doc. dr. sc. Marku Švaci te Branimiru Ćaranu na danim savjetima, komentarima i usmjerenjima tijekom izrade ovog rada. Također želim izraziti iskrenu zahvalu kolegi Luki Grdenu koji mi je pružio neprocjenjivu pomoć tijekom procesa validacije algoritama na mobilnom robotu. Njegova pomoć i podrška ključni su faktori validacije ovog rada.

Nadalje, posebno se zahvaljujem svojim roditeljima, Jasminki i Ivanu, na pruženoj podršci i pomoći bez koje ne bih uspješno završio studij. Također, zahvaljujem se bratu Franku i svim bližnjim na podršci, a veliko hvala mojoj djevojci Doris na pruženoj podršci, ljubavi i razumijevanju tijekom studija.

Na posljetku, hvala Ti Bože za Tvoju milost i snagu koja me vodila kroz izazove studija i omogućila mi uspjeh na fakultetu.

Filip Miklaužić

”No race has ever been won in the first corner; many have been lost there.“

Garth Stein

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment, mehatronika i robotika, autonomni sustavi i računalna inteligencija

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: **602 - 04 / 24 - 06 / 1**	
Ur.broj: **15 - 24 -**	

DIPLOMSKI ZADATAK

Student: **Filip Miklaužić** JMBAG: **0035219360**

Naslov rada na hrvatskom jeziku: **Navigacija mobilnog robota hodača primjenom globalnog sustava satelitske navigacije (GNSS)**

Naslov rada na engleskom jeziku: **Outdoor navigation of a quadruped walking mobile robot using global navigation satellite system (GNSS)**

Opis zadatka:

Primjena globalnog sustava satelitske navigacije (GNSS - eng. *Global Navigation Satellite System*) sve je značajnija u procesima lokalizacije te navigacije mobilnih robota jer nudi mogućnosti dobivanja informacija o trenutnoj lokaciji robota u svijetu. Jedan od najpoznatijih sustava satelitske navigacije je GPS (eng. *Global Positioning System*). Ako se izvrši fuzija informacija dobivenih iz GNSS sustava s informacijama iz inercijalne mjerne jedinice (IMU – eng. *Inertial Measurement Unit*) i odometrije robota, moguće je povećati točnost lokalizacije robota u prostoru. Roboti hodači jedna su vrsta mobilnih robota, a danas pronalaze značajniju primjenu u vanjskim prostorima zbog mogućnosti prilagodbe i kretanjima na raznovrsnim vrstama i konfiguracijama podloga. Kombinacijom svih prethodnih tehnologija moguće je povećati autonomiju robotskih sustava te povećati točnost i robusnost rada u vanjskim prostorima.

U sklopu diplomskog rada potrebno je:

- Istražiti mogućnosti primjene mobilnog robota hodača *Boston Dynamics SPOT* s odabranim GNSS sustavom,
- Ostvariti komunikaciju robota i nadređenog upravljačkog računala s ciljem upravljanja prostornim položajem robota u zadanoj mapi,
- Istražiti i primijeniti sustav GNSS navigacije primjenom GPS-RTK sustava povećane apsolutne točnosti,
- Primjenom GNSS sustava i odometrije robota potrebno je omogućiti kontinuiranu lokalizaciju robota u vanjskom prostoru i u zadanoj mapi,
- Istražiti i primijeniti globalne planere putanja za navigaciju temeljenu na GNSS koordinatama,
- Eksperimentalno validirati sustave lokalizacije i navigacije robota na zadanoj mapi.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:	Datum predaje rada:	Predviđeni datumi obrane:
18. siječnja 2024.	21. ožujka 2024.	25. – 29. ožujka 2024.

Zadatak zadao: Predsjednik Povjerenstva:

Doc. dr. sc. Marko Švaco Prof. dr. sc. Ivica Garašić

SADRŽAJ

SADRŽAJ I
POPIS SLIKA IV
POPIS TABLICA VI
POPIS TEHNIČKE DOKUMENTACIJE VII
POPIS OZNAKA VIII
POPIS KRATICA X
SAŽETAK XII
SUMMARY XIII
1. UVOD 1
2. MOBILNA ROBOTIKA 3
2.1. Karakteristike mobilnih robota 3
2.2. Osnovni podsustavi mobilnih robota 4
2.3. Robot hodač 5
2.4. Robotski operativni sustav 6
2.4.1. Motivacija univerzalnog robotskog sustava 7
2.4.2. Generalni pregled ROS i ROS2 sustava 8
2.4.3. Kompatibilnost i prijelaz 9
2.4.4. Zajednica i podrška 10
2.4.5. Primjena 10
2.5. Implementacija GNSS sustava na sustav mobilnog robota 11
3. SUSTAV GNSS NAVIGACIJE 12
3.1. Konfiguracija RTK sustava 13
3.2. IZBOR BAZNE STANICE 14
3.2.1. Integrirani sustavi baze i rovera 15
3.3. Komunikacija RTK opreme i ROS2 sustava 17
3.3.1. Ublox čvor 17
3.3.1.1. Parametri ublox_gps_node čvora 19
3.3.2. Serveri zaduženi za prikaz satelitskih snimaka 20
3.3.3. Mapviz paket 21
3.3.3.1. Parametri mapviz_node čvora 22
3.4. Testiranje rada RTK sustava 22
4. BOSTON DYNAMICS SPOT 26
4.1. Kinematika i dinamika Spot mobilnog robota 27
4.2. Specifikacije Spot mobilnog robota 28
4.2.1. Određivanje brzina kretanja Spot mobilnog robota 29
4.3. Senzorika integrirana na mobilnom robotu 31
4.3.1. Stereo kamere 31
4.3.1.1. realsense-ros paket 33
4.3.2. Lidar senzori 33
4.3.3. IMU senzori 34

4.3.3.1. Kalibracija IMU-a ... 35
4.4. Upravljačko računalo Intel Nuc ... 36
4.4.1. Komunikacija putem DDS protokola ... 38
4.4.2. Kvaliteta servisa ROS sustava ... 40
4.5. Spot SDK ... 41
4.6. Spot_ros2 paket ... 41
4.6.1. spot_driver ... 42
4.7. Transformacije koordinatnih sustava robota i opreme ... 43
4.8. Implementacija sustava na mobilnog robota ... 45
4.8.1. Konstrukcijska rješenja implementacije sustava ... 45
4.8.2. Napajanje implementiranih sustava ... 49
5. LOKALIZACIJA ROBOTA ... 51
5.1. Kalmanov filtar ... 51
5.1.1. Prošireni Kalmanov filtar ... 53
5.1.2. Neprikriveni Kalmanov filtar ... 54
5.2. Fuzija međusobno nezavisnih podataka lokalizacije ... 54
5.3. Robot_localization paket ... 55
5.3.1. Navsat_transform_node čvor ... 58
5.3.2. Ekf_filter_node čvor ... 60
5.3.3. Ukf_filter_node čvor ... 65
5.4. Lokalizacija robota u vanjskom okruženju ... 66
6. GLOBALNO PLANIRANJE PUTANJE ... 67
6.1. Nav2 paket ... 67
6.1.1. Modularni pristup nav2 paketa ... 68
6.1.1.1. Stabla ponašanja ... 69
6.1.1.2. Akcijski poslužitelji ... 70
6.1.1.3. Navigacijski poslužitelji ... 71
6.1.1.4. Otisak robota ... 73
6.1.1.5. Reprezentacija okoline ... 73
6.2. Navigacija korištenjem GNSS lokalizacije ... 74
6.2.1. Postavljanje stabla ponašanja globalne navigacije ... 74
6.2.2. Postavljanje poslužitelja globalne navigacije ... 75
6.2.2.1. Upravljački poslužitelji globalne navigacije ... 76
6.2.2.2. Poslužitelji planiranja putanje pri globalnoj navigaciji ... 81
6.2.2.3. Poslužitelji lokalnih mapa troškova pri globalnoj navigaciji ... 83
6.2.2.4. Poslužitelji globalnih mapa troškova pri globalnoj navigaciji ... 85
6.2.2.5. Poslužitelji zaglađivanja putanje pri globalnoj navigaciji ... 87
6.2.2.6. Poslužitelji zaglađivanja brzine pri globalnoj navigaciji ... 88
6.2.3. Postavljanje karte troškova lokalnog planera putanje ... 90
6.2.3.1. Spot algoritam za izbjegavanje prepreka ... 91
6.2.3.2. Izbjegavanje prepreka temeljeno na reprezentaciji okoline korištenjem oblaka točaka ... 91
6.2.4. Postavljanje karte troškova globalnog planera putanje ... 93
6.2.4.1. Planeri putanje temeljeni na zadanim putnim točkama ... 94
6.2.4.2. Poslužitelji za upravljanje ponašanjima u zadacima globalne navigacije ... 95
6.3. Zadavanje referentne trajektorije ... 97
7. NAVIGACIJA MOBILNOG ROBOTA ... 98
7.1. Priprema algoritama navigacije na simulacijskom modelu mobilnog robota ... 98

7.1.1. Postavljanje simulacijskog okruženja 99
7.1.1.1. Prikaz simulacijskog okruženja 99
7.1.1.2. Podešavanje čvorova i poslužitelja lokalizacije i globalne navigacije 101
7.1.2. Praćenje putnih točaka bez prepreka u simulacijskom okruženju 101
7.1.3. Praćenje putnih točaka s preprekama u simulacijskom okruženju 103
7.2. Validacija algoritama navigacije na mobilnom robotu 104
7.2.1. Praćenje putnih točaka Spot mobilnim robotom u okolišu bez prepreka 105
7.2.2. Praćenje putnih točaka Spot mobilnim robotom u okolišu s preprekama..... 107
7.3. Prijedlog budućeg rada..... 109
8. ZAKLJUČAK..... 111
LITERATURA..... 112
PRILOZI..... 116

POPIS SLIKA

Slika 1. Mobilni robot hodač kompanije Notum Robotics [8] 6
Slika 2. Ilustracija orbita najpoznatijih zemljinih satelita [13] 12
Slika 3. Prikaz GNSS RTK sustava baze i rovera [14] 13
Slika 4. SparkFun RTK Facet integrirani sustav [15] 15
Slika 5. SparkFun RTK Facet u ulozi bazne stanice postavljena 16
Slika 6. Holybro odašiljač/prijamnik za slanje korekcijskih podataka između baze i rovera .. 17
Slika 7. u-center korisničko sučelje 18
Slika 8. Testiranje rada servera distribucije satelitskih snimaka 21
Slika 9. Prikaz Mapviz sučelja 22
Slika 10. Rasipanje podataka globalne pozicije u slučaju korištenja rovera bez primjene korekcijskih podataka 23
Slika 11. Rezultat testiranja RTK opreme – opći prikaz 24
Slika 12. Rezultat testiranja RTK opreme – detaljniji prikaz 24
Slika 13. SPOT robot u CRTA okruženju 27
Slika 14. Prikaz koordinatnih sustava i zglobova četveronožnog mobilnog robota hodača [26] 28
Slika 15. Prikaz Realsense D435 stereo kamere [28] 31
Slika 16. Dubinska slika okruženja Spota u Rviz okruženju [29] 32
Slika 17. Spot s dodatkom poboljšane autonomije (EAP) [32] 34
Slika 18. Intel Nuc 13 Pro računalo [35] 37
Slika 19. Struktura komunikacije između upravljačkih računala 39
Slika 20. shematski prikaz spajanja komponenti korištenih u zadacima globalne navigacije mobilnog robota 39
Slika 21. Prikaz koordinatnih sustava Intel RealSense D400 serije kamera 43
Slika 22. Statičke transformacije koordinatnog sustava IMU senzora u odnosu na bazu sustava za dodatke 44
Slika 23. Statičke transformacije koordinatnog sustava GNSS prijamnika u odnosu na bazu sustava za dodatke 45
Slika 24. Model sklopa nosača GNSS sustava 46
Slika 25. Model sklopa kučišta Intel NUC računala 47
Slika 26. GNSS sustav postavljen na nosač 47
Slika 27. Intel NUC računalo postavljeno na nosač 48

Slika 28. Nosač Intel RealSense D435i kamere postavljena na nosač GNSS prijamnika 48
Slika 29. Oprema postavljena na Spot robota 49
Slika 30. Spot dodatak za proširenje montirani na robotu 50
Slika 31. Dijagram nastanka signala poboljšane odometrije [59] 55
Slika 32. Transformacije koordinatnih sustava u zadatku globalne lokalizacije [59] 56
Slika 33. Prikaz koordinatnih sustava mobilnih robota [61] 56
Slika 34. Koordinatni sustavi robota nakon provedene globalne lokalizacije 66
Slika 35. Koncept djelovanja poslužitelja nav2 paketa [68] 68
Slika 36. Prikaz međusobne interakcije čvorova i poslužitelja korištenih u zadacima globalne navigacije 69
Slika 37. Prikaz pretvorbe oblaka točaka u očitanja lasera 93
Slika 38. Prikaz modela TurtleBot Waffle robota u stvorenom Gazebo okruženju 100
Slika 39. Prikaz Mapviz sučelja simulacijskog okruženja 101
Slika 40. Praćenje putnih točaka bez prepreka u Gazebo simulacijskom okruženju prikazano u videozapisu [81] 102
Slika 41. Praćenje putnih točaka s postavljenim preprekama u Gazebo simulacijskom okruženju prikazano u videozapisu [82] 103
Slika 42. Prikaz okruženja validacije algoritama navigacije 104
Slika 43. Prikaz putnih točaka korištenih za validaciju navigacije Spot mobilnim robotom. 105
Slika 44. Praćenje putnih točaka Spot mobilnog robota bez postavljenih prepreka prikazano u videozapisu [84] 106
Slika 45. Praćenje putnih točaka Spot mobilnog robota s jednom postavljenom preprekom prikazano u videu [85] 107
Slika 46. Praćenje putnih točaka Spot mobilnog robota s dvije postavljene prepreke prikazano u videu [86] 108
Slika 47. Kompilacija praćenja putnih točaka Spot mobilnog robota s postavljenim preprekama prikazana u videu [87] 109

POPIS TABLICA

Tablica 1. Specifikacije Boston Dynamics Spot robota hodača 29
Tablica 2. Prikaz graničnih brzina Boston Dynamics Spot mobilnog robota hodača 30
Tablica 3. Karakteristike Intel NUC 13 računala 37

POPIS TEHNIČKE DOKUMENTACIJE

NGNSS-00-00	Nosač GNSS prijamnika
NNUC-00-00	Nosač Nuc 13 Pro računala

POPIS OZNAKA

Oznaka	Jedinica	Opis
$\vec{a}_{bias}$	m/s	Iznos ubrzanja kojim se uklanjaju utjecaji nepravilnosti senzora u nultom položaju
$\vec{a}_{raw}$	m/s	Iznos sirovog ubrzanja dobivenog od senzora
$\vec{a}_{true}$	m/s	Iznos stvarnog ubrzanja kojeg daje kamera
$\vec{\omega}_{bias}$		Iznos kuta zakreta kojim se uklanjaju utjecaji nepravilnosti senzora u nultom položaju
$\vec{\omega}_{raw}$		Iznos sirovog kuta zakreta dobivenog od senzora
$\vec{\omega}_{true}$		Iznos stvarnog kuta zakreta kojeg daje kamera
c_{xy}		Faktor korekcije ortogonalnosti x osi u odnosu na y os
c_{xz}		Faktor korekcije ortogonalnosti x osi u odnosu na z os
c_{yx}		Faktor korekcije ortogonalnosti y osi u odnosu na x os
c_{yz}		Faktor korekcije ortogonalnosti y osi u odnosu na z os
c_{zx}		Faktor korekcije ortogonalnosti z osi u odnosu na x os
c_{zy}		Faktor korekcije ortogonalnosti z osi u odnosu na y os
$pitch$	rad	kut nutacije
$\dot{pitch}$	rad/s	derivacija kuta nutacije
$roll$	rad	kut precesije
$\dot{roll}$	rad/s	derivacija kuta precesije
s_x		Koeficijent umnoška sirovog signala mjerenja ubrzanja uzduž x osi kojim se osigurava izlaz senzora u jedinicama SI sustava
s_y		Koeficijent umnoška sirovog signala mjerenja ubrzanja uzduž y osi kojim se osigurava izlaz senzora u jedinicama SI sustava
s_z		Koeficijent umnoška sirovog signala mjerenja ubrzanja uzduž z osi kojim se osigurava izlaz senzora u jedinicama SI sustava
$\mathbf{T}_{GNSS}$		Matrica transformacije koordinatnog sustava GNSS prijamnika u odnosu na koordinatni sustav prednje vodilice

$\mathbf{T}_{IMU}$		Matrica transformacije koordinatnog sustava IMU senzora u odnosu na koordinatni sustav prednje vodilice
X	m	x koordinata robota u relativnom koordinatnom sustavu
$\dot{X}$	m/s	brzina robota u smjeru x osi
$\ddot{X}$	m/s^2	ubrzanje robota u smjeru x osi
yaw	rad	kut rotacije
$\dot{yaw}$	rad/s	derivacija kuta rotacije
Y	m	y koordinata robota u relativnom koordinatnom sustavu
$\dot{Y}$	m/s	brzina robota u smjeru y osi
$\ddot{Y}$	m/s^2	ubrzanje robota u smjeru y osi
Z	m	z koordinata robota u relativnom koordinatnom sustavu
$\dot{Z}$	m/s	brzina robota u smjeru z osi
$\ddot{Z}$	m/s^2	ubrzanje robota u smjeru z osi

POPIS KRATICA

API – sučelje za razvoj aplikacija *(*engl. *Application Programming Interface)*

CAD – računalno potpomognuto oblikovanje (engl. *Computer Aided Design*)

DDS – poslužitelj za distribuciju podataka (engl. *Data Distribution Service*)

DWB – osnovni pristup dinamičkom prozoru (engl. *Dynamic Window-approach Base*)

EAP – dodatak poboljšane autonomije (engl. *Enhanced Autonomy Payload*)

EKF – Prošireni Kalmanov filtar (engl. *Extended Kalman Filter*)

GNSS – globalni sustav satelitske navigacije (engl. *global navigation satelite system*)

GPS – sustav globalnog pozicioniranja (engl. *Global Positioning System*)

GRV – nasumične vrijednosti Gaussove raspodjele (engl. *Gaussian Randoom Variable*)

GXP – dodatak za proširenje (engl. *General Expansion Payload*)

HAA – abdukcija/adukcija kuka (engl. *Hip Abduction/Adduction*)

HFE – savijanje/produljenje kuka (engl. *Hip Flexion/Extension*)

IMU – jedinica za mjerenje inercija (engl. *Inertial Measurement Unit*)

KFE – savijanje/produljenje koljena (engl. *Knee Flexion/Extension*)

LIDAR – otkrivanje i domet svjetlosti (engl. *Light detection and ranging*)

LiPo – polimer litija (engl. *Lithium Polymer*)

L/R F – lijeva/desna prednja noga (engl. *Left/Right Foreleg*)

L/R H – lijeva/desna stražnja noga (engl. *Left/Right Hind leg*)

PETG – polietilen tereftalat glikol (engl. *Polyethylene Terephthalate Glycol*)

PLA – polimliječna kiselina (engl. *Polylactic acid*)

RCL – ROS biblioteka podrške klijentima (engl. *ROS Client Library*)

QoS – kvaliteta ROS servisa (engl. *Quality of Service*)

REP – prijedlog poboljšanja ROS-a (engl. *ROS Enhancement Proposal*)

ROS – robotski operativni sustav (engl. *Robot Operating System*)

RPC – pozivna daljinska procedura (engl. *Remote Procedure Call*)

RTK – kinematika u stvarnom vremenu (engl. *Real-Time Kinematic*)

SDK – alat za razvoj softvera (engl. *Software Development Kit*)

TCP – protokol za kontrolu prijenosa (engl. *Transmission Control Protocol*)

UDP – korisnički protokol prijenosa podataka (engl. *User Datagram Protocol*)

UKF – neprikriveni Kalmanov filtar (engl. *Unscented Kalman Filter*)

UTM – univerzalni transverzalni merkator (engl. *Universal Transverse Mercator*)

VESA – udruženje standarda video elektronike (engl. *Video Electronics Standards Association*)

WGS – svjetski geodetski sustav (engl. *World Geodetic System*)

WMTS – distribucija kartografskih podataka u obliku pločica zaduženih za prikaz karata na web stranicama (engl. *Web Map Tile Service*)

XML – proširivi označni jezik (engl. *Extensible Markup Language*)

YAML – još jedan označni jezik (engl. *yet another markup language*)

SAŽETAK

Primjena globalnog sustava satelitske navigacije postaje sve važnija u procesima lokalizacije i navigacije mobilnih robota, pružajući informacije o trenutnoj globalnoj poziciji robota. Jedan od najpoznatijih satelitskih radionavigacijskih sustava za određivanje položaja na zemlji ili u njezinoj neposrednoj blizini je sustav za globalno pozicioniranje (engl. *Global Positioning System*, GPS). Fuzijom podataka dobivenih od strane globalnog sustava satelitske navigacije (engl. global navigation satelite system, GNSS) s podacima inercijalne mjerne jedinice (engl. *Inertial Measurement Unit*, IMU) te odometrije robota, moguće je povećati točnost lokalizacije robota u prostoru. Roboti hodači predstavljaju inovativnu kategoriju mobilnih robota te su zbog svoje prilagodljivosti i sposobnosti kretanja na različitim vrstama površina i konfiguracija terena posebice relevantni u vanjskom okruženju. Integracijom ovdje navedenih tehnologija moguće je povećati autonomiju robotskih sustava te unaprijediti točnost i otpornost na rad u vanjskim prostorima. U okviru diplomskog rada istražena je mogućnost primjene mobilnog robota Boston Dynamics Spot u kombinaciji s odabranim GNSS sustavom. Nadalje, uspostavljena je komunikacija između robota i nadređenog upravljačkog računala s ciljem upravljanja prostornim položajem robota unutar zadane mape. Također, istraženi su sustavi GNSS navigacije, pri čemu je odabran sustav poboljšane kinematike u stvarnom vremenu (engl. *Real Time Kinematic*, RTK) namijenjen povećanju apsolutne točnosti, što ga čini prikladnim za dobiveni zadatak. Fuzijom podataka GPS-RTK sustava povećane točnosti, odometrije robota te IMU jedinice omogućena je neprekidna lokalizacija robota u vanjskom prostoru unutar zadane mape. Također, istraženi su i primijenjeni globalni planeri putanje za navigaciju temeljenu na GNSS koordinatama, uz eksperimentalnu validaciju sustava lokalizacije i navigacije robota unutar zadane mape.

Ključne riječi: robotski operativni sustav, globalni sustav satelitske navigacije, sustavi navigacije poboljšane apsolutne točnosti, fuzija podataka

SUMMARY

The application of the global satellite navigation system is becoming increasingly important in the localization and navigation processes of mobile robots, providing information about the robot's current global position. One of the most well-known satellite radio navigation systems for determining position on or near the Earth is the Global Positioning System (GPS). By fusing data obtained from the global navigation satellite system (GNSS) with data from the Inertial Measurement Unit (IMU) and robot odometry, it is possible to increase the accuracy of robot localization in space. Quadruped robots represent an innovative category of mobile robots and are particularly relevant in outdoor environments due to their adaptability and ability to move on various types of surfaces and terrain configurations. Integrating the technologies mentioned in this master thesis can increase the autonomy of robotic systems and improve accuracy and resilience to outdoor workspaces. In the context of this thesis, the possibility of applying the Boston Dynamics Spot mobile robot in combination with a selected GNSS system was investigated. Furthermore, communication was established between the robot and the superior control computer to control the robot's spatial position within the specified map. Additionally, GNSS navigation systems were explored, with a focus on the Real Time Kinematic (RTK) system for improved real-time kinematics, making it suitable for the given task. By fusing data from the GPS-RTK system, increased odometry, and IMU unit, continuous localization of the robot in outdoor spaces within the specified map was enabled. Furthermore, global path planning algorithms for navigation based on GNSS coordinates were explored and applied, with experimental validation of the robot's localization and navigation systems within the specified map.

Keywords: robotic operating system, global navigation satellite system, enhanced absolute accuracy navigation systems, data fusion

1. UVOD

Projekti mobilnih robota hodača koji koriste GNSS sustave za zadatke kretanja kroz prostor obuhvaćaju značajan udio inovacija u području autonomnih robotskih sustava. Uzimajući u obzir brz tehnološki napredak u području robotike, ovim diplomskim radom prikazana je integracija GNSS sustava u mobilne robote hodače, otvarajući vrata raznolikim primjenama kako u industriji i istraživanju, tako i u svakodnevnom životu.

Mobilni roboti hodači, oponašajući organizme iz prirode predstavljaju značajnu inovaciju u području robotike. Korištenjem kombinacija nogu s naprednim senzorima i aktuatorima postiže se visoka prilagodljivost različitim tipovima terena. Njihova kinematska struktura omogućava savladavanje prepreka poput stepenica, neravnina, stijena te ostalih nepredvidivih površina te ih čini vrlo sposobnim u kretanju u stvarnim uvjetima. U doba sveopćeg ubrzanog razvoja tehnologije, inovativna rješenja koja sjedinjuju mobilnost, preciznost i autonomiju postaju ključna u raznim područjima. Projekt mobilnih robota hodača, koji koriste GNSS sustave za navigaciju i kretanje kroz prostor, predstavlja značajan korak u procesu stvaranja inteligentnih sustava sposobnih za autonomno istraživanje okruženja uz sposobnost prilagodbe istom. Kroz istraživanje i eksperimentalnu validaciju dolazi se do novih mogućnosti primjena kao i rješenja izazova koji proizlaze iz unaprijed nepoznatog dinamičnog okruženja robota što za posljedicu ima značajniju prilagodljivost autonomnih robotskih sustava.

GNSS tehnologija, koja je u posljednjih nekoliko desetljeća postala neizostavan dio naše svakodnevice koristi se također u sustavima mobilnih robota kako bi im omogućila precizno određivanje položaja i navigaciju kroz kompleksna okruženja. Ovim diplomskim radom istražene su mogućnosti te izazovi povezani s implementacijom GNSS sustava u mobilne robote hodače, uz naglasak na ostvarivanje sposobnosti planiranja putanja te dinamičko prilagođavanje u nepredvidivim situacijama. Nadalje, analizirani su tehnološki aspekti mobilnih robota hodača te njihova sposobnost kretanja i navigacije s pomoću GNSS sustava uz sagledavanje potencijalne primjene u raznim sektorima. Također, sagledani su potencijalni izazovi s kojima se susreću mobilni roboti hodači prilikom korištenja GNSS sustava, uključujući preciznost određivanja položaja, osjetljivost na okolišne uvjete te potrebu za fuzije različitih senzora. Kao motivacija ovog diplomskog rada navodi se komercijalni paket praćenja GNSS putnih točaka [1] kompanije Clearpath Robotics koji donosi mogućnost autonomne

navigacije mobilnog robota na temelju zadanih putnih točaka. Svi paketi korišteni tijekom izrade ovog diplomskog rada priloženi su na repozitorij diplomskog rada [2].

U prvom dijelu rada dan je kratki uvod o trenutnom stanju područja mobilne robotike. Potom je dan opis GNSS sustava pozicioniranja objekata na planetu Zemlji čiji je detaljniji opis dan u Projektu MiR [3]. Nadalje, dan je opis mobilnog robota hodača kompanije Boston Dynamics, industrijskog naziva Spot koji je korišten kao mobilna platforma na kojoj se bazira implementacija svih algoritama korištenih u ovom radu. Potom su opisani postupci lokalizacije robota na zemlji uz detaljan opis korištenih algoritama globalnih planera putanje korištenih u ovom radu. U posljednjem dijelu rada dani su eksperimentalni rezultati provođenja lokalizacije i navigacije uz navođenje metoda validacije implementiranih algoritama.

2. MOBILNA ROBOTIKA

Mobilna robotika, odnosno grana robotike posvećena pokretnim robotskim sustavima predstavlja složeni spoj brojnih tehnologija. Ovo dinamično područje istražuje mogućnosti stvaranja robotskih sustava sposobnih za samostalno kretanje i interakciju s okolinom. Mobilni roboti postali su zastupljeni u širokom spektru industrija, istraživanju, medicini, aeronautici, kućanstvu te drugim područjima koja zahtijevaju kombinaciju preciznost, mobilnost te određeni stupanj autonomije [4].

Autonomija u novije vrijeme slovi kao ključna karakteristika mobilnih robota te se njome označava sposobnost donošenja odluka i izvršavanja zadataka bez neprestane ljudske intervencije. Autonomni robotski sustavi koriste razne senzorske sustave napredne algoritme poput neuronskih mreža kako bi analizirali okolinu, planirali svoje kretanje te ispravno reagirali na promjene u stvarnom vremenu. Ovisno o stupnju autonomije, mobilni roboti mogu se prilagoditi se raznim okruženjima i situacijama čime postaju sposobni nositi se sa situacijama gdje je potrebna brza reakcija na promjenljive uvjete. Kroz integraciju autonomije, mobilna robotika postavlja temelje razvoja inteligentnih tehničkih sustava sposobnih samostalnom obavljanju zadataka. Razvoj takvih sustava potiče inovacije u područjima koje u svom radu koriste tehnologije naprednih sustava mobilne robotike.

2.1. Karakteristike mobilnih robota

U novije vrijeme, mobilna robotika igra značajnu ulogu u suočavanju s izazovima i zadacima, posebice onima koji predstavljaju opasnost za ljudsko zdravlje. Ova tehnologija dobiva svoj puni smisao u situacijama gdje je ljudska prisutnost nepoželjna ili iznimno rizična, bilo da se radi o istraživanju neprijateljskih teritorija, spašavanju u hitnim situacijama ili pak obavljanju zadataka u kontaminiranom okruženju. Korištenje mobilnih robota u takvim situacijama pruža niz prednosti, dok istovremeno postavlja niz izazova koje je potrebno savladati. Nadalje, mobilni roboti često moraju biti oblikovani s visokom prilagodljivošću različitim vrstama terena i podloga. Njihova prilagodljivost doprinosi povećanju sposobnosti kretanja kroz teško dostupne prostore ili neravne površine što ih čini pogodnim za hitne intervencije ili istraživačke zadatke. Također, mobilni roboti često moraju biti načinjeni od izdržljivih materijala kako bi bili sposobni podnijeti ekstremne vanjske uvjete. Za razliku od ljudi, roboti nisu ograničeni radnim vremenima ili fizičkim ograničenjima poput umora. Mogu

raditi neprekidno, pružajući kontinuiranu prisutnost i praćenje u uvjetima gdje ljudi teško održavaju stalnu razinu prisutnosti potrebnu za izvršavanje pojedinih zadataka. Njihova opremljenost naprednim senzorima omogućava im izvršavanje zadataka s visokom preciznošću, čime se povećava učinkovitost uz smanjenje rizika od nepotrebnih grešaka.

Iako su roboti sve autonomniji, neki zadaci i dalje zahtijevaju povremenu ljudsku intervenciju i nadzor. Samostalnost robota može biti ograničena u nepredvidivim situacijama ili okruženjima, time manje što je algoritam upravljanja prilagođeniji raznim mogućim scenarijima i situacijama u kojima se robot može naći tijekom svoj rada.

2.2. Osnovni podsustavi mobilnih robota

Mobilni robot je sustav koji ima sposobnost slobodnog kretanja u fizičkom okolišu, obično bez potrebe za fiksnom infrastrukturom kao što su tračnice, sajle ili nekog drugog sustava vodilica. Roboti u svom radu koriste razne sustave koji su zaduženi za kretanje, percepciju okoline te donošenje odluka o sljedećim aktivnostima koji rezultiraju kretanju bez kolizije s okolinom. Prema funkcionalnim cjelinama, mobilni robot dijeli se na četiri glavna podsustava, odnosno podsustav aktuatora, senzora, upravljački te energetski podsustav [5].

Energetski podsustav mobilnog robota odnosi se na sustav čija je zadaća osiguravanje potrebne energije zadužene za pokretanje i napajanje različitih sustava robota. Ovaj podsustav igra ključnu ulogu u osiguravanju kontinuiranog rada robota, dok se dobava energije može vršiti iz različitih izvora, ovisno o vrsti robota te njegovoj primjeni. Sam pojam mobilne robotike za sebe veže korištenje izvora energije koji robotu omogućuje određenu dozu mobilnosti. U mobilnoj robotici se kao najčešći izvor energije, zbog povoljne gustoće energije, navode punjive baterije [6]. Također, u članku [6] detaljno je uspoređena primjena baterija izrađenih od raznih materijala uz navođenje prednosti i nedostataka svakog od korištenih materijala ćelija. Zadaća upravljačkog podsustava mobilnog robota je uspostavljanje i održavanje komunikacije između svih podsustava te upravljanje istima. Glavni element upravljačkog podsustava je mikroupravljač, upravljački element koji je zadužen za zadatke prikupljanja informacija sa senzora, obradu istih, upravljanje aktuatorima temeljeno na predodređenim zakonitostima te ostale zadatke upravljanja i nadzora robotskih sustava. Upravljački sustav se najčešće sastoji od dva glavna dijela. Prvi dio upravljačkog podsustava je sustav upravljanja visoke razine (engl. *high level control system*) koji je najčešće zadužen za planiranja putanje bez kolizija ili procesiranje podataka dobivenih od strane upravljačkog sustava niske razine (engl. *low level*

control system). Upravljački sustav visoke razine zatim šalje referentne vrijednosti upravljačkom sustavu niske razine čija je glavna zadaća očitavanje podataka sa senzora te upravljanje aktuatorima korištenjem zakonitosti koje su prethodno implementirane u programski kod sustava upravljanja niske razine [7]. Podsustavi aktuatora se kod mobilnih robota sastoje od fizičkih komponenata kojima se omogućuje kretanje ili izvođenje određenog zadatka u radnom prostoru robota. Sklop aktuatora odgovoran je za preslikavanje upravljačkih signala dobivenih od upravljačkog sustava u stvarne pokrete ili radnje robota. Kod mobilnih robota se, kao aktuatori zaduženi za kretanje, najčešće koriste servo i koračni motori zbog visoke upravljivosti i preciznosti. Na posljetku, ključan sustav potreban za ispravan rad robota je podsustav senzora. Glavna uloga sustava senzora jest akvizicija informacija iz okoline robota te njihova transformacija u referentni naponski ili strujni signal. Signali dobiveni iz sustava senzora služe kao referentni ulazi upravljačkom sustavu visoke razine, prenoseći informacije koje su prethodno obrađene u upravljačkom sustavu niske razine. Podaci prikupljeni s pomoću senzora djeluju kao ključni parametri u zadacima interpretacije okoline. Prilikom odabira senzora preferiraju se oni koji omogućuju što jednostavniju dobavu podataka iz okoline. Primjerice, senzori udaljenosti i temperature, akcelerometri, žiroskopi, razni enkoderi, lidari te slični uređaji predstavljaju optimalan izbor za postizanje preciznih informacija o okolini na što jednostavniji način [7].

2.3. Robot hodač

Četveronožni mobilni roboti predstavljaju inovativnu kategoriju mobilnih robota, pružajući široku lepezu prednosti u području istraživanja okoline. Primjerice, Boston Dynamics Spot, jedan od najpoznatijih četveronožnih mobilnih robota, ilustrira ove karakteristike te je detaljnije opisan u poglavlju *4. BOSTON DYNAMICS SPOT*.

Jedna od ključnih prednosti četveronožnih mobilnih robota jest njihova izvanredna pristupačnost raznim terenima. Inspirirani biomehaničkim karakteristikama četveronožnih životinja, ovim robotima omogućeno je kretanje kroz različite tipove terena, uključujući neravne površine, stepenice kao i neujednačena područja poput šljunčane podloge ili kamenja, koja su u velikoj većini slučajeva potpuno nepristupačna mobilnim robotima s kotačima. Ova prilagodljivost čini ih nezamjenjivim pri manevriranju tijekom izvođenja zadataka istraživanja okoline, gdje se često susreću s dinamičkim i nepredvidivim situacijama. Unatoč brojnim prednostima, četveronožni mobilni roboti suočavaju se s brojnim izazovima koji proizlaze iz

složene kinematike i dinamike njihovog kretanja. Modeliranje kao i upravljanje pokretima četveronožnih robota zahtijeva rješavanje izuzetno kompleksnih kinematičkih i dinamičkih problema. Složene jednadžbe kojima se opisuje ponašanje robota moraju biti rješavane u stvarnom vremenu kako bi se osigurala stabilnost, preciznost i sigurnost tijekom kretanja. Složeni zapis dinamike robota predstavlja tehnički zahtjevan aspekt u razvoju četveronožnih mobilnih robota, posebno u situacijama gdje je potrebna brza i precizna prilagodba pokreta. Održavanje ravnoteže ovdje spomenutih prednosti i izazova ključno je za daljnji napredak u razvoju četveronožnih mobilnih robota te ujedno njihovu integraciju u različite aplikacije u kojima mogu biti od velike koristi. Slika 1. prikazuje biološki inspiriran mobilni robota hodača Hrvatske tvrtke Notum Robotics iz Velike Gorice [8]:

Slika 1. Mobilni robot hodač kompanije Notum Robotics [8]

2.4. Robotski operativni sustav

U suvremenom svijetu robotike, robotski operativni sustav, odnosno ROS (engl. *Robot Operating System*) predstavlja ključnu komponentu koja olakšava razvoj i upravljanje robotskih sustava. ROS donosi infrastrukturu i alate kojima se ostvaruje integracija, komunikacija te

upravljanje različitih dijelova robota, čime se olakšava rad s velikom većinom robotskih sustava. Otvorenost i modularni pristup čine ROS sustav popularnim izborom u istraživanju, razvoju i primjeni robotskih tehnologija u gotovo svim područjima koji se na bilo koji način mogu povezati s robotskim tehnologijama.

2.4.1. Motivacija univerzalnog robotskog sustava

ROS predstavlja značajnu prekretnicu u području robotike, stvarajući sveobuhvatan okvir za olakšani i ubrzani razvoj robotskih sustava. Sastoji se od seta alata, biblioteka i konvencija korištenih u razvoju programske podrške robotskih sustava .S obzirom na sveprisutnu upotrebu robota, kako u industriji tako i kućanstvima javila se potreba za standardizacijom i pojednostavljenjem složenih procesa izrade robotskih sustava. Ideja o modularnom sustavu osvanula je 2007. godine kada je Morgan Quigley, student Kalifornijskog sveučilišta Stanford sa svojim kolegama pokrenuo projekt Switchyard (hrv. *rasklopno postrojenje*). Iz tog projekta proizašao je koncept današnjeg ROS-a, tvoreći tako okolinu koja predstavlja svojevrsni predložak prilikom izrade robotskih sustava različitih namjena. ROS sustav se sastoji od velike baze paketa, alata, biblioteka te konvencija, pojednostavljujući tako razvojni proces i omogućavajući kreiranje složenih robotskih sustava bez potrebe za pojedinačnim razvojem svakog aspekta sustava. Primjena ROS-a pri izradi robotskih sustava omogućava korištenje unaprijed razvijenih i usavršenih rješenja, dok istovremeno omogućava relativno brzu prilagodbu istih prema specifičnim zahtjevima raznih zadataka.

Jedna od ključnih prednosti ROS-a kao operativnog robotskog sustava leži u modularnom pristupu, gdje svaki segment sustava može biti razvijen zasebno te na posljetku sjedinjen s ostalim sustavima kreirajući tako jednu veliku funkcionalnu cjelinu. Ovaj modularni pristup omogućuje jednostavnu integraciju različitih komponenti u upravljački sustav robota. Konceptualno, takvim pristupom je omogućena podjela zadataka među različitim stručnostima unutar tima, rezultirajući kvalitetnijim rješenjima problema te povećanom produktivnošću [7][9]. Nadalje, sljedeća prednost ROS sustava leži u bogatom izboru alata koji značajno unapređuju razvoj, dijagnostiku i upravljanje robotskim sustavima. Alati za vizualizaciju i simuliranje u ROS-u omogućavaju testiranje ponašanje robotskih sustava prije same fizičke implementacije. Ova mogućnost uvelike doprinosi smanjenju troškova i vremena razvoja, pružajući detaljan uvid u mnoge funkcionalnosti prije fizičke izrade sustava robota te samog izvođenja eksperimenata u stvarnom okruženju. Dodatno, alati za dijagnozu i analizu

pogrešaka unutar ROS-a donose okruženje za precizno praćenje i rješavanje problema tijekom procesa razvoja robotskih aplikacija. To je od ključne važnosti za optimizaciju performansi i identifikaciju grešaka u realnom vremenu, omogućujući značajno ubrzanje razvojnog ciklusa. Sučelja za procesiranje različitih tipova podataka, poput slika, laserskih podataka, podataka mjerenih inercija, oblaka točaka i drugih, predstavljaju značajnu vrijednost ROS-a. Postojanje različitih sučelja za obradu podataka omogućuje integraciju raznolikih senzorskih informacija, ključnih za kompleksno shvaćanje okoline i precizno upravljanje robotima u stvarnom svijetu. Relativno jednostavan razvoj pojedinih komponenti sustava uz korištenje koda razvijenog od strane zajednice kao i otvorenost i dostupnost zajedničkih resursa omogućavaju učinkovito korištenje postojećih implementacija, ubrzavajući time proces razvoja modularnih i skalabilnih robotskih sustava. Nadalje, ROS sustav sadrži alate korisne za rad s geometrijskim transformacijama, pridonoseći preciznom pozicioniranju i praćenju položaja robota u trodimenzionalnom prostoru. Ova sposobnost podržava integraciju različitih senzora i aktuatora, čime se postiže koherentna percepcija i kontrola robotskog sustava.

Nakon značajnog porasta zajednice koja je doprinijela razvoju sustava izradom brojnih paketa te analizom nedostataka ROS sustava stvoren je koncept unaprijeđenog, odnosno ROS2 sustava. ROS2 uvodi brojne prilagodbe koje sustavu omogućavaju rad sa zahtjevnim sustavima te raznovrsnim aplikacijama u području robotike.

2.4.2. Generalni pregled ROS i ROS2 sustava

ROS svoj rad temelji na sustavu oglašivač-pretplatnik (engl. *publisher-subscriber*), gdje čvorovi, odnosno procesi koji provode izračune izmjenjuju poruke u obliku struktura podataka putem raznih tema, odnosno kanala komunikacije. ROS sustav koristi centraliziranu arhitekturu koja podrazumijeva pokretanje glavnog sustava zaduženog za upravljanje komunikacijom između pojedinih čvorova. S druge pak strane, ROS2 nadilazi ograničenja komunikacije uvođenjem takozvane među programske apstrakcije koja sustavu omogućava korištenje raznih komunikacijskih protokola, kao što je primjerice DDS (engl. *Data Distribution Service*) [10]. Za razliku od ROS sustava, ROS2 sustav koristi decentraliziranu arhitekturu koja uklanja potrebu za centraliziranim sustavom. Promjene u arhitekturi sustava doprinose stabilnosti i robusnosti sustava uz istovremeno omogućavanje bolje skalabilnosti i primjenjivosti u raznim procesima [10]. Arhitektura ROS2 sustava s XML-RPC (engl. *Extensible Markup Language of Remote Procedure Call, hrv. proširivi označni jezik pozivne daljinske procedure*)

međuprogramskom opremom se ističe u zadacima izvođenja jednostavnijih procesa, dok nadodana kompleksnost DDS (engl. *Data Distribution Service, hrv. servis za distribuciju podataka*) međuprogramske opreme pokazuje značajnije stabilnije rješenje za upravljanje procesima u stvarnom vremenu. Nadalje, zbog svoje distribuirane prirode te izostanka centralnih elemenata sustava, DDS programska potpora uvelike doprinosi u zadacima uklanjanja kvarova nastalih uslijed gubitka komunikacije [11]. Također, kao značajna prednost ROS2 sustava je potpuna prilagođenost za rad na MacOS i Windows uz dosadašnju prilagođenost isključivo Linux operativnom sustavu.

2.4.3. Kompatibilnost i prijelaz

Jedna od glavnih stvari koje treba imati na umu pri radu s dvama verzijama sustava jest da isti nisu u potpunosti međusobno kompatibilni. Takvo što za posljedicu ima nemogućnost izravne instalacije paketa koji nisu namijenjeni korištenoj verziji. Međutim, postoje razni alati koji se koriste za povezivanje sučelja koje omogućava prijenos poruka između oba sustava ili pak naredbi ROS2 sustava koje se koriste za pokretanje čvorova ROS sustava te samim time omogućavaju zajednički rad oba sustava [10]. Nadalje, ROS u svojoj bazi sadržava dvije odijeljene knjižnice koda, odnosno roscpp koji je zadužen za C++ te rospy koji služi kao baza koda za Python programski jezik. Python i C++ u pravilu korisniku ne nude jednake značajke te se njihova primjena može značajnije razlikovati zbog potrebe nezavisnog razvoja koda za obje platforme. Za razliku od ROS sustava, ROS2 ima središnju knjižnicu pisanu u C – rcl (engl. *ROS client support library, hrv. biblioteka podrške klijentima*) uz izgradnju knjižnica na vrhu te arhitekture. Takav pristup omogućava bržu dostupnost temeljnih funkcija u različitim API (engl. *Application Programming Interface, hrv. sučelje za razvoj aplikacija*) sučeljima. Korištenje središnje knjižnice ROS2 sustava istovremeno doprinosi širenju programske podrške za jezike poput Java-e i C# [11].

Također, ROS2 rosbag sučelje omogućava veću fleksibilnost po pitanju serijalizacije u usporedbi s ROS sustavom koji u svom radu koristi vlastiti format serijalizacije. S druge pak strane, prelazak na standardne kanale komunikacije isključilo je mogućnost izravne interakcije ROS i ROS2 sustava, što je naknadno riješeno raznim paketima koji pretvaraju poruke iz jednog formata u drugi te tako do neke mjere omogućavaju istovremeno korištenje oba sustava [11].

2.4.4. Zajednica i podrška

Sljedeći bitni aspekt koji je potrebno razmotriti prije odabira jednog od dva robotska sustava je dostupnost i kvaliteta podrške zajednice. Zbog svojeg dužeg postojanja, ROS ima značajnije veću zajednicu i podršku nego ROS2, što se očituje u potpunijoj dokumentaciji, detaljnim uputama, člancima s opisom primjene, forumima koji rješavaju probleme s kojima se korisnici često susreću, pisane literature te ostale dokumentacije koja može biti od iznimne važnosti prilikom rada na određenom zadatku. Također, ROS posjeduje daleko brojniju listu dostupnih paketa, knjižnica, alata i drivera što kao posljedicu ima korištenje pojedinih paketa bez potrebe za značajnijim prilagodbama. S druge pak strane, ROS2 u posljednje vrijeme privlači mnogo pozornosti te se s velikom sigurnošću može reći da će u naredno vrijeme baza paketa redovito ažurirati te da će svaki noviji projekt ići u smjeru korištenja novije i pouzdanije tehnologije, odnosno ROS2 sustava [10].

2.4.5. Primjena

Prvi korak u procesu gradnje robotskog sustava je odabir robotskog operativnog sustava koji zadovoljava sve potrebe nekog zadatka. Primjerice, ako je cilj zadatka razviti pojedini modul ili pak hobistički robot koji ne zahtjeva visoke performanse te značajnu razinu sigurnosti i pouzdanosti, izbor će vjerojatno prevagnuti na stranu ROS sustava zbog svih prednosti koje su navedene u prethodnim poglavljima. Ukoliko je pak cilj projekta izgraditi kompleksan sustav koji će raditi s velikim količinama podataka uz istovremeno korištenje raznih platformama, izbor bi najvjerojatnije pao na ROS2 koji nudi više funkcionalnosti uz značajniju robusnost sustava [10]. Za razliku od ROS sustava, ROS2 sustav donosi značajno unaprijeđenu metodu istovremenog procesiranja velikog broja poruka i događaja istovremeno što doprinosi nesmetanom radu više paralelnih procesa bez međusobnog utjecanja na performanse sustava [12]. Unaprijeđena metoda paralelizacije kao i korištenje DDS protokola značajno doprinose korištenju ROS2 sustava kod zadataka u stvarnom vremenu koji zahtijevaju niska kašnjenja tijekom komunikacije [11].

U ovom radu je zbog nadilaženja ograničenja koja su postavljena korištenjem ROS sustava prve generacije odlučeno korištenje ROS2 sustava. Također, u vrijeme početka rada na diplomskom zadatku, ROS2 je već sadržavao većinu paketa koji su bili analizirani kao potencijalno bitni za zadatke koji je potrebno riješiti u istom. Uz to, posljednja verzija ROS

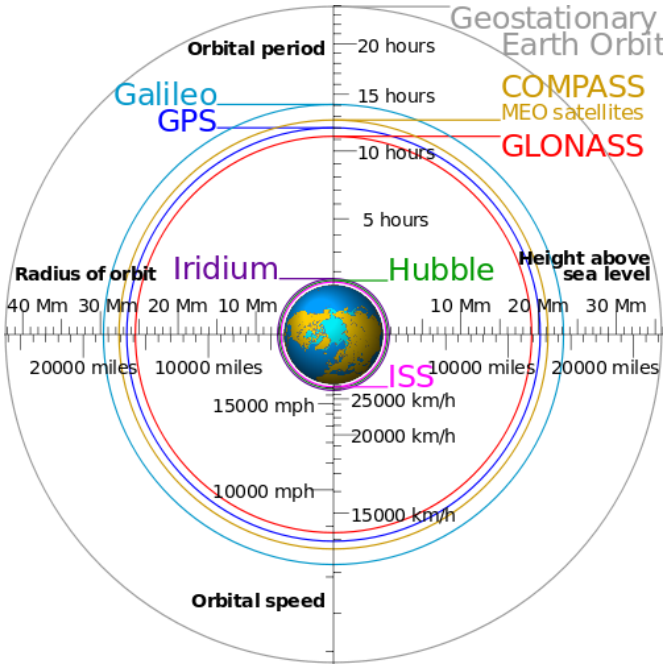
sustava ima podršku od strane ROS razvojnog tima do 2025. godine te će svako naknadno korištenje ovisiti samo o podršci zajednice, dok će se ROS2 nastaviti kontinuirano razvijati.

2.5. Implementacija GNSS sustava na sustav mobilnog robota

Implementacija GNSS komponenata u sustave mobilnih robota dodatno proširuju njihove mogućnosti, posebno u kontekstu precizne lokalizacije i navigacije. Postojanje metode precizne lokalizacije preduvjet je preciznom pozicioniranju, optimalnom planiranju putanje, autonomnoj navigaciji te samim time efikasnom obavljanju zadataka. GNSS sustav pruža mobilnim robotima visoku preciznost u određivanju njihovog globalnog položaja, čime se poboljšava lokalizacija što samim time smanjuje rizik od sudara ili nesreća nastalih pri autonomnom kretanju mobilnog robota. Također, implementacija GNSS sustava omogućava robotima planiranje optimalnih putanja na temelju zadanih kriterija, čime se može značajno povećati učinkovitost pri izvođenju zadataka kretanja kroz prostor. GNSS sustavi implementirani u mobilnoj robotici omogućuje brzo i precizno pozicioniranje u hitnim situacijama. To je od iznimne važnosti u situacijama poput spašavanja ili intervencija u opasnim okolinama, gdje druge metode lokalizacije često nije moguće provesti. U nastavku rada opisana je korištena GNSS oprema uz prikaz implementacije na mobilnog robota hodača.

3. SUSTAV GNSS NAVIGACIJE

Globalni satelitski navigacijski sustav (GNSS) omogućava određivanje točne geografske pozicije na Zemlji i u neposrednoj blizini iste uz korištenje satelitskih signala. Najpoznatiji GNSS sustavi su Američki GPS sustav, Ruski GLONASS sustav, Europski Galileo sustav te Kineski Beidou sustav. Glavni elementi svakog od globalnih navigacijskih sustava su sateliti koji emitiraju te prijemnika koji primaju signale o trenutnom vremenu očitanja atomskog sata. Prijamnici u pravilu koriste podatke tri ili više satelita kako bi odredili svoju točnu poziciju u prostoru. Globalni satelitski navigacijski sustavi se koriste u različitim područjima, kao što su navigacija vozila, plovila, aviona, precizno kartiranje, zemljopisno informacijski sustavi te mnogi drugi sustavi gdje je potrebno brzo određivanje točne lokacije na Zemlji. Svaki sustav globalne navigacije ima svoje karakteristike te prednosti nad ostalim alternativama, dok je osnovni princip rada svih sustava jednak. Sateliti iz mreže emitiraju signale koji sadržavaju informacije o satelitu, njegovom trenutnom položaju u orbiti te točno vrijeme koje označava vrijeme atomskog sata satelita iz kojeg je isti poslan. Prijemnik na Zemlji prima signale od više satelita te korištenjem istih u stvarnom vremenu računa svoju poziciju na Zemlji. Slika 2. prikazuje radijus orbite kao te orbitalni period najpoznatijih skupina satelita:



Slika 2. Ilustracija orbita najpoznatijih zemljinih satelita [13]

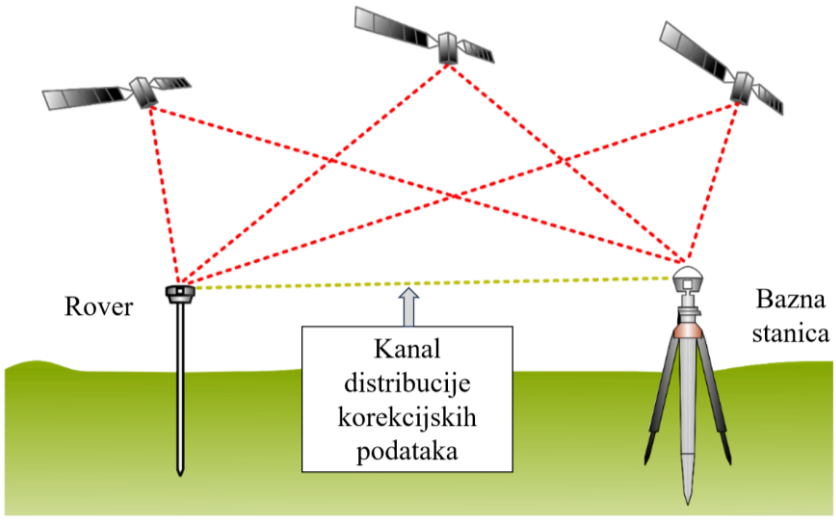
Kao tehnologija, GNSS sustavi se stalno razvijaju kako bi bili još precizniji i robusniji te kako bi se kao takvi mogli primjenjivati u područjima u kojima još uvijek nisu zastupljeni zbog svojih

trenutnih nedostataka. U novije vrijeme su svjetlo dana ugledali i neki noviji sustavi koji se trenutno u fazi razvoja, kao što su QZSS (Japanski sustav) i NavIC (Indijski sustav).

3.1. Konfiguracija RTK sustava

Sustav kinematike u stvarnom vremenu (engl. *Real Time Kinematic,* RTK) vrsta je GNSS sustava koji se koristi za precizno pozicioniranje prijamnika. Kao tehnologija koristi se kako bi se povećala preciznost konvencionalnih GNSS prijamnika čija se preciznost u normalnim radnim uvjetima rasipa unutar nekoliko metara. Tehnologija RTK sustava koristi se u raznim područjima poput geodezije, kartografije, graditeljstva, poljoprivrede, robotike te drugih područja gdje je potrebna subcentimetarska preciznost globalno određene pozicije.

RTK sustav se sastoji od tri glavna dijela: rovera, baze i komunikacijskog kanala. Baza je fiksni uređaj koji prima signale sa satelita te generira podatke koji se koriste za kompenzaciju poremećaja signala u atmosferi uzrokovane lomovima u sloju ionosfere. Bazni prijamnik se postavlja na mjesto kojemu je poznata točna lokacija, bilo određena nekim drugim nezavisnim uređajem ili određena raznim matematičkim metodama, poput metoda analiza prirode rasipanja tijekom duljeg perioda koje se potom uklanja iz mjerenja kako bi se dobila točna lokacija. Tijekom određivanja svog položaja, baza uspoređuje svoju poznatu lokaciju s lokacijom očitanom iz primljenih satelitskih signala te prema njihovoj razlici stvara podatke koji se potom koriste kao korekcijski podaci za prijamnike u neposrednoj blizini s bazom. Slika 3. prikazuje GNSS RTK sustav baze i rovera:



Slika 3. Prikaz GNSS RTK sustava baze i rovera [14]

Rover je mobilni dio RTK sustava koji prima signale sa satelita te ujedno koristi informacije dobivene od baze s pomoću kojih se ostvaruje subcentimetarsko pozicioniranje. Uređaj koristi signale dobivene od satelita kako bi približno odredio svoju poziciju, dok uz pomoć signala dobivenih od baze sustav provodi korekciju atmosferskih utjecaja te samim time postiže željenu točnost. Bazna stanica se kod RTK sustava koristi za generiranje popravaka (eng. *corrections*) GNSS signala koje prima. Rover i baza su u neprestanoj komunikaciji koja se najčešće ostvaruje putem interneta ili radio vezom. Komunikacijski kanal roveru u stvarnom vremenu osigurava korekcijske podatke te samim time sustav komunikacije mora biti iznimno robustan i stabilan u radnom prostoru robotskog sustava kako bi se mogao iskoristiti puni potencijal RTK sustava.

3.2. IZBOR BAZNE STANICE

Prvi korak u procesu izrade RTK sustava je postavljanje sustava dobave korekcijskih signala, odnosno sustava bazne stanice. Kao što je detaljnije opisano u Projektu MiR [3], baznu stranicu moguće je konfigurirati na više načina. Prvi način je postavljanje privremene bazne stanice koju karakterizira relativno brzo i jednostavno postavljanje te je odmah spremna za korištenje. Nedostatak privremenih baznih stanica je manja preciznost u odnosu na ostale dvije metode. Drugi način je postavljanje stacionarne bazne stanice čije su karakteristike dugotrajni proces inicijalizacije dok im je glavna prednost osiguravanje najveće preciznosti koju takvi sustavi mogu isporučiti. Treći način dobave korekcijskih signala nije vezan uz korisničku baznu stanicu, već podrazumijeva povezivanje na lokalnu mrežu repetitora koji distribuiraju korekcijske podatke, što najčešće sa sobom nosi potrebu za pretplatom na spomenute usluge.

Pri procesu konfiguracije baze, važno je detaljno obraditi četiri glavna koraka: planiranje, postavljanje, konfiguraciju i validaciju rezultata. U koraku planiranja odabiru se komponente potrebne za željene performanse sustava te mjesto za postavljanje baze uz detaljnu analizu okolišnih uvjeta koji mogu utjecati na točnost mjerenja. Sljedeći korak, odnosno korak postavljanja, uključuje postavljanje antene GNSS prijamnika na odabrano mjestu te povezivanje antene s prijemnikom. Glavni cilj ovog koraka je osigurati fiksnu poziciju antene u prostoru kako bi baza bila sposobna osigurati što točnije korekcijske podatke mobilnim uređajima u neposrednoj blizini. Nakon uspješnog postavljanja opreme potrebno je izvršiti konfiguraciju postavljene opreme, odnosno prijemnika s ciljem omogućavanja povezivanja s drugim GNSS uređajima koji su namijenjeni za izvršavanje određenih zadataka. Ovaj korak obično uključuje postavljanje IP adrese baze te konfiguriranje protokola prijenosa podataka ili

spajanje s uređajem za prijenos telemetrijskih podataka radiovalovima. Posljednji korak pri konfiguraciji baze je validacija rezultata. Nakon postavljanja opreme i provedene konfiguracije, potrebno je izvršiti ispitivanja dobivenih korekcijskih podataka. Iste je moguće validirati ukoliko se poznaje lokacija na kojoj se smješta baza. Drugi način validacije ispravnosti dobivenih korekcijskih datoteka je mjerenje nekoliko različitih točaka uz pomoć prijamnika koji prima korekcijske datoteke od postavljene stanice te potom provođenje analize snimljenih točaka i usporedba rezultata sa stvarnih točkama kako bi se ocijenila kvaliteta postavljene bazne stanice. Druga metoda validacije ispravnosti dobivenih korekcijskih podataka prikazana je u nastavku odlomka.

3.2.1. Integrirani sustavi baze i rovera

Za razliku od prethodno spomenutih sustava, integrirani sustav rovera predstavlja spoj svih komponenti potrebnih za postizanje visoke razine točnosti uz korištenje GNSS signala, zbog čega je i odabran kao komponenta u izradi diplomskog rada. Integrirani sustavi se najčešće sastoje od GNSS RTK pločice, LiPo (engl. *Lithium Polymer, hrv. polimer litija*) baterije, modula za spajanje, antena te mikroupravljača koji se koristi za upravljanje sustava te se može jednostavno modificirati korištenjem Arduino IDE ili srodnih sučelja. Korištenjem Bluetooth modula otvara se široki spektar mogućnosti nakon spajanja GNSS uređaja s pametnim telefonom ili računalom gdje pametni telefon može služiti kao distributer korekcijskih podataka bazne stanice ili za promjenu parametara GNSS-RTK sustava u stvarnom vremenu, bez potrebe za fizičkim povezivanjem GNSS uređaja s Arduino sučeljem mikroupravljača prijamnika. Kao ishod Projekta MiR [3] za RTK sustav odabrana su dva GNSS Faceta kompanije SparkFun [15].

Slika 4. SparkFun RTK Facet integrirani sustav [15]

Slika 4. prikazuje RTK Facet, GNSS uređaj koji može raditi u četiri načina rada: GNSS pozicioniranje kao samostalni Rover, GNSS pozicioniranje uz korištenje korekcijskih RTK podataka, samostalna bazna stanica te NTRIP bazna stanica. Ukoliko se RTK Facet koristi kao samostalni Rover, apsolutna točnost koju isti može postići iznosi 300 mm. Uz korištenje RTK podataka Facet postiže točnost od 14 mm. Facet se također može koristiti kao bazna stanica te koristiti za izračun i slanje korekcijskih podataka do RTK uređaja iz okoline putem radio signala ili interneta. U odlomku *4.8.1. Konstrukcijska rješenja implementacije sustava* prikazan je SparkFun Facet prijamnik konfiguriran kao rover, spreman za montiranje na mobilnog robota.

Zbog svoje robusnosti, jednostavnosti postavljanja, široke mogućnosti primjene te izostanka potrebe za pretplatom na razne servise za dobavu korekcijskih podataka, integrirani sustavi baze i rovera su vrlo često primjenjivi u slučajevima kada je potrebno precizno određivanje pozicije u zadacima vezanim za sustave mobilnih robota. Slika 5. prikazuje SparkFun RTK Facet konfiguriran kao baznu stanicu, fiksno montiran na poziciju s nesmetanim pogledom na nebo koristeći stalak kamere:

Slika 5. SparkFun RTK Facet u ulozi bazne stanice postavljena

3.3. Komunikacija RTK opreme i ROS2 sustava

Komunikacija rovera i upravljačkog računala moguće je ostvariti putem SPI, UART, I2C i USB priključka. Zbog jednostavnosti i pokazane robusnosti sustava rover i upravljačko računalo mobilnog robota komuniciraju korištenjem serijske komunikacije preko USB priključka. Komunikacija rovera i baze sa svrhom prijenosa korekcijskih podataka provodi se putem radio telemetrijskih odašiljača kompanije Holybro [16] na frekvenciji od 915 MHz, prikazanih na slici 7:

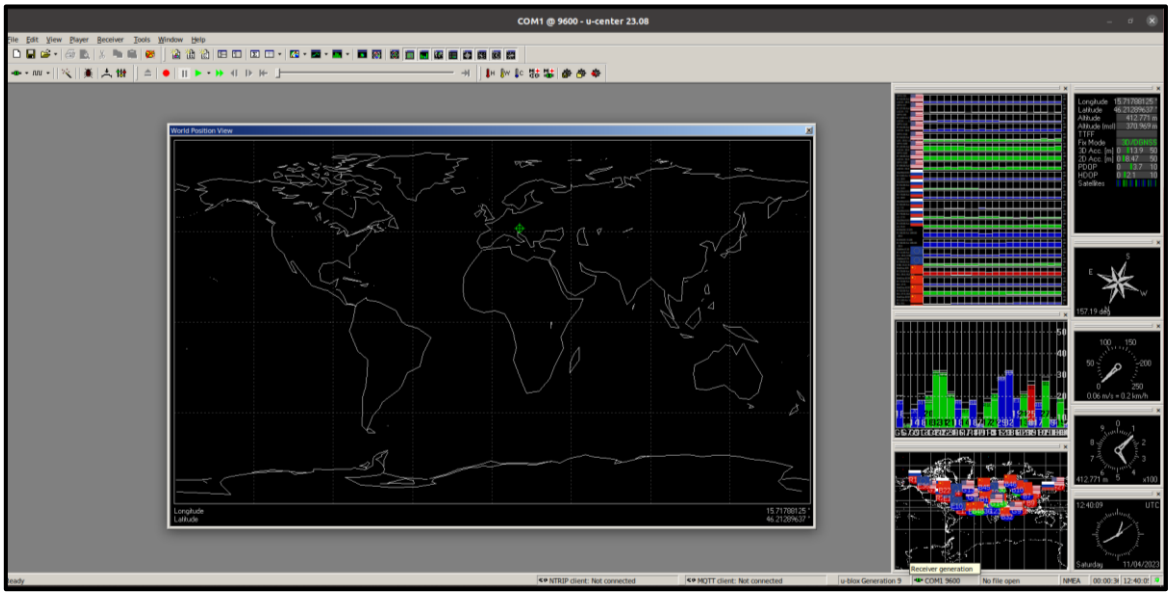
Slika 6. Holybro odašiljač/prijamnik za slanje korekcijskih podataka između baze i rovera

Korištenjem prikazanih radio telemetrijskih odašiljača moguće je postići komunikaciju baze i rovera udaljenih do 300 metara bez dodatno napajanja dok se uz dodatno napajanje može postići komunikacija i do desetak kilometara, što se u literaturi navodi kao gornja granica kvalitetnih korekcijskih podataka.

3.3.1. Ublox čvor

U-blox je Švicarska kompanija koja svoj rad temelji na proizvodnji i razvoju GNSS sustava te komponenata bežične komunikacije. Zbog svoje robusnosti i sveopće primjene izrađen je ROS2 ublox paket [18] koji je u ovom radu korišten za prijenos signala s mikroupravljača

GNSS prijamnika na upravljačko računalo mobilnog robota te za dodatnu konfiguraciju sustava globalne lokalizacije. Slika 7. prikazuje u-center korisničko sučelje Ublox čvora:



Slika 7. u-center korisničko sučelje

Korisničko sučelje ublox čvora pruža najbitnije informacije za korištenje spomenute opreme, kao što je broj satelita s kojima prijamnik ima interakciju te rasipanje rješenja pomoću kojeg se računa preciznost samog sustava. Čvor istovremeno daje pregled putanja satelita kao i grafičke prikaze trajektorija uređaja uz prikaz podataka o rasipanju rješenja iz kojih je potom moguće odrediti pouzdanost korištenog sustava. Također, putem ublox čvora mogu se podesiti pretplate na korekcijske podatke kao i sva ostala podešavanja koja je nužno provesti pri postavljanju RTK sustava. Dodatne informacije o u-center sučelju kao i moguće konfiguracije sustava moguće je pronaći u ublox službenoj dokumentaciji [19].

Glavna zadaća ublox čvora na ROS2 sustavu je pokretanje u-center sučelja koje prilagođenog korištenoj opremi. U datoteci za pokretanje čvora potrebno je odabrati konfiguracijsku datoteku u kojoj se nalaze osnovni podaci korištene opreme kao što su brzina prijenosa podataka, naziv priključka, naziv novoizrađenog koordinatnog sustava te ostali podaci koje je potrebno prilagoditi prilikom postavljanja sustava. Pokretanjem čvora započinje se proces slušanja odabranih ulaznih priključaka kao i interpretacija očitanih podataka. Poruka o globalnoj poziciji dobivena od ublox čvora sastoji se od podataka o geografskoj širini, dužini, visini i kovarijanci određene pozicije s pomoću kojih se određuje razina povjerenja pojedinoj komponenti koju daje prijamnik. Također, ublox čvor provodi derivaciju podataka o poziciji na temelju kojih se određuje brzina kretanja prijamnika kroz prostor. Kao i kod poruka o poziciji,

poruke brzine također posjeduju parametar kovarijanci kojima se određuje pouzdanost pojedinih komponenti objavljivane teme brzine. U nastavku je prikazan primjer teme lokacije korištene pri globalnoj lokalizaciji robota:

```
header:
    stamp:
        sec: 1704985587
        nanosec: 305943
    frame_id: gps
status:
    status: 0
    service: 1
latitude: 45.7953981
longitude: 15.9729367
altitude: 163.472
Position_covariance:
- 5.262436
- 0.0
- 0.0
- 0.0
- 5.262436
- 0.0
- 0.0
- 0.0
- 10.336224999999999
position_covariance_type: 2
```

3.3.1.1. Parametri ublox_gps_node čvora

Nadalje, potrebno je provesti konfiguraciju ublox_gps_node čvora. To podrazumijeva podešavanje parametara kako bi odgovarali konfiguraciji korištene GNSS opreme. Pri postavljanju ublox_gps_node čvora potrebno je definirati sljedeće parametre:

- *debug* – Parametar u rasponu od nula do četiri koji označava stupanj ispisa poruka za otklanjanje poteškoća gdje se nula odabire kada se poruke ne žele primati dok se četiri odabire za ispis svih poruka.
- *device* – Parametar kojim se odabire priključak na kojemu se nalazi GNSS uređaj.

- *frame_id* – Parametar kojim se određuje naziv koordinatnog sustava kojim se prikazuje GNSS uređaj.
- *uart1*
    - *baudrate* – Parametar kojim se određuje brzina prijenosa podataka kojom GNSS uređaj šalje informacije putem odabranog priključka.
- *tmode3* – Parametar kojim se odabire moda rada sustava. U biblioteci ublox paketa [18] nalazi se detaljniji opis konfiguracijskih parametara s pomoću kojih se odabire pojedini mod sustava.
- *sv_in.reset* – Parametar kojim se omogućava ponovno postavljanje GNSS uređaja prilikom novog pokretanja čvora.
- *sv_in.min_dur* – Parametar minimalnog trajanja određivanja pozicije do početka objavljivanja iste u uvjetima lošijeg prijema signala.
- *sv_in.acc_lim* – Parametar određivanja točnosti sustava. Nakon što sustav postigne definiranu točnost, čvor započinje objavljivanje podataka o globalnoj poziciji.
- *inf.all* – Parametar prikaza svih poruka o lokaciji u terminalu sustava u kojemu se čvor poziva.
- *publish.all* – Parametar kojim omogućava slanje svih tema na teme koje su definirane čvorom

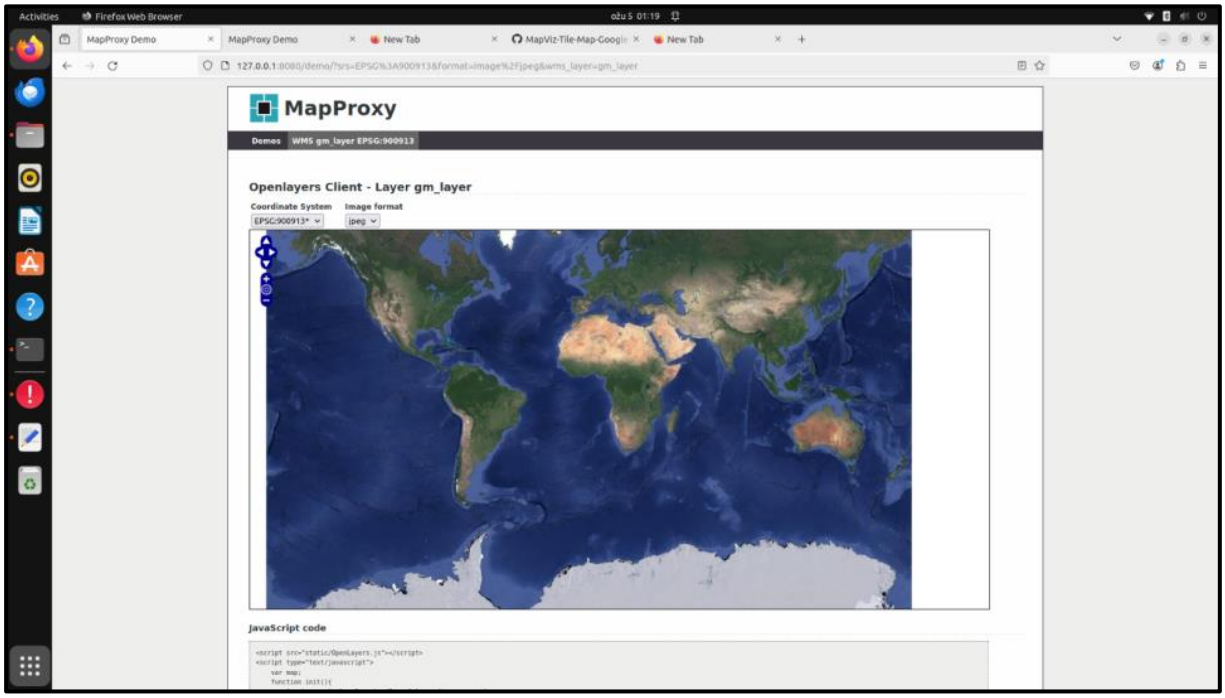
Detaljnu lista parametara za konfiguraciju ublox_gps_node čvora moguće je pronaći unutar datoteka programskog koda paketa [18].

3.3.2. Serveri zaduženi za prikaz satelitskih snimaka

Kako bi se omogućio prikaz terena u Mapviz okruženju potrebno je osigurati vezu sa servisima za dobavu satelitskih snimaka. Kao što je spomenuto u odlomku *3.3.3. Mapviz paket* u ovom radu korišteni su poslužitelji distribucije satelitskih snimaka [20] koji koriste besplatne MapProxy Python servise [21] putem kojih se ostvaruje povezivanje podataka Google Maps satelitskim snimaka u Mapviz okruženje. Podaci se zatim šalju servisima distribucije kartografskih podataka u obliku pločica zaduženih za prikaz karata na web stranicama *(*engl. *Web Map Tile Service,* WMTS) koji mogu biti izravno prikazani u ROS2 okruženju putem dodataka opisanih u nastavku. Servis također omogućava korištenje prethodno očitanih karata bez potrebe pristupa mreži.

3.3.3. Mapviz paket

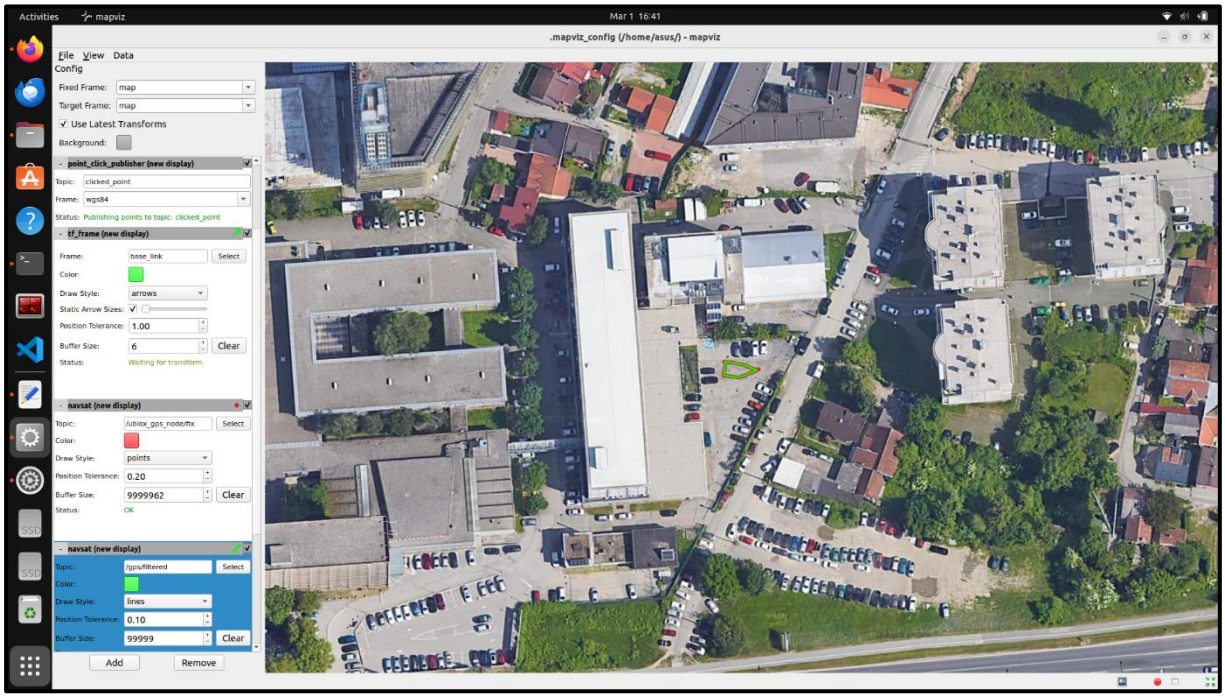
Mapviz paket [22] je visoko prilagodljiv ROS2 alat koji je pogodan za vizualizaciju velikih količina 2D podataka uz mogućnost dodavanja raznih priključaka (engl. *plugins*). Za razliku od Rviz okruženja koje se najčešće koristi za manja područja i unutrašnju upotrebu, Mapviz okruženje je pogodno za vizualizaciju velikih područja s pomoću kojeg je moguće prikazati i mapu svijeta. Kako bi se omogućio prikaz terena u Mapviz okruženju, na računalu se pokreće server [20] koji vrši dobavu slika terena. Slika 8. prikazuje server distribucije satelitskih snimaka temeljen na geografskoj širini, dužini te nadmorskoj visini:



Slika 8. Testiranje rada servera distribucije satelitskih snimaka

Serveru se potom prosljeđuju podaci o trenutnoj poziciji na Zemlji korištenjem Tile_map_plugin dodatka unutar Mapviz sučelja. Sučeljem se omogućava korištenje brojnih značajki kao što su prikaz lokacija većeg broja GNSS prijamnika, smjerova kretanja, trajektorija dobivenih od poslužitelja (engl. *servers*) zaduženih za planiranje putanja i slično. Također, putem grafičkog sučelja moguće je prikazati aktivne koordinatne sustave koji postoje u ROS2 okolini uz mogućnost određivanja relativnih položaja pojedinih točaka na mapi u odnosu na odabrani koordinatni sustav. Time se moguće kreirati razne poligone koji se potom mogu izvoziti u obliku tekstualne datoteke ili pak izravno proslijediti čvorovima navigacije putem raznih čvorova. Sučelje također omogućava prikaz raznih očitanja senzora, ruta, transformacija i ostalih značajki što je detaljnije opisano unutar datoteka programskog koda

paketa [22]. Slika 9. prikazuje Mapviz sučelje pretplaćeno na server distribucije snimaka terena iz zraka:



Slika 9. Prikaz Mapviz sučelja

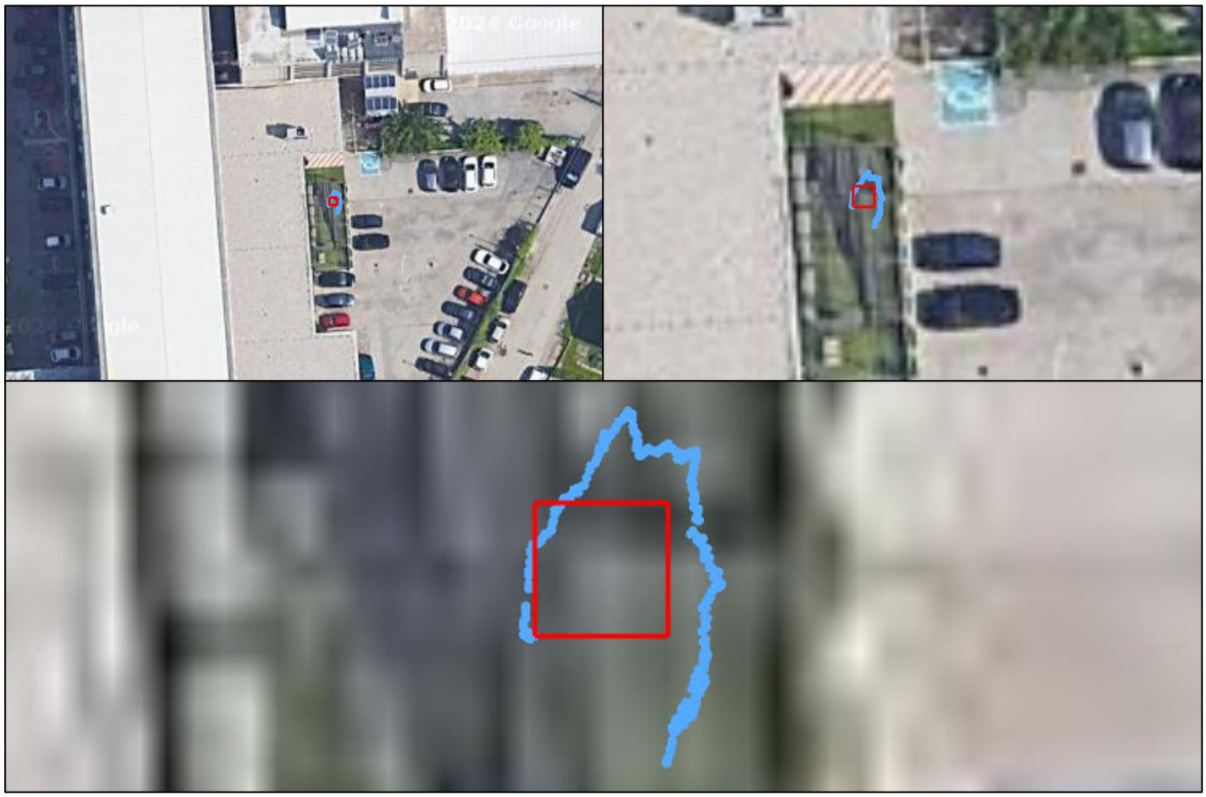
3.3.3.1. Parametri mapviz_node čvora

Nadalje, potrebno je provesti konfiguraciju mapviz_node čvora. To podrazumijeva podešavanje parametara kako bi odgovarali željenoj konfiguraciji sučelja za prikaz satelitskih snimaka. Pri postavljanju mapviz_node čvora potrebno je definirati putanju do prethodno podešene konfiguracije Mapviz sučelja te koordinate ishodišta novo učitanog terena. Prikaz konfiguracijske datoteke mapviz_node čvora moguće je pronaći unutar datoteka programskog koda paketa [22].

3.4. Testiranje rada RTK sustava

Prije implementacije RTK sustava testiran je rad GNSS prijamnika u ulozi rovera bez korekcijskih podataka. Slika 10. plavom bojom prikazuje podatke o lokaciji dobivene od samostalnog rovera, odnosno sustava prije implementacije RTK korekcijskih podatka snimane tijekom perioda u trajanju od 60 sekundi. Također, u projektu MiR [3] detaljno je opisana mogućnost samostalnog korištenja Sparkfun Faceta uz opis tehnika kojima se uklanja rasipanje rezultata koje je svojstveno GNSS prijamnicima pri takvom korištenju. Iako je prijamnik stajao

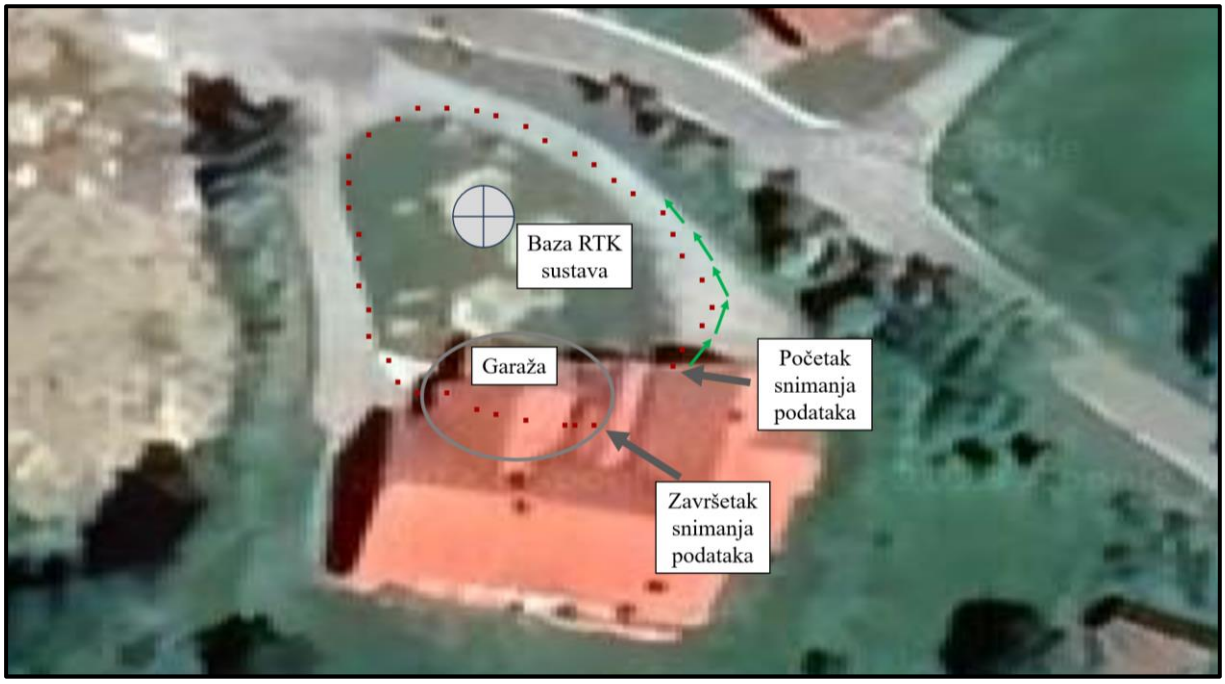
na mjestu, vidljiv je značajni pomak vrijednosti tijekom vremena uzrokovan nedostatkom RTK funkcionalnosti kao i neposrednom blizinom objekata, bez obzira na to što je uređaj imao nesmetani pogled na nebo. S ciljem stvaranje slike o redu veličine prikazan je crveni pravokutnik stranica veličina jednog metra. Takvo rasipanje podataka može imati negativan utjecaj u zadacima pozicioniranja robotskih sustava od kojih se očekuje visoka točnost i preciznost.



Slika 10. Rasipanje podataka globalne pozicije u slučaju korištenja rovera bez primjene korekcijskih podataka

Provođenjem testa koji je prikazan u nastavku rada provjerena je vjernost prikaza lokalizacijskog sustava uz korištenje korekcijskih podataka RTK sustava. Slika 11. i Slika 12. prikazuju testiranje korištene GNSS-RTK opreme:

Slika 11. Rezultat testiranja RTK opreme – opći prikaz



Slika 12. Rezultat testiranja RTK opreme – detaljniji prikaz

Slika 12. prikazuje položaj smještene baze RTK sustava, lokaciju početka kao i završetka snimanja putanje. Tolerancija pozicije postavljena je na dva metra što znači da će točkicom biti prikazana svaka promjena koja po euklidskoj udaljenosti iznosi više od tog iznosa. Na prvoj

polovini testa kretanje se odvijalo što više sredinom asfaltirane ceste, što je prilično vjerno prikazano provedenim testom. Na drugoj polovini testa kretanje se odvijalo rubom asfalta kako bi se dodatno validirala točnost podataka RTK sustava uparenog sa satelitskim snimkama. U oba slučaja se prikaz vrlo dobro preklapa sa stvarnim kretanjem iz čega se može donijeti zaključak da su satelitske snimke i lokacija koju daje RTK sustav u ispravnom međusobnom odnosu. Pred kraj testa prikazan je ulazak u garažu u kojoj je GNSS sustav rovera, zbog ometanog pogleda na nebo, mogao pristupiti podacima samo 14 satelita. Iako se broj satelita koji prate Rovera prepolovio, rover je i dalje emitirao ispravnu poziciju gdje mu je rasipanje iz prethodno spomenutih 14 mm poraslo na odstupanje većeg reda veličine, odnosno ~ 100 mm, što je za sustav bez IMU senzora koji se kreće u zatvorenom prostoru i više nego zadovoljavajuće. Ta spoznaja je vrlo bitna za daljnji nastavak korištenja podataka o globalnoj lokalizaciji jer će sve lokacije i poligoni biti definirani korištenjem uparenih satelitskih snimaka. Također valja napomenuti da se tijekom cijelog snimanja podataka prijamnik kretao brzinom ljudskog hoda, odnosno otprilike 1 m/s.

Iz provedenog testa vidljivo je da se pozicije dobivene RTK sustavom relativno dobro poklapaju sa stvarnom putanjom te da tijekom provedenog testa nije bilo neočekivanih skokova pozicije, koji u pravilu karakteriziraju GNSS sustave. Iz toga se može zaključiti da je testirani RTK sustav određivanja pozicije moguće koristiti za potrebe lokalizacije mobilnog robota u zadacima globalne navigacije, što je detaljnije opisano u nastavku rada.

4. BOSTON DYNAMICS SPOT

Spot je mobilni robot hodač razvijen od strane američke kompanije Boston Dynamics [23]. Zbog svojih tehničkih karakteristika postao je istaknut u mnogim područjima gdje postoje zahtjevi kombinacije mobilnosti, autonomije te sposobnosti manipulacije raznim predmetima. U industrijskom okruženju, Spot se često koristi pri rutinskim inspekcijama postrojenja pri održavanju optimalnog rada opreme [24]. Jedna od ključnih značajki Spota je njegova iznimna agilnost koja se postiže korištenjem strukture zatvorenog kinematskog lanca koja mu omogućuje brzo i precizno kretanje po različitim terenima. Modularna konstrukcija doprinosi prilagodbi u raznim situacijama, dok mu njegova sposobnost samostalnog kretanja omogućuje pristup teško dostupnim ili opasnim terenima bez izlaganja ljudi rizicima. Zahvaljujući modularnosti i relativno jednostavno proširivom električnom sučelju, omogućeno je opremanje robota dodatnom opremom poput raznih senzora, kamera, svjetala ili drugih uređaja na za to predviđena mjesta na leđima robota. Visoka konfigurabilnost omogućava prilagodbu robota za različite zadatke, uključujući istraživanje okoline, nadzor, inspekciju industrijskih postrojenja, ili bilo koju drugu primjenu gdje je potrebna mobilnost i mogućnost opremanja različitim senzorima ili alatima za izvršavanje određenih zadataka. Primjerice, u graditeljstvu, Spot se pokazao kao koristan uređaj za praćenju napretka gradilišta [24]. Periodičkim snimanjem visokokvalitetnih slika i 3D modela okoline, Spot pruža informacije koje mogu uvelike pripomoći u planiranju i vođenju građevinskih projekata [24]. Također, provođenjem periodičkog snimanja mogu se na vrijeme uočiti eventualne nepravilnosti ili oštećenja na objektima, doprinoseći tako povećanju kvalitete samih projekata [25]. U sigurnosnim situacijama, Spot se koristi prilikom istraživanja opasnih područja, poput ruševina oštećenih uslijed potresa, požara ili eksplozija, pružajući relevantne informacije koje olakšavaju pravovremeno donošenje odluka važnih po ljudsko zdravlje. Njegova sposobnost brze reakcije i prilagodbe dinamičkom okruženju čini ga korisnim alatom u hitnim situacijama [25]. Brojne istraživačke zajednice također su prepoznale potencijal Spota korištenog u proučavanju prirodnih staništa i terena koji su teško dostupni ljudima. Spot se također može koristiti za praćenje divljih životinja te za sakupljanje uzoraka tijekom ekoloških istraživanja [25]. Fleksibilnost u postavljanju dodataka čini Spota svestranim i prilagodljivim rješenjem u raznim domenama.

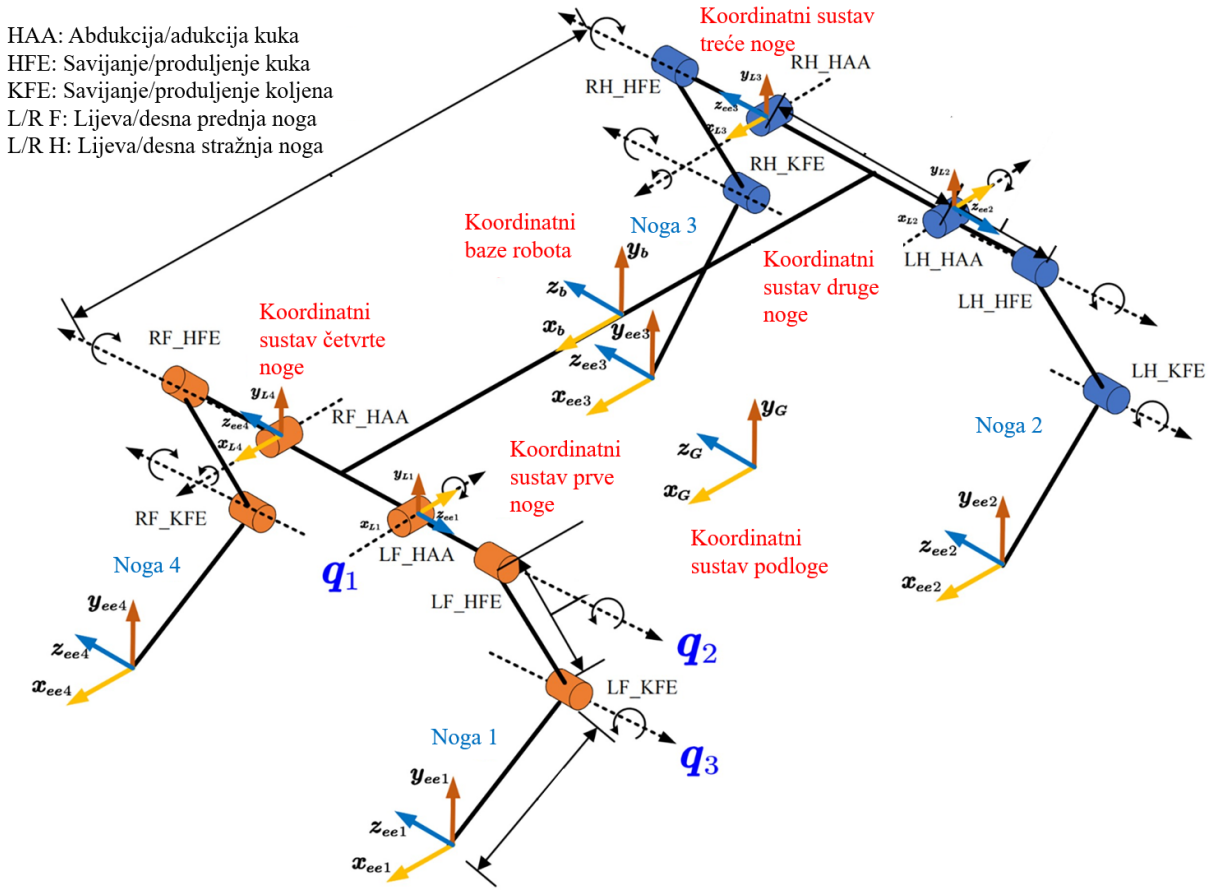
Sva potencijalno podobna područja primjene spomenuta u prethodnom dijelu odlomka ukazuju na to kako Spot, kao predstavnik novije generacije mobilnih robota zbog svoje sposobnosti kretanja uvelike doprinosi rješavanju složenih izazova u različitim sektorima. Također, Spot istovremeno doprinosi poboljšanju učinkovitosti i sigurnosti operacija koje se odvijaju u stvarnom svijetu. Slika 13. prikazuje Spot robota u CRTA okruženju:

Slika 13. SPOT robot u CRTA okruženju

4.1. Kinematika i dinamika Spot mobilnog robota

Kinematika i dinamika mobilnih robota hodača predstavljaju složene aspekte u području robotike. Sustavi robota hodača karakteriziraju sposobnost izvođenja širokog spektra pokreta, uključujući hodanje, okretanje te interakciju s okolinom. U kontekstu analize ponašanja robota, kinematika se odnosi na proučavanje geometrije kretanja, dok se dinamika bavi analizom sila i momenta koji utječu na kretanje robota. Analizom kinematike i dinamike složenih sustava mobilnih robota hodača s 12 rotacijskih stupnjeva slobode omogućava se bolje razumijevanje njihovih sposobnosti čime se pridonosi razvoju naprednih sustava autonomnog kretanja robota.

Također, valja naglasiti da prema kinematičkoj strukturi Spot mobilni robot hodač predstavlja zatvoreni kinematički lanac gdje su noge međusobno povezane preko podloge, uz zanemarivanje proklizavanja. Takva struktura značajno povećava točnost i ponovljivost pokreta koje je moguće ostvariti s istim uz postizanje odometrije visoke točnosti. Slika 14. prikazuje koordinatne sustave i zglobove korištene prilikom opisa kinematike četveronožnih mobilnih robota hodača:



Slika 14. Prikaz koordinatnih sustava i zglobova četveronožnog mobilnog robota hodača [26]

Detaljan raspis direktne i inverzne kinematike kao i dinamike četveronožnog robota hodača uz prikaz raznih mehanizama i topologija hodanja dan je u [26].

4.2. Specifikacije Spot mobilnog robota

Konstrukcija Spot mobilnog robota temelji se na pažljivoj analizi biomehaničkih karakteristika psa, što rezultira vjernim oponašanjem pokreta ovih četveronožnih životinja. Spot tijekom svog gibanja replicira hod psa koristeći visoko precizne elektromehaničke aktuatore koji su povezani s nogama. Sloboda kretanja svakog od aktuatora omogućava mu

ostvarivanje složenih pokreta kojima se postiže iznimna prilagodljivost raznim terenima. Tablica 1. prikazuje osnovne specifikacije Spot mobilnog robota hodača:

Tablica 1. Specifikacije Boston Dynamics Spot robota hodača

Dužina robota	1100 mm
Širina robota	500 mm
Maksimalna visina pri hodanju	700 mm
Minimalna visina pri hodanju	520 mm
Masa robota s baterijom	32,7 kg
Broj stupnjeva slobode gibanja	12 (tri rotacijska zgloba po svakoj nozi)
Zaštita od prodora krutih tijela i tekućina	IP 54
Radna temperatura okoliša	-20 °C do 45 °C
Maksimalna visina koraka na ravnom terenu	300 mm
Kapacitet baterije	564 Wh
Maksimalni napon baterije	58,5 V
Propisano tipično vrijeme rada	90 min
Maksimalna snaga punjenja baterije	400 W
Maksimalna struja punjenja baterije	7 A
Percepcija okoline	pet stereo kamera sa dodatnim dubinskim senzorima
Raspon dubinske slike: minimalna/maksimalna udaljenost	0,2/10 m

Detaljnu listu specifikacija Spot mobilnog robota moguće je pronaći u tehničkoj dokumentaciji robota [27].

4.2.1. Određivanje brzina kretanja Spot mobilnog robota

Određivanje postizivih brzina robota igra ključnu ulogu u određivanju njegovog sigurnog i učinkovitog kretanja. Tablica 2. pruža informacije o minimalnim i maksimalnim brzinama kretanja robota unaprijed i unatrag kao i rotaciji oko svoje osi, čime se definira opseg izvedivih brzina robota. Određivanje graničnih iznosa brzina od velikog je značaja pri postavljanju upravljačkih poslužitelja jer omogućuju precizno prilagođavanje parametara kako bi se osiguralo stabilno i točno kretanje robota. Iznosi ostvarivih brzina određeni su validacijom na robotu.

Tablica 2. Prikaz graničnih brzina Boston Dynamics Spot mobilnog robota hodača

Maksimalna brzina uzduž x osi, odnosno maksimalna brzina kretanja unaprijed/unatrag.	1,6 m/s
Minimalna brzina uzduž x osi, odnosno minimalna brzina kretanja unaprijed/unatrag.	0,15 m/s
Maksimalna brzina uzduž y osi, odnosno maksimalna brzina kretanja u stranu.	1,2 m/s
Minimalna brzina uzduž y osi, odnosno minimalna brzina kretanja u stranu.	0,15 m/s
Maksimalna kutna brzina oko z osi, odnosno maksimalna brzina rotacije robota oko svoje osi.	1,1 rad/s
Minimalna kutna brzina oko z osi, odnosno minimalna brzina rotacije robota oko svoje osi.	0,22 rad/s

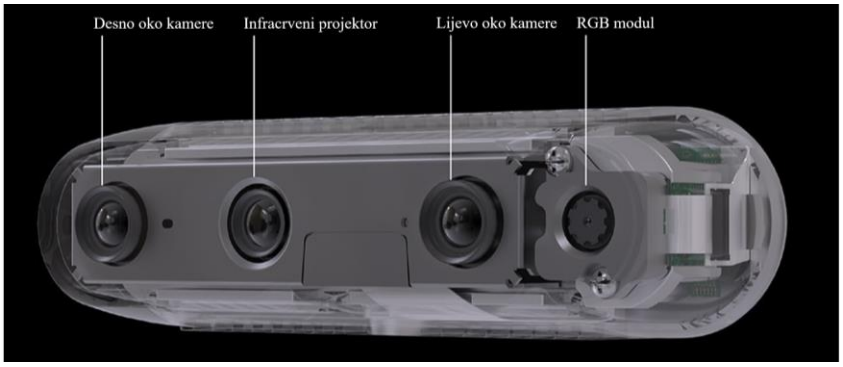
Informacije o granično izvedivim brzinama pružaju temelj za prilagodbu upravljačkih poslužitelja, omogućujući precizno definiranje dinamičkog ponašanja robota. Kroz ove granične brzine, sustav upravljanja može biti optimiziran za različite zadatke, bilo da je riječ o preciznom pozicioniranju, agilnom manevriranju pri zaobilaženju prepreka ili navigaciji kroz složeno okruženje. Detaljni opisi prilagodbi upravljačkih poslužitelj dan je pri podešavanju upravljačkih poslužitelja u odlomku *6.2.2. Postavljanje poslužitelja globalne navigacije*. Pošto se upravljanje robotom vrši putem referenci brzina, Spot sadržava implementiran regulator kruga ubrzanja što uklanja potrebu definiranja graničnih iznosa izvedivih ubrzanja.

4.3. Senzorika integrirana na mobilnom robotu

Spot je opremljen nizom senzora, uključujući kamere visoke razlučivosti, senzore dubine te senzore mjerenja inercije (IMU) koji Spotu omogućava stvaranje detaljne mape okoline. Korištenjem integriranog softvera za obradu podataka omogućava se autonomno prepoznavanje prepreka, navigaciju kroz kompleksna okruženja kao i donošenje odluka u stvarnom vremenu. Implementacija takvog softvera za posljedicu ima uspješno kretanje kroz prostor što ga čini sposobnim za interakciju s dinamičkom okolinom i izvršavanje različitih zadataka s visokom pouzdanošću. Njihova integracija omogućuje robotu stvaranje i održavanje sveobuhvatne svijesti o dinamičkoj okolini, što je ključno za njegovu autonomiju i prilagodljivost. U nastavku odlomka opisane su osnovne komponente integrirane na robotu čija je zadaća stvaranje detaljne slike o okolini robota.

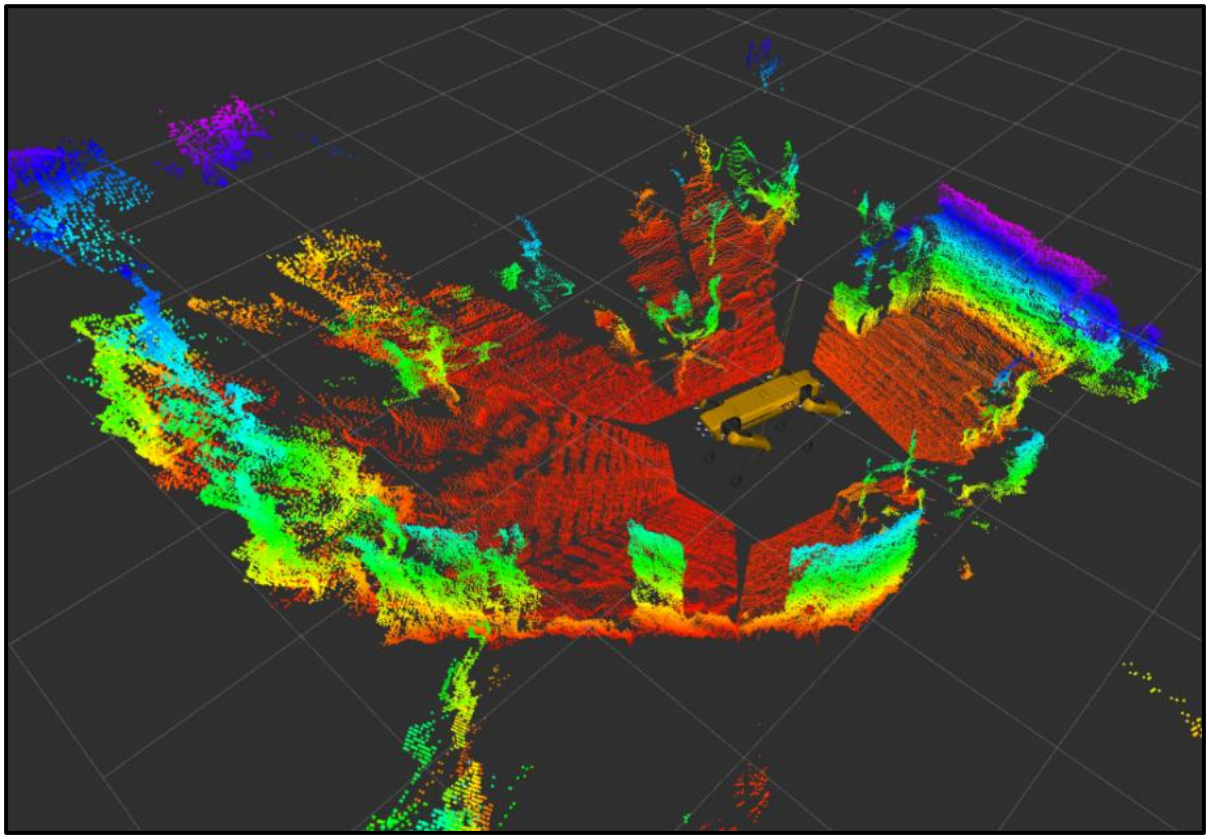
4.3.1. Stereo kamere

Spot u serijskoj izvedbi dolazi s prethodno ugrađenih 5 stereo kamera od kojih su dvije smještene na prednju stranu, po jedna kamera na svakoj strani te jedna na stražnjoj strani robota. S pomoću stereo kamera ostvaren je binokularni vid, sličan ljudskom gdje se stereo vizija koristi za stvaranje trodimenzionalne slike okoline nastale analizom razlika u perspektivi između dva kadra. Iz statičkih transformacija koordinatnih sustava kamere poznati su međusobni odnosi kamera kojima se ostvaruje binokularni vid na temelju čega je moguće ostvariti trodimenzionalnu sliku prostora. Tako dobivenom trodimenzionalnom slikom prostora omogućeno je precizno procjenjivanje udaljenosti do objekata kao i detaljno mapiranje okoline koje je od velikog značenja prilikom autonomnog kretanja robota kroz prostor. Slika 15. prikazuje senzore RealSense D435 kamera koje dolaze prethodno ugrađene na Spot robota:



Slika 15. Prikaz Realsense D435 stereo kamere [28]

Stereo kamere implementirane na Spota su također opremljene senzorom dubine koji se koristi za dobivanje oblaka točaka (engl. *point cloud,* PCL). Reprezentacija okoline u obliku oblaka točaka često je korišteni način interpretacije okoline prilikom kretanja robota kroz prostor. Oblak točaka je model zapisa objekata iz okoline koji sadrži opis dubine točaka iz prostora. Podaci oblaka točaka sastoje se od x, y i z koordinate kao i informacije o udaljenosti interpretirane bojom određenu analizom odsjaja infracrvene svjetlosti koju emitira kamera. Korištenjem senzora dubine poboljšava se sposobnost robota da precizno procijeni prostor oko sebe što rezultira kvalitetnijom metodom izbjegavanja kolizija. To se izravno može vidjeti prilikom penjanja po stepenicama ili pak pri obilaženju prepreka prilikom kretanja kroz prostor. Slika 16. prikazuje okruženje robota reprezentiranog oblakom točaka u Rviz2 alatu:



Slika 16. Dubinska slika okruženja Spota u Rviz okruženju [29]

Na Spota je dodatno postavljena RealSense D435i kamera. Ista posjeduje IMU senzor koji je u radu iskorišten u zadacima filtriranja odometrije mobilnog robota te filtriranja globalne pozicije dobivene putem GNSS prijamnika što je podrobnije opisano u poglavljima *4.3.3. IMU senzori* i *5.3. Robot_localization paket.* Ista također može biti korištena za akviziciju RGB slike te oblaka točaka prostora iznad robota pošto Spot nema nikakav uvid u stanje okoliša iznad njega.

4.3.1.1. realsense-ros paket

Paket realsense-ros [30] omogućuje integraciju Intel RealSense kamera u ROS okruženje pružajući robotskim sustavima podatke o slikama te dubinski prikaz objekata iz prostora. Uz podršku za IMU senzore unutar realsense-ros paketa omogućeno je očitavanje signala istih čime se ostvaruje dodatno poboljšanje preciznosti i stabilnosti robotskog kretanja. Integracijom IMU senzora omogućeno je praćenje orijentacije i ubrzanja, odnosno ključnih parametara za preciznu lokalizaciju u stvarnom svijetu. U okviru korištenja realsense2_camera čvora za podešavanje konfiguracije IMU senzora korišteni su sljedeći parametri:

- *enable_gyro* – Parametar kojim se čvoru kamere omogućuje slanje podataka očitanih s žiroskopa.
- *enable_accel* – Parametar kojim se čvoru kamere omogućuje slanje podataka očitanih s senzora ubrzanja.
- *gyro_fps* – Parametar stope osvježavanja očitanja žiroskopa.
- accel_fps – Parametar stope osvježavanja očitanja senzora ubrzanja.
- *unite_imu_method* – Parametar kojim se određuje metoda objavljivanja IMU podataka. Ukoliko je parametar postavljen na vrijednost nula, parametri s IMU senzora se ne objavljuju. Za odabranu vrijednost parametra 1, čvor između dva dobivena signala objavljuje stare vrijednosti ubrzanja i usmjerenja. Ako pak odabrana vrijednost parametra iznosi 2, čvor između dva dobivena signala provodi linearnu interpolaciju temeljenu na prethodnim očitanjima.

Detaljne parametre konfiguracije realsense2_camera čvora moguće je pronaći unutar datoteka programskog koda paketa [30].

4.3.2. Lidar senzori

Lidar (engl. *light detection and ranging, hrv. otkrivanje i domet svjetlosti*) je često korištena metoda preciznog mapiranja okoline. Koristi se za stvaranje planarnog oblaka točaka iz kojih se dobiva opis obrisa objekata iz prostora u kojemu se nalazi robot. Visoka preciznost lidara omogućava Spotu da s visokom pouzdanošću identificira i razlikuje različite predmete iz svog okruženja čineći ga sposobnim za složene zadatke navigacije, što je ključno za većinu autonomnih zadataka koji se stavljaju pred mobilne robote. Implementacijom Spot nosivog dodatka poboljšane autonomije (engl. *Enhanced Autonomy Payload*, EAP) [31] koji sadrži lidar

povećava se radijus perspektive Spot robota na otprilike 120 metara dok je bez lidara, odnosno prilikom korištenja isključivo senzora dubine ista ograničena u rasponu od dva do četiri metra, ovisno o uvjetima te predmetima iz okoline. Slika 17. prikazuje Spot robota s dodatkom poboljšane autonomije:

Slika 17. Spot s dodatkom poboljšane autonomije (EAP) [32]

Pošto Spot mobilni robot koji je korišten prilikom izrade ovog diplomskog rada nema dostupan dodatak poboljšane autonomije, podaci o okolini bit će generirani korištenjem oblaka točaka dobivenih od dubinskih kamera te paketa za pretvorbu istih u očitanja lidara korištenjem paketa pointcloud_to_laserscan [33]. Tako korištena planarna reprezentacija okoline korištena je u ovom radu za potrebe lokalnog planiranja zaduženog za izbjegavanje prepreka. Postavljanje čvora pretvorbi dubinskih informacija u očitanja lidara detaljnije je opisano u odlomku *6.2.3.2. Izbjegavanje prepreka temeljeno na reprezentaciji okoline korištenjem oblaka točaka*.

4.3.3. IMU senzori

IMU senzori igraju ključnu ulogu u mobilnoj robotici pružajući informacije o ubrzanju i brzini te orijentaciji robota koje su od velikog značaja prilikom održavanja ravnoteže pri gibanju robota. U pravilu se sastoje od akcelerometra, žiroskopa i magnetometra. Spot je opremljen s četiri IMU senzora, po jedan na svakoj nozi. Podaci dobiveni od IMU senzora nisu samo ključni za održavanje ravnoteže robota, već se često koriste i za određivanje točnije odometrije korištenjem Kalman filtera s ciljem preciznijeg praćenja kretnji robota u stvarnom vremenu pri njegovom upravljanju i navigaciji.

Akcelerometri su važne komponente zadužene za nadzor linearnih brzina i ubrzanja. Iste je potrebno kontinuirano pratiti kako bi algoritmi kontrole stabilnosti uvijek imali točne informacije o ponašanju robota te postupili ispravno održavajući ga u stabilnoj poziciji. Žiroskopi su ključne komponente u zadacima detekcije nagiba kao i određivanja kuta skretanja robota. Informacije o nagibu robota dobivene s žiroskopa također su od velikog značaja pri održavanju ravnoteže, pogotovo prilikom kretanja po neravnom terenu. Magnetometri kao komponente IMU senzora pružaju informacije o smjeru kretanja u odnosu na magnetni sjever. Informacije odnosa orijentacije robota u odnosu na magnetski sjever su od velikog značenja u zadacima navigacije, posebno kada robot istražuje nove terene ili svoje gibanje treba provoditi u odnosu na određenu orijentaciju kako bi se izbjegle neželjene rotacije robota te omogućilo ostvarivanje željenih gibanja. Nabrojene komponente od kojih se sastoji sklop IMU senzora uvelike doprinose preciznosti i dosljednosti kretanja robota te uvelike doprinose robusnosti cijelog sustava.

Na robota je ugrađena Intel RealSense D435i kamera s ugrađenim IMU senzorom koji se sastoji od žiroskopa i akcelerometra dok u istome nedostaje magnetometar. Izostanak magnetometra potrebno je kompenzirati, što je detaljnije objašnjeno u odlomku *5. LOKALIZACIJA ROBOTA*, gdje je također detaljnije opisana upotreba IMU senzora u zadacima lokalizacije.

4.3.3.1. Kalibracija IMU-a

Kalibracija IMU senzora neizostavan je korak prilikom provođenja precizne lokalizacije korištenjem Kalmanovog filtra kako bi se ostvarila visoka točnost i stabilnost praćenja kretnji. Kalibracija IMU senzora provodi se s ciljem eliminacije pogrešaka koje mogu proizaći iz različitih faktora, uključujući mehaničke deformacije komponenti, temperaturne varijacije kao i postojanje inherentnih odstupanja senzora. Pošto Intel RealSense kamere iz D400 serije nemaju unutarnje mehanizme kalibracije, prije korištenja kamere potrebno je provesti kalibraciju te tako dobivene kalibracijske parametre pohraniti u memoriju kamere. Procesi kalibracije koji koriste alate poput Python skripti dostupnih od strane proizvođača ili zajednice omogućuju korisnicima podešavanje parametara komponenti s pomoću kojih se postižu točnija očitanja mjerenih veličina. U dokumentaciji kalibracije Intel kamera iz D400 serije [34] dan je detaljan opis procesa podešavanja sustava, instalacije potrebnih paketa kao i alata potrebnih za provođenje kalibracije kamere. Također, detaljno su opisani koraci koje je potrebno provesti te

orijentacije u koje je potrebno postaviti kameru kako bi se referentni podaci mogli ispravno prikupiti te kako bi se sam proces kalibracije uspješno proveo.

Korištena kalibracijska metoda temeljena na Python skripti [34] uključuje kalibraciju isključivo intrinzičnih parametara senzora ubrzanja i žiroskopa. Intrinzični parametri senzora ubrzanja podrazumijevaju faktor skaliranja odnosno osjetljivosti, bias odnosno inicijalni odmak kao i odmake osi. Faktor skaliranja se koristi kako bi se izlazni signal preslikao u jedinice SI sustava te samim time mogao izravno koristiti nakon provedene kalibracije. Sastoji se od tri elementa od kojih je svaki zadužen za skaliranje ubrzanja uzduž jedne osi, odnosno s_x, s_y, s_z. Faktor inicijalnog odmaka se koristi kako bi se uklonio utjecaj svih poremećaja u slučajevima kada bi senzor trebao očitavati nulu kao stanje, matematički zapisanog vektorom $\vec{a}_{bias}$. Faktor korekcije odmaka osi se koristi za ispravljanje osi koje nisu ortogonalne. Sastoji se od šest elemenata, odnosno c_{xy}, c_{yx}, c_{xz}, c_{zx}, c_{zy}, c_{yz}. Kao intrinzični parametar žiroskopa navodi se bias odnosno inicijalni odmak kojim se uklanja utjecaj svih poremećaja kada bi senzor trebao pokazivati nulu kao stanje, matematički zapisan vektorom $\vec{\omega}_{bias}$. Intrinzičnim parametrima modeliraju se unutrašnje nepravilnosti senzora kako bi se sirovi podaci sa senzora ispravno očitali. Matematički zapis implementacije intrinzičnih parametara senzora ubrzanja prikazan je izrazom [34]:

$$\vec{a}_{true} = \begin{bmatrix} s_x & c_{xy} & c_{xz} \\ c_{yx} & s_y & c_{yz} \\ c_{zx} & c_{zy} & s_z \end{bmatrix} \vec{a}_{true} - \vec{a}_{bias}. \quad (1)$$

Matematički zapis intrinzičnih parametara žiroskopa prikazan je izrazom [34]:

$$\vec{\omega}_{bias} = \vec{\omega}_{raw} - \vec{\omega}_{bias} \quad (2)$$

Ekstrinzični parametri IMU senzora, odnosno matrica transformacije i rotacije senzora u odnosu na bazu robota definiraju se putem funkcija transformacija ROS2 sustava, detaljnije opisanih u poglavlju *4.7. Transformacije koordinatnih sustava robota i opreme*.

4.4. Upravljačko računalo Intel Nuc

Intel NUC 13 predstavlja kompaktno i snažno računalo pogodno za izvođenje upravljačkih zadataka visoke razine. Njegov mali dizajn ključan je za prilagodbu različitim okolinama u relativno ograničenom prostoru što ga čini idealnim kandidatom za ugradnju u sustave mobilnih robota. Jednostavno montiranje ubrzava proces integracije u robotske platforme što omogućava brzu implementaciju upravljačkih zadatka bez potrebe za složenim prilagodbama ili dodatnim

konstrukcijskim preinakama. Komponente ugrađene u sustav računala doprinose brzini izvršavanja složenih upravljačkih algoritama i upravljačkih zadataka visoke razine. Inteligentni procesori i količina ugrađene memorije računalu omogućavaju brzu i učinkovitu obradu podataka, pridonoseći brzini izvršavanja stvarnih zadataka koji se stavljaju pred robotske sustave. Računalo dolazi s nizom priključaka kojima se ostvaruje fleksibilnost povezivanja s različitim senzorima, aktuatorima i drugom perifernom opremom koja se često koristi u brojnim robotskim aplikacijama. Slika 18. prikazuje korišteno Intel NUC 13 Pro računalo:

Slika 18. Intel Nuc 13 Pro računalo [35]

Tablica 3. prikazuje karakteristike Intel Nuc 13 računala [36]:

Tablica 3. Karakteristike Intel NUC 13 računala

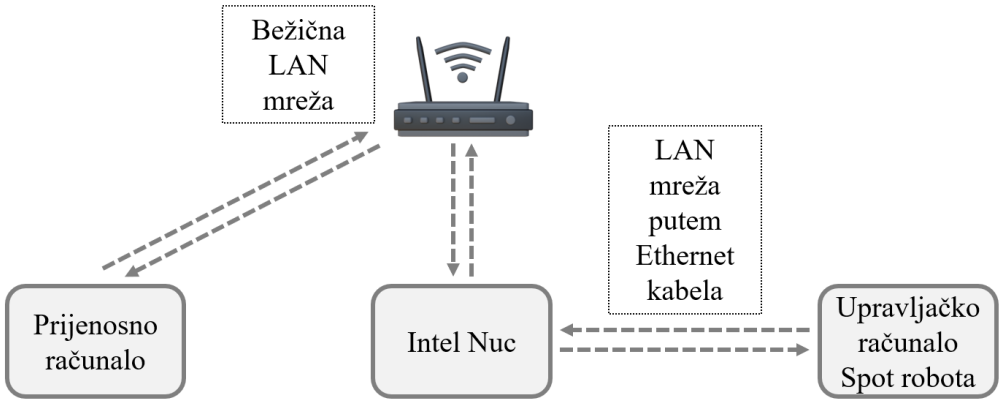
Procesor	13th Gen Intel Corei7-1360P, 12 jezgra (4P+8E), 16 threads, 18MB Intel L3 Cache, 35W TDP
Napajanje	19V 120W DC, bačvasti priključak vanjskog/unutarnjeg promjera 5,5/2,5 mm
Memorija	16GB DDR4 2666 RAM, proširiv do 64GB DR4-3200 SODIMMs, dvo-kanalni ulaz
Pohrana	256GB PCle X4 NVMe SSD, proširiv do 2TB sa 2.5-inch PCleX4 Gen4 SSD ili SATA SSD
Zaslon	Izlaz 4K 60Hz HD putem DP2.1, HDMI2.1, Thunderbolt4 priključka, mogućnost spajanja 4x4K proširena ekrana ili jedan 8k ekran
Grafički procesor	Integrirana 35W Intel Iris Xe grafika

Bežično povezivanje	Intel Wi-Fi 6E AX211 na M.2 priključka, podržava 802.11ax i Bluetooth 5.3
Ethernet	Intel i225-v 10/100/1000/2500 Mbps RJ45
Priključci	2x prednja strana i 1x stražnja strana USB 3.2 Gen 2 tip A; 1x stražnja strana tip A i 2x interna USB 2.0 zaglavlja; 2x HDMI 2.1 TMDS (4K@60Hz); 3.5mm prednji stereo priključak za slušalice
Audio izlazi	Do osam kanala 7.1 digitalnih audio izlaza putem HDMI priključka i DP putem USB C porta
Kompatibilnost za postavljanje	Mogućnost postavljanja na VESA nosač

4.4.1. Komunikacija putem DDS protokola

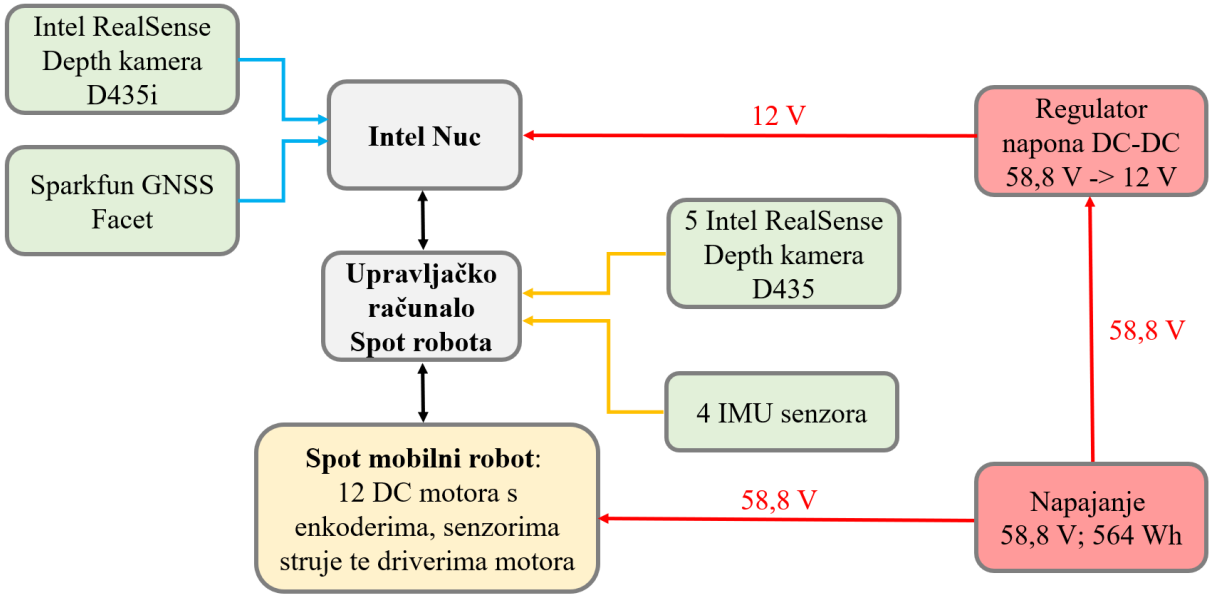
Komunikacija putem protokola servisa za distribuciju podataka (engl. *Data Distribution Service,* DDS) protokola ključan je aspekt učinkovite suradnje između dva upravljačka računala koja međusobno komuniciraju pute mreže. Za potrebe ovog diplomskog rada korištena su tri upravljačka računala, odnosno upravljačko računalo robota, nadodano Intel Nuc računalo te prijenosno računalo zaduženo za pokretanje poslužitelja navigacije te samim time i upravljanje robotom zadavanjem referenci. Na prvom računalu, smještenom na samom robotu, odvija se kontinuirana razmjena informacija između nadodanog Intel Nuc računala te upravljačkog računala robota. Nadodano Intel Nuc računalo upravlja različitim zadacima uključujući programske pakete robota, kamera, GNSS prijamnika te drugih senzora. Putem DDS protokola korisnik s prijenosnog računala može zadavati trajektorije, putne točke te vršiti nadzor rada sustava, dok se komunikacija sa sustavom robota vrši putem nadodanog Intel Nuc računalu. DDS protokol osigurava pouzdanu i distribuiranu komunikaciju između računala, čime se ostvaruje pouzdano sučelje za upravljanje i navigaciju robota. Zbog velike količine podataka koje je potrebno prenijeti u stvarnom vremenu postoji mogućnost desinkronizacija podataka te zapunjenje međuspremnika (engl. *buffer*) memorijskih lokacija tema na računalima koja međusobno komuniciraju. Kako bi se izbjeglo rušenje kanala komunikacije zbog dolaska do povremenih desinkronizacija, u ovom radu korišteni su dodaci za brze eProsima DDS servise [37][38]. Korištenje brzih DDS sustava omogućava se održavanje sinkroniziranog prijenosa podataka između sustava koji rezultira stabilnijim radom te se također izbjegava rušenje sustava prilikom privremene desinkronizacije što je od velikog značenja u zadacima upravljanja

stvarnim robotskim sustavima. Također, prilikom postavljanja komunikacije upravljačkog računala i dodatne opreme podešen je i isproban Eclipse Cyclone DDS servis [39]. Servis pruža izravnu implementaciju DDS servisa koja se navodi kao osnova pri ostvarivanju komunikacije te razmjene podataka između raznih dijelova distribuiranih softverskih sustava. Cyclone DDS u svojoj dokumentaciji [40] naglašava postizanje visoke stabilnosti, pouzdanosti i niske latencije što je od posebnog značaja u zadacima gdje je potrebna brza i sigurna razmjena podataka u distribuiranim okruženjima. Slika 19. shematski prikazuje strukturu komunikacije između upravljačkih računala zaduženi za upravljanje mobilnim robotom:



Slika 19. Struktura komunikacije između upravljačkih računala

Slika 20. shematski prikazuje povezivanje komponenti potrebnih za izvršavanje zadatka globalne navigacije mobilnog robota:



Slika 20. shematski prikaz spajanja komponenti korištenih u zadacima globalne navigacije mobilnog robota

4.4.2. Kvaliteta servisa ROS sustava

Kvaliteta servisa (engl. *Quality of Service,* QoS) u ROS2 sustavu predstavlja ključan element za prilagodbu parametara komunikacije prilikom komunikacije raznih elemenata robotskih sustava. Parametri kvalitete servisa mogu biti istovremeno određeni za oglašivače, pretplatnike, servise poslužitelja i klijente. Svaki profil kvalitete servisa primjenjuje se nezavisno te je prilikom svake promijene kvalitete servisa zbog moguće nekompatibilnosti potrebno provjeriti kvalitete poruka koje se šalju među čvorovima. Pri radu sustava, čvorovi koji obavljaju različite zadatke mogu imati različite zahtjeve u vezi s pouzdanošću, trajanjem, zadrškom i veličinom reda poruka koje se šalju u kanalu komunikacije. U svojoj bogatoj lepezi dodatnih mogućnosti, ROS2 sustav nudi široki izbor konfiguracija podešavanja ponašanja komunikacija među čvorovima. Uz pravilno podešavanje kvaliteta servisa može se postići pouzdanost i brzina razmjerna UDP (engl. *User Datagram Protocol, hrv. korisnički datagramski protokol*) protokolu [41].

Za razliku od ROS sustava, koji prvenstveno podržava samo TCP protokol, ROS2 koristi fleksibilnost osnovnog DDS prijenosa u okruženjima s neidealnim bežičnim mrežama koje karakteriziraju brojni gubitci. Podešavanjem pouzdanosti putem QoS servisa određuje se postojanje potrebe primanja informacije o primitku poruka što ima značajan utjecaj na broj poruka koje se mogu izmijeniti u jedinici vremena. U situacijama gdje je brza reakcija ključna, postavljanje trajanja poruka može biti od suštinske važnosti dok parametar zadrške određuje koliko poruka može čekati u redoslijedu prije dostave. Pri podešavanju QoS parametara za rad servisa i akcijskih poslužitelja odabire se kompromis između brzine izvođenja komunikacije i vjernosti podataka koji se šalju u kanalu komunikacije između servisa i poslužitelja. Parametri profila QoS servisa podešeni su pri konfiguraciji čvorova paketa koji su priloženi u repozitoriju diplomskog rada [2].

4.5. Spot SDK

Spot SDK [42] predstavlja ključni alatni skup kojim se uvelike olakšava razvoj aplikacija i dodataka za Spot robota. Koristi se za razvoj specifičnih funkcionalnosti koje odgovaraju potrebama različitih industrija i aplikacija u kojima Spot pronalazi primjenu. Sastoji se od pet ključnih dijelova, odnosno konceptualne dokumentacije, Python knjižnice, dokumentacije za razvoj dodataka, Spot API definicije protokola te Spot SDK repozitorija. Unutar konceptualne dokumentacije opisane su ključne značajke Spot SDK alata. U dijelu koji opisuje Python knjižnicu nalazi se opis aplikacija kojima se ostvaruje upravljanje robota kao i čitanje podataka sa senzora te općih informacija o stanju robota. Dokumentacija za razvoj dodataka uključuje skripte i predloške koji se koriste za integraciju dodatnih senzorskih jedinica te jedinica za komunikaciju. Spot API definicija protokola obuhvaća pojedinosti o protokolima s pomoću kojih se vrši komunikacija sa Spot robotom. Definicije protokola se najčešće koriste kako bi se za rad sustava robota implementirao neki drugi programski jezik umjesto inicijalno postavljenog Pythona.

4.6. Spot_ros2 paket

Paket spot_ros2 [43] predstavlja ključan alat pri integraciji brojnih značajki koje Spot donosi. Paket je izveden od spot_ros paketa razvijenog za ROS sustav [44] uz sve potrebne prilagodbe arhitekture zahtijevane od strane ROS2 sustava. Paket u velikoj mjeri olakšava komunikaciju između upravljačkog sustava robota i ROS2 sustava omogućavajući relativno jednostavnu i robusnu međusobnu izmjenu informacija i naredbi.

Paket u svojoj biblioteci sadržava čvorove, servise kao i predloške izvođenja zadataka kojima se ostvaruje upravljanje i komunikacija s robotom te raznim dodacima koji su montirani na istog. Također, paket sadrži sve potrebne programske dodatke koji se koriste za akviziciju slika kamere, oblaka točaka te ostalih senzora i uređaja koji su implementirani na istom. Hardverska integracija koju donosi spot_ros2 paket ključna je za prikupljanje informacija o okolini, lokaciji i stanju robota gdje elementi paketa donose razna sučelja za prijenos senzorskih podataka kroz ROS2 teme i servise, čineći ih dostupnim u cijelom ROS2 okruženju. Spot paket također podržava vizualizaciju senzorskih podataka i stanja robota kroz ROS2 alate poput Rviz2 alata, pružajući intuitivan način praćenja i analize rada robota. Postojanje mogućnosti vizualizacije je od velike važnosti prilikom razvoja, testiranja te otklanjanja pogrešaka pri izradi robotskih aplikacija. Također valja spomenuti da spot_ros2 paket prati trenutni razvoj spot-sdk

alata za razvoj sustava [42] te se periodički obnavlja kako bi u svojoj biblioteci koda sadržavao najnovije značajke koje donosi isti. Za pokretanje svih čvorova i poslužitelja koji su potrebni za uspješnu komunikaciju s robotom te upravljanje putem referenci brzina i položaja pokreće se spot_driver datoteka koja je priložena u repozitoriju diplomskog rada [2].

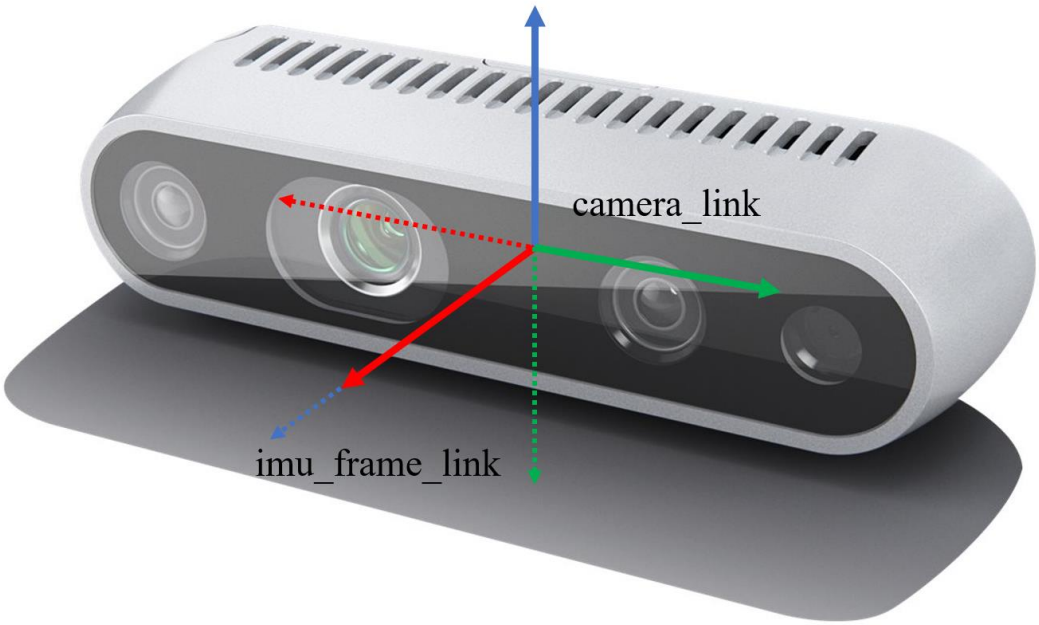
4.6.1. spot_driver

S ciljem ostvarivanja veze između Spot robota i ROS2 sustava pokreće se spot_driver datoteka. Datoteka omogućava konfiguraciju podataka za autorizaciju pri povezivanju, odnosno parametre korisničkog imena, lozinke te IP adrese na kojoj se robot nalazi. Konfiguracijskom datotekom paketa također je moguće definirati parametre korištenih dodataka poput robotske ruke, GXP (engl. *General Expansion Payload, hrv. dodatak za proširenje*) dodatka te postavke korištenih RealSense kamera montiranih na tijelo robota. Unutar konfiguracijske datoteke paketa nalaze se putanje datoteka spot_ros čvorova u kojima se nalaze detaljni opisi transformacija, zglobova te ostalih postavki vezanih za opis kretanja mobilnog robota. U biblioteci koda [43] nalazi se detaljniji pregled mogućnosti konfiguracije čvora drivera. Robot se s Intel NUC 13 Pro računalom spaja putem ethernet priključka koji je detaljnije opisan u odlomku *4.2. Specifikacije Spot mobilnog robota*. Razlog žičnog spajanja robota i upravljačkog računala umjesto spajanjem putem bežičnog mrežnog protokola leži u minimizaciji kašnjenja paketa koji se šalju između dva upravljačka računala. Anuliranje kašnjenja u komunikaciji upravljačkog računala s mobilnim robotom može imati značajan utjecaj u ponašanju cijelog sustava.

4.7. Transformacije koordinatnih sustava robota i opreme

Poznavanje transformacija koordinatnih sustava je nužno kako bi se mogao odrediti relativni položaj senzora u odnosu na bazu robota te kako bi se samim time očitanja senzora ispravno interpretirala. Transformacije kao i opis zglobova Spota su definirani u konfiguracijskim datotekama pri pokretanju skupa čvorova iz spot_driver paketa.

U zadacima globalne navigacije za potrebe lokalizacije koriste se dva dodatna izvora informacija. To su informacija o globalnoj poziciji dobivena od strane GNSS prijamnika te podaci dobiveni od strane IMU senzora smještenog u RealSense D435i kameri. Kako bi isti mogli biti korišteni u zadacima lokalizacije potrebno je odrediti međusobni odnos njihovih koordinatnih sustava s koordinatnim sustavom baze robota. U prilogu I. nalazi se izvadak iz tehničke dokumentacije Spot robota [45] uz detaljniji prikaz međusobnog položaja vodilica za montiranje dodataka i baze sustava. Korištenjem ključnih točaka korištene opreme određene su vrijednosti statičkih transformacija korištenog IMU senzora i GNSS prijamnika u odnosu na vodilice za montiranje dodataka. Slika 21. prikazuje koordinatne sustave RealSense D400 serije kamera:



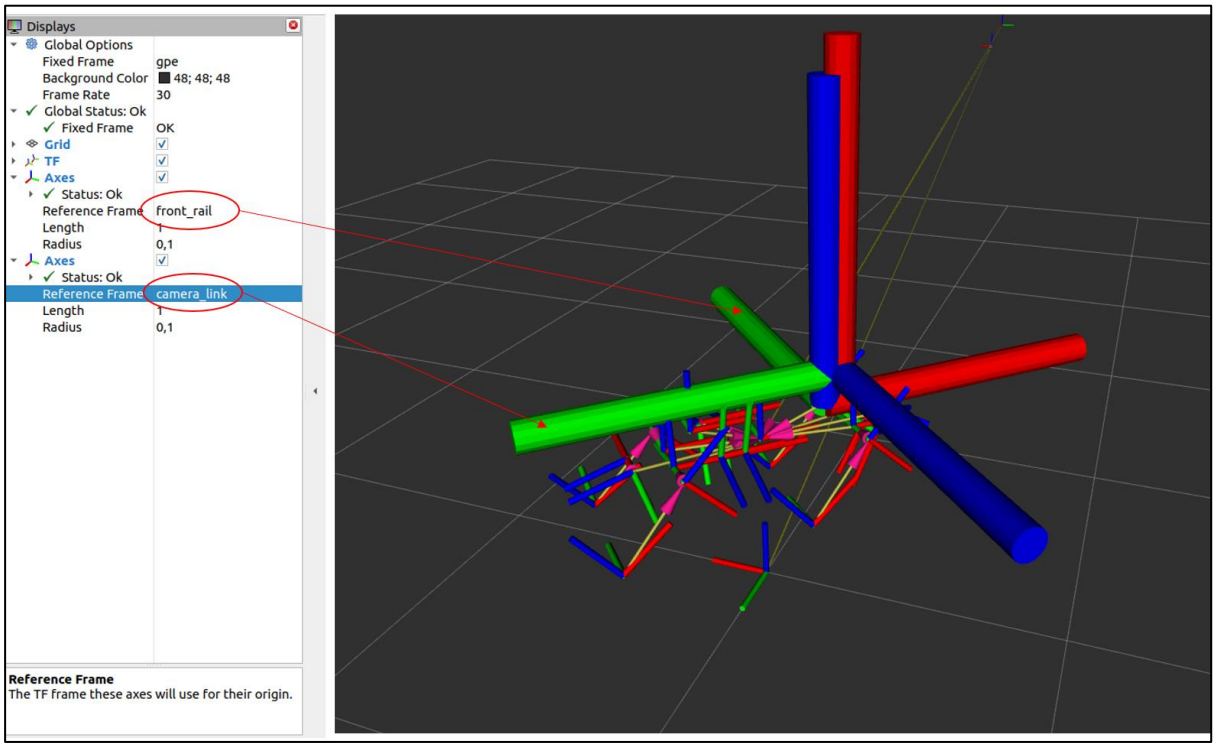
Slika 21. Prikaz koordinatnih sustava Intel RealSense D400 serije kamera

Sa slike je vidljivo da koordinatni sustav baze kamere prikazan punim linijama (*camera_link*) nije jednak koordinatnom sustavu IMU senzora prikazanom isprekidanim linijama (*imu_frame_link*). Zbog toga je, uz potrebne translacije koordinatnog sustava kamere, potrebno

odrediti rotacije kako bi osi koordinatnog sustava IMU senzora odgovarale onima propisanim REP 105 [46] i REP 103 [47] konvencijama koje su detaljnije opisane u odlomku *5. LOKALIZACIJA ROBOTA*. Transformacije koordinatnih sustava određene su mjerenjem položaja IMU senzora u odnosu na referentne značajke robota dane u prilogu I.. Matrice rotacija koordinatnog sustava kamere određene su korištenjem dostupnih alata za određivanje rotacija u quaternion zapisu [48] kako bi se ispoštovali zahtjevi gore spomenutih konvencija. Matrica transformacija koordinatnog sustav kamere u odnosu na koordinatni sustav prednje vodilice prikazana je u nastavku:

$$\mathbf{T}_{IMU} = \begin{bmatrix} 0 & -1 & 0 & 0{,}04 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0{,}06 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Prema standardnim mjernim jedinicama ROS2 sustava, iznosi translacija koordinatnog sustava izražene su u metrima. Slika 22. prikazuje koordinatni sustav IMU senzora podešenog uz korištenje čvora statičke transformacije:



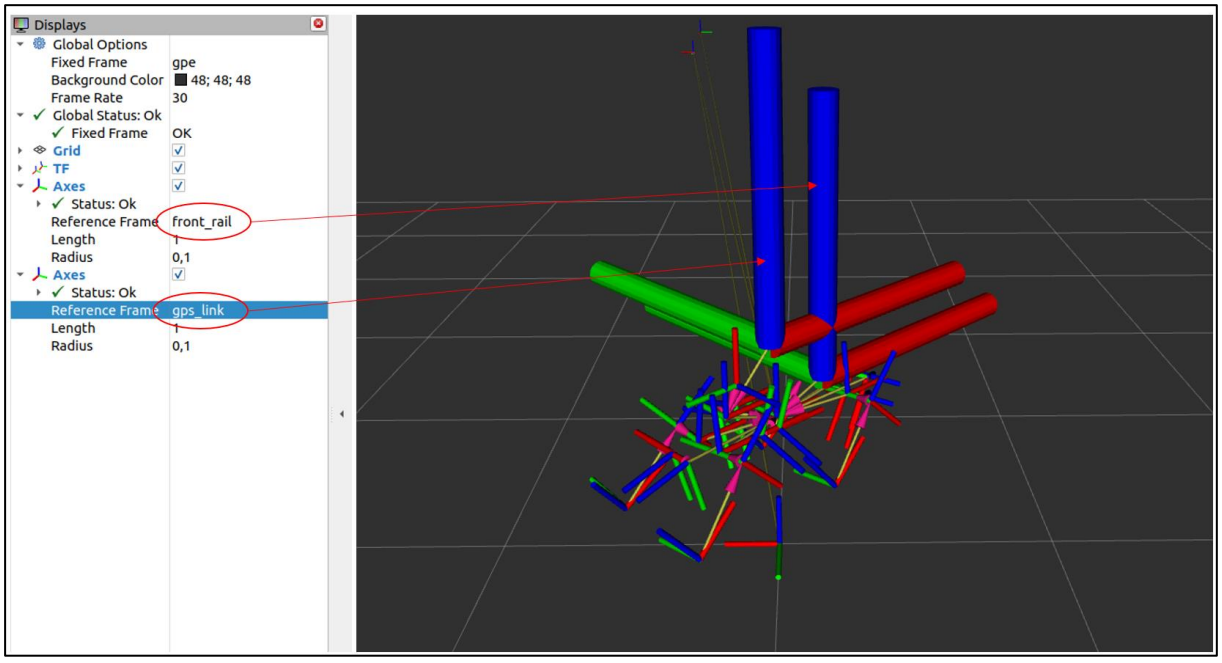
Slika 22. Statičke transformacije koordinatnog sustava IMU senzora u odnosu na bazu sustava za dodatke

Kao i kod IMU senzora, pri podešavanju statičkih transformacija koordinatnog sustava GNSS prijamnika određeni su iznosi mjerenjem položaja GNSS prijamnika u odnosu na

referentne značajke robota dane u prilogu I.. Matrica transformacija koordinatnog sustav GNSS prijamnika u odnosu na koordinatni sustav prednje vodilice prikazana je u nastavku:

$$\mathbf{T}_{GNSS} = \begin{bmatrix} 1 & 0 & 0 & -0{,}185 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0{,}31 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

Slika 23. prikazuje koordinatni sustav GNSS prijamnika podešenog korištenjem čvora statičke transformacije:



Slika 23. Statičke transformacije koordinatnog sustava GNSS prijamnika u odnosu na bazu sustava za dodatke

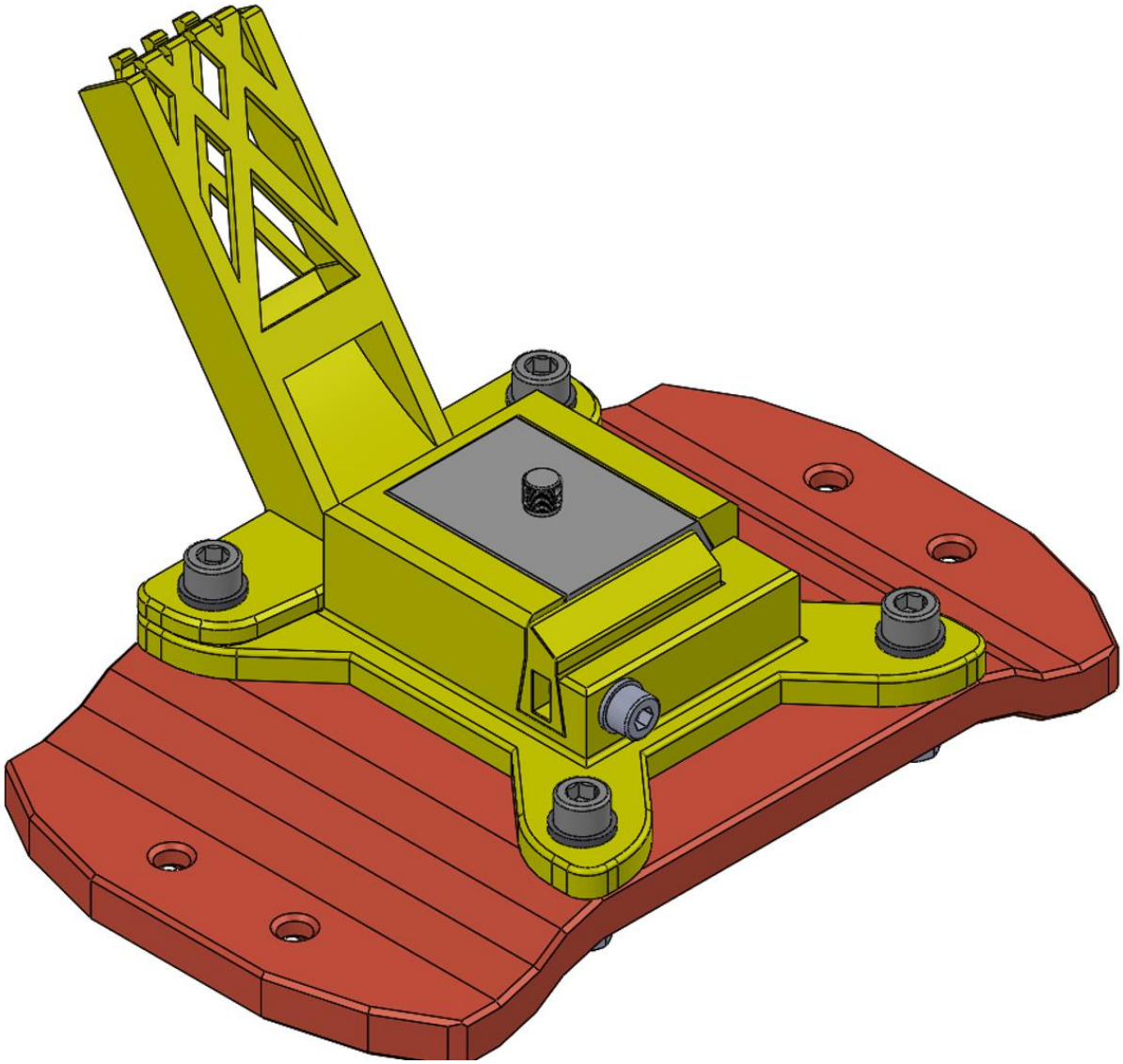
4.8. Implementacija sustava na mobilnog robota

U sljedećim odlomcima opisani su postupci izrade i montiranja kao i metode napajanja dodatne opreme korištene u zadacima globalne navigacije mobilnog robota.

4.8.1. Konstrukcijska rješenja implementacije sustava

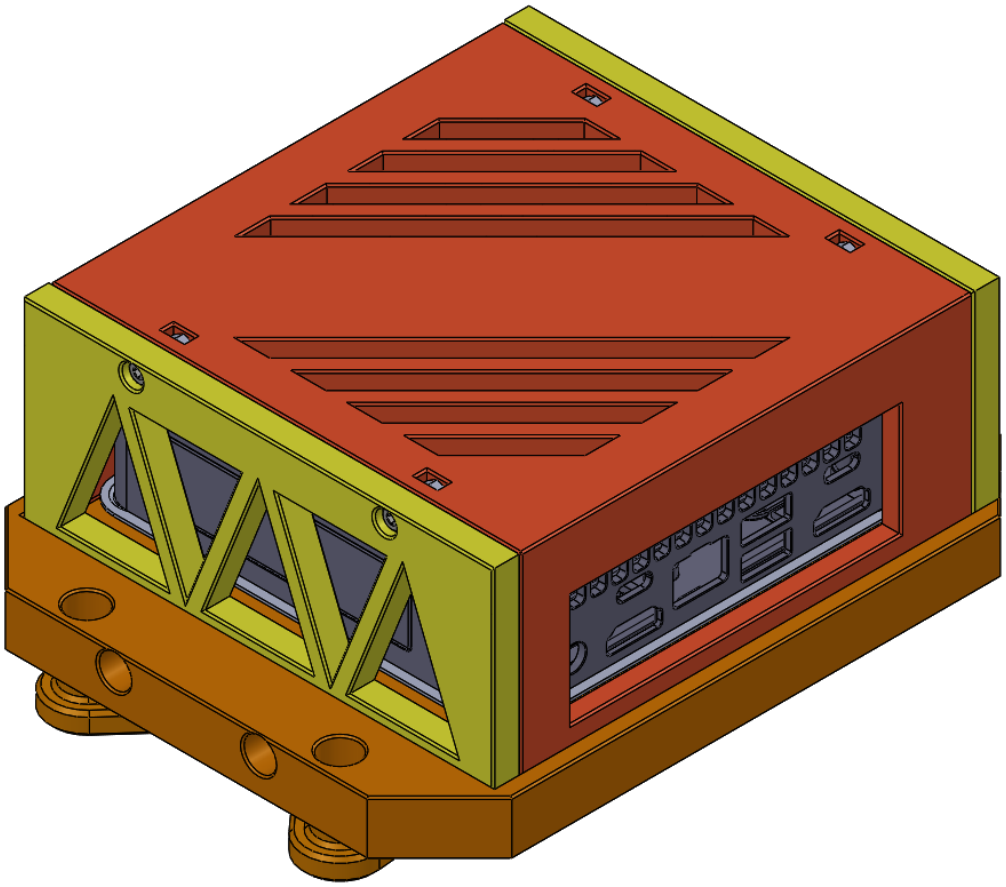
Modeli su oblikovani u Dassault Systemes programskom paketu SolidWorks. Izrađeni su tehnologijom aditivne proizvodnje koristeći Prusa MK3s pisače te PLA i PETG filamente. Nosači su konstrukcijski načinjeni tako da omogućavaju brzo postavljanje opreme na robota te samim time i skidanje s istog što je od velikog značaja posebice u ranim fazama testiranja

algoritama koje u pravilu zahtijevaju česte intervencije na opremi. Također, nosači su konstruirani tako da osiguravaju nesmetani protok zraka na gornjim površinama robota kako svojim prisutnošću ne bi ometali hlađenje robota. Dodavanjem zračnosti između nosača i robota dodatno se izbjegava nepotrebno zagrijavanje opreme kao i korištenih nosača. Slika 24. prikazuje sklop nosača GNSS prijamnika i telemetrijske antene:



Slika 24. Model sklopa nosača GNSS sustava

Slika 25. prikazuje sastavljeno kučište Intel NUC računala:

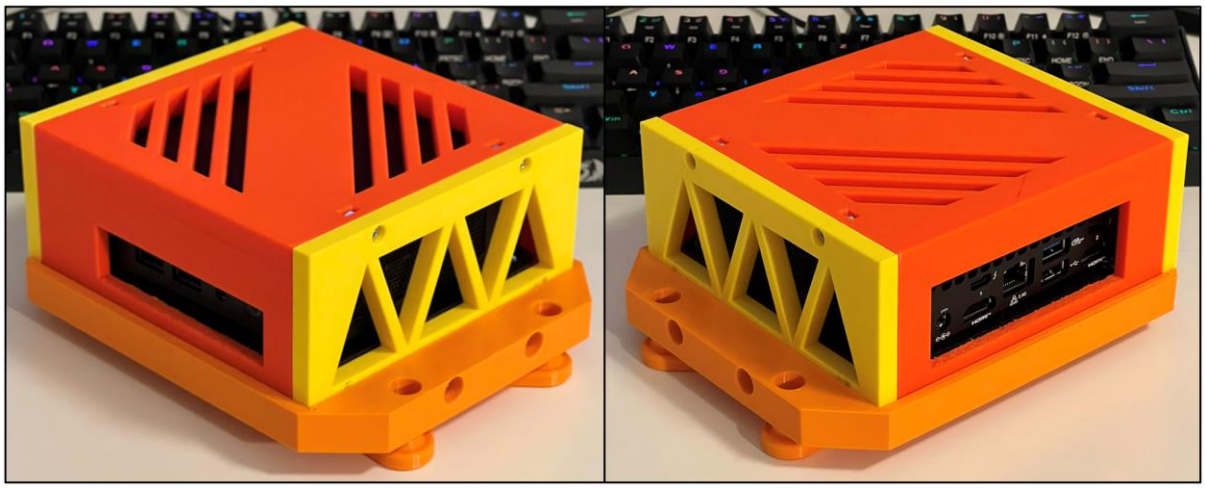


Slika 25. Model sklopa kučišta Intel NUC računala

Slika 26. prikazuje GNSS sustav montiran na nosač:

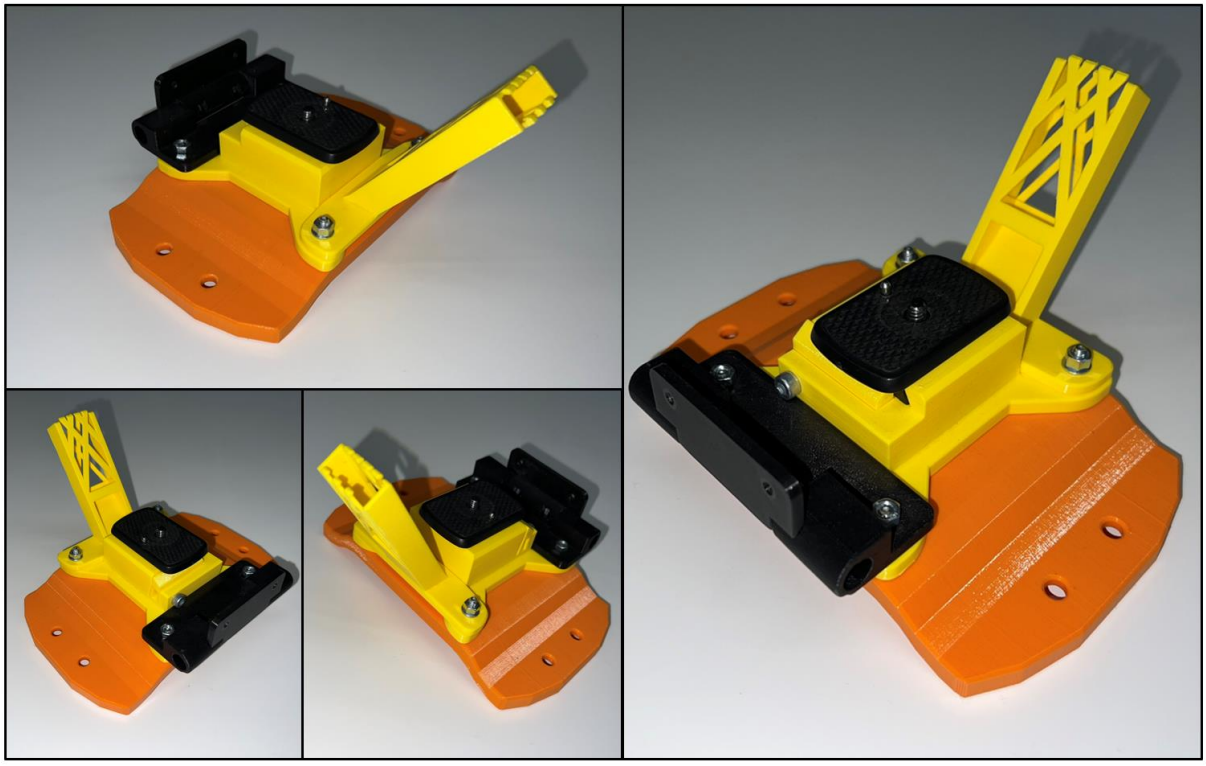
Slika 26. GNSS sustav postavljen na nosač

Slika 27. prikazuje Intel NUC računalo montirano na nosač:



Slika 27. Intel NUC računalo postavljeno na nosač

Kao nosač kamere iskorišten je postojeći nosač konstruiran za potrebe drugih projekata sa Spot mobilnim robotom. Slika 28. prikazuje nosač RealSense D435i kamere montiran na nosač GNSS prijamnika:



Slika 28. Nosač Intel RealSense D435i kamere postavljena na nosač GNSS prijamnika

Slika 29. prikazuje Spot robota na kojem je montirana oprema korištena u ovom radu:

Slika 29. Oprema postavljena na Spot robota

Komponente nosača konstruiranih za potrebe ovog rada nalaze se na repozitoriju diplomskog rada [2]. Sklopni crteži nosača GNSS prijamnika te Intel Nuc računala nalaze se u prilogu.

4.8.2. Napajanje implementiranih sustava

U pogledu napajanja opreme na robotu, ključno je osigurati pouzdano i učinkovito snabdijevanje električnom energijom za svaku komponentu. Kao što je spomenuto u ovom odlomku, robotski sustav sastoji se od nekoliko ključnih dodataka, uključujući upravljačko računalo, GNSS uređaj i kameru s integriranim IMU senzorom, svaki od njih s različitim energetskim potrebama. Spot donosi mogućnost integracije dodatne opreme putem Payload sučelja [49]. Robot na leđima donosi žljebove za montažu između kojih se nalaze dva DB25 priključka.

Korištenjem Spot GXP dodatka [50] moguće je vršiti napajanje dodataka robota. Oba prisutna DB25 priključka mogu pojedinačno isporučiti do 150 W snage pri 12 i 24 V te 10 W

pri 5 V izvoru istosmjernog napajanja putem HD15 priključka. Također, GXP dodatak omogućava povezivanje dodatne opreme s upravljačkim sustavom robota uz korištenje RJ45 priključka. Upravljačko računalo Intel NUC radi u rasponu od 12 do 20 V izvora istosmjernog napona uz vršnu snagu od 120 W [51], što je moguće izravno postići korištenjem Spot GXP dodatka sa serijskim napajanjem kojeg robot može isporučiti. Na upravljačko računalo spajaju se GNSS prijamnik i kamera s integriranim IMU senzorom. Prijamnik GNSS signala zahtjeva 5 V istosmjerni naponski izvor uz vršnu struju od 2 A pri brzom punjenju baterije. Korišteni GNSS prijamnik posjeduje LiPo bateriju kapaciteta 6 Ah kojom se osigurava korištenje prijamnika u minimalnom trajanju od 25 sati [3]. Postojanje baterije u sklopu GNSS uređaja smanjuje opterećenost napajanja Intel NUC računala te samim time nema značajan utjecaj na degradaciju performansi pri korištenju prijamnika. Intel RealSense D435i kamera zahtjeva istosmjerni izvor napajanja od 5 V uz vršnu struju od 0,7 A pri korištenju svih funkcionalnosti kamere. Kao i za GNSS prijamnik, korištenje kamere kao izvora IMU podataka nema značajniji utjecaj na povećanje opterećenosti Intel NUC računala. Detaljna dokumentacija napajanja dodatnih uređaja putem Spot robota nalazi se u Boston Dynamics SDK biblioteci [49]. Slika 30. prikazuje dodatak za proširenje (engl. General Expansion Payload, GXP) montiran na žljebove za montažu:

Slika 30. Spot dodatak za proširenje montirani na robotu

5. LOKALIZACIJA ROBOTA

Lokalizacija robota, odnosno određivanje položaja mobilnog robota u prostoru ključan je aspekt mobilne robotike. Glavna zadaća lokalizacije svodi se na određivanje položaja robota u odnosu na poznatu okolinu u koju je isti postavljen. U zadacima autonomnog kretanja robota, lokalizacija predstavlja osnovnu komponentu na temelju koje se donose odluke za sljedeće radnje koje će robot izvršiti. Ako se robot kreće po poznatoj okolini, algoritmi lokalizacije imaju na pristup mapu okruženja te uz pomoć senzora postavljenih na robota stvaraju sliku okruženja te prate kretanje, što kao cilj ima određivanje pozicije i orijentacije robota u poznatoj mapi. U slučajevima kada se robot kreće u nepoznatom okruženju, lokalizacija se koristi kako bi se odredio trenutni položaj robota u odnosu na neki otprije poznati položaj, najčešće početnu točku od koje je robot započeo svoje gibanje ili pak referentnu točku definirana navigacijskim sustavom. Zbog korištenja više senzora koji generiraju zašumljeni signal, algoritmi lokalizacije su uz estimaciju zaduženi i za mjerenje nesigurnosti estimacije stanja.

Globalna lokalizacija provodi se korištenjem informacija temeljenih na poziciji dobivenoj od GPS RTK sustava uz korištenje satelita koji kruže oko Zemlje. Zbog odstupanja položaja dobivenih od sustava globalne navigacije kao i stope osvježavanja od 1 Hz, provodi se fuzija sa sustavom lokalizaciji na lokalnoj razini koji je zadužen za anuliranje skokova koje GPS RTK sustav može isporučiti zbog ulaska u područje slabijeg GNSS signala ili ostalih problema navedenih u odlomku *3. SUSTAV GNSS NAVIGACIJE*. Također, mjerenja senzora potrebno je filtrirati korištenjem estimatora stanja kako bi se u velikoj mjeri reducirao ili pak uklonio negativan utjecaj zašumljenih podataka mjerenja. U ovom radu korišteni su prošireni i neprekriveni Kalmanov filtar čiji koncept je detaljnije opisan u sljedećim odlomcima.

5.1. Kalmanov filtar

Zbog raznih prednosti i nedostataka pojedinih izvora signala potrebno je provesti matematički algoritam fuzije signala raznih izvora što na posljetku rezultira stabilnijom i točnijom odometrijom robota tijekom kretanja kroz prostor. Primjerice, IMU povećava svoje odstupanje (engl. *drift)* razmjerno vremenu proteklom od pokretanja, dok odometrija dobivena iz podataka enkodera degradira udaljavanjem od početne lokacije. Nadalje, podaci dobiveni od sustava globalne navigacije pružaju informacije o globalnoj poziciji robota uz rasipanje vrijednosti koje ovisi o više faktora što je detaljnije opisano u odlomku *3.SUSTAV GNSS*

NAVIGACIJE. Zbog prednosti u raznim situacijama, fuzija različitih senzora dobiva svoj smisao u zadacima preciznijeg određivanja pozicije i orijentacije robota.

Kalmanov filtar odnosno linearni kvadratni estimator vrsta je estimatora stanja projektiranog za upotrebu u stohastičkim sustavima gdje se javljaju vanjski poremećaji te pogreške mjerenja. Kao što je spomenuto u uvodu ovog odlomka, svoj rad temelji na prikupljanju podataka iz više međusobno nezavisnih izvora kako bi njihovom fuzijom davao točnije podatke od svakog izvora zasebno ili pak estimirao podatke koje nije moguće izravno mjeriti. Spada u kategoriju optimalnih rekurzivnih filtera te je zbog korištenja rekurzivnih funkcija primjenjiv isključivo u vremenski-diskretnoj domeni [53][55].

Kalmanov filtar omogućava fuziju poznate nesigurnosti očitanja trenutnog stanja robota te poznatih nesigurnosti očitanja pojedinih senzora s ciljem smanjenja sveukupne nesigurnosti robota. Nesigurnost trenutnog stanja kao i nesigurnost očitanja senzora se pri radu Kalmanovog filtra najčešće opisuju Gaussovom raspodjelom. Gaussova, odnosno normalna raspodjela ima dva parametra, odnosno srednju vrijednost i kovarijancu kojima se opisuje pouzdanost svake od korištenih komponenata sustava. Srednjom vrijednosti opisana je distribucija koja ima najveće izglede da bude točna, dok se varijancama opisuje nesigurnost pojedinih izvora informacija. Kalmanov filter se u osnovi sastoji od tri glavna koraka, odnosno koraka predikcije i korekcije koji se iterativno provode u vremenu kako bi se procijenilo stanje sustava na osnovu nove serije mjerenja kojima prethodi korak inicijalizacije sustava. Korak inicijalizacije podrazumijeva postavljanje inicijalnih vrijednosti vektora stanja, odnosno početnih uvjeta sustava. Korak predikcije odnosi se na upotrebu dinamičkog modela sustava te matrica kovarijanci kako bi se predvidjelo sljedeće stanje sustava, bazirano na mjerenjima dobivenim u prethodnom koraku. Faza korekcije podrazumijeva mjerenje stvarnih vrijednosti sustava nakon čega slijedi računanje razlike između stvarnog mjerenja i estimata određenih Kalmanovim filtrom. Na temelju te razlike kao i odnosa nesigurnosti mjerenja i predviđenih vrijednosti računaju se pojačanja Kalmanovog filtra koja služe kao ocjena estimacije varijabli sustava. Na osnovu dobivenih pojačanja Kalmanovog filtra te podacima o nesigurnosti mjerenja i dinamičkom modelu sustava ažurira se matrica kovarijanci. Na posljetku faze korekcije procjenjuje se stanje estimiranih veličina koristeći nova pojačanja Kalmanovog filtra kao i trenutnu razliku mjerenih i predviđenih vrijednosti varijable sustava. Ponavljanje koraka predikcije i korekcije tijekom vremena provodi se s ciljem ponašanja dinamičkog sustava i održavanja tražene točnosti procjene varijabli koje se estimiraju. Kao što je spomenuto,

algoritam Kalmanovog filtra se provodi rekurzivno te se može izvoditi u stvarnom vremenu koristeći samo trenutna mjerenja te prethodno izračunato stanje i nesigurnost te kao takav ne zahtijeva poznavanje nikakvih drugih informacija.

Linearni Kalmanov filtar primjenjuje se na sustave čija dinamika i mjerenja mogu biti matematički opisani linearnim modelima. Ovaj filter učinkovito funkcionira u linearnim uvjetima, ali gubi na preciznosti prilikom implementacije s nelinearnim sustavima te pri djelovanju poremećaja nelinearnih karakteristika. Zbog prilagođenosti radu s nelinearnostima u procesu i adaptaciji na nelinearne poremećaje često se koriste prošireni i neprikriveni Kalmanovi filtri [56]. U ovom radu je kao matematički algoritam za procjenu stanja dinamičkog sustava, zbog prirode nelinearnih poremećaja i varijanci te brzine izvođenja algoritma putem DDS servisa korišten prošireni Kalmanov filtar dok je neprekriveni Kalmanov filtar opisan i isproban.

5.1.1. Prošireni Kalmanov filtar

Prošireni Kalmanov filtar (engl. *Extended Kalman Filter*, EKF) je izvedenica linearnog Kalmanovog filtra prilagođena radu sa sustavima čiji su modeli ili mjerenja nelinearni. Prošireni Kalmanov filtar koristi Kalmanov filtar na lineariziranoj funkciju u blizini procijenjenog stanja čime se dobiva aproksimacija optimalnog rješenja za filtrirano stanje. Kao što je spomenuto u uvodu ovog odlomka, glavna zadaća Kalmanovog filtra je provlačenje nasumičnih vrijednosti Gaussove raspodjele (engl. *Gaussian Random Variable*, GRV) kroz dinamiku sustava [57]. U slučaju proširenog Kalmanovog filtra, distribucija stanja je aproksimirana koristeći nasumične vrijednosti Gaussove raspodjele koje zatim podliježu linearizaciji oko trenutnog stanja robota. Provođenjem linearizacije omogućava se korištenje linearnog Kalmanovog filtra u svakom koraku, dok prije svakog novog koraka procjene stanja slijedi linearizacija oko novo estimirane radne točke robota. Provođenje postupaka linearizacije u svakom koraku rada Kalmanovog filtra povećava zahtjevnost algoritma što se ne oslikava značajno na računalno vrijeme potrebno za izvršavanje istog [54]. Također, provođenjem postupka linearizacije uvodi se pogreška aproksimacije koju filtar ne uzima u obzir u svom radu, odnosno koracima predviđanja i korekcije. Nedostatak metoda određivanja pogrešaka temeljenih na lineariziranom modelu često se preslikavaju na podcjenjivanje nesigurnosti estimiranog stanja odnosno značajnije pogreške u estimaciji sljedećeg stanja kao i matrici kovarijanci. Pogreške u ulaznim podacima filtra nerijetko dovode do suboptimalnih

performansi kao i divergiranja estimiranih vrijednosti od stvarnog ponašanja sustava. Detaljna analiza rada proširenog Kalmanovog filtra je dana u [58]. Parametri i upotreba proširenog Kalmanovog filtra detaljnije su opisani u odlomku *5.3.2. Ekf_filter_node čvor*.

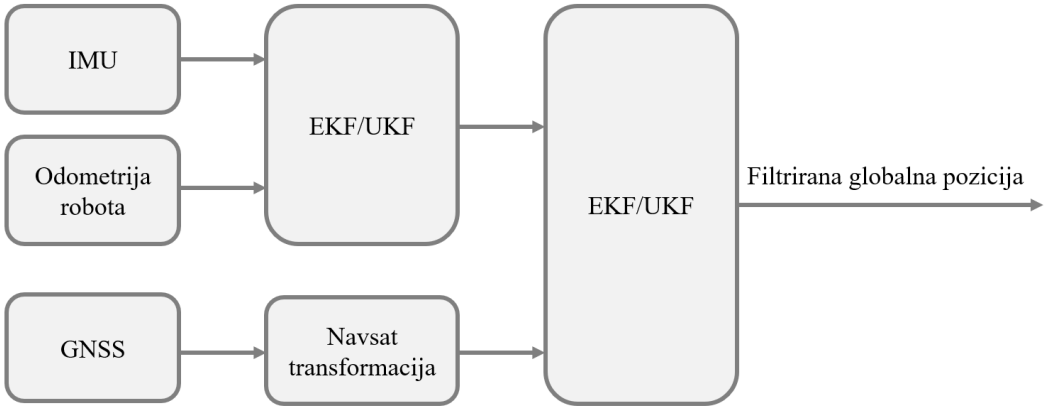
5.1.2. Neprikriveni Kalmanov filtar

U usporedbi s proširenim Kalmanovim filtrom, neprikriveni Kalmanov filtar (engl. *Unscented Kalman Filter*, UKF) bira razlike uzoraka, takozvane sigma uzorke točaka iz distribucije filtriranja te ih ažurira kroz nelinearne modele stanja i mjerenja. Za razliku od proširenog Kalmanovog filtra u čijem se radu zanemaruje utjecaj linearizacije, neprikriveni Kalmanov filtar koristi metodu deterministčkog uzorkovanja stanja sustava. Distribucija varijabli stanja se također određuje korištenjem Gaussove nasumične raspodijele koja se od proširenog Kalmanovog filtra razlikuje u izboru točaka uzorkovanja. Korištenje pomno izabranih točaka uzorkovanja značajno utječe na kvalitetu estimacije točnijih srednjih vrijednosti i kovarijanci lineariziranog sustava. Matematička pozadina odabira točaka uzorkovanja nalazi se u [58]. Propagacijom točaka uzorkovanja kroz promatrani nelinearni sustav postiže se točnost razmjernu razvojem do trećeg člana Taylorovog reda [57], dok se korištenjem proširenog Taylorovog reda postiže točnost razmjerna raspisom samo prvog člana reda. Takva metoda rezultira procjenama stanja koje bolje predstavljaju nesigurnost stanja od procjena dobivenih korištenjem proširenog Kalmanovog filtra. Također, korištenje takvog pristupa dodatno povećava opterećenje sustava fuzije podataka što se oslikava na porast računalnog vremena izvršavanja algoritma. Detaljna analiza rada neprekrivenog Kalmanovog filtra dana je u [58]. Parametri i upotreba neprekrivenog Kalmanovog filtra detaljnije su opisani u odlomku *5.3.3. Ukf_filter_node čvor*.

5.2. Fuzija međusobno nezavisnih podataka lokalizacije

Zbog složene arhitekture robota kao i većeg broja međusobno povezanih zglobova, odometriju Spota koja se dobiva očitanjima enkodera potrebno je sjediniti s podacima dobivenih od IMU senzora korištenjem jedne od navedenih inačica Kalmanovog filtra. Time se dobiva stabilnija i točnija odometrija koja može izravno biti korištena u zadacima lokalizacije i navigacije. Također, podatke dobivene od GNSS prijamnika potrebno je pretvoriti u referentni koordinatni sustav robota kako bi isti mogli biti korišteni pri globalnoj lokalizaciji robota.

Nakon pretvaranja u referentni koordinatni sustav provodi se fuzija točnije odometrije, transformiranog GNSS signala u koordinatni sustav robota te IMU senzora korištenjem proširenog ili neprikrivenog Kalmanovog filtra. Time se omogućava postizanje stabilnije i točnije lokalizacije robota u usporedbi s lokalizacijom provedenom korištenjem svakog od senzora zasebno. Slika 31. prikazuje tok signala prilikom lokalizacije robota čiji je ishod dobivanje filtrirane globalne pozicije često korištene u zadacima navigacije:



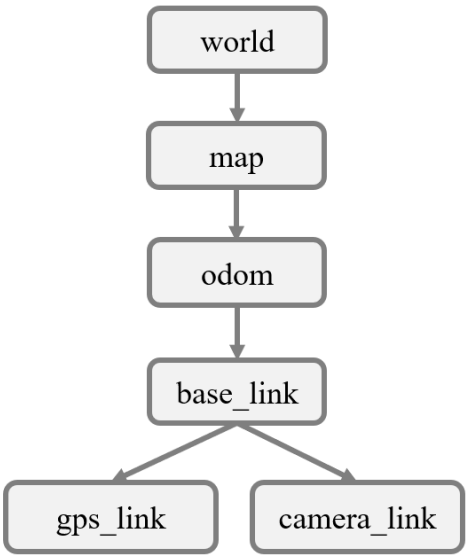
Slika 31. Dijagram nastanka signala poboljšane odometrije [59]

5.3. Robot_localization paket

Robot_localization [59] je paket ROS2 sustava koji donosi alate namijenjene fuziji i filtriranju podataka mjerenih senzorima s ciljem osiguravanja preciznih i stabilnih podataka u mobilnoj robotici. U suštini, robot_localization paket sadržava dva čvora namijenjenih nelinearnoj estimaciji stanja, odnosno ekf_localization_node i ukf_localization_node čvorove. Čvorovi za estimaciju stanja prate 15-dimenzionalni vektor stanja robotskog sustava, odnosno $X, Y, Z, roll, pitch, yaw, \dot{X}, \dot{Y}, \dot{Z}, \dot{roll}, \dot{pitch}, \dot{yaw}, \ddot{X}, \ddot{Y}, \ddot{Z}$ putem kojeg se podešava ponašanje cijelog paketa.

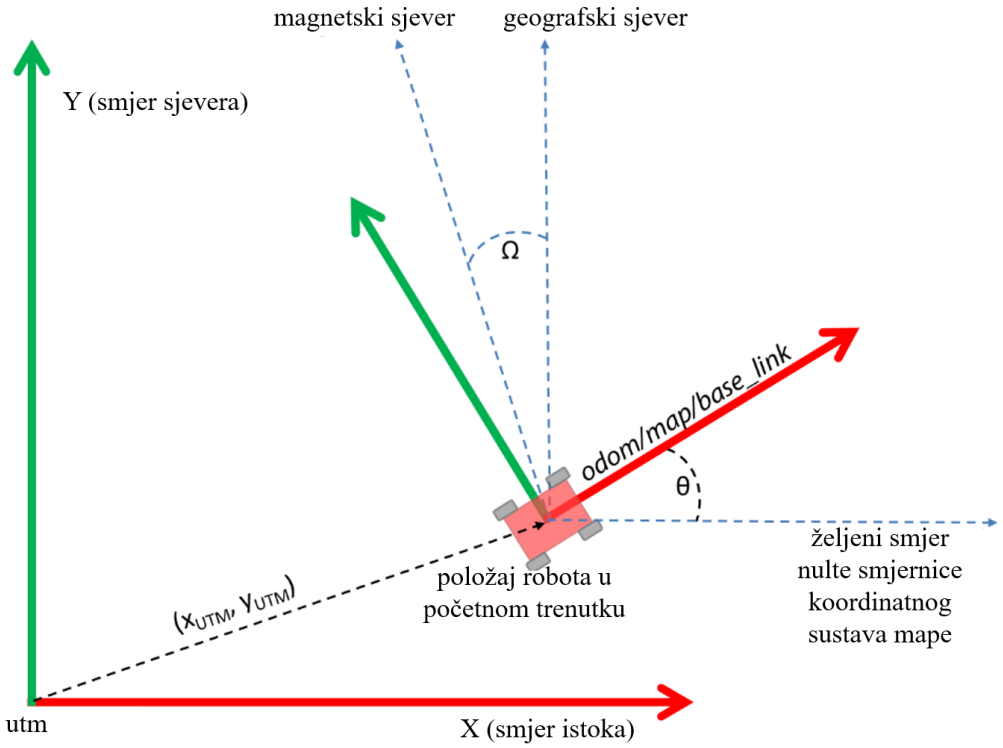
Paket koristi REP-105 konvenciju [46] postavljanja koordinatnih sustava koji uvodi standarde prilikom izrade softverskih komponenata kako bi iste mogle biti preglednije integrirane i ponovno korištene. Također, korištenje konvencija doprinosi jednostavnoj implementaciji razvijenih algoritama na većinu mobilnih platformi koje su kompatibilne s korištenom opremom. Nadalje, paket ima mogućnost fuzije podataka s više izvora od kojih svaki ima poznatu mjeru nesigurnosti što za posljedicu ima dobivanje stabilnijeg i točnijeg signala nego signal svakog od senzora zasebno. Dodatno, paket sadržava

navsat_transform_node čvor koji se koristi prilikom integracije GNSS podataka u sustav lokalizacije robota [60]. Slika 32. prikazuje nazive koordinatnih sustava korištenih u globalnoj lokalizaciji, podešeni prema REP-105 konvenciji:



Slika 32. Transformacije koordinatnih sustava u zadatku globalne lokalizacije [59]

Nadalje, Slika 33. prikazuje koordinatne sustave mobilnog robota prilagođene konvencijama korištenih u paketu robot_localization:



Slika 33. Prikaz koordinatnih sustava mobilnih robota [61]

Transformacija između koordinatnog sustava svijeta, odnosno *world* te lokalne mape robotskog sustava, odnosno *map* provodi se korištenjem navsat_transform_node čvora. Odnos koordinatnih sustava koordinatnog sustava odometrije robota, odnosno *odom* te baze robota odnosno *base_link* određen je korištenjem čvora proširenog Kalmanovog filtra čija je zadaća fuzija podataka odometrije i IMU senzora. Transformacija koordinatnih sustava *map* i *base_link* dobiva se fuzijom podataka o globalnoj poziciji dobivenoj od navsat_transform_node čvora, podacima dobivenim od IMU senzora te filtrirane odometrije korištenjem jedne od spomenutih inačica Kalmanovog filtra.

Base_link koordinatni sustav predstavlja sustav pričvršćen na robota koji se kreće dok su odom i map koordinatni sustavi fiksirani u prostoru te u pravilu imaju svoja ishodišta u početnoj poziciji i orijentaciji robota. To znači da u početnom trenutku kada je robot upaljen, *base_link, odom* i *map* koordinatni sustavi su međusobno poravnati. Koordinatni sustav zemlje nazvan je *utm* (engl. *Universal Transverse Mercator, hrv. univerzalni transverzalni merkator*) te se asocira s podacima dobivenim od GNSS prijamnika kojim se određuje pomak robota od nulte paralele i nultog meridijana. Time se dobiva informacija o geografskoj širinu i dužinu robota. Svrha navsat_transform_node čvora je stvaranje transformacije koordinatnog sustava zemlje, odnosno *utm* u lokalni koordinatni sustav robota, odnosno *map* koordinatni sustav. Kao što je spomenuto, transformacija inicijalne pozicije robota u *utm* koordinatnom sustavu zahtjeva poznavanje geografske duljine odnosno x te geografske širine odnosno y kao i kuta zakreta u odnosu na pravac koji je usmjeren prema istoku. Istovremeno y os *utm* koordinatnog sustava usmjerena je prema sjeveru, dok je z os usmjerena u podlogu, što odgovara desnokretnom koordinatnom sustavu propisanom REP-105 konvencijom koordinatnih sustava mobilnih platformi [46] kao i REP-103 konvenciji odnosno konvenciji standardnih jedinica mjere te koordinatnih sustava [47].

5.3.1. Navsat_transform_node čvor

Navsat_transform_node [61] je čvor paketa robot_localization [59] čija je glavna namjena transformacija GNSS podataka o poziciji u koordinatni sustav robota. Čvor zahtjeva poznavanje tri izvora informacija, odnosno trenutnu poziciju robota u njegovom nazivnom koordinatnom sustavu, orijentaciju u odnosu na magnetski sjever te geografske koordinate izražene kao par zemljopisna širina/dužina uz opcionalno dodavanje geografske visine. Dobava podataka se može vršiti na tri različita načina. Prvi način podrazumijeva da čvor tijekom cijelog vremena očitava senzore te djeluje prema podacima koje dobiva od istih, što je korišteno u nastavku ovog rada. Drugi način dobave podataka odnosi se na zadavanje referenci putem ROS2 *datum* parametara, što je detaljnije opisano u [61]. Posljednji način definiranja podataka navsat_transform_node čvoru je korištenjem ručnog unosa podataka putem servisa što može biti korisno u procesu testiranja rada sustava.

Prvi korak u konfiguraciji paketa je konfiguracija čvorova koji su korišteni u zadacima lokalizacije. U prilogu I. simbolički su prikazane komponente kao i teme putem kojih se ostvaruje globalna lokalizacija korištenjem navsat_transform_node čvora. Čvor obrade GNSS podataka zahtijeva podešavanje ulaznog priključka na kojeg je spojen prijamnik kao i brzinu prijenosa podataka kako bi isti bio sposoban ispravno interpretirati dobivene podatke. Obrada GNSS podataka provodi se korištenjem ros2_ublox paketa [18] koji kao izlaz osigurava poruku tipa *sensor_msgs/NavSatFix* potrebnu za rad navsat_transform_node čvora. Nadalje, tijekom upotrebe IMU podataka treba voditi računa da korištena konfiguracija IMU senzora odgovara REP-105 konvenciji [46] kako bi se osigurao ispravan rad sustava. Ako korišteni IMU uređaj ne odgovara REP-105 konvenciji, dobivene podatke potrebno je prilagoditi prije implementacije u navsat_transform_node čvoru kako bi isti ispravno radio. Nadalje, potrebno je provesti konfiguraciju navsat_transform_node čvora. To podrazumijeva podešavanje imena gore spomenutih tema te podešavanje statičkih transformacija komponenata kako bi odgovarali korištenoj konfiguraciji robota. Pri postavljanju navsat_transform_node čvora potrebno je definirati sljedeće parametre:

- *frequency* [Hz]– Parametar kojim se definira učestalost objavljivanja filtrirane vrijednosti globalne pozicije.
- *delay* [s]– Parametar kojim se definira vrijeme odgode početka objavljivanja globalne transformacije.

- *magnetic_declination_radians* [rad] – Parametar kojim se definira magnetska deklinacija trenutne lokacije robota.
- *yaw_offset* [rad]– Parametar kojim se definira kut skretanja dobivenog od IMU senzora kako bi se zadovoljila konvencija propisana paketom robot_localization [56].
- *zero_altitude* – Parametar kojim se uključuje planarni model robota, odnosno iz jednadžbe gibanja robota uklanja se parametar o nadmorskoj visini.
- *broadcas_utm_transform* – Parametar kojim se uključuje objavljivanje transformacije sustava *world_frame* → *utm* čija svrha je detaljnije opisana na početku ovog odlomka.
- *broadcast_utm_transform_as_parent_frame* – Ako je ovaj parametar uključen, čvor će objavljivati *utm* → *world_frame* transformaciju.
- *publish_filtered_gps* – Ukoliko je parametar uključen, filtrirana globalna odometrija se pretvara u zemljopisnu širinu i dužinu.
- *use_odometry_yaw* – Ukoliko je parametar uključen, čvor ignorira podatke dobivene od IMU senzora te se smjer određuje samo korištenjem odometrije. Odometrija robota mora biti usklađena s koordinatnim sustavom svijeta što je opisano u uvodu ovog odlomka.
- *wait_for_datum* – Ukoliko je parametar uključen, sustav uzima predefinirane podatke o lokaciji. Ako pak podaci o lokaciji nisu definirani, sustav čeka pozivanje istih preko poziva servisa, dok se do poziva servisa kao ishodište uzima prva pristigla koordinata dobivena od strane GNSS uređaja.
- *datum* – Umjesto korištenja prve pristigle GNSS lokacije te orijentacije određene korištenjem IMU senzora, datum parametar omogućuje ručno postavljanje zemljopisne širine i dužine na kojoj se nalazi robot.

Nadalje, potrebno je razmotriti dva korekcijska faktora odnosno korekcijski faktor magnetske deklinacije te odmak u rotaciji oko z osi. Pri korištenju razne opreme postoji mogućnost da IMU driver ne dozvoljava korištenje korekcijskog faktora magnetske deklinacije ili pak da isti izravno ne dopušta implementaciju odmaka rotacije. Čvor navsat_transform_node izravno nudi rješenje za nadilaženje tih nedostataka dodavanjem dvaju parametara u opis čvora koji su potom korišteni za spomenute namjene. Parametar magnetske deklinacije moguće je odrediti korištenjem dostupnih alata poput NOAA kalkulatora magnetskog polja [63]. U prilogu III. nalazi se izračun magnetske deklinacije za lokaciju testiranja sustava što bi moglo biti izravno

implementirano ukoliko bi se trenutno korišteni IMU senzor zamijenio sa senzorom koji posjeduje implementirani magnetometar. Pokretanjem navsat_transform_node čvora dobiva se transformacija *utm* koordinatnog sustava svijeta u lokalni koordinatni sustav robota naziva *map*. Ona je od značaja jer je će sve buduće lokacije dobivene s GNSS prijamnik biti prikazane u lokalnom koordinatnom sustavu okruženja robota kako bi se iste mogle izravno koristiti [61].

5.3.2. Ekf_filter_node čvor

Pri radu s više izvora informacija, potrebno je voditi računa o načinu fuzije istih. Podaci dobiveni od GNSS prijemnika u većini slučajeva ne mogu biti izravno primijenjeni u zadacima navigacije. Zbog osvježavanja pozicije frekvencijom od 1 Hz te diskontinuiranih skokova iste je potrebno filtrirati kako bi isti mogli biti korišteni u radu algoritama navigacije. Uz problem diskontinuiranih skokova, pri lokalizaciji robota značajan negativni utjecaj ima povećanje pogreške odometrije proporcionalno odmaku od početnog položaja te prolasku vremena od resetiranja iste. Ti nedostaci se nastoje ukloniti korištenjem proširenih Kalmanovih filtra. Kao što je spomenuto na početku ovog odlomka, prilikom fuzije podataka globalne pozicije bit će korištena dva Kalmanova filtera. Prvi filtar se koristi za fuziju odometrije te IMU podataka, dok je drugi filter zadužen za fuziju GNSS podataka s IMU podacima te odometrijom. Pri postavljanju ekf_filter_node čvora potrebno je definirati sljedeće parametre:

- *frequency* [Hz] – Parametar kojim se definira učestalost objavljivanja filtrirane vrijednosti estimirane pozicije.
- *sensor_timeout* [s] – Period kojim je definiran maksimalni vremenski interval odnosno minimalna frekvencija dobave podataka sa senzora za koju filter estimira izlazne varijable.
- *two_d_mode* – Ako je parametar uključen, čvor svoje estimacije temelji na 2D modelu gibanja umjesto zadanog 3D svesmjernog modela gibanja. Često se koristi u planarnom gibanju kako bi se izbjegle nepotrebne varijacije uslijed korištenja zašumljenih signala.
- *transform_time_offset* [s] – Parametar kojim se definira vremenski odmak generiranih transformacija ekf_filter_node čvora. Koristi se ako neki paket zahtjeva kašnjenje transformacija.
- *transform_timeout* [s] – Parametar kojim se određuje period koji je potrebno čekati do pojave transformacije između koordinatnih sustava korištenih senzora.

- *print_diagnostics* – Parametar kojim se uključuje slanje dijagnostičkih poruka. Čvor potom objavljuje dijagnostičke poruke na */diagnostics* temu s pomoću kojih je potom moguće jednostavnije otkriti greške u postavkama i ulaznim podacima.
- *debug* – Parametar kojim se uključuje objavljivanje matrica korištenih prilikom računanja pojačanja Kalmanovog filtra. Korištenje ove značajke ima značajan utjecaj na smanjenje performansi čvora te se u pravilu koristi samo pri otklanjanju pogrešaka.
- *debug_out_file* – Parametar kojim se određuje putanja do lokacije na kojoj se spremaju dijagnostički podaci.
- *permit_corrected_publication* – Parametar kojim se omogućava ažuriranje podataka o stanju koristeći stara mjerenja senzora.
- *publish_acceleration* – Parametar kojim se objavljuju varijable stanja vezane za ubrzanje robota.
- *publish_tf* – Parametar kojim se omogućava slanje transformacija koordinatnih sustava na /tf temu što naknadno može biti prikazano u Rviz2 sučelju.
- *map_frame* – Parametar kojim se određuje fiksni koordinatni sustav mape (Slika 32.).
- *odom_frame* – Parametar kojim se određuje fiksni koordinatni sustav odometrije robota (Slika 32.).
- *base_link_frame* –Parametar kojim se određuje pomični koordinatni sustav vezan za bazu robota (Slika 32.).
- *world_frame* – Parametar kojim se određuje globalni koordinatni sustav (Slika 32.).
- odom0 – Parametar kojim se određuje tema na koju se objavljuju podaci o odometriji.
- *odom0_config* – Parametar kojim se određuje korištenje pojedinih podataka odom0 senzora prilikom estimacije varijabli stanja.
- *odom_queue_size* – Parametar kojim se definira sjedinjenje većeg broja poruka pristiglih na temu odometrije, odnosno podešava se nisko propusni filtar kojim se anulira utjecaj visokofrekventnih šumova senzora.
- *odom0_differentia*l – Parametar kojim se omogućava integriranje podataka više senzora koji mjere iste varijable stanja.
- *odom0_relative* – Parametar kojim se određuje inicijalno očitanje teme odometrije koje je u nastavku tretirano kao referenca za sva ostala mjerenja. Ako je parametar uključen,

čvor prilikom pokretanja uzima trenutnu vrijednost odometrije koja u nastavku predstavlja referentno stanje.

- *odom0_pose_rejection_threshold* – Parametar kojim se definiraju granice odbacivanja očitanja odometrije koji odstupaju od očekivanih mjerenja.
- *odom0_twist_rejection_threshold* – Parametar kojim se definiraju granice odbacivanja očitanja brzina koji odstupaju od očekivanih mjerenja.
- *twist0* – Parametar kojim se određuje tema na koju se objavljuju podaci o brzini.
- *twist0_config* – Parametar kojim se određuje korištenje pojedinih podataka senzora mjerenja brzine prilikom estimacije varijabli stanja.
- *twist0_queue_size* – Parametar kojim se definira sjedinjenje većeg broja poruka objavljenih na temu brzine, odnosno podešava se nisko propusni filtar kojim se anulira utjecaj visokofrekventnih šumova senzora.
- *twist0_rejection_threshold* – Parametar kojim se definiraju granice za odbacivanje očitanja podataka o brzini koji odstupaju od očekivanih mjerenja.
- *imu0* – Parametar kojim se određuje tema na koju se objavljuju podaci IMU senzora.
- *imu0_config* – Parametar kojim se određuje korištenje pojedinih podataka *imu0* senzora prilikom estimacije varijabli stanja.
- *imu0_differential* – Parametar kojim se omogućava integriranje podataka više senzora koji mjere iste varijable stanja.
- *imu0_relative* – Parametar kojim se određuje inicijalno očitanje teme mjerenja IMU senzora koje je u nastavku tretirano kao referenca za sva ostala mjerenja.
- *imu0_queue_size* – Parametar kojim se definira sjedinjenje većeg broja poruka objavljenih na temu IMU očitanja, odnosno podešava se nisko propusni filtar kojim se anulira utjecaj visokofrekventnih šumova senzora.
- *imu0_pose_rejection_threshold* – Parametar kojim se definiraju granice odbacivanja očitanja mjerenja pozicije korištenjem IMU senzora koji odstupaju od očekivanih mjerenja.
- *imu0_twist_rejection_threshold* – Parametar kojim se definiraju granice za odbacivanje očitanja mjerenja brzine korištenjem IMU senzora koji odstupaju od očekivanih mjerenja.

- *imu0_linear_acceleration_rejection_threshold* – Parametar kojim se definiraju granice za odbacivanje očitanja mjerenja ubrzanja korištenjem IMU senzora koji odstupaju od očekivanih mjerenja.
- *imu0_remove_gravitational_acceleration* – Parametar kojim se uključuje uklanjanje utjecaja gravitacije na IMU senzor.
- *use_control* – Parametar koji omogućava estimaciju akceleracije između dva koraka predikcije/korekcije. Koristi se ako sustavu nije dostupan izvor podataka kojim se određuje akceleracija robota.
- *stamped_control* – Parametar koji omogućava korištenje podataka o brzini u formi twist poruka.
- *control_timeout* [s] – Parametar kojim se određuje period držanja posljednjeg pristiglog upravljačkog parametra.
- *control_config* – Vektor parametara kojim se određuju komponente vektora brzine kojima se upravlja. Vektor brzine definira se prema sljedećoj strukturi: $\dot{X}$, $\dot{Y}$, $\dot{Z}$, $\dot{roll}$, $\dot{pitch}$, $\dot{yaw}$.. Analizom parametara komponenti brzina određuje se model robota kojim se potom koristi pri estimaciji stanja Kalmanovim filterom.
- *acceleration_limits* – Parametar kojim se određuje maksimalna vrijednost ubrzanja koju određeni robot može postići tijekom svog gibanja.
- *deceleration_limits* – Parametar kojim se određuje maksimalna vrijednost usporenja koju određeni robot može postići tijekom svog gibanja.
- *acceleration_gains* – Vektor parametara kojima se omogućava skaliranje intenziteta ubrzanja pojedinih varijabli stanja.
- *deceleration_gains* – Vektor parametara kojima se omogućava skaliranje intenziteta usporenja pojedinih varijabli stanja.
- *process_noise_covariance* – Matrica parametara kojom se određuju kovarijance mjerenja pojedinih stanja.
- *initial_estimate_covariance* – Matrica parametara kojom se određuju inicijalne vrijednosti matrice kovarijance pogreške procjene stanja.

Čvor Proširenog Kalmanovog filtra može biti korišten, kako za lokalnu tako i za globalnu fuziju podataka lokalizacije. Detaljni opis parametara korištenih za globalno i lokalno filtriranje

podataka dan je u biblioteci koda paketa [56]. Nadalje, čvor omogućava korištenje više ulaza za pojedini tip poruke. U praksi se često koriste sustavi s dva GNSS prijamnika čime se sustavu osigurava dobava informacije o globalnoj orijentaciji robota. Također, pravilna kombinacija više IMU senzora može rezultirati točnijom estimacijom parametara vektora stanja robota. Ako se neka varijabla stanja dobiva od više senzora ili estimatora potrebno im je ispravno definirati kovarijance. Nepravilno definirane kovarijance u tom slučaju rezultiraju skokove u estimiranim stanjima što je izrazito nepoželjno u zadacima lokalizacije. Također, *ekf_filter_node* čvor nudi mogućnost kombinacije više izvora informacija gdje je izlaz nastao kao kombinacija istih.

Kovarijancom smetnji u procesu određuje se intenzitet promjene pojedinih parametara u vremenu. Ako se parametar postavi na mali broj, filtar će biti osjetljiviji na promjene u stanju sustava čime se mogu preciznije pratiti mali pomaci, dok istovremeno isti može biti previše osjetljiv na šum i nesavršenosti mjerenja. S druge pak strane, ako je kovarijanca postavljena visoko, filtar će biti manje osjetljiv na šumove, ali može propustiti stvarne promjene u stanju sustava. Podešavanje ovog parametra često se svodi na metodu pokušaja i pogrešaka prilikom postavljanja optimalnog balansa između prilagodljivosti sustava na promjene i odbacivanja šuma ili nesavršenih mjerenja. Inicijalnom kovarijancom određuje se nesigurnost sustava u početnom stanju. Isto kao i kod matrice kovarijanci smetnji, pri podešavanju inicijalnih kovarijanci manje vrijednosti definiraju veću pouzdanost pojedinim mjerenjima, dok veće vrijednosti parametara definiraju veću nesigurnost estimacije varijabli stanja. Ispravno podešavanje matrice inicijalnih kovarijanci osigurava brže konvergiranje estimiranih varijabli što se izravno preslikava na performanse korištenog filtera. Za potrebe ovog rada, analizirana je priroda šuma svakog od izvora podataka te su parametri kovarijanci iterativno podešeni kako bi se postiglo zadovoljavajuće ponašanje sustava. Također, dobra praksa je izbjegavanje korištenja kutnih brzina dobivenih od IMU senzora ako se ne koriste visokokvalitetni senzori. Razlog tome je što su mjerenja IMU senzora u pravilu značajno zašumljena. Integracijom takvog signala bez prethodno implementiranog nisko-pojasnog filtera značajno povećava šumove te sama estimacija temeljena na takvim ulaznim signalima gubi svoj smisao. Parametri matrice kovarijanci smetnji te matrice inicijalnih kovarijanci postavljene su prema preporukama danih u [60], uz dodatno podešavanje tijekom testiranja na sustavu. Detaljan opis svih mogućnosti čvora proširenog Kalmanovog filtra dan je u biblioteci koda paketa [56].

5.3.3. Ukf_filter_node čvor

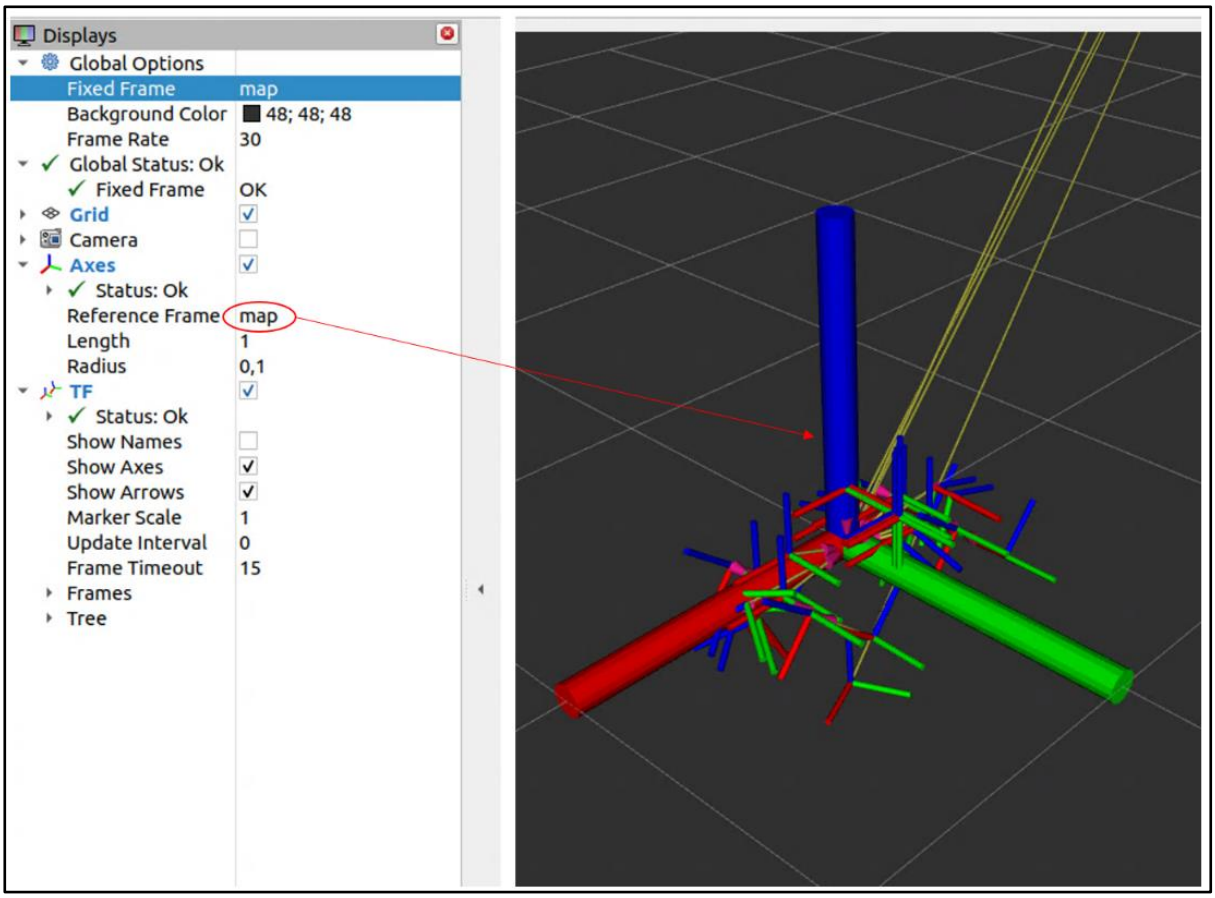
Konfiguracijski parametri ukf_filter_node čvora u velikoj mjeri odgovaraju onima opisanim u odlomku *5.3.2. Ekf_filter_node čvor* uz tri nadodana parametra koja su svojstvena samo neprikrivenom Kalmanovom filtru:

- *alpha* – Parametar kojim se određuje faktor širenja sigma točaka pri linearizaciji neprikrivenim Kalmanovim filtrom. Povećanjem vrijednosti ovog parametra smanjuje se računalno vrijeme izvršavanja koraka Kalmanovog uz posljedicu porasta varijanci pri estimiranju novih stanja sustava.
- *kappa* – Parametar kojim se određuje raspored sigma točaka oko srednje vrijednosti. Povećanje ovog parametra rezultira približavanju sigma točaka njihovoj srednjoj vrijednosti, što može ubrzati rad filtra, dok istovremeno filtar može postati nestabilan zbog loše procjene srednje vrijednosti stanja sustava.
- *beta* – Parametar koji se koristi pri estimaciji kovarijanci mjerenja u procesu filtriranja. Često se koristi u sustavima višeg reda kako bi se postigla željena distribucija stanja sustava.

Detaljan opis svih mogućnosti čvora neprikrivenog Kalmanovog filtra dan je u biblioteci koda paketa [56].

5.4. Lokalizacija robota u vanjskom okruženju

Nakon pokretanja čvorova opisanih u prethodnim odlomcima uspostavlja se lokalizacija robota u koordinatnom sustavu svijeta. Prikazani koordinatni sustav u inicijalnom stanju ujedno predstavlja koordinatne sustave odometrije, baze robota kao i bazu koordinatnog sustava mape, što je detaljnije opisano u prethodnim odlomcima. Žute strelice prikazuju transformaciju u odnosu na globalni koordinatni sustav određenu podacima dobivenim od strane GNSS prijamnika. Slika 34. prikazuje koordinatne sustave robota nakon provedene lokalizacije:



Slika 34. Koordinatni sustavi robota nakon provedene globalne lokalizacije

Na repozitoriju diplomskog rada nalazi se prikaz stabla transformacija korištenih koordinatnih sustava nakon pokretanja čvorova lokalizacije mobilnog robota temeljem na temelju globalne pozicije.

6. GLOBALNO PLANIRANJE PUTANJE

U zadacima globalne navigacije mobilnog robota, planiranje putanje ima ključnu ulogu prilikom osiguravanja učinkovitog i sigurnog kretanja robota od početne točke do željenog cilja u dinamičnom okruženju. Globalno planiranje putanje odnosi se na strategiju koja na temelju informacija okolini, generira optimalnu putanju koju robot treba slijediti za postizanje zadanog cilja. Proces globalne navigacije obuhvaća analizu globalne mape okoline, prepoznavanje prepreka kao i drugih karakteristika terena uz kontinuirano provođenje planiranja sigurnog puta koji zadovoljava specifične zahtjeve robota i okoline. U ovom odlomku prikazane su metode globalnog i lokalnog planiranja putanje kao ključnih aspekata uspješne navigacije odnosno autonomnog kretanja robota. Paketi koji su navedeni kao osnova za izvođenje zadataka navigacije, uz potrebne modifikacije priloženi su na repozitoriju diplomskog rada [2].

6.1. Nav2 paket

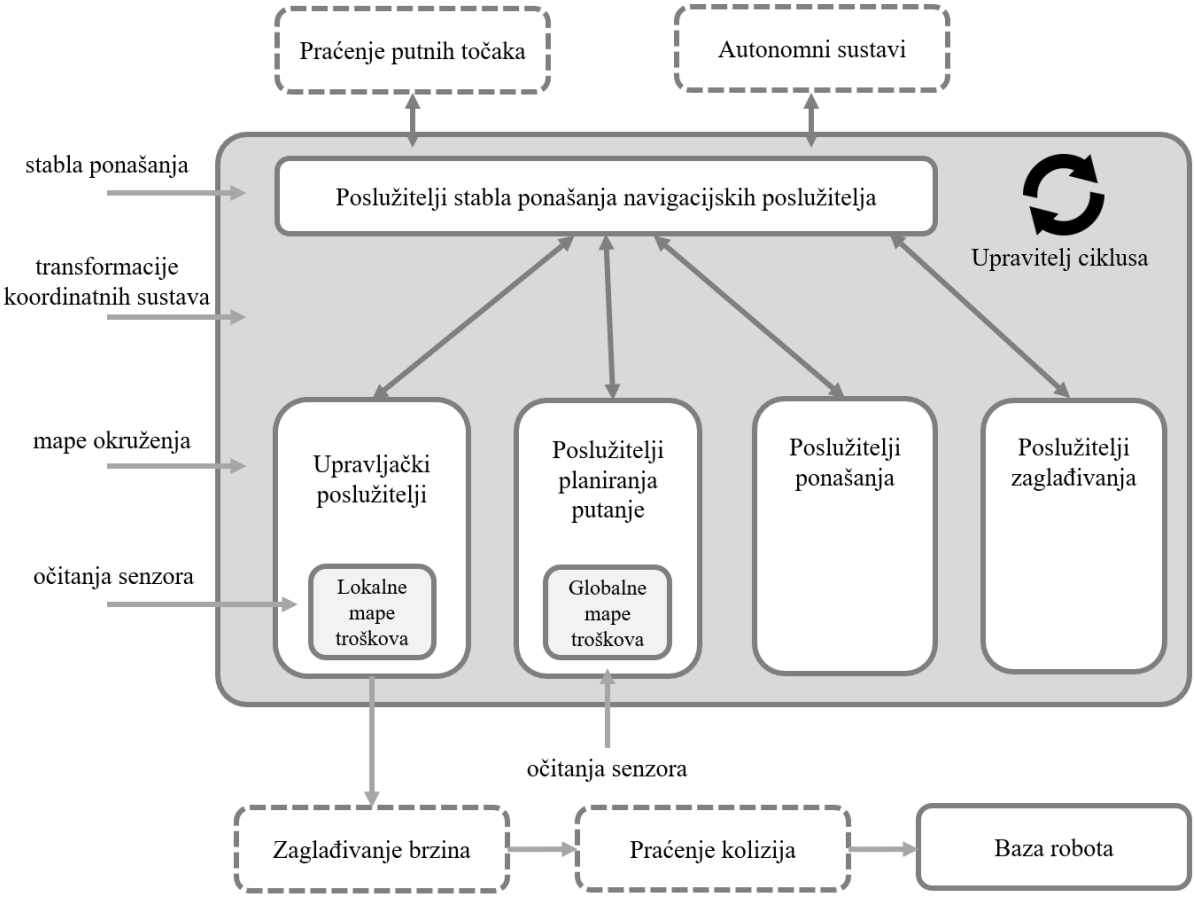
Paket nav2 [64] sastoji se od biblioteka, čvorova i alata namijenjenih autonomnoj navigaciji mobilnih robota. Paket pruža brojne funkcionalnosti koje robotima omogućavaju planiranje puta, izbjegavanje prepreka, praćenje željene putanje što zajedno doprinosi autonomnom kretanju robota koji se navodi kao temelj za izvođenje određenih zadaća bez potrebe za stalnim ljudskim nadzorom. Konceptima koje donosi nav2 paket nadilaze se nedostaci koji su primijećeni tijekom dugogodišnje upotrebe prve inačice paketa navigacije [65] namijenjenog ROS okruženju. Paket sadržava mape servera, globalne i lokalne planere, servise težinskih mapa, servise za navigaciju, čvorove za transformaciju, čvorove za rad sa senzorima te pakete za vizualizaciju. Također, nav2 paket koristi stablo ponašanja što uvelike olakšava primjenu raznih obrazaca ponašanja namijenjenih za rad u dinamičkom okolišu uz korištenje brojnih senzora. U svom radu paket koristi okruženja sigurnih poruka koje donosi ROS2 sustav što doprinosi sigurnosnom aspektu primjene kao i upravljanju životnim ciklusima programa. Također, algoritmi koje donosi nav2 paket podvrgnuti su dugotrajnim odnosno maratonskim testiranjima u dinamičkom okruženju kako bi se validirala robusnost istih [66].

Paket nav2 u svom radu koristi sučelje za rad sa stablima ponašanja koji su zaduženi za vođenje zadataka planiranja, upravljanja kao i zadataka oporavka. Model cijelog sustava navigacije leži na izvršavanju čvorova stabla ponašanja koji potom pozivaju poslužitelje koji su zaduženi za izvršavanje nekog zadatka koristeći jedan od brojnih dostupnih algoritama koje

paket donosi. Svaki poslužitelj koristi standardno sučelje za dodatke koje olakšava jednostavnu integraciju novih algoritama u radno okruženje paketa nav2. Arhitektura paketa temeljena na radu nezavisnih poslužitelja omogućava podjelu zadataka, odnosno izvođenje algoritama na više jezgri što je osnova rada algoritma u stvarnom vremenu uz niska kašnjenja [67]. U nastavku odlomka opisani su osnovni elementi nav2 paketa.

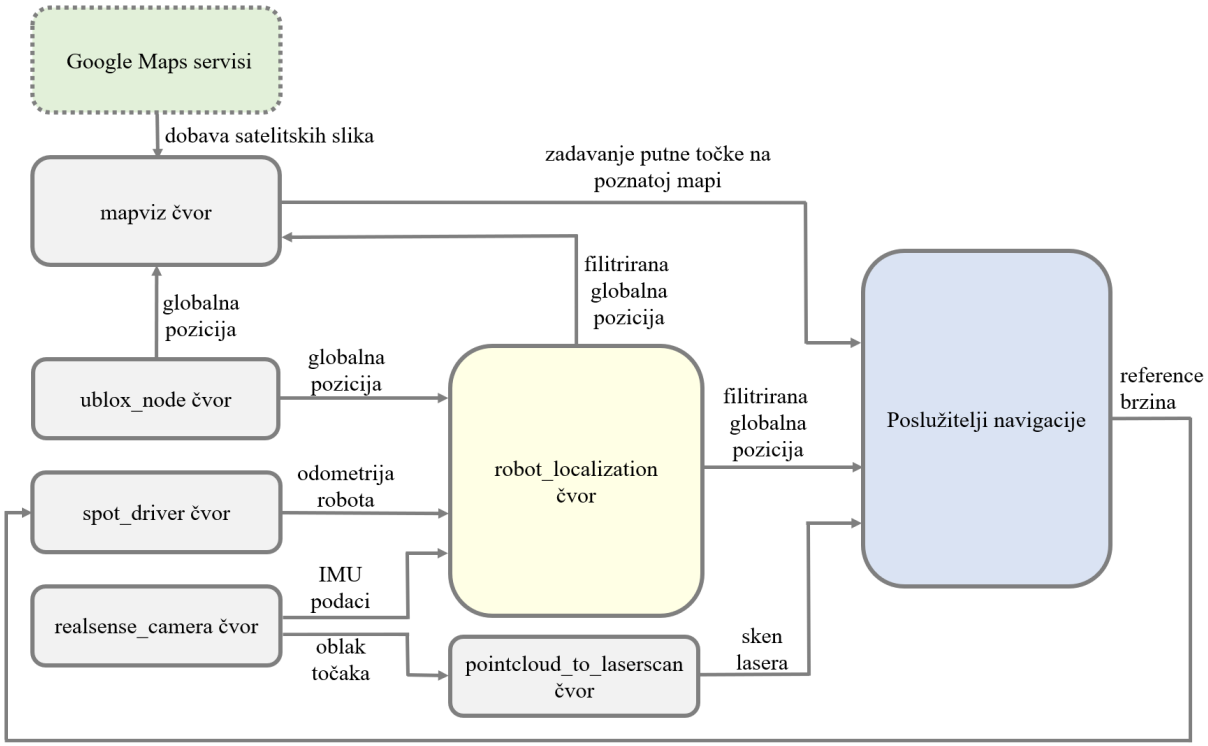
6.1.1. Modularni pristup nav2 paketa

Kao što je spomenuto u ovom odlomku, od zadataka navigacije namijenjenih za rad s velikim brojem raznih robotskih sustava zahtjeva se značajan stupanj modularnosti kako bi se isti mogli bez značajnih promjena implementirati u razne sustave. Paket nav2 donosi dva obrasca izrade sustava s pomoću kojih se ostvaruje modularnost zadataka. To su redom navigatori korištenjem stabla ponašanja te asinkroni poslužitelji prilagođeni zadatku [67]. Slika 35. konceptualno prikazuje međudjelovanje poslužitelja, odnosno servera nav2 paketa:



Slika 35. Koncept djelovanja poslužitelja nav2 paketa [68]

Slika 36. shematski prikazuje korištenje navigacijskih poslužitelja sa čvorovima spomenutim u prethodnim poglavljima uz navođenje tema putem kojih se komunikacija odvija:



Slika 36. Prikaz međusobne interakcije čvorova i poslužitelja korištenih u zadacima globalne navigacije.

Osnovni pristup kao i arhitektura nav2 paketa dani su u nastavku odlomka. Također, u prilogu IV nalazi se dijagram toka koji konceptualno prikazuje rad poslužitelja koji su opisani u nastavku poglavlja.

6.1.1.1. Stabla ponašanja

Stabla ponašanja (engl. *behaviour trees*) ključne su komponente u nav2 paketu. Koriste se pri definiranju ponašanja robota tijekom izvršavanja dugotrajnih zadataka kao što su zadaci navigacije [70]. Njihova hijerarhijska struktura olakšava modeliranje složenog ponašanja te je zaslužna za donošenje odluka na osnovu trenutnog stanja robota i okoline. Iz perspektive nav2 paketa, stabla ponašanja koriste se za definiranje ponašanja robota tijekom svih faza navigacije, što uključuje planiranje i praćenje putanje, izbjegavanje stacionarnih te prepreka u pokretu kao i ostalih radnji zaduženih za oporavak sustava uslijed djelovanja raznih poremećaja. U svom radu, stabla ponašanja pozivaju planere putanje, kontrolere te servise oporavka (engl. *recovery services*) putem ROS2 sučelja za upravljanje radnjama (engl. *action interface*). Promjenu ponašanja algoritama navigacije moguće je postići izmjenom stabla ponašanja. Takav pristup

omogućava relativno jednostavnu promjenu ponašanja algoritma navigacije bez potrebe za izmjenom programskog koda poslužitelja. Također, korištenjem stabala ponašanja omogućava se upravljanje radnjama za oporavak što za ishod ima jedinstveni odgovor pri pojavi kvara pojedinih poslužitelja te samim time izvlačenje robotskog sustava iz nepovoljne situacije. Prednost korištenja stabla ponašanja u zadacima navigacije je mogućnost proširenja sustava te promjena konfiguracije robota bez potrebe za promjene programskog koda niske razine zaduženih za izvođenje raznih zadataka. Paket nav2 koristi BehaviourTree.CPP [71] knjižnicu stabla ponašanja s pomoću koje se omogućuje pregledni prikaz stabla opisanog XML datotekom te također pruža mogućnost stvaranja i korištenja podstabala. Drugim riječima, stabla ponašanja načinjena korištenjem nav2 paketa mogu biti izravno implementirana kao dodatni čvorovi u neka druga stabla ponašanja više razine [68].

6.1.1.2. Akcijski poslužitelji

Akcijski poslužitelji (engl. *action servers*) često su korištena metoda upravljanja dugotrajnih zadataka između različitih čvorova, primjerice različitih zadataka navigacije. Upotrebom akcijskih poslužitelja omogućeno je izvršavanje pojedinih radnji bez potrebe postojanja sučelja za teme. U principu, akcijski poslužitelji su vrlo slični servisima kanonskih poslužitelja gdje korisnik od sustava zahtijeva izvođenje nekog zadatka koji zahtijeva dulje vrijeme izvođenja kao što je primjerice pomak robota 20 metara u smjeru sjevera. Prilikom izvođenja takvih zadataka, akcijski poslužitelji omogućavaju izvršavanje istih u pozadini pružajući povratne informacije o stanju zadatka koji se trenutno izvodi. Nadalje, akcijski poslužitelji su također korišteni od strane stabla ponašanja prilikom komunikacije s malim podsustavima za izračun popratnih stvari kao što je analiza utrošaka energije pojedinih aktuatora i slično. Uz akcijske poslužitelje, ROS2 omogućava korištenje čvorova životnog vijeka (engl. *lifecycle nodes*) koji sadržavaju algoritme prijelaza stanja (engl. *state machines*). Najčešća upotreba algoritama prijelaza stanja očitava se prilikom pokretanja i gašenju ROS sustava te pri strukturiranju ponašanja sustava, otklanjanje poteškoća (engl. *debugging)* te određivanje prijelaznih pojava prilikom prekida komunikacije s poslužiteljem kako bi se spriječile daljnje greške te samim time nepovoljno ponašanje sustava [68]. U nastavku su prikazani akcijski poslužitelji namijenjeni upotrebi u zadacima navigacije.

6.1.1.3. Navigacijski poslužitelji

Navigacijski poslužitelji (engl. *navigation servers*) predstavljaju osnovnu komponentu zadatka navigacije. Koriste se za rad s priključcima namijenjenih izvršavanju pojedinih zadataka. Nav2 paket nudi četiri poslužitelja, odnosno poslužitelje planiranja putanje (engl. *planning servers),* poslužitelje ponašanja (engl. *behaviour servers*), upravljačke (engl. *control servers*) te poslužitelje za zaglađivanje (engl. *smoother servers)* koji su u nastavku ovog odlomka detaljnije objašnjeni. Detaljna konfiguracija poslužitelja dana je u odlomku *6.2. Navigacija korištenjem GNSS lokalizacije*.

Zadatak poslužitelja planiranja, odnosno globalnih planera je izračunati ostvarive te potencijalno optimalne putanju kroz prostor na temelju zadane ciljne funkcije. Ovisno o informacijama o početnoj točki te željenom cilju kao i detaljnih informacija o okolini, poslužitelj planiranja stvara optimalnu putanju koja opisuje trajektoriju kretanja kao i ključne točke kroz koje robot treba proći. Također, poslužitelj planiranja u obzir uzima dinamične uvjete okoline sagledavajući prepreke i druge faktore koji utječu na izvođenje zadataka navigacije. Drugi značajan aspekt zadatka poslužitelja planiranja je upravljanje izbjegavanjem prepreka. Poslužitelji planiranja putanje analiziraju globalnu mapu okoline kako bi identificirali prepreke te izbjegli kolizije na predloženoj putanji što rezultira sigurnim kretanjem robota. Također, poslužitelji planiranja prate napredak izvršavanja globalnog plana, pružajući informacije o postignutim točkama na putanji te eventualnim izazovima ili poteškoćama koje se javljaju prilikom kretanja robota. Algoritam planiranja putanje može biti napisan na tri osnovna načina [68]. Prvi način podrazumijeva određivanje najkraće putanje do određene pozicije u prostoru. Drugi način odabire putanje prema otprije definiranim putanjama kojima se robot smije kretati. Treća način odnosi se na određivanja putanje koja će osigurati potpunu pokrivenost slobodnog prostora, što se primjerice koristi prilikom inspekcije terena između dvije zadane točke gdje se od robota zahtijeva da prođe cijeli podobni prostor.

Poslužitelji ponašanja glavna su komponenta sustava otpornih na pogreške prilikom pojave sustavu nepoznatog vanjskog podražaja. Ponašanja prilikom pojave pogreške u sustavu kao i pojave nepoznatog stanja moraju biti pokrivena implementiranim algoritmima oporavka kako bi se omogućio nastavak normalnog rada robota nakon pojave istih. Kao primjer takvog ponašanja navode se pokreti robota prilikom zastoja zbog prisutnosti dinamičkih objekata. Robot nakon zastoja treba promijeniti svoju poziciju u nastojanju nastavka gibanja što postiže pomakom prema natrag ili rotiranjem oko svoje osi ako to okolnosti dozvoljavaju. Ova

dinamička prilagodba mobilnim robotima omogućava suočavanje s nepredviđenim situacijama što rezultira uspješnim nastavkom izvršavanja danog zadatka. Kroz sustav poslužitelja ponašanja, roboti postaju sposobni reagirati na kompleksne scenarije, pružajući pouzdanost i sigurnost prilikom navigacije kroz razna okruženja.

Upravljački poslužitelji, u ROS sustavu prve inačice nazvani kao lokalni planeri, korišteni su za obavljanje lokalnog planiranja podobne putanje u zadacima praćenja globalno određene trajektorije. Svoj rad temelje na pristupu lokalnim podacima senzora s robota kako bi se odredile upravljačke veličine kojima je moguće ostvariti praćenje reference. Poslužitelji planiranja će određivati kretanje robota u prostoru putem zadanih kriterija optimalnosti, dok će upravljački poslužitelji u svakom koraku računati lokalno ostvarivu poziciju za sljedeći korak navigacije. Upravljački poslužitelji mogu biti definirani na tri različita načina, odnosno namijenjeni praćenju trajektorije, pozicioniranje u odnosu na poznati objekt korištenjem značajki poput Aruco markera ili pak namijenjeni ponašanju koje mu određuje alat s kojim je korišteni robotski sustav u interakciji. Glavna zadaća nav2 paketa u zadaćama upravljačkih poslužitelja je izračun valjanih upravljačkih varijabli kako bi se postiglo željeno ponašanje sustava te praćenje stvarnog stanja robota u odnosu na planiranu putanju. Ako prilikom navigacije robota dođe do odstupanja ili neočekivanih prepreka, upravljački poslužitelji prilagođavaju upravljačke naredbe kako bi se osiguralo sigurno i učinkovito kretanje robota. Shodno tome, paket pruža mogućnost interakcije s raznim dostupnim lokalnim planerima kako bi isti mogli biti implementirani u upravljački poslužitelj pri izvršavanju raznih zadataka.

Poslužitelj za zaglađivanje koriste se u zadacima zaglađivanja putanje s pomoću kojih se izbjegavaju skokovi u zadanoj putanji kao i isprekidane rotacije uzrokovane zadanom stopom osvježavanja korištenih poslužitelja. Kao glavna zadaća poslužitelja zaglađivanja navodi se obrada putanje dobivene od strane upravljačkih algoritama prema definiranim kriterijima što kao ishod ima postizanje zaglađene trajektorije te samim time uglađeniji rad robotskog sustava. Također, primjenom poslužitelja zaglađivanja u velikoj mjeri doprinosi uklanjanju oscilacija i neželjenih vibracija što se također oslikava mirnijim i uglađenijim ponašanjem robota prilikom kretanja [64]. Potpuna lista dostupnih poslužitelja uz prikaz parametara podešavanja ponašanja robotskog sustava u zadacima navigacije nalazi se u [69].

6.1.1.4. Otisak robota

U većini zadataka navigacije, otisak robota (engl. *robot footprint)* prikazan je kružnicom određenog radijusa ili pak vektorom točaka koje opisuju proizvoljni poligon korišten u algoritmima planiranja i upravljanja. Paket nudi mogućnost promjene otiska robota tijekom vremena kako bi se što točnije opisao kolizijski prostor robota što može biti korisno nakon promjene položaja aktuatora ili pak hvatanja predmeta.

6.1.1.5. Reprezentacija okoline

Mape troškova (engl. *costmaps*) bitan su element u domeni navigacije mobilnih robota. Putem mapa troškova ostvaruje se vizualna reprezentacija okoline robota prikazujući troškove kretanja različitim dijelovima prostora. Troškovi se obično izražavaju numeričkim vrijednostima koje reflektiraju koliko je određeno područje pogodno ili nepogodno za kretanje robota. Mape troškova se u zadacima vanjske navigacije dijele u dvije glavne skupine, odnosno globalne i lokalne mape troškova. Globalne mape troškova (engl. *global costmaps*) obuhvaćaju širu sliku okoline i označavaju područja koja su prikladna za kretanje uz istovremeni prikaz područja koja treba izbjegavati, poput prepreka ili zidova koji su navedeni kao otprije poznatih karakteristika terena. S druge pak strane, lokalne mape troškova (engl. *local costmaps*) specifične su za trenutnu poziciju robota te se kontinuirano osvježavaju prilikom kretanja robota. Njihova zadaća svodi se na stvaranje reprezentacije trenutnog stanja u neposrednoj blizini robota s ciljem adresiranja prepreka koje se mogu pojaviti tijekom kretanja.

Mape reprezentacije okolina od velikog su značaja pri autonomnom kretanju robota jer pružaju informacije potrebne za planiranje optimalnih i podobnih putanja robotskog sustava. Algoritmi navigacije koriste ove mape kako bi odabrali najprikladniji put od trenutačne pozicije robota do ciljne točke, uzimajući u obzir troškove svakog područja. Detaljniji opis globalnih i lokalnih mapa troškova dan je u poglavlju *6.2. Navigacija korištenjem GNSS lokalizacije*.

6.2. Navigacija korištenjem GNSS lokalizacije

Prilikom konfiguracije GNSS navigacijskih poslužitelja potrebno je prilagoditi parametre u skladu s karakteristikama radnog okruženja i specifičnostima korištenog mobilnog robota. To uključuje parametre kao što su brzina osvježavanja, kalibracije te rasponi dozvoljenih odstupanja kako bi se maksimizirala preciznost navigacije robota. Također, prilikom postavljanja poslužitelja potrebno je odabrati i konfigurirati dodatke koji su korišteni od strane poslužitelja prilikom izvođenja zadataka navigacije. U nastavku odlomka definirani su parametri koje je potrebno podesiti za ispravan rad poslužitelja u zadacima navigacije. Zbog ekspeditivnosti i preglednosti ovog rada, vrijednost svih podešenih parametara dane su na repozitoriju diplomskog rada [2].

6.2.1. Postavljanje stabla ponašanja globalne navigacije

Postavljanje stabla ponašanja zadataka globalne navigacije mobilnog robota ključno je za definiranje ponašanja robota tijekom različitih navigacijskih scenarija. U ovom kontekstu, dodaci navigacijskog sustava igraju ključnu ulogu pri određivanju ponašanja robota tijekom navigacije do određene točke te kako će se nositi s globalnim putanjama.

Prilikom postavljanja stabla ponašanja, najprije je potrebno odrediti dodatke navigatora koji će utjecati na ponašanje robota za vrijeme navigacije do željene točke. Ovi dodaci mogu uključivati prioritete kao i uvjete aktivacije različitih obrazaca ponašanja, kao što su izbjegavanje prepreka ili pak održavanje sigurne udaljenosti od prepreka. Organizacija ovih ponašanja u hijerarhijsku strukturu omogućava fleksibilnost i prilagodljivost robota različitim scenarijima. Također, prilikom podešavanja stabla ponašanja odabiru se dodaci koji će biti korišteni tijekom globalne navigacije, odnosno prilikom planiranja i slijeđenja globalne putanje. Ovi dodaci mogu uključivati planiranje optimalne putanje, prilagodbu putanje na temelju dinamičkih promjena u okolini te prilagodbu ponašanja robota kako bi se smanjili rizici od sudara ili neočekivanih prepreka. Kroz postavljanje i podešavanje stabla ponašanja nastoji se optimizirati ponašanje robota u skladu sa specifičnostima zadatka globalne navigacije. Testiranje i iterativno prilagođavanje parametara stabla omogućava postizanje željene razine učinkovitosti, sigurnosti i prilagodljivosti u različitim okruženjima. Pri postavljanju stabla ponašanja potrebno je definirati sljedeće parametre:

- *global_frame* – Parametar kojim se određuje globalni koordinatni sustav. U slučaju fuzije s GNSS signalima, ovaj parametar se postavlja na koordinatni sustav mape.
- *robot_base_frame* – Parametar kojim se postavlja koordinatni sustav baze robota.
- *odom_topic* – Parametar kojim se odabire tema odometrije.
- *bt_loop_duration* [ms] – Parametar kojim se određuje trajanje svake iteracije stabla ponašanja.
- *server_timeout* [ms] – Vremenski okvir čekanja akcijskog čvora stabla ponašanja na potvrdu akcijskog poslužitelja.
- *navigators* – Parametar kojim se odabiru dodaci koje je moguće implementirati u sučelje stabla ponašanja. U ovom diplomskom radu korišteni su dodaci za navigaciju do pozicije (engl. *navigate to pose*) te navigaciju kroz definirane pozicije (engl. *navigate through poses*). Detaljna lista dostupnih dodataka nalazi se u dokumentaciji nav2 paketa [68].
- *plugin_lib_names* – Parametar kojim se određuje lista dodataka korištenih u stablu ponašanja. Detaljnu listu dostupnih dodataka korištenih za potrebe ovog rada nalazi se na repozitoriju diplomskog rada [2].
- *error_code_names* – Parametar kojim se određuju dodaci korišteni za identifikaciju pogrešaka koje nastupaju tijekom izvođenja zadataka navigacije.

Detaljnu listu parametara stabla ponašanja globalne navigacije moguće je pronaći u datotekama repozitorija nav2 paketa [64] te u dokumentaciji nav2 paketa [68], dok su vrijednosti podešenih parametara priložene na repozitoriju diplomskog rada [2].

6.2.2. Postavljanje poslužitelja globalne navigacije

U ROS2 sustavu, poslužitelji globalne navigacije predstavljaju ključnu infrastrukturu koja mobilnim robotima omogućava postizanje visoke razine autonomije te preciznosti prilikom kretanja unutar različitih okoliša. Ovi poslužitelji uključuju niz komponenti koje zajednički rade na ostvarivanju inteligentnih navigacijskih sustava, pružajući robotima sposobnost sigurnog i efikasnog kretanja u nepoznatim i dinamičnim okruženjima.

Kao što je već spomenuto u ovom odlomku, globalni planeri putanja unutar navigacijskih poslužitelja odgovorni su za generiranje optimalnih ruta od početne točke do željenog cilja,

uzimajući u obzir globalnu strukturu okoline. S druge pak strane, lokalni planeri putanja fokusiraju se na prilagodbu globalne putanje u stvarnom vremenu, omogućujući robotu prilagođavanje dinamičkim promjenama u okolini kao i izbjegavanje otprije poznatih statičkih prepreka koje mu se nađu na putu prilikom kretanja. Uz to, navigacijski poslužitelji obuhvaćaju i sustave percepcije okoline koji analiziraju senzorske podatke sa ciljem prepoznavanja prepreka, procjene položaja robota te osiguranja sigurnog kretanja robota. Ova integracija senzora čini navigacijski sustav sveobuhvatnim i prilagodljivim, dok poslužitelji za navigaciju u ROS2 sustavu pružaju okvir za jednostavno i modularno upravljanje ovim kompleksnim procesima. U nastavku odlomka opisane su ključne značajke i uloge poslužitelja za navigaciju u ROS2 sustavu uz isticanje prednosti modularne gradnje kompleksnih sustava poput sustava navigacije korištenjem međusobno nezavisnih poslužitelja raznih namjena.

6.2.2.1. Upravljački poslužitelji globalne navigacije

Upravljački poslužitelji globalne navigacije, odnosno lokalni planeri putanje spadaju u kategoriju poslužitelja izvršitelja zadataka koje donosi nav2 paket. Poslužitelji zaduženi za upravljanje položajem robota u svom radu koriste akcijski poslužitelj praćenja putanje (engl. *follow path server).* Glavna namjena poslužitelja upravljanja je provođenje izračuna upravljačkih brzina na temelju putanja dobivenih od modula planiranja globalnih i lokalnih mapa troškova. Paket navigacijskog kontrolera predstavlja sučelje namijenjeno integraciji raznih dodataka za određivanje podobne trajektorije mobilnog robota do željene pozicije u prostoru bez kolizija. Putem upravljačkih poslužitelja moguće je integrirati prethodno razvijene napredne alate globalnih i lokalnih planiranja putanje te percepciju okoline integracijom senzorskih informacija koji su jedan od preduvjeta pri gradnji sustava precizne i autonomne navigacije.

Kao što je već spomenuto u ovom odlomku, globalni planeri putanja unutar nav2 poslužitelja odgovorni su za generiranje podobnih ruta na temelju globalne strukture okoline, uzimajući u obzir početnu i ciljnu točku robota. S druge pak strane, lokalni planeri putanja korišteni su za prilagođavanje globalno generiranih putanja u stvarnom vremenu, uzimajući u obzir lokalne uvjete, dinamičke promjene u okolini i detekciju prepreka. Korištenjem ovih planera omogućeno je strateško planiranje kretanja robota što za posljedicu ima stvaranje globalno optimalnih putanja prema odabranim kriterijima koje se potom koriste se kao vodilja za kretanje kroz šire prostorne domene. Dinamička prilagodba globalne putanje trenutnom

okruženju robota omogućava ostvarivanje sigurnog i efikasnog kretanja unutar nepoznatih ili dinamičkih prostora.

U ovom radu je kao algoritam lokalnog planiranja korišten DWB upravljač (engl. *Dynamic Window Approach Base, hrv. osnovni pristup dinamičkom prozoru*). DWB upravljač je nastao kao izvedenica David Lu upravljača prilagođenog za rad u ROS2 sustavu uz korištenje dinamičkog prozora. Glavna namjena DWB planera se očitava u prilagodbi putanje robota na temelju informacija dobivenih od raznih senzora u stvarnom vremenu. Planiranje se vrši u okviru dinamičkog prozora koji predstavlja dopuštene brzine i ubrzanja robota u određenom vremenskom intervalu čime se ostvaruju optimalno ponašanje robota prilikom gibanja uz poštivanje fizikalnih ograničenja [68]. Prilikom postavljanja upravljačkih poslužitelja potrebno je definirati sljedeće parametre:

- *controller_frequency* [Hz] – Parametar kojim se određuje frekvencija izvođenja algoritma upravljanja.
- *min_x_velocity_threshold* [m/s] – Parametar kojim se ostvaruje filtriranje poruka o potrebnoj brzini robota u smjeru x osi prije slanja upravljaču. Sve vrijednosti ispod granice određene ovim parametrom bit će zanemarene odnosno neće biti korištene u algoritmu upravljanja.
- *min_y_velocity_threshold* [m/s] – Parametar kojim se ostvaruje filtriranje poruka o potrebnoj brzini robota u smjeru y osi prije slanja upravljaču. Sve vrijednosti ispod granice određene ovim parametrom bit će zanemarene odnosno neće biti korištene u algoritmu upravljanja.
- *min_theta_velocity_threshold* [rad/s] – Parametar kojim se ostvaruje filtriranje poruka o potrebnoj brzini rotacije robota oko z osi prije slanja upravljaču. Sve vrijednosti ispod granice određene ovim parametrom bit će zanemarene odnosno neće biti korištene u algoritmu upravljanja.
- *failure_tolerance* [s] – Parametar kojim se određuje vrijeme čekanja servera upravljanja kada jedan ili više korištenih dodataka objavljuje grešku u radu prije nego što pojedini poslužitelji objave grešku u radu. Postavljanjem ovog parametra na -1 odabire se beskonačno čekanje, odnosno poslužitelj neće reagirati na pogreške u radu pojedinih dodataka. Za isključenje ovog parametra odabire se 0 dok svaka odabrana pozitivna

vrijednost određuje vrijeme čekanja poslužitelja na nepravilnosti uočene u radu pojedinih dodataka.

- *progress_checker_plugin* – Parametar kojim se odabire dodatak pomoću kojeg se prati napredak rada poslužitelja. Korištenje dodataka kojim se prati stanje poslužitelja uvelike doprinosi kvaliteti analize rada cijelog sustava te samim time uvelike olakšava uklanjanje nepravilnosti u radu pojedinih poslužitelja.
- *goal_checker_plugins* – Parametar kojim se odabire dodatak dobave informacija o uspješnosti postizanja željenog stanja zadanog od strane upravljačkog poslužitelja.
- *controller_plugins* – Parametar kojim se određuje upravljački dodatak koji se koristi u upravljačkim poslužiteljima. Detaljna listu dostupnih dodataka korištenih za potrebe ovog rada nalazi se na repozitoriju diplomskog rada [2].
- *progress_checker.plugin* – Parametar kojim se odabire programska datoteka odabranog dodatka za praćenje napretka rada poslužitelja upravljanja.
- *progess_checker.required_movement_radius* [m] – Parametar kojim se određuje minimalni iznos pomaka kojeg robot mora izvesti kako bi mogao započeti kretanje do cilja.
- *progress_checker.movement_time_allowance* [s] – Parametar kojim se određuje vremenski okvir u kojem je robotu dozvoljeno da napravi minimalni pomak, odnosno pomak opisan u prethodnom parametru.
- *general_goal_checker.stateful* – Parametar kojim se omogućava provjera tolerancija pozicije robota u X i Y osi nakon provedene rotacije do ciljne orijentacije. Ako je parametar isključen, upravljački algoritam zanemaruje promjene lokacije robota u uzduž X i Y osi koje su nastupile tijekom izvođenja rotacije.
- *general_goal_checker.plugin* – Parametar kojim se određuje dodatak koji će biti korišten tijekom provjere konačne pozicije robota.
- *general_goal_checker.xy_goal_tolerance* [m] – Parametar kojim se određuje tolerancija ciljne pozicije mobilnog robota, ujedno za X i Y os.
- *general_goal_checker.yaw_goal_tolerance* [rad] – Parametar kojim se određuje tolerancija ciljne orijentacije pozicije mobilnog robota.

- *FollowPath.plugin* – Parametar kojim se određuje dodatak za praćenje putanje. Detaljnu listu dostupnih dodataka korištenih za potrebe ovog rada moguće je pronaći u repozitoriju diplomskog rada [2].
- *FollowPath.debug_trajectory_details* – Parametar kojim se uključuje slanje poruka za uklanjanje pogrešaka (engl. *debugging messages)*
- *FollowPath.min_vel_x* [m/s] – Parametar kojim se određuje minimalna brzina robota uzduž X osi.
- *FollowPath.min_vel_y* [m/s] – Parametar kojim se određuje minimalna brzina robota uzduž Y osi.
- *FollowPath.max_vel_x* [m/s] – Parametar kojim se određuje maksimalna brzina robota uzduž x osi.
- *FollowPath.max_vel_y* [m/s] – Parametar kojim se određuje maksimalna brzina robota uzduž Y osi. Ako je ovaj parametar postavljen na vrijednost različitu od nula, robotu se omogućava holonomno ravninsko gibanje.
- *FollowPath.max_vel_theta* [rad/s] – Parametar kojim se određuje maksimalna kutna brzina robota oko Z osi.
- *FollowPath.acc_lim_x* [m/s^2] – Parametar kojim se određuje maksimalni iznos ubrzanja uzduž X osi.
- *FollowPath.acc_lim_y* [m/s^2] – Parametar kojim se određuje maksimalni iznos ubrzanja uzduž Y osi.
- *FollowPath.acc_lim_theta* [rad/s^2] – Parametar kojim se određuje maksimalni iznos kutnog ubrzanja oko Z osi.
- *FollowPath.decel_lim_x* – Parametar kojim se određuje maksimalni iznos usporavanja uzduž X osi.
- *FollowPath.decel_lim_y* – Parametar kojim se određuje maksimalni iznos usporavanja uzduž Y osi.
- *FollowPath.acc_lim_theta* – Parametar kojim se određuje maksimalni iznos kutnog usporavanja oko Z osi.
- *FollowPath.vx_samples* – Parametar kojim se određuje broj uzoraka brzine uzduž X osi.
- *FollowPath.vy_samples* – Parametar kojim se određuje broj uzoraka brzine uzduž Y osi.

- *FollowPath.vtheta_samples* – Parametar kojim se određuje broj uzoraka kutne brzine oko Z osi.
- *FollowPath.sim_time* – Parametar kojim se određuje vrijeme za koje simulacija prethodi stvarnom očitanju robotskog sustava. Parametar je koristan ukoliko se algoritam izvodi na temelju modela iz Gazebo simulacijskog okruženja, dok se isti isključuje pri korištenju stvarnog robota.
- *FollowPath.linear_granularity* [m] – Parametar kojim se određuje udaljenost ispred robota za koju se predviđa lokalno planiranje trajektorije.
- *FollowPath.angular_granularity* [rad] – Parametar kojim se određuje kut otklona robota za kojeg se predviđa lokalno planiranje trajektorije.
- *FollowPath.transform_tolerance* [s] – Parametar kojim se određuje maksimalno dopušteno vrijeme između dvije pristigle transformacije.
- *FollowPath.xy_goal_tolerance* [m] – Parametar kojim se određuje tolerancija ciljne pozicije mobilnog robota, ujedno za X i Y os.
- *FollowPath.trans_stopped_velocity* [m/s] – Parametar kojim se određuje minimalna brzina koju kontroler prosljeđuje robotu pri zaustavljanju robota kako bi se osiguralo postizanje željene pozicije. Svaka brzina ispod ovdje definirane rezultira mirovanjem robota. Podešavanje ovog parametra od velikog je značenja prilikom korištenja stvarne opreme koja zbog svojih mehaničkih karakteristika ne može ostvariti male brzine, odnosno brzine ispod ovdje definiranih.
- *FollowPath.short_circuit_trajectory_evaluation* – Parametar kojim se omogućava prestanak izračuna putanje nakon što se postigne izvedivo rješenje globalnog planera putanje. Uključenjem ove značajke može se značajno ubrzati rad algoritma lokalnog planera putanje uz posljedicu dobivanja rješenja koje nije optimalno po svim kriterijima postavljenih u ciljnu funkciju planera putanje.
- *FollowPath.stateful* – Parametar kojim se omogućava provjera tolerancija globalne pozicije robota u X i Y osi nakon provedene lokalne navigacije. Ukoliko je parametar isključen, upravljački algoritam zanemaruje promjene lokacije robota u uzduž X i Y osi te kuta oko Z osi koje su nastupile tijekom izvođenja lokalne navigacije.
- *FollowPath.critics* – Parametar kojim se određuje lista dodataka koji će biti korišteni tijekom lokalnog planera putanje, poput algoritama rotacije do ciljne pozicije, algoritme

određivanja ponašanja pri pojavi oscilacija, poravnavanja na ciljnoj poziciji te ostalih navedenih dodataka navedenih u [68].

- *FollowPath.BaseObstacle.scale* – Parametar kojim se određuje utjecaj kritičkog poslužitelja određivanja putanja dobivenih od lokalnog planera koje su podobne za kretanje robota uzimajući u obzir otisak robota.
- *FollowPath.PathAlign.scale* – Parametar kojim se određuju utjecaj kritičkog servera određivanja putanje dobivene od lokalnog planera zaduženog za poravnavanje trenutne točke robota i izračunate trajektorije.
- *FollowPath.PathAlign.forward_point_distance* – Parametar kojim se određuje broj točaka ispred robota koji je potrebno uzimati u obzir prilikom lokalnog planiranja putanje robota.
- *FollowPath.PathDist.scale* – Parametar kojim se određuje utjecaj udaljenosti dvaju točaka pri izračunu putanje mobilnog robota.
- *FollowPath.GoalDist.scale* – Parametar kojim se određuje utjecaj udaljenosti trenutne i konačne točke pri izračunu putanje mobilnog robota.
- *FollowPath.RotateToGoal.scale* – Parametar kojim se određuje utjecaj trenutne orijentacije robota i orijentacije koju je potrebno ostvariti kako bi se robot poravnao sa zadanom ciljnom točkom.
- *FollowPath.RotateToGoal.slowing_factor* – Parametar kojim se određuje usporavanje robota prilikom rotacije kako bi se osiguralo glatko zaustavljanje na željenoj orijentaciji robota bez prebačaja iste.
- *FollowPath.RotateToGoal.lookahead_time* – Parametar kojim se određuje period unaprijednog računanja kolizija.

Detaljnu listu parametara upravljačkih poslužitelja moguće je pronaći u datotekama repozitorija nav2 paketa [64][73] te u dokumentaciji nav2 paketa [68], dok su vrijednosti podešenih parametara priložene na repozitoriju diplomskog rada [2].

6.2.2.2. Poslužitelji planiranja putanje pri globalnoj navigaciji

Poslužitelji planiranja putanje, odnosno globalni planeri u svom radu koriste algoritme planiranja ruta kako bi pronašli put koji minimizira troškove uzrokovane upravljanjem aktuatora ili pak vremena putovanja, uzimajući u obzir prepreke i ograničenja okoline. U svojoj

arhitekturi, poslužitelji za planiranje implementiraju sučelja stabla ponašanja zadužena za izračun putanje do željene pozicije (engl. *behavior tree compute path to pose interface*). Sučelje poslužitelja planiranja zaduženo je za određivanje svih mogućih podobnih putanja robota koji zadovoljavaju postavljenim uvjetima.

U ovom radu je kao globalni planer putanje korišten dodatak upravljanja temeljen na rešetkama (engl. *grid based*). Kao što mu samo ime nalaže, planer putanje koristi reprezentaciju radnog prostora u obliku rešetke s pomoću koje se dobiva podjela radnog prostora na diskretne ćelije. Svaka ćelija može biti označena kao slobodna, zauzeta ili neka druga kategorija koja odražava specifično stanje okoline. Primjenjujući algoritme pretraživanja podobnih područja poput A* [74] ili Dijkstrinog algoritma [75], ovaj pristup omogućuje efikasno generiranje optimalnih ili suboptimalnih putanja kroz diskretiziranu reprezentaciju okoline. A* algoritam, primjerice, koristi heuristiku za pronalaženje optimalnih putanja, uzimajući u obzir trošak prijelaza između susjednih ćelija gdje u funkciju cilja ulazi jedino ukupna udaljenost susjednih ćelija od početne do krajnje točke. S druge pak strane, Dijkstrin algoritam se koristi za pronalaženje najkraćih putanja u grafu korištenjem težina pojedinih bridova. Ovaj algoritam počinje od trenutne pozicije robota i postepeno širi pretraživanje, dodjeljujući trenutne udaljenosti do svakog novog čvora u grafu. Na početku, sve udaljenosti postavljaju se na beskonačno, osim početnog čvora koji ima udaljenost 0. Nadalje, algoritam bira susjedne čvorove s najmanjim trenutnim udaljenostima uz vršenje provjera je li moguće smanjiti udaljenost do ciljne točke putem tog čvora. Ako je moguće smanjiti ukupnu udaljenost korištenjem određenog čvora, taj čvor se dodaje na putanju do cilja. Iterativni postupak provjere najkraće putanje završava kada se provjere svi čvorovi ili dok se ne postigne postavljeni cilj. Korištenje dodataka planera putanji temeljenih na rešetci uvelike olakšava planiranje putanja u situacijama gdje je radni prostor kompleksan. Unatoč relativnoj jednostavnosti i brzini planiranja putanja, pristup planera putanje temeljenih na rešetkama može izazvati poteškoće u situacijama s velikim radnim prostorom i složenim oblicima zbog značajnog porasta potrebne memorije kao i vremena izvođenja algoritma.

Pošto pri globalnoj navigaciji često izostaje postojanje globalne mape, korišteni su planeri putanje temeljeni na zadanim putnim točkama (engl. *waypoints*) koji su opisani u nastavku ovog odlomka. Prepreke određene lokalnim planerom preslikavaju se u globalnu mapu troškova na kojoj se nakon toga provode algoritmi globalnog planiranja putanje temeljenih na rešetkama. Putne točke koje robot mora proći kako bi došao do cilja postavlja korisnik ili neka nadređena

strategija upravljanja koja korištenjem raznih metoda analizira satelitske snimke i određuje točke kojima mobilni robot mora proći tijekom kretanja. Tako se nadilazi nepostojanje globalne mape što je česti slučaj kod globalne navigacije u nepoznatom okolišu. Pri postavljanju poslužitelja planiranja putanje potrebno je definirati sljedeće parametre:

- *Expexted_planner_frequency* [Hz] – Parametar kojim se određuje očekivana frekvencija objavljivanja planera putanje. Ukoliko je frekvencija objavljivanja planera putanje manja od postavljene, poslužitelj će javiti poruku upozorenja.
- *Planner_plugins* – Parametar kojim se određuje korišteni dodatak za planiranje putanje, u slučaju ovog rada korišten je planer temeljen na rešetkama. Listu dodataka planera putanje moguće je pronaći u dokumentaciji nav2 paketa [68].
- *GridBased.plugin* – Parametar kojim se određuje skripta u kojoj je opisan dodatak za planiranje temeljen na mreži generiranoj na temelju globalne mape troškova.
- *GridBased.use_astar* – Parametar kojim se određuje korištenje A* algoritma određivanja najkraćeg puta između dvije točke u prostoru. Ako parametar nije uključen, algoritam koristi Dijkstrin algoritam za pronalazak najkraće putanje. Za razliku od Dijkstrinog algoritma koji u svom radu ravnomjerno pretražuje sve podobne putanje, A* algoritam koristi heuristiku s pomoću koje se procjenjuje trenutna udaljenost do cilja što rezultira bržim konvergiranjem optimalnom rješenju.
- *GridBased.allow_unknown* – Parametar kojim se dopušta planiranje putanje po nepoznatom prostoru, odnosno prostoru za kojeg do trenutka planiranja ne postoji lokalna i globalna mapa troškova.

Detaljnu listu parametara poslužitelja planiranja putanje moguće je pronaći u datotekama repozitorija nav2 paketa [64] te u dokumentaciji nav2 paketa [68] dok su vrijednosti podešenih parametara priložene na repozitoriju diplomskog rada [2].

6.2.2.3. Poslužitelji lokalnih mapa troškova pri globalnoj navigaciji

Poslužitelji lokalnih mapa troškova jedna su od ključnih komponenti sustava navigacije mobilnih robota. Njihova uloga u zadacima navigacije je pružanje detaljnih informacija o okolini robota na lokalnoj razini, naglašavajući troškove povezane s preprekama. Ovi poslužitelji provode aktivno mapiranje prepreka iz okoline robota, dodjeljujući im odgovarajuće troškove kretanja. Svakom segmentu lokalne mape pridružen je trošak koji

predstavlja težinu kretanja kroz taj prostor, omogućujući serverima planiranja putanje generiranje podobnih putanja koje rezultiraju kretanje bez kolizija. Kao što im samo ime nalaže, lokalne mape troškova fokusirane su na detaljnu reprezentaciju okoline u neposrednom blizini robota, pružajući detaljne informacije o trenutačnom stanju prostora, uključujući dinamičke promjene, prepreke i druge lokalne značajke. Pri postavljanju poslužitelja lokalnih mapa potrebno je definirati sljedeće parametre:

- *update_frequency* [Hz] – Parametar kojim se postavlja stopa osvježavanja lokalne mape troškova.
- *publish_frequency* [Hz] – Parametar kojim se postavlja stopa osvježavanja teme lokalne mape troškova koju objavljuje algoritam navigacije.
- *global_frame* – Parametar kojim se postavlja referentni koordinatni sustav globalne mape troškova.
- *robot_base_frame* – Parametar kojim se određuje bazni koordinatni sustav mobilnog robota, odnosno koordinatni sustav na kojeg su vezane sve komponente mobilnog robota.
- *robot_footprint* [m] – Parametar kojim se određuju rubne točke koje opisuju otisak robota. Također, otisak robota može biti zadan putem teme s pomoću čega se omogućuje promjena otiska robota tijekom rada što pronalazi primjenu u zadacima gdje se na mobilnog robota postavlja razna dodatna oprema koja mijenja svoj položaj i oblik tijekom rada. Primjer takve dodatne opreme je robotska ruka postavljena na leđa robota.
- *robot_radius* [m] – Parametar kojim se određuje radijus robota ako koordinate otiska robota nisu definirane.
- *resolution* – Parametar kojim se definira rezolucija jednog piksela u karti troškova.
- *rolling_window* – Parametar kojim se određuje ponašanje pomičnog prozora (engl. *rolling window*) pri kretanju kroz prostor. Ako je parametar uključen, karta troškova lokalnog planera će se kretati sukladno bazi mobilnog robota te će rubovi pomičnog prozora biti uvijek jednako udaljeni od baze robota.
- *width* [m] – Parametar kojim se opisuje širina mape troškova lokalnog planera.
- *height* [m] – Parametar kojim se opisuje visina mape troškova lokalnog planera.

- *track_unknown_space* – Parametar kojim se određuje ponašanje lokalnih planera pri radu u nepoznatom prostoru. Ukoliko je parametar uključen, nepoznati prostor je u mapi lokalnih troškova prikazan kao neistraženi odnosno nepodobni prostor, dok se pri isključenom parametru nepoznati prostor smatra kao slobodni prostor.
- *plugins* – Parametar kojim se određuju dodaci korišteni pri radu s mapama troškova lokalnih planera. Listu dodataka lokalnih mapa troškova moguće je pronaći u dokumentaciji nav2 paketa [68].
- *inflation_layer.plugin* – Parametar kojim se odabire skripta kojom se opisuje dodatak inflacije. Dodatak se koristi kako bi se eksponencijalno povećale vrijednosti mape troškova pri približavanju preprekama. Na taj način ostvaruje se dodatna penalizacija kretanja u neposrednoj blizini prepreka.
- *voxel_layer.publish_voxel_map* – Parametar kojim se određuje objavljivanje mreže voksela koja može biti korisna u zadacima otklanjanja poteškoća. Vokseli su korišteni za preslikavanje trodimenzionalnog prostora u dvodimenzionalni model temeljem kojeg se potom provode algoritmi izbjegavanja prepreka. Uključivanje parametra prikazivanja mapa voksela ima značajan utjecaj na porast računalne zahtjevnosti poslužitelja lokalnih mapa troškova te izravno utječe na degradaciju performansi servisa izrade lokalnih mapa troškova.
- *voxel_layer.scan.topic*– Parametar kojim se određuje tema reprezentacije okoline koji će biti korišten prilikom izrade mape troškova reprezentirane vokselima.
- *always_send_full_costmap* – Parametar kojim se određuje način osvježavanja mape troškova lokalne mape vrijednosti. Ukoliko je parametar uključen, u svakoj novoj iteraciji server lokalnih mapa troškova objavljuje cijelu mapu, dok se pri isključenom parametru objavljuju samo promijene nastale od prošlog objavljivanja iste.

Detaljnu listu parametara poslužitelja lokalnih mapa troškova moguće je pronaći u datotekama repozitorija nav2 paketa [64] te u dokumentaciji nav2 paketa [68] dok su vrijednosti podešenih parametara priložene na repozitoriju diplomskog rada [2].

6.2.2.4. Poslužitelji globalnih mapa troškova pri globalnoj navigaciji

Kao što je spomenuto u uvodu ovog odlomka, globalne i lokalne mape troškova predstavljaju ključne komponente u sustavima navigacije mobilnih robota, dok njihova osnovna

razlika leži u razini apstrakcije i širini prostora kojeg iste predstavljaju. Za razliku od lokalnih mapa troškova koji predstavljaju relativno uzak pojas oko robota, globalne mape troškova pružaju cjelovitu reprezentaciju okoline na razini cijelog radnog prostora robota. One obuhvaćaju širu sliku terena s fokusom na različite troškove povezane s kretanjem poput udaljenosti od cilja ili statičkih prepreka. Globalne mape troškova korištene su prilikom izrade ovog rada kao osnova za algoritme planiranja globalnih ruta, omogućujući robotima generiranje optimalne putanje za postizanje globalnih ciljeva. Parametri opće inicijalizacije poslužitelja globalnih mapa troškova jednaki su kao kod poslužitelja lokalnih mapa troškova prikazanih u prethodnom odlomku, dok su specifični parametri za postavljanje poslužitelja globalnih mapa troškova prikazani u nastavku:

- *plugins* – Parametar kojim se određuju dodaci korišteni pri radu s mapama troškova globalnih planera. Listu dodataka globalnih mapa troškova moguće je pronaći u dokumentaciji nav2 paketa [68].
- *obstacle_layer.plugin* – Parametar kojim se određuje skripta u kojoj je opisan dodatak za opis prepreka. Dodatak se koristi kako bi se integrirala očitanja lidara, dubinske slike prikazane u obliku oblaka točaka ili ostalih izvora informacija kojim se opisuje okolina. Na taj se način globalna mapa troškova proširuje s informacijama o preprekama dobivenih od strane korištenih senzora.
- *obstacle_layer.observation_source* – Parametar kojim se određuje izvor podataka na koju se objavljuje reprezentacija okoline. Izvor može biti postavljen na sken lasera ili pak dubinsku sliku predstavljenu oblakom točaka.
- *obstacle_layer.scan.topic* – Parametar kojim se određuje tema na koju se objavljuje reprezentacija okoline.
- *obstacle_layer.scan.max_obstacle_height* – Parametar kojim se određuje maksimalna visina prepreka koje ulaze u globalnu mapu troškova.
- *obstacle_layer.scan.data_type* – Tip podataka kojim se predstavlja okolina. Ovisno o odabranom izvoru, parametar može biti postavljen na standardne ROS2 poruke *LaserScan* ili *PointCloud2* tipa.
- *obstacle_layer.scan.raytrace_max_range* [m] – Parametar kojim se određuje maksimalni raspon koji se uzima u obzir prilikom reprezentacije okoline.

- *obstacle_layer.scan.raytrace_min_range* [m] – Parametar kojim se određuje minimalni raspon koji se uzima u obzir prilikom reprezentaciji okoline.
- *obstacle_layer.scan.obstacle_max_range* [m] – Maksimalna udaljenost od prepreke za koju se iste uzimaju u obzir pri generiranju globalne mape troškova.
- *obstacle_layer.scan.obstacle_min_range* [m] – Minimalna udaljenost od prepreke za koju se iste uzimaju u obzir pri generiranju globalne mape troškova.
- *obstacle_layer.scan.inf_is_valid* [m] – Parametar kojim se određuje objavljivanje beskonačne udaljenosti ukoliko se na temi reprezentacije okoline u zadanom rasponu ne nalazi nikakva prepreka.

Detaljnu listu parametara poslužitelja globalnih mapa troškova moguće je pronaći u datotekama repozitorija nav2 paketa [64] te u dokumentaciji nav2 paketa [68], dok su vrijednosti podešenih parametara dane na repozitoriju diplomskog rada [2]. Kombinacija globalnih i lokalnih mapa troškova često se koristi kako bi se postigla potpuna slika okoline, gdje globalne mape pružaju strategiju i smjer kretanja, dok lokalne mape omogućuju precizno izvođenje planirane putanje uzimajući u obzir trenutačne uvjete u neposrednoj blizini robota. Ova integracija mobilnim robotima omogućuje autonomnu i navigaciju kroz razna okruženja uz prisutnost kako statičkih tako i dinamičkih prepreka.

6.2.2.5. Poslužitelji zaglađivanja putanje pri globalnoj navigaciji

Poslužitelji za zaglađivanje implementiraju sučelja zadužena za uglađivanje putanje. Poslužitelji planiranja putanje kao ulazni podatak primaju putanju određenu poslužiteljima planera putanje koja u većini slučajeva ne može izravno biti primijenjena na mobilnom robotu. Zbog toga su u zadacima navigacije često korišteni poslužitelji zaglađivanja putanje kako bi se u velikoj mjeri smanjile oscilacije kao i nagle promjene smjerova u putanji robota čime se značajno povećava stabilnost kao i uglađenost kretanja robota. Nadalje, korištenjem poslužitelja za zaglađivanje putanje anulira se pojava oštrih kutova čime se uvelike olakšava izvođenje kretanja, posebice za robote s ograničenom okretnošću. Također, primjena poslužitelja za zaglađivanje doprinosi smanjenju vibracija i oscilacija tijekom kretanja robota što može uvelike doprinijeti kvaliteti preciznog pozicioniranja robota. Pri postavljanju poslužitelja zaglađivanja putanje potrebno je definirati sljedeće parametre:

- *smoother_plugins* – Parametar kojim se odabire dodatak za zaglađivanje putanje. U ovom radu korišten je jednostavni algoritam zaglađivanja (engl. *simple smoother*) kojim

se efikasno i računalno jeftino uklanjaju oscilacije i nazubljenja nastala tijekom planiranja putanje.

- *simple_smoother.tolerance* – Parametar kojim se određuje dopuštena promjena putanje. Ukoliko se tijekom zaglađivanja vrijednost promijeni više od vrijednosti propisane ovim parametrom, proces zaglađivanja se prekida.
- *simple_smoother.max_its* – Maksimalni broj iteracija rekurzivnih pokušaja zaglađivanja putanje.
- *simple_smoother.do_refinement* – Parametar kojim se određuje rekurzivno zaglađivanje otprije zaglađene putanje koje se često koristi radi daljnjeg povećanja kvalitete putanje.
- *simple_smoother.refinement_num* – Parametar kojim se određuje broj rekurzivnih zaglađivanja putanji.

Detaljnu listu parametara poslužitelja zaglađivanja putanje moguće je pronaći u datotekama repozitorija nav2 paketa [64] te u dokumentaciji nav2 paketa [68], dok su vrijednosti podešenih parametara dane na repozitoriju diplomskog rada [2].

6.2.2.6. Poslužitelji zaglađivanja brzine pri globalnoj navigaciji

Poslužitelji zaglađivanja brzine u kontekstu globalne navigacije mobilnih robota, uz poslužitelje zaglađivanja putanje igraju ključnu ulogu u poboljšanju kvalitete i izvedbe planirane putanje. Implementacijom ovih poslužitelja izravno se utječe na zaglađivanje brzina, ubrzanja te mrtvog pojasa istih. Time se u velikoj mjeri smanjuju ili čak anuliraju trzaji koji uzrokuju brže trošenje aktuatora robota te samim time doprinose povećanju stabilnosti sustava. Također, poslužitelji zaglađivanja brzina mogu ujedno biti korišteni za interpolaciju referenci brzine većom frekvencijom od frekvencije kojom objavljuje upravljački poslužitelji. Pri postavljanju poslužitelja zaglađivanja brzine potrebno je definirati sljedeće parametre:

- *smoothing_frequency* [Hz] – Parametar kojim se određuje frekvencija izvršavanja dodataka za zaglađivanje brzine kretanja mobilnog robota. Ukoliko je parametar podešen na frekvenciju kojom radi lokalni planer putanje, isti naredbama referenci brzina zaglađuje ubrzanja robotskog sustava. Ukoliko je pak parametar postavljen na vrijednost značajnije veću od vrijednosti lokalnog planera, servis zaglađivanja brzine će provoditi interpolaciju kojom će se osigurati glatki skup naredbi algoritmu zaduženom za kretanje robota.

- *scale_velocities* – Parametar kojim se određuje postojanje potrebe za prilagodbom komponenata referenci brzina kako bi se zadovoljila ograničenja ubrzanja. Ukoliko je parametar uključen, upravljač će provoditi prilagodbu brzina robota kako bi se zadovoljila ograničenja ubrzanja čak i kad to podrazumijeva mala odstupanja od zadane putanje.
- *feedback* – Parametar kojim se određuje vrsta povratne veze koja je korištena za određivanje trenutnog stanja brzine robota. Ako je parametar postavljen na vrijednost povratne veze koja označava otvorenu petlju (engl. *open loop*), poslužitelj zaglađivanja brzine u svakom novom koraku uzima posljednju referentnu brzinu. Otvorena petlja može biti uspješno implementirana jedino u slučaju dobrog postavljanja limita ubrzanja. U slučaju zatvorene petlje (engl. *closed loop*), server zaglađivanja brzine koristi odometriju kako bi estimirao trenutnu brzinu robota. Pri korištenju zatvorene petlje važno je osigurati brzi odziv teme odometrije kako bi server zaglađivanja brzine u svakom trenutku imao stvarni uvid u trenutno ponašanje robota.
- *max_velocity* [m/s, m/s, rad/s] – Parametar kojim se određuje maksimalna brzina kretanja robota. Parametar se zadaje u obliku vektora s tri člana, gdje prva dva člana predstavljaju maksimalne brzine uzduž X i Y osi, dok treći parametar predstavlja maksimalnu kutnu brzinu oko Z osi.
- *min_velocity* [m/s, m/s, rad/s] – Parametar kojim se određuje minimalna brzina kretanja robota. Parametar se zadaje u obliku vektora s tri člana gdje prva dva člana predstavljaju minimalne brzine uzduž X i Y osi dok treći parametar predstavlja minimalnu kutnu brzinu oko Z osi.
- *max_accel* [m/s^2, m/s^2, rad/s^2] – Parametar kojim se određuje maksimalno ubrzanje robota. Parametar se zadaje u obliku vektora s tri člana, gdje prva dva člana predstavljaju maksimalno ubrzanje uzduž X i Y osi, dok treći parametar predstavlja maksimalno kutno ubrzanje oko Z osi.
- *max_decel* [m/s^2, m/s^2, rad/s^2] – Parametar kojim se određuje maksimalno usporavanje robota. Parametar se zadaje u obliku vektora s tri člana, gdje prva dva člana predstavljaju maksimalno usporavanje uzduž X i Y osi, dok treći parametar predstavlja maksimalno kutno usporavanje oko Z osi.
- *odom_topic* – Parametar kojim se određuje tema odometrije u slučaju zatvorene petlje.

- *odom_duration* [s] – Parametar kojim se određuje vrijeme trajanja dobave nove poruke na temu odometrije. Parametar trajanja odometrije se potom koristi pri estimaciji brzine mobilnog robota u slučaju zatvorene petlje.
- *deadband_velocity* [m/s^2, m/s^2, rad/s^2] – Parametar kojim se određuje minimalna brzina koja će biti slana upravljačkom sklopu robota kako bi se spriječilo oštećenje opreme ukoliko određena brzina ne može biti postignuta zbog momenta zastoja (engl. *stall torque*).
- *velocity_timeout* [s] – Parametar kojim se određuje vremenski okvir nakon kojega upravljač šalje nultu referencu kao izlaz pri prestanku normalnog rada servisa koji može biti uzrokovan prestankom objavljivanja neke od korištenih tema ili sličnim događajem.

Detaljnu listu parametara poslužitelja zaglađivanja brzine moguće je pronaći u datotekama repozitorija nav2 paketa [64] te u dokumentaciji nav2 paketa [68], dok su vrijednosti podešenih parametara dane na repozitoriju diplomskog rada [2].

6.2.3. Postavljanje karte troškova lokalnog planera putanje

Postavljanje karte troškova lokalnog planera putanje predstavlja ključnu fazu u osiguravanju sigurnog i učinkovitog kretanja robota kroz dinamično okruženje. Ovaj proces uključuje odabir odgovarajuće reprezentacije okoline, čime se omogućuje pravovremena identifikacija potencijalnih opasnosti radi izbjegavanja kolizija. Jedan od često korištenih načina reprezentacije okoline je korištenje mape troškova, gdje se svakoj lokaciji dodjeljuje specifičan trošak ovisno o karakteristikama tog područja. Slobodan prostor pri takvoj reprezentaciji poprima nizak, dok će prepreke i teško prohodna područja poprimiti visok trošak. Ovakva reprezentacija troškova omogućuje planeru putanje odabir optimalne staze koje minimiziraju ukupni trošak kretanja kroz neki prostor.

Reprezentacija okoline putem karte troškova ključna je za prevenciju kolizija i osiguravanje sigurnog prolaska kroz prostor. Korištenje ovakvog pristupa omogućuje lokalnom planeru putanje donošenje brzih i prilagodljivih odluka, čime se optimizira kretanje robota u realnom vremenu. U nastavku su prikazane neke od metoda reprezentacija okoline pomoću kojih se gradi lokalna mapa troškova korištena pri planiranju putanje mobilnog robota.

6.2.3.1. Spot algoritam za izbjegavanje prepreka

U algoritmima izbjegavanja prepreka dobiva se slika o okolini koristeći informacije dobivene sa stereo kamera. Kao što je opisano u poglavlju *4.3. Senzorika integrirana na mobilnom robotu*, Spot dolazi s pet integriranih RealSense D435 kamera čiji međusobni prostorni položaj omogućava potpuni pregled okoline robota. Korištenjem dostupnih paketa drivera Spot robota [43], izostavljena je dodatna mogućnost konfiguracije algoritma izbjegavanja prepreka. Zbog toga je prilikom dodatne konfiguracije lokalnih planera putanje potrebno algoritmima navigacije vršiti dobavu podataka o okolini robota. Time se omogućava podešavanje parametara kao što je maksimalna udaljenost na koju se robot smije približiti prepreci ili pak maksimalna visina prepreke preko koje robot može prijeći bez pojave kolizija. U nastavku je naveden paket s pomoću kojeg je moguće dodatno konfigurirati reprezentaciju okoline mobilnog robota.

6.2.3.2. Izbjegavanje prepreka temeljeno na reprezentaciji okoline korištenjem oblaka točaka

Algoritmi navigacije predstavljeni nav2 paketom imaju mogućnost izravnog korištenja dubinskih informacija u obliku oblaka točaka iz koje se potom generira lokalna mapa troškova putem koje se provodi izbjegavanje prepreka. Zbog korištenja oblaka točaka dobivenog od pet nezavisnih dubinskih kamera, takav pristup značajno bi opteretio poslužitelje lokalnih planera putanje pred koje bi bila stavljena značajna količina podataka koju je potrebno obraditi u stvarnom vremenu. Samim time, uz izravno korištenje oblaka točaka postoji velika vjerojatnost značajnije degradacije performansi izvršavanja navigacijskih servisa. Jedno od mogućih rješenja je izbjegavanje izravnog korištenja izvora koji donose veliku količinu informacija, u ovom slučaju oblaka točaka. Kao alternativno rješenje ovdje spomenutom problemu moguće je provođenje pretvorbe tako zahtjevnog izvora informacija u minimalni zapis okoline robota, odnosno zapis sukladan skenu lasera.

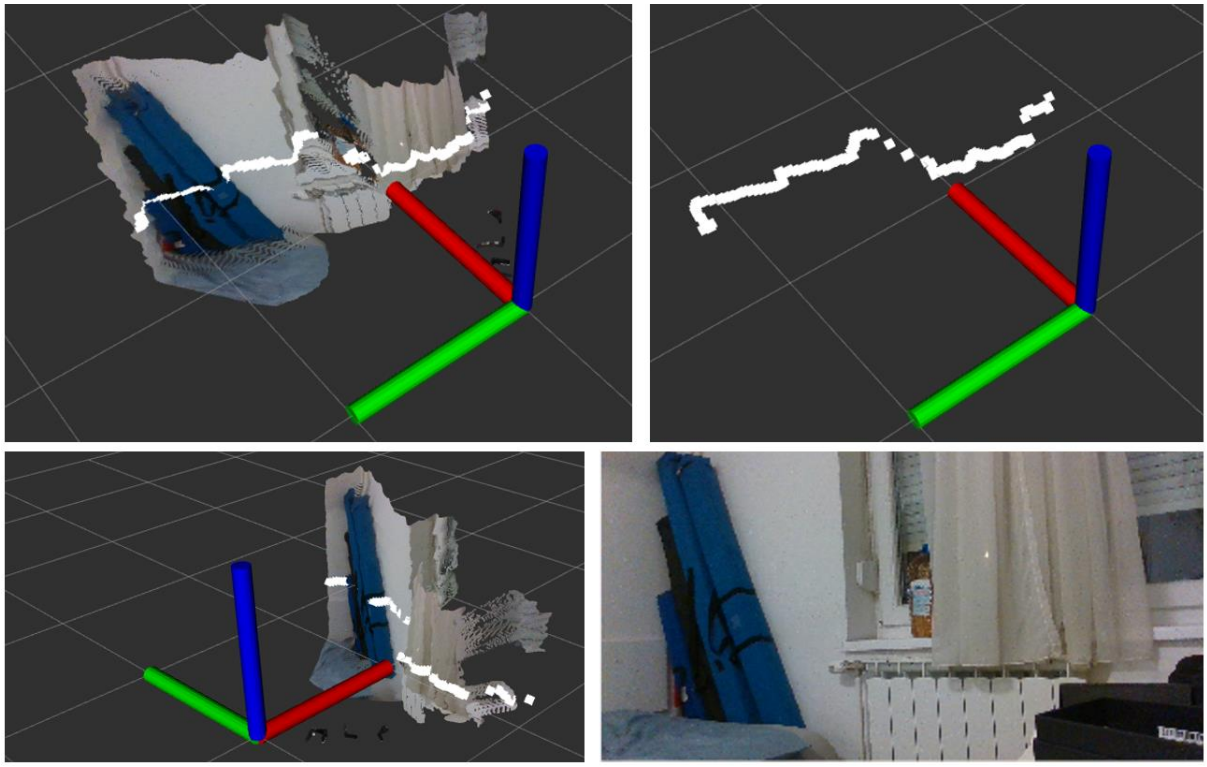
Paket pointcloud_to_laserscan [33] donosi mogućnost pretvorbe dubinskih podataka zapisanih u formi oblaku točaka u očitanje lasera te obratno. U svom radu, paket uzima opis okoline prikazan oblakom točaka kojeg obrađuje te na temelju kojega se generira 2D reprezentacija okoline kojim se detektira potencijalno podobno područje za kretanje robota bez

kolizija. Pri postavljanju parametara čvora pretvorbe dubinskih poruka u zadacima pretvorbe oblaka točaka u očitanje lasera potrebno je definirati sljedeće parametre:

- *scan* – Parametar kojim se određuje tema na koju se objavljuje očitanje lasera.
- *cloud_in* – Parametar kojim se određuje tema oblaka točaka iz kojeg je potrebno generirati poruku u obliku očitanja lasera.
- *min_height* [m] – Parametar kojim se određuje minimalna visina iz oblaka točaka koja se uzima u obzir prilikom generiranja očitanja lasera. Najčešće se postavlja na visinu koju robot može sigurno prekoračiti u svom radu.
- *max_height* [m] – Parametar kojim se određuje maksimalna visina iz oblaka točaka koja se uzima u obzir prilikom očitanja lasera. Najčešće se podešava na visinu robota uvećanu za sigurnosnu visinu koja osigurava da robot sigurno prođe ispod očitane prepreke.
- *angle_min* [rad] – Parametar kojim se određuje minimalni kut iz oblaka točaka koji se uzima pri pretvorbi u sken lasera.
- *angle_max* [rad] – Parametar kojim se određuje maksimalni kut iz oblaka točaka koji se uzima pri pretvorbi u sken lasera.
- *angle_increment* [rad] – Parametar kojim se određuje razlika između susjednih očitanja. Time se oponaša poruka dobivena od pravih LIDAR senzora koji svoju poruku generiraju prilikom rotiranja senzora.
- *queue_size* – Parametar kojim se definira sjedinjenje većeg broja poruka oblaka točaka, odnosno podešava se nisko propusni filtar kojim se anulira utjecaj visokofrekventnih šumova senzora.
- *scan_time* [s] – Parametar kojim se određuje brzina objavljivanja poruke na temu laserskog skena, ekvivalentno trajanju jedne potpune rotacije senzora lasera.
- *range_min* [m] – Parametar kojim se određuje minimalna udaljenost za koju se generira očitanje lasera. Svaka vrijednost ispod ovdje propisane bit će zanemarena.
- *range_max* [m] – Parametar kojim se određuje maksimalna udaljenost za koju se generira očitanje lasera. Svaka vrijednost iznad ovdje propisane bit će zanemarena.

Detaljnu listu parametara čvora pretvorbe oblaka točaka u očitanje lasera moguće je pronaći u datotekama repozitorija pointcloud_to_laserscan paketa [33], dok su vrijednosti podešenih

parametara dane na repozitoriju diplomskog rada [2]. Slika 37. prikazuje pretvorbu dubinske slike iz donjeg desnog kuta u očitanje lasera korištenjem pointcloud_to_laserscan čvora:



Slika 37. Prikaz pretvorbe oblaka točaka u očitanja lasera

Parametri potrebni za ispravan rad pointcloud_to_laserscan čvora podešeni su prema karakteristikama korištenih RealSense D435i kamera [28]. Paketi pretvorbe podataka, uz potrebne modifikacije priloženi su na repozitoriju diplomskog rada [2].

6.2.4. Postavljanje karte troškova globalnog planera putanje

Kartu troškova globalne putanje moguće je postaviti putem tri najčešće korištene metode. Prva metoda zasniva se na karti troškova s kontinuiranim ažuriranjem. Vanjski okoliš je u pravilu vrlo velik što uvelike otežava prikaz koristeći samo jednu kartu troškova. Iz tog razloga se u zadacima navigacije često koristi metoda karta troškova s kontinuiranim osvježavanjem kojoj je osiguran horizont za smještanje nekoliko putnih točaka. Smještanjem putnih točaka u bližoj okolini robota osigurava se uspješno izvođenje zadataka vanjske navigacije uz zaobilaženje prepreka. Druga metoda se odnosi na umetanje globalne karte troškova u obliku statičkih slojeva u mapu okoliša. Kretanjem robota kroz prostor sustav provodi osvježavanje statičkih slojeva prema trenutnoj lokaciji robota kako bi istom u svakom trenutku bio dostupan

pregled okoliša. Treća metoda podrazumijeva poznavanje fiksne mape troškova u cijelom okolišu u kojem se robot može kretati tijekom izvršavanja svojih operacija. Poznavanje fiksne mape moguće je provesti u situacijama kada se kretanje robota odvija u poznatom prostoru te ako zadatak navigacije ne zahtijeva izlazak iz tog prostora. U ovom radu je, zbog spomenutih prednosti korištenja u nepoznatom okolišu, korištena metoda karata troškova s kontinuiranim osvježavanjem.

6.2.4.1. Planeri putanje temeljeni na zadanim putnim točkama

Praćenje putnih točaka (engl. *waypoint following*) predstavlja glavnu značajku sustava namijenjenih navigaciji. Tom značajkom određuje se metoda korištenja sustava navigacije s ciljem postizanja željene pozicije i orijentacije u prostoru. Čvor nav2_waypoint_folower sadržava servise za praćenje putnih točaka uz mogućnost implementacije dodataka za izvršavanje zadataka prilikom dolaska u određenu putnu točku. Uz to, čvor pruža mogućnost rada s flotama robota, pružajući mogućnost kontrole robota visokog stupnja autonomnosti s jednostavnom upravljačkom strategijom te obrnuto, odnosno upravljanje robota bez sposobnosti autonomije s naprednom upravljačkom strategijom. Više o radu s flotama robota opisano je u dokumentaciji nav2 paketa [68]. Nadalje, čvor nav2_waypoint_folower također podržava rad s GNSS putnim točkama ako je globalna lokalizacija robota provedena korištenjem navsat_transform čvora unutar robot_localization paketa. Paket sadržava follow_gps_wapoints poslužitelja koji omogućava izravno korištenje koordinata dobivenih od strane GNSS prijamnika. Te koordinate čvor potom pretvara u kartezijev koordinatni sustav te ih izvršava kao takve [68]. Pri postavljanju planera putanje temeljenog na putnim točkama potrebno je definirati sljedeće parametre:

- *loop_rate* [Hz] – Parametar kojim se određuje stopa provjere stanja trenutnog zadatka navigacije, u ovom slučaju zadatka praćenja putnih točaka.
- *stop_on_failure* – Parametar kojim se određuje potrebna radnja prilikom nemogućnosti postizanja pojedine putne točke. Ako je parametar uključen, zadatak navigacije pri nemogućnosti dostizanja pojedine točke javlja grešku koja rezultira zaustavljanjem zadatka navigacije.
- *action_server_result_timeout* [s] – Parametar kojim se određuje vremenski okvir trajanja pojedine radnje navigacije prije odbacivanja zadatka od strane poslužitelja.

Time se omogućava nastavak provođenja ostalih zadataka navigacije ukoliko pojedinu radnju nije moguće izvršiti u konačnom vremenu.

- *waypoint_task_executor_plugin* – Parametar kojim se određuje dodatak koji definira ponašanje robota pri postizanju određene putne točke. U slučaju dodatka slijeđenjem putnih točaka, robot može čekati određeni period prije nego što nastavi kretanje dalje. Također, okidač za nastavak gibanja može biti neki vanjski signal odnosno izlaz iz nekog drugog algoritma.
- *wait_at_waypoint.plugin* – Parametar kojim se određuje dodatak koji će biti korišten prilikom postizanja zadane putne točke. U ovom diplomskom radu korišten je dodatak čekanja na svakoj putnoj točki jer u ovom stadiju rada nema nadređenog algoritma koji bi robotu mogao narediti nastavak gibanja.
- *wait_at_waypoint.waypoint_pause_duration* [ms] – Parametar kojim se određuje vremenski okvir čekanja robota od trenutka dolaska u putnu točku do nastavka kretanja.

Detaljnu listu parametara planera putanje temeljenih na putnim točkama moguće je pronaći u datotekama repozitorija nav2 paketa [64] te u dokumentaciji nav2 paketa [68], dok su vrijednosti podešenih parametara dane na repozitoriju diplomskog rada [2].

6.2.4.2. Poslužitelji za upravljanje ponašanjima u zadacima globalne navigacije

Poslužitelji upravljanja raznim ponašanjima često su korišteni u zadacima navigacije u zadacima kao što su zadaci oporavka ili pak spajanja na razne dodatke kao što su punjači ili ostala dodatna oprema. Također, korištenjem jedinstvenog poslužitelja za upravljanje pojedinačnim ponašanjima omogućeno je dijeljenje sredstva kao što su mape troškova ili pak međuspremnika transformacija koordinatnih sustava. Takav pristup uzrokuje sjedinjenje dodataka koji koriste približno jednake podatke što rezultira nižim utroškom računalnih resursa prilikom dodavanja novih ponašanja te stabilnijim ponašanjem rada cijelog sustava kao cjeline. Pri postavljanju poslužitelja za upravljanje ponašanjima potrebno je definirati sljedeće parametre:

- *local_costmap_topic* – Parametar kojim se određuje tema na koju se objavljuju sirovi podaci lokalne mape troškova.
- *global_costmap_topic* – Parametar kojim se određuje tema na koju se objavljuju sirovi podaci globalne mape troškova.

- *local_footprint_topic* – Parametar kojim se određuje tema na koju se objavljuju podaci o otisku robota u lokalnoj mapi troškova.
- *global_footprint_topic* – Parametar kojim se određuje tema na koju se objavljuju podaci o otisku robota u globalnoj mapi troškova.
- *cycle_frequency* [Hz] – Parametar kojim se određuje frekvencija izvršavanja dodataka za praćenje ponašanja algoritama navigacije.
- *behaviour_plugins* – Parametar kojim se određuju dodaci koji su korišteni u zadacima praćenja ponašanja algoritama globalne navigacije. Kao dodaci ponašanja najčešće su korišteni dodaci za rotaciju prilikom nemogućnosti pronalaska podobne trajektorije, dodaci za daljinski potpomognuto teledirigiranje, dodaci za čekanja temeljena na raznim uvjetima i ostali. Cijelu listu dostupnih dodataka namijenjenih praćenju ponašanja algoritama globalne navigacije moguće je pronaći u [68].
- *local_frame* – Parametar kojim se određuje fiksni lokalni koordinatni sustav korištenog mobilnog robota. Najčešće se postavlja na vrijednost koordinatnog sustava odometrije.
- *global_frame* – Parametar kojim se određuje fiksni globalni koordinatni sustav korištenog mobilnog robota. Najčešće se postavlja na vrijednost koordinatnog sustava mape dobivene od lokalizacijskog sustava, u ovom slučaju sustava globalne navigacije putem GNSS podataka.
- *robot_base_frame* – Parametar kojim se određuje pomični koordinatni sustav s pomoću kojeg se definiraju pomaci mobilnog robota u odnosu na fiksni lokalni koordinatni sustav.
- *transform_tolerance* [m] – Parametar kojim se određuje tolerancija transformacija koordinatnih sustava korištenih u zadacima navigacije.
- *simulate_ahead_time* [s] – Parametar kojim se određuje vremenski okvir za koji se provjeravaju kolizije. Povećanjem ovog parametra povećava se horizont kontrolera što dodatno opterećuje simuliranje samog sustava.

Detaljnu listu parametara poslužitelja upravljanja raznim ponašanjima algoritama navigacije moguće je pronaći u datotekama repozitorija nav2 paketa [64] te u dokumentaciji nav2 paketa [68], dok su vrijednosti podešenih parametara dane na repozitoriju diplomskog rada [2].

6.3. Zadavanje referentne trajektorije

Pokretanjem navigacijskih poslužitelja opisanih gore pokreću se sve komponente zadužene za stvaranje sučelja korištenog u zadacima navigacije. Nadalje, potrebno je implementirati sučelje s pomoću kojega se ostvaruje kretanje robota na temelju eksplicitnog određivanja pozicije zadavanjem GNSS koordinata korištenjem Mapviz *pose* dodatka [76]. Za potrebe ovog rada korišteno je sučelje interaktivnog praćenja putnih točaka predstavljeno u [77] koje je dorađeno prema potrebama ovog zadatka i priloženo u repozitorij diplomskog rada [2]. Sučelje omogućava odabir ciljne točke pritiskom na mapu u Mapviz sučelju. Sučelje potom koristi dodatak koji objavljuje koordinate odabrane točke u WGS84 (engl. *World Geodetic System 1984, hrv. svjetski geodetski sustav 1984*) koordinatnom sustavu s pomoću kojega je moguće očitavati pozicije na satelitskoj snimci [78]. Nakon određivanja pozicije u WGS84 koordinatnom sustavu, korišteni algoritam sučelja praćenja putnih točaka objavljuje poruku s zapisom pozicije označene točke na mapi. Tako objavljeni zapisi pozicije se potom koriste kao ulazni parametri u poslužiteljima navigacije s pomoću kojih se provodi praćenje putnih točaka.

Također, paket prikazan u [77] donosi dva čvora koji omogućavaju manipulaciju s podacima o lokaciji očitanih putem GNSS prijamnika, odnosno čvor zapisa putnih točaka (engl. *waypoint logging*) kao i čvor praćenja GNSS putnih točaka. Čvor namijenjen zapisu putnih točaka pretplaćuje se na teme koje objavljuju podatke o globalnoj poziciji i orijentaciji robota. Nadalje, čvor zapisa putnih točaka donosi grafičko sučelje na kojem se nalazi gumb za spremanje trenutnih koordinata robota u yaml (engl. *yet another markup language*, hrv. *još jedan označni jezik*) datoteku. Sljedeći čvor, odnosno čvor praćenja GNSS putnih točaka vrši dobavu putnih točaka zapisanih u yaml datoteci poslužiteljima navigacije. Time se putne točke redom šalju akcijskim poslužiteljima koji potom započinju proces navigacije što naposljetku rezultira postizanjem putnih točaka zapisanih u danoj yaml datoteci.

7. NAVIGACIJA MOBILNOG ROBOTA

U ovom odlomku prikazana je validacija algoritama lokalizacije i navigacije temeljene na GNSS koordinatama. Prije nego što se algoritmi lokalizacije i globalne navigacije primijene u stvarnom okruženju, korisno je provesti podešavanje parametara čvorova i poslužitelja u simulacijskom okruženju. Simulacije pružaju kontrolirano i sigurno okruženje namijenjeno testiranju različitih aspekata ponašanja robota bez potrebe za fizičkim prisustvom na terenu. Također, simulacijskim okruženjem omogućeno je ugrubo podešavanje dodataka i parametara korištenih pri lokalizaciji i navigaciji, prateći osnovne principe ponašanja robota. Istodobno, proces podešavanja parametara se daleko brže provodi u simulacijskom okruženju jer nema ograničenja stvarnog vremena, što omogućuje ubrzanje rane faze razvoja sustava. Nadalje, simulacijskim okruženjem olakšano je ispitivanje različitih scenarija i situacija koje se mogu pojaviti prilikom kretanja robota u stvarnom okruženju čime se ostvaruje dublje razumijevanje ponašanja robota u različitim scenarijima. Nakon podešavanja provedenih u simulacijskom okruženju, sustav se postavlja u vanjski prostor gdje se testiranja provode na stvarnoj opremi. Tijekom terenskih ispitivanja na stvarnom robotu vrše se fina podešavanja, uzimajući u obzir stvarne uvjete, senzore kao i ostale specifičnosti okoliša. Ovakav pristup sjedinjuje prednosti korištenja simulacije kao i testiranja na terenu, čime se olakšava brže postavljanje sustava lokalizacije i navigacije mobilnog robota.

7.1. Priprema algoritama navigacije na simulacijskom modelu mobilnog robota

Kao prvi korak validacije algoritama navigacije i lokalizacije navodi se podešavanje poslužitelja, parametara i čvorova modela mobilnog robota u Gazebo simulacijskom okruženju. Radi ekspeditivnosti kao i minornih razlika u ponašanju sustava, kao simulacijski model mobilnog robota korišten je mobilni robot s diferencijalnim pogonom. Time se uvelike olakšao proces okvirnog postavljanja sustava lokalizacije i navigacije dok su svi parametri dodatno proanalizirani i podešeni prilikom terenskog testiranja na Spotu. Osnova programskog koda korištenog u ovom poglavlju preuzeta je iz [68], dok su sve modifikacije paketa priložene na repozitoriju diplomskog rada [2]. Također, dijagrami referenci brzina u odnosu na vrijeme dani su na repozitoriju diplomskog rada.

Kao simulacijski model sustava korišten je TurtleBot3 *Waffle* robot [79] s diferencijalnim pogonom. Kako se diferencijalni robot ne može kretati holonomno, prilikom testiranja sustava

u simulacijskom okruženju poslužiteljem regulatora brzine onemogućeno je kretanje uzduž Y osi, dok se pri upotrebi na Spot robotu omogućava holonomno kretanje, odnosno kretanje uzduž X i Y osi. Time se omogućava puni potencijal kretanja mobilnog robota Spota, odnosno sposobnost kretanja u svakom smjeru na ravnini bez obzira na orijentaciju robota. U nastavku je prikazano postavljanje simulacijskog okruženja kao i validacija algoritama navigacije u Gazebo simulacijskom okruženju temeljenih na zadanim putnim točkama. Zatim je prikazana validacija algoritama navigacije na Spot mobilnom robotu s nadodanom opremom opisanom u ovom radu.

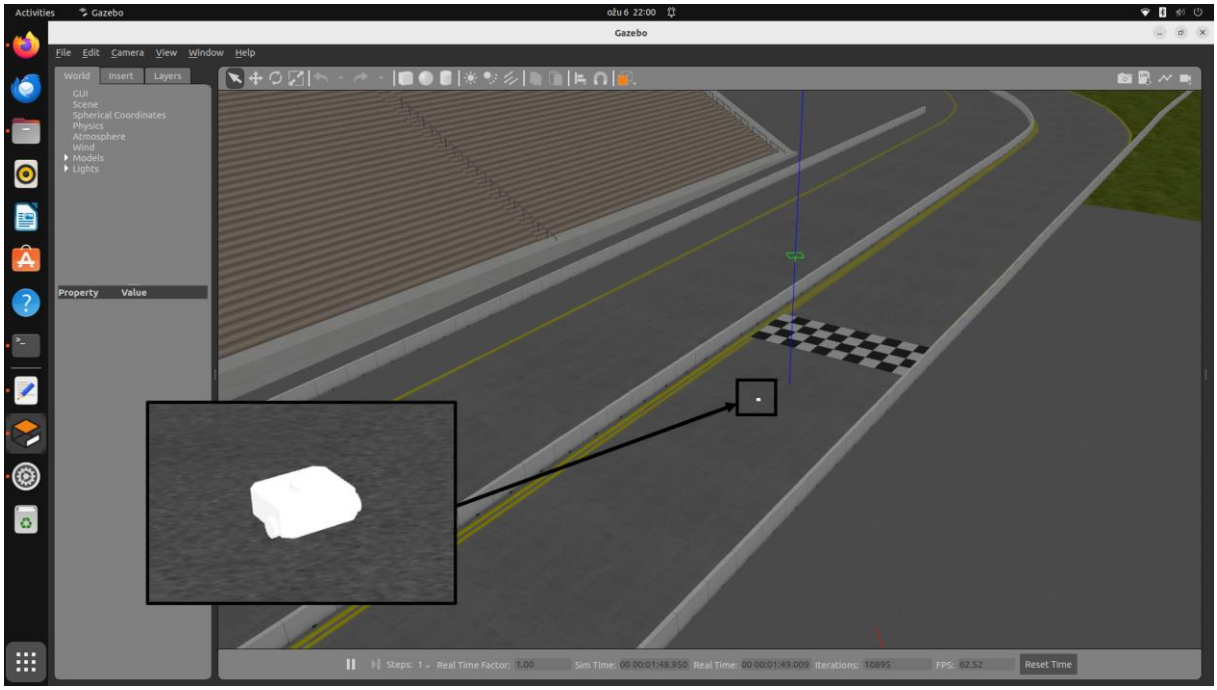
7.1.1. Postavljanje simulacijskog okruženja

U sljedećim odlomcima detaljno je opisano postavljanje simulacijskog okruženja koje je potom korišteno za validaciju algoritama lokalizacije te navigacije na temelju postavljenih putnih točaka.

7.1.1.1. Prikaz simulacijskog okruženja

Postavljanje robota u Gazebo okruženje predstavlja ključan korak u simulaciji i testiranju robotskih sustava prije stvarne implementacije na terenu. Kao što je spomenuto u prethodnom dijelu odlomka, u simulacijskom okruženju je kao model diferencijalnog robota korišten TurtleBot3 *Waffle* robot s diferencijalnim pogonom koji je potom korišten za validaciju algoritama lokalizacije i navigacije. Ovaj robot opremljen je s nekoliko senzora, uključujući GNSS prijamnik, IMU senzor, lidar te enkodere smještene na pogonske kotače. Također valja naglasiti da su modelima senzora iz Gazebo sučelja nadodani šumovi kako bi se što vjernije validirala ispravnost rada algoritama lokalizacije i navigacije. IMU senzor postavljen na model mobilnog robota mjeri kutnu brzinu i ubrzanje robota, pružajući podatke kako o orijentaciji tako i o dinamici kretanja. Ovi podaci bitni su za precizno modeliranje ponašanja robota, posebice pri promjenama brzine i smjera kretanja. Lidar senzor koristi laserske zrake za mjerenje udaljenosti do okolnih objekata, stvarajući dvodimenzionalnu mapu okoline na kojoj je moguće adresirati potencijalne prepreke kao i podobna područja kojima je omogućeno kretanje. Enkoderi, koji se često koriste u diferencijalnim pogonima, pružaju informacije o rotaciji kotača što se koristi pri određivanju relativnog položaja robota u koordinatnom sustavu odometrije.

Kao simulacijski model okruženja korišten je model Sonoma trkače staze [80] iz Kalifornije, Sjedinjene Američke Države. Korištena staza opisuje otvoreni prostor dimenzija koje odgovaraju terenu koji je naknadno korišten za validaciju ponašanja sustava na stvarnom robotu hodaču. Također, korištenjem dodatka sferičnih koordinata stvorena je lokalna ravnina smještena u središtu korištenog Gazebo modela okoline. Tako stvorena lokalna ravnina potom se koristi pri oponašanju koordinata dobivenim od strane GNSS prijamnika. Slika 38. prikazuje model TurtleBot3 Waffle robota smještenog u Gazebo simulacijskom okruženju:



Slika 38. Prikaz modela TurtleBot Waffle robota u stvorenom Gazebo okruženju

7.1.1.2. Podešavanje čvorova i poslužitelja lokalizacije i globalne navigacije

Nadalje, kako bi se koristili paketi navigacije opisani u odlomku *6. GLOBALNO PLANIRANJE PUTANJE* potrebno je podesiti sustav lokalizacije detaljnije opisan u odlomku *5. LOKALIZACIJA ROBOTA*. Nav2 paket u svom radu koristi elemente robot_localization paketa [59], točnije navsat_transform_node čvor prilikom pretvorbe GNSS koordinata u Kartezijev koordinatni sustav robota. Iz lokalizacijskog paketa se također koriste čvorovi estimatora stanja, odnosno proširenog Kalmanovog filtera s ciljem fuzije podataka s više nezavisnih izvora, detaljnije opisano u odlomku *5.3. Robot_localization paket*. Slika 39. prikazuje Mapviz sučelje simulacijskog okruženja u kojemu su u nastavku rada isprobani algoritmi lokalizacije i globalne navigacije:

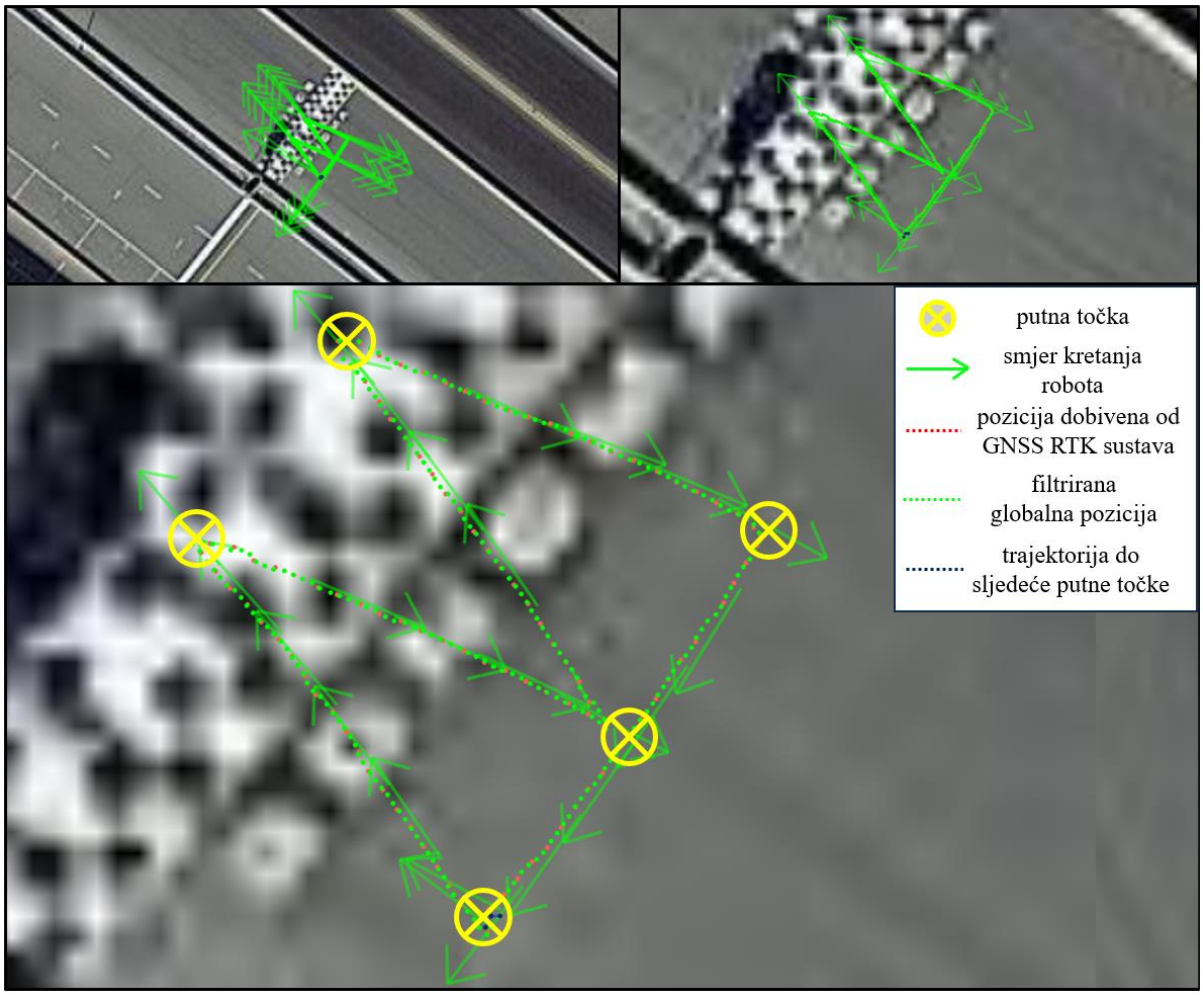
Slika 39. Prikaz Mapviz sučelja simulacijskog okruženja

Kako bi se replicirao slučaj korištenog IMU senzora bez magnetometra, unutar simulacijskog okruženja se zanemaruje postojanje IMU senzora te se orijentacija robota također provodi korištenjem isključivo podataka o odometriji. Zbog korištenja odometrije robota kao glavnog izvora orijentacije robota potrebno je istog usmjeriti prema istoku prije pokretanja čvorova zaduženih za lokalizaciju.

7.1.2. Praćenje putnih točaka bez prepreka u simulacijskom okruženju

Za validaciju gore opisanih algoritama koristi se pet međusobno povezanih putnih točaka unaprijed definiranih u Mapviz sučelju. U repozitoriju diplomskog rada [2] nalazi se .yaml

datoteka koja sadrži koordinate korištenih putnih točaka. Nadalje, videozapis [81] prikazuje praćenje putnih točaka Turtlebot3 robotom u Gazebo simulacijskom okruženju bez prepreka. Slika 40. prikazuje trajektorije mobilnog robota u Gazebo okruženju prilikom slijeđenja zadanih putnih točaka bez postavljenih prepreka:



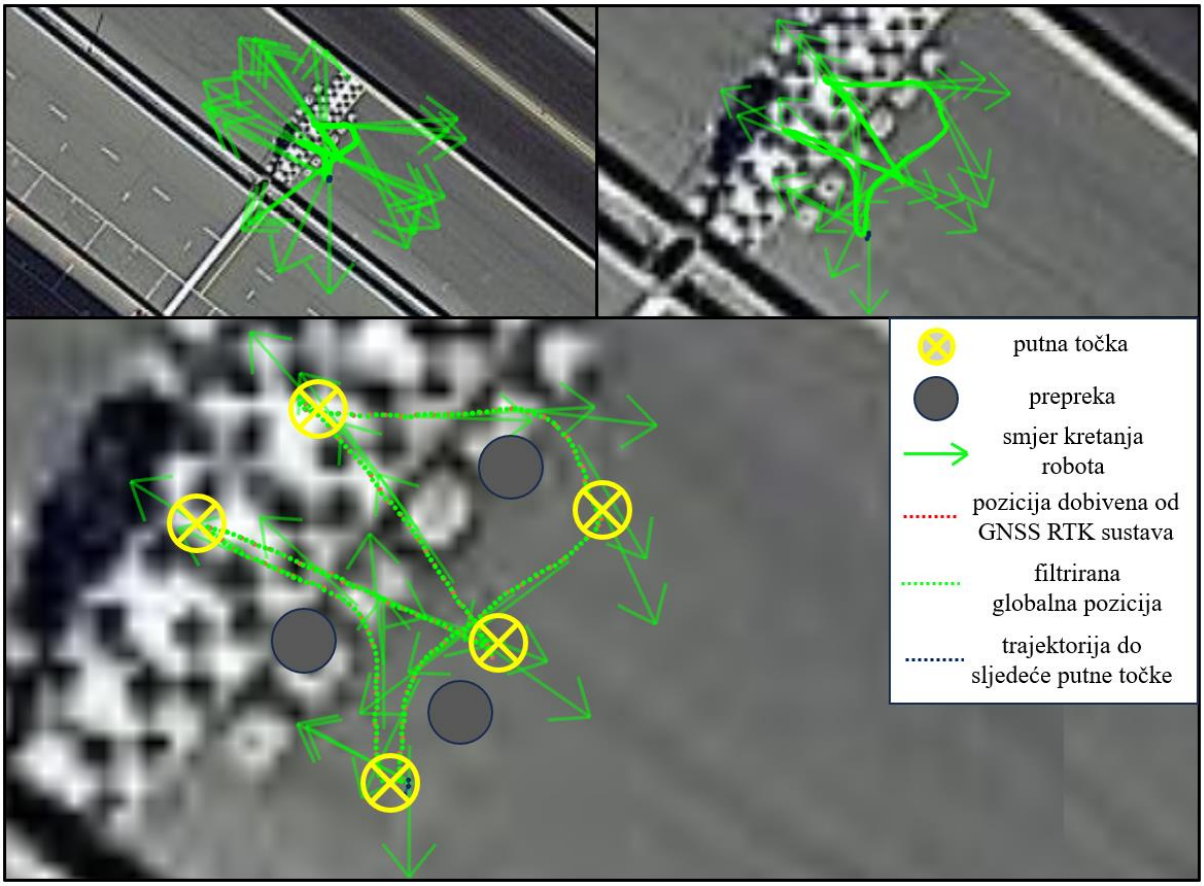
Slika 40. Praćenje putnih točaka bez prepreka u Gazebo simulacijskom okruženju prikazano u videozapisu [81]

Zelene strelice predstavljaju smjer robota koji je nastao kao fuzija podataka odometrije i IMU senzora. Crvenim točkama opisuje se globalna pozicija očitana putem GNSS prijamnika. Zelene točke predstavljaju filtriranu globalnu poziciju, odnosno poziciju koja je nastala fuzijom podataka o globalnoj poziciji, IMU senzora te odometrije. Žutim kružićima sa oznakom X označene su putne točke pri navigaciji, dok sivi puni kružići označavaju prepreke postavljene u područje potencijalnog kretanja robota. Tamno plavim točkama prikazana je potencijalna trajektorija mobilnog robota do slijedeće putne točke koja se kontinuirano osvježava prilikom približavanja robota istoj. Iz videozapisa [81] vidljivo je da robot uspješno slijedi predodređene

putne točke. Također, iz videozapisa [81] vidljivo je povremeno zastajkivanje prikaza u Mapviz sučelju koje je uzrokovano desinkronizacijom komunikacije između upravljačkog računala robota te računala na kojem se provodila navigacija. Zbog podešavanja servisa parametara kvalitete poslužitelja (QoS) s ciljem očuvanja kvalitete komunikacijskog kanala dešavaju se povremena kašnjenja vizualizacijskih podataka primjerice elemenata Mapviz sučelja. Time se doprinosi stabilnosti rada poslužitelja krucijalnih za uspješnu navigaciju te samim time i radu cijelog sustava te nema utjecaj na rad poslužitelja koji se odvijaju u pozadini.

7.1.3. Praćenje putnih točaka s preprekama u simulacijskom okruženju

Videozapis [82] prikazuje praćenje putnih točaka Turtlebot3 robotom u Gazebo simulacijskom okruženju s postavljenim preprekama. Slika 41. prikazuje trajektorije mobilnog robota u Gazebo okruženju prilikom slijeđenja zadanih putnih točaka s postavljenim preprekama:

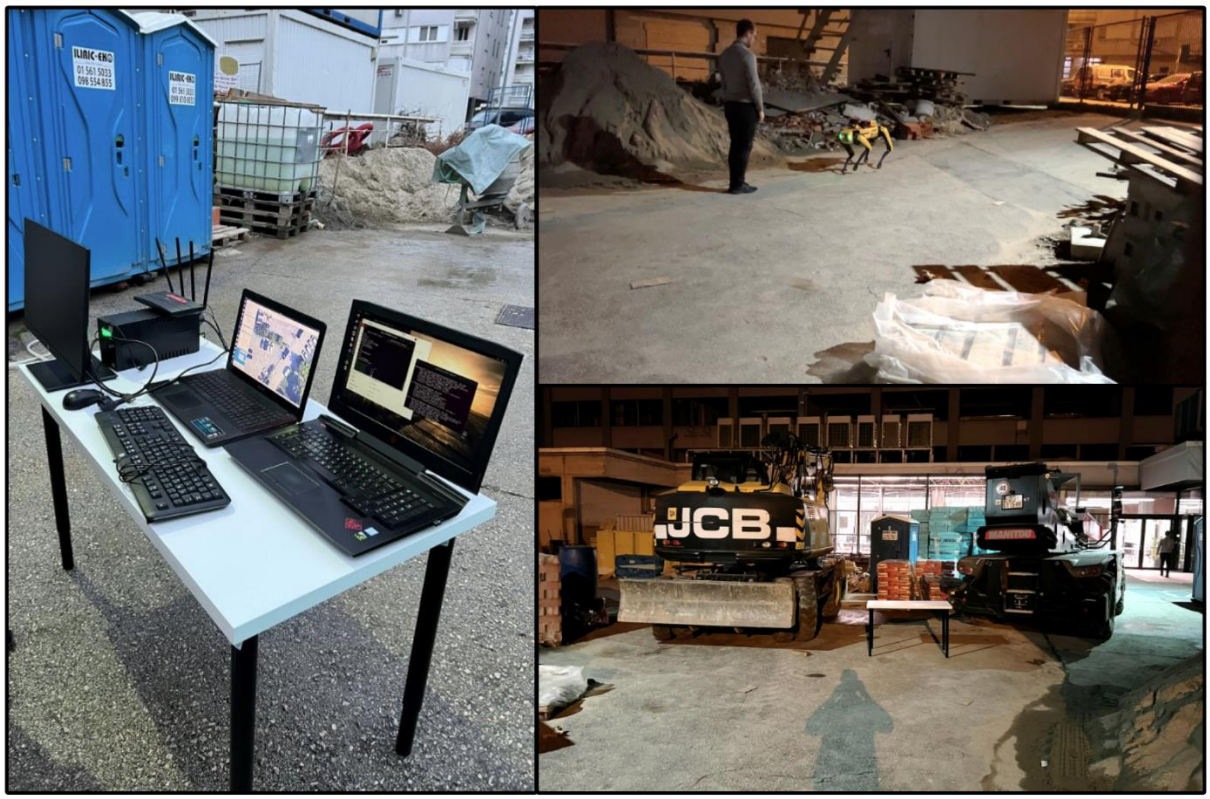


Slika 41. Praćenje putnih točaka s postavljenim preprekama u Gazebo simulacijskom okruženju prikazano u videozapisu [82]

Opis scene jednak je onomu opisanom u odlomku *7.1.2. Praćenje putnih točaka bez prepreka u simulacijskom okruženju*. Iz videozapisa [82] vidljivo je da se potencijalna trajektorija prikazana tamno plavim točkama mijenja prilikom pojave prepreke između dvije putne točke što na posljetku rezultira uspješnim slijeđenjem predodređenih putnih točaka uz izbjegavanje kolizija tijekom kretanja. Također, u videozapisu [82] je vidljivo povremeno zastajkivanje prikaz Mapviz sučelja čiji uzrok jednak onome opisanom u *7.1.2. Praćenje putnih točaka bez prepreka u simulacijskom okruženju*.

7.2. Validacija algoritama navigacije na mobilnom robotu

U nastavku je opisana validacija algoritama navigacije i lokalizacije na Spot mobilnom robotu. Za razliku od simulacijskog okruženja, okoliš provedenih testiranja karakteriziraju smetnje GNSS signala uzrokovane prisutnosti okolnih građevina te strojeva smještenih u prostorima oko Fakulteta strojarstva i brodogradnje. Slika 42. prikazuje okruženje u kojemu su testirani algoritmi navigacije i lokalizacije. Zbog prisutnosti zgrada kao i građevinskih strojeva očekuju se djelomično nepogodni rezultati uzrokovani smetnjama neposredne blizine istih što je potrebno riješiti podešavanjem parametara čvorova lokalizacije i navigacije.

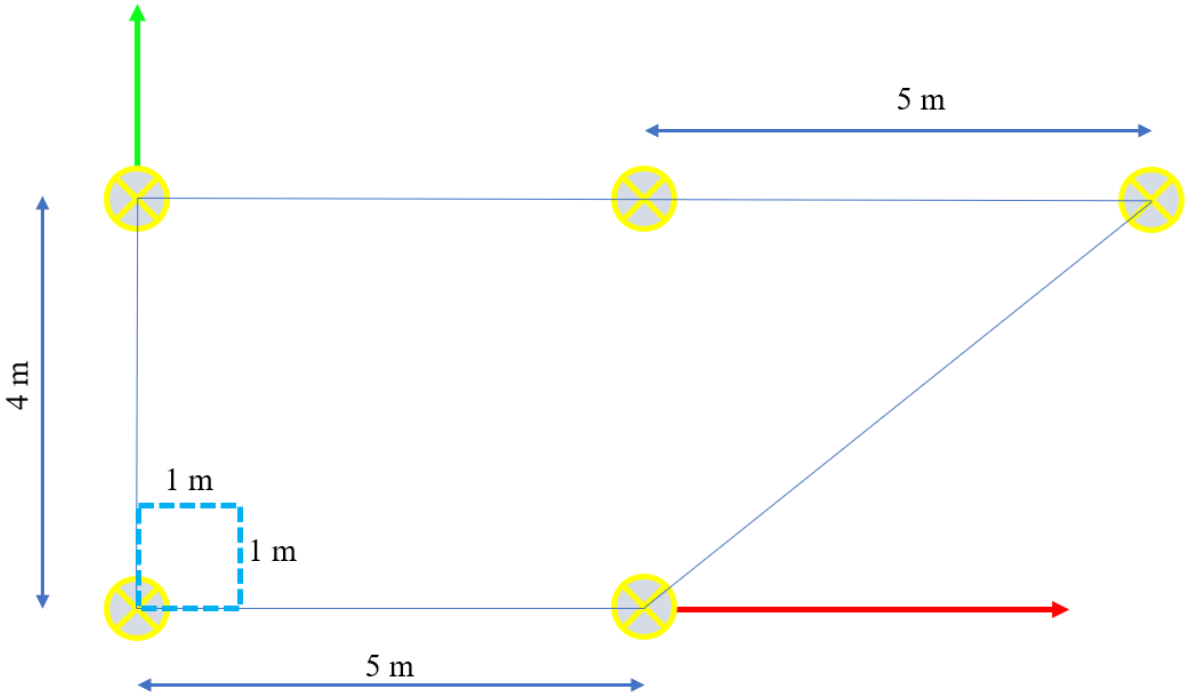


Slika 42. Prikaz okruženja validacije algoritama navigacije

Također, u videozapisu [83] detaljnije je prikazano okruženje u kojemu su validirani algoritmi lokalizacije i navigacije opisani u ovom radu.

7.2.1. Praćenje putnih točaka Spot mobilnim robotom u okolišu bez prepreka

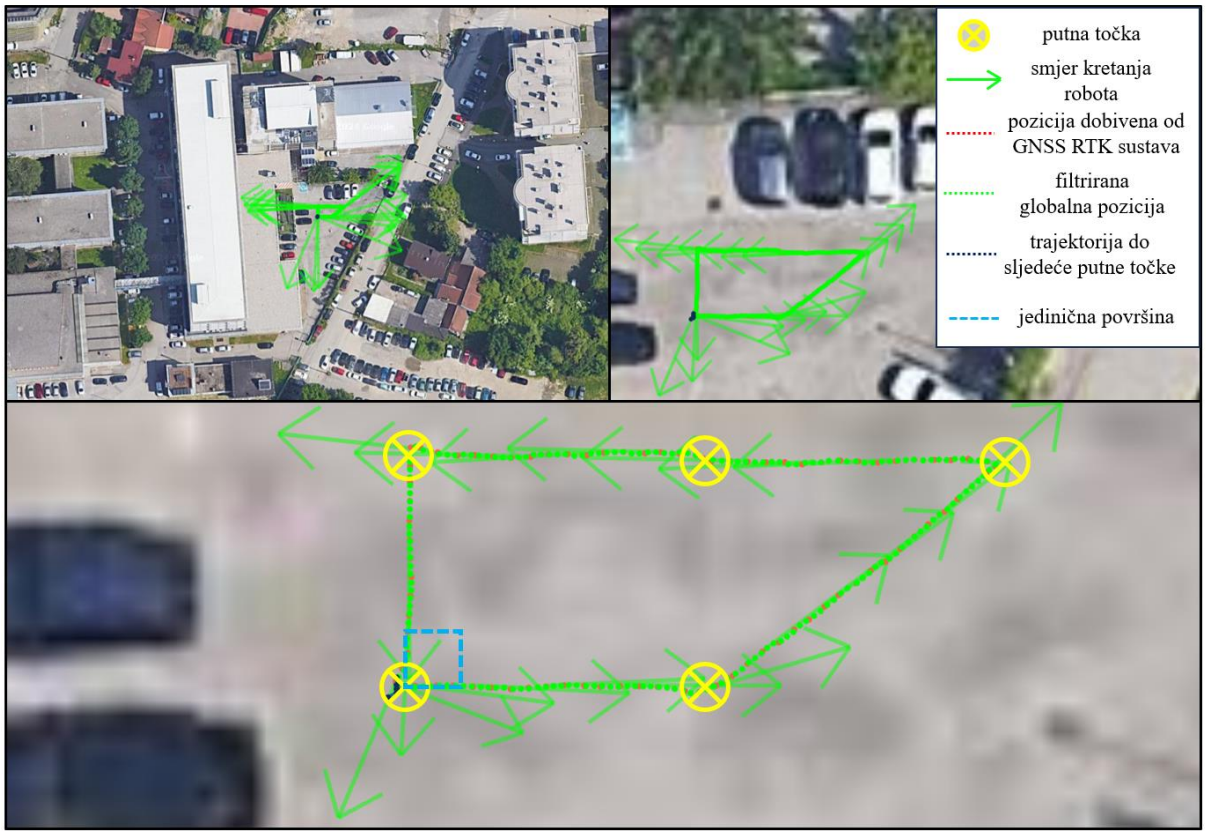
Za validaciju gore opisanih algoritama na Spot robotu koristi se pet međusobno povezanih putnih točaka (Slika 43) priloženih u repozitoriju diplomskog rada [2] unaprijed definiranih u Mapviz sučelju.



Slika 43. Prikaz putnih točaka korištenih za validaciju navigacije Spot mobilnim robotom

Na slici su žutim kružićima s oznakom x prikazane zadane putne točke. Također, plavim kvadratom s isprekidanim linijama prikazana je jedinična površina od 1 m^2 koja će u nastavku služiti kao referenca prilikom analize kretanja robota.

Videozapis [84] prikazuje praćenje putnih točaka Spot mobilnim robotom. Slika 44. prikazuje trajektorije mobilnog robota Spota prilikom slijeđenja zadanih putnih točaka bez postavljenih prepreka:



Slika 44. Praćenje putnih točaka Spot mobilnog robota bez postavljenih prepreka prikazano u videozapisu [84]

Opis scene jednak je onomu opisanom u odlomku *7.1.2. Praćenje putnih točaka bez prepreka u simulacijskom okruženju*. Dodatno, plavim kvadratom s isprekidanim linijama prikazana je jedinična površina od 1 m^2 koja će u nastavku služiti kao referenca prilikom analize kretanja robota. Iz videozapisa [84] vidljivo je da robot uspješno slijedi predodređene putne točke. Nadalje, u videozapisu [84] također je vidljivo povremeno zastajkivanje prikaza u Mapviz sučelju koje je uzrokovano desinhronizacijom komunikacije između upravljačkog računala robota te računala na kojem se provodila navigacija. Zbog podešavanja servisa parametara kvalitete poslužitelja (QoS) s ciljem očuvanja kvalitete komunikacijskog kanala dešavaju se povremena kašnjenja vizualizacijskih podataka kao što su elementi Mapviz sučelja. Time se doprinosi stabilnosti rada poslužitelja krucijalnih za uspješnu navigaciju te samim time i radu cijelog sustava. U usporedbi s navigacijom u Gazebo okruženju vidljiva su značajnija povremena kašnjenja u trajanju reda veličine sekunde zbog prijenosa podataka većeg broja

čvorova zaduženih za akviziciju informacija sa senzora kao i prijenos podataka upravljačkih algoritama mobilnog robota. Iako postoji povremeno kašnjenje u prikazu, nisu uočena kašnjenja kao ni ostale poteškoće u provođenju zadataka navigacije.

7.2.2. Praćenje putnih točaka Spot mobilnim robotom u okolišu s preprekama

Videozapis [85] prikazuje praćenje putnih točaka Spot mobilnim robotom uz prisutnost jedne prepreke. Slika 45. prikazuje trajektorije mobilnog robota Spota prilikom slijeđenja zadanih putnih točaka s postavljenom preprekom:

Slika 45. Praćenje putnih točaka Spot mobilnog robota s jednom postavljenom preprekom prikazano u videu [85]

Opis scene jednak je onomu opisanom u odlomku *7.1.2. Praćenje putnih točaka bez prepreka u simulacijskom okruženju*. Iz videozapisa [85] vidljivo je da robot uspješno slijedi predodređene putne točke uz uspješno izbjegavanje prepreke. Također, vidljivo je da robot postiže putnu točku nakon postavljene prepreke s određenim odmakom. Maksimalni dozvoljeni odmak robota od zadane putne točke je iskustveno postavljen na 280 mm kako bi se izbjegla nepoželjna gibanja prilikom približavanja putnoj točki ispred koje je slijedila prepreka te kako bi se kompenzirali nedostaci korištenja stvarne opreme. Nadalje, videozapis [86] prikazuje

praćenje putnih točaka Spot mobilnim robotom uz prisutnost dvije prepreke. Slika 46. prikazuje trajektorije mobilnog robota Spota prilikom slijeđenja zadanih putnih točaka s postavljenim preprekama:

Slika 46. Praćenje putnih točaka Spot mobilnog robota s dvije postavljene prepreke prikazano u videu [86]

Opis scene jednak je onomu opisanom u odlomku *7.1.2. Praćenje putnih točaka bez prepreka u simulacijskom okruženju*. Iz videozapisa [86] vidljivo je da robot uspješno slijedi predodređene putne točke uz uspješno izbjegavanje prepreka.

Kako bi se detaljnije prikazalo ponašanje opisanog sustava pri kretanju prema zadanim putnim točkama, u [87] dan je videozapis kompilacije četiri nezavisna slučaja praćenja putnih točaka u okruženju s dinamičkim preprekama. Slika 47. prikazuje više nezavisnih slijeđenja zadanih putnih točaka Spot mobilnog robota s postavljenim preprekama:

Slika 47. Kompilacija praćenja putnih točaka Spot mobilnog robota s postavljenim preprekama prikazana u videu [87]

Opis scene jednak je onomu opisanom u odlomku *7.1.2. Praćenje putnih točaka bez prepreka u simulacijskom okruženju*. Iz kompilacije slučajeva praćenja putnih točaka [87] vidljivo je da robot uspješno slijedi predodređene putne točke uz uspješno izbjegavanje svih prepreka.

7.3. Prijedlog budućeg rada

Tijekom rada s robotskim sustavom opisanim u ovom radu uočeni su potencijalni smjerovi za unaprjeđenje sustava. Kao prvi prijedlog budućeg rada na sustavu navodi se nadogradnja sustava mobilnog robota s komunikacijskim modulom kako bi se povezivanje upravljačkih računala moglo provoditi s udaljenih lokacija. Nadalje, potrebno je postaviti metodu određivanja postignute točke u prostoru kako bi se omogućila validacija točnosti sustava uz korištenje ovdje opisanih sustava navigacije i lokalizacije. Isto je moguće postići postavljanjem Aruco markera u neposrednoj blizini svake od putnih točaka gdje bi se provođenjem transformacija koordinatnih sustava markera u odnosu na bazu robota odredio položaj robota u

koordinatnom sustavu markera. Također, moguće je postaviti 3D vizijski sustav praćenja, primjerice OptiTrack kojim bi se odredila pozicija robota u odnosu na zadanu putnu točku s točnosti reda veličine milimetra. Nadalje, ponašanje robota moguće je dodatno ugladiti provođenjem podešavanja parametara poslužitelja navigacije. Također, nedostatak magnetometra na korištenom IMU senzoru dovodi do nemogućnosti određivanja orijentacije robota u odnosu na magnetski sjever. Isto može biti riješeno korištenjem IMU senzora koji u svojoj arhitekturi sadržava magnetometar ili pak korištenjem GNSS sustava s dva prijamnika, čime se također može precizno odrediti globalna orijentacija mobilnog robota.

8. ZAKLJUČAK

Osnovna ideja ovoga rada bila je pokazati upotrebu algoritama lokalizacije i globalne navigacije u sklopu robotskog operativnog sustava. Kako bi se omogućio ispravan rad autonomnog vozila, potrebno je bilo uspostaviti komunikaciju između korištenih upravljačkih računala te korištene dodatne opreme. Također, tijekom izrade rada istražen je i primijenjen globalni navigacijski satelitski sustav (GNSS) primjenom sustava povećane točnosti kinematike u stvarnom vremenu (RTK) kako bi se anulirao utjecaj rasipanja podataka globalne pozicije autonomnog sustava. Nadalje, fuzijom podataka globalne pozicije s podacima odometrije kao i podacima dobivenih od inercijalne mjerne jedinice (IMU) omogućena je kontinuirana lokalizacija autonomnog vozila u vanjskom prostoru. Također, prilikom izrade rada istraženi su i primijenjeni planeri putanje temeljeni na poziciji dobivenoj od strane algoritama globalne lokalizacije. Po završetku provedene lokalizacije korištenjem GNSS signala provedena je validacija navigacije autonomnog vozila u zadanoj mapi pri kojoj su podešeni parametri korištenih poslužitelja i čvorova zaduženih za lokalizaciju i globalnu navigaciju. Iz validacije opisane u ovom radu vidljivo je da se korištenjem RTK sustava u fuziji s podacima od IMU jedinice te odometrije omogućava lokalizacija robota u subcentimetarskom području što se smatra pogodnim za većinu zadataka globalne navigacije. Također, prilikom izrade ovog rada konstruirani su i izrađeni nosači pomoću kojih se osigurava brza, jednostavna i sigurna integracija korištene dodatne opreme na Spot mobilnog robota.

Kao ishod ovog diplomskog rada navodi se konfiguracija platforme koja u budućim radovima može biti korištena kao baza pri testiranju raznih postojećih algoritama, razvoju novih algoritama globalnog planiranja putanje te sličnih zadataka koji se stavljaju pred autonomna vozila namijenjena za upotrebu u vanjskom prostoru. Također, kao svrha ovog diplomskog rada navodi se stvaranje šire slike o mogućnostima korištenja dostupnih paketa otvorenog koda (engl. *open source*) u području autonomne mobilnosti, s ciljem povećavanja sposobnosti autonomnih sustava u stvarnoj primjeni.

LITERATURA

[1] https://clearpathrobotics.com/gps-waypoint-navigation-package/ (29.2.2024.)

[2] https://github.com/fmiklauzic/global_navigation_spot (14.3.2024.)

[3] F. Miklaužić, “Pregled i odabir GNSS sustava subcentimetarske točnosti“, Projekt MiR, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, Zagreb, 2023, https://github.com/fmiklauzic/global_navigation_spot

[4] https://www.techtarget.com/iotagenda/definition/mobile-robot-mobile-robotics (1.3.2024.)

[5] L. Jurišica, R. Murár, “Mobile robots and their subsystems“, MOBILNÉ ROBOTY, 2008

[6] T. Mikołajczyk et al., “Energy Sources of Mobile Robot Power Systems: A Systematic Review and Comparison of Efficiency“, Applied Sciences 13, 2023

[7] F. Miklaužić, "Praćenje objekta mobilnim robotom primjenom računalnog vida", Završni rad, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, Zagreb, 2022

[8] https://notumrobotics.com/ (8.3.2024.)

[9] https://www.ros.org/ (20.12.2023.)

[10] https://www.linkedin.com/advice/0/how-can-you-decide-between-ros-1-2-skills-ros (20.12.2023.)

[11] https://www.model-prime.com/blog/ros-1-vs-ros-2-what-are-the-biggest-differences (20.12.2023.)

[12] https://docs.ros.org/en/foxy/Concepts/About-Executors.html (3.1.2024.)

[13] https://www.masterclock.com/gps-vs-gnss.html (3.1.2024.)

[14] https://commons.wikimedia.org/wiki/File:Real_time_kinematic.svg (12.12.2023.)

[15] https://www.sparkfun.com/products/20000 (12.12.2023.)

[16] https://www.sparkfun.com/products/19032 (13.12.2023.)

[17] https://holybro.com/products/sik-telemetry-radio-v3 (19.12.2023.)

[18] https://github.com/KumarRobotics/ublox/tree/ros2 (13.12.2023.)

[19] https://www.u-blox.com/en/product/u-center (3.1.2024.)

[20] https://github.com/danielsnider/MapViz-Tile-Map-Google-Maps-Satellite/tree/master (3.1.2024.)

[21] https://github.com/danielsnider/docker-mapproxy-googlemaps (5.2.2024.)

[22] https://github.com/swri-robotics/mapviz (13.12.2023.)

[23] https://bostondynamics.com/ (13.2.2024.)

[24] https://parametric-architecture.com/how-can-boston-dynamics-spot-have-a-role-in-construction-and-architecture-sector/ (3.1.2024.)

[25] https://bostondynamics.com/products/spot/ (3.1.2024.)

[26] W. Yan, Y. Pan et al.,"Whole-body kinematic and dynamic modeling for quadruped robot under different gaits and mechanism topologies", PeerJ Computer Science, 2021

[27] https://support.bostondynamics.com/s/article/Robot-specifications (3.1.2024.)

[28] https://www.intelrealsense.com/depth-camera-d435/ (4.1.2024.)

[29] https://www.clearpathrobotics.com/assets/guides/melodic/spotros/ros_usage.html#cameras-and-depth-clouds (4.1.2024.)

[30] https://github.com/IntelRealSense/realsense-ros (17.1.2024.)

[31] https://support.bostondynamics.com/s/article/Spot-Enhanced-Autonomy-Package-EAP (28.2.2024.)

[32] https://www.intuitive-robots.com/spot-robot-payloads-and-accessories/spot-core-i-o-payload/ (4.1.2024.)

[33] https://github.com/ros-perception/pointcloud_to_laserscan/tree/humble (4.2.2024.)

[34] D. Mirota, J. Scaife, Intel RealSense Depth Camera D435i IMU Calibration, Intel, 2019

[35] https://www.asus.com/displays-desktops/nucs/nuc-kits/nuc-13-pro-kit/ (27.2.2024.)

[36] https://history-computer.com/intel-nuc-13-pro-mini-pc-review/ (28.1.2024.)

[37] https://www.eprosima.com/index.php (6.2.2024.)

[38] https://github.com/ros2/rmw_fastrtps/tree/humble (6.2.2024.)

[39] https://docs.ros.org/en/iron/Installation/DDS-Implementations/Working-with-Eclipse-CycloneDDS.html (28.2.2024.)

[40] https://projects.eclipse.org/projects/iot.cyclonedds/who (28.2.2024.)

[41] https://docs.ros.org/en/iron/Concepts/Intermediate/About-Quality-of-Service-Settings.html (28.2.2024.)

[42] https://dev.bostondynamics.com/ (9.1.2024.)

[43] https://github.com/bdaiinstitute/spot_ros2 (9.1.2024.)

[44] https://github.com/heuristicus/spot_ros (30.1.2024.)

[45] https://support.bostondynamics.com/s/article/Spot-robot-mounting-rails (30.1.2024.)

[46] https://www.ros.org/reps/rep-0105.html (5.1.2024.)

[47] https://www.ros.org/reps/rep-0103.html (8.1.2024.)

[48] https://www.andre-gaschler.com/rotationconverter/ (27.2.2024.)

[49] https://dev.bostondynamics.com/docs/payload/robot_electrical_interface (12.2.2024.)

[50] https://support.bostondynamics.com/s/article/Spot-General-Expansion-Payload-GXP (12.2.2024.)

[51] https://download.intel.com/newsroom/2023/client-computing/Intel-NUC-13-Pro-Tech-Product-Spec.pdf (13.2.2024.)

[52] S. Huang, D, Dissanayake, "Robot Localization: An Introduction", Wiley Encyclopedia of Electrical and Electronics Engineering, 2016

[53] https://thekalmanfilter.com/kalman-filter-explained-simply/ (4.1.2024.)

[54] https://jannik-zuern.medium.com/robot-localization-with-kalman-filters-and-landmarks (13.2.2024.)

[55] http://bilgin.esme.org/BitsAndBytes/KalmanFilterforDummies (4.1.2024.)

[56] https://math.stackexchange.com/questions/2338643/advantages-and-disadvantages-of-extended-kalman-filter-vs-unscented-kalman-filte (8.2.2024)

[57] E. Wan, R. van der Merwe, The Unscented Kalman Filter for Nonlinear Estimation, Oregon Graduate Institute of Science and Technology, 2000

[58] L. Yan, L. Zhao, "An approach on advanced unscented Kalman filter from mobile robot-slam", XXIV ISPRS Congress 2020

[59] https://github.com/cra-ros-pkg/robot_localization (4.1.2024.)

[60] https://docs.ros.org/en/melodic/api/robot_localization/html/index.html (4.1.2024.)

[61] https://docs.ros.org/en/jade/api/robot_localization/html/integrating_gps.html (4.1.2024.)

[62] https://wiki.ros.org/nmea_navsat_driver (5.1.2024.)

[63] https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml (10.1.2024.)

[64] https://github.com/ros-planning/navigation2 (18.2.2024.)

[65] https://github.com/ros-planning/navigation (10.3.2024.)

[66] E. Berger et al., "The Office Marathon: Robust navigation ina n indoor office environment", International Conference on Intelligent Robots and Systems, 2020

[67] S. Macenski et al., "The Marathon 2: A Navigation System", International Conference on Intelligent Robots and Systems, 2020

[68] https://navigation.ros.org/index.html (17.2.2024.)

[69] https://navigation.ros.org/configuration/index.html (26.2.2024.)

[70] M. Colledanchise, P. Örgen, Behaviour Trees in Robotics and AI: An Introduction, Chapman & Hall/CRC Artificial Intelligence and Robotics Series 2018

[71] https://www.behaviortree.dev/ (6.1.2024.)

[72] https://github.com/ros-planning/navigation2/tree/main/nav2_controller (18.2.2024.)

[73] https://navigation.ros.org/plugins/index.html (18.2.2024.)

[74] S. Persson, I. Sharf, "Sampling-based A* algorithm for robot path-planning", The International Journal of Robotics Research, 2014

[75] J. Liu et al., " Fast Planning of Globally Optimal Paths for All Points in 2D Continuous Space", Cornell University, 2023.

[76] https://swri-robotics.github.io/mapviz/plugins/ (29.2.2024.)

[77] https://github.com/ros-planning/navigation2_tutorials (29.2.2024.)

[78] https://www.linz.govt.nz/guidance/geodetic-system/coordinate-systems-used-new-zealand/geodetic-datums/world-geodetic-system-1984-wgs84 (29.2.2024.)

[79] https://github.com/ROBOTIS-GIT/turtlebot3/tree/ros2 (29.2.2024.)

[80] https://dev.px4.io/v1.11_noredirect/en/simulation/gazebo_worlds.html (29.2.2024.)

[81] https://youtu.be/nmoWdBqOi2Q

[82] https://youtu.be/gmgwzJu7m_A

[83] https://youtu.be/QcdZK2d_sl4

[84] https://youtu.be/vPDoR0G0X_4

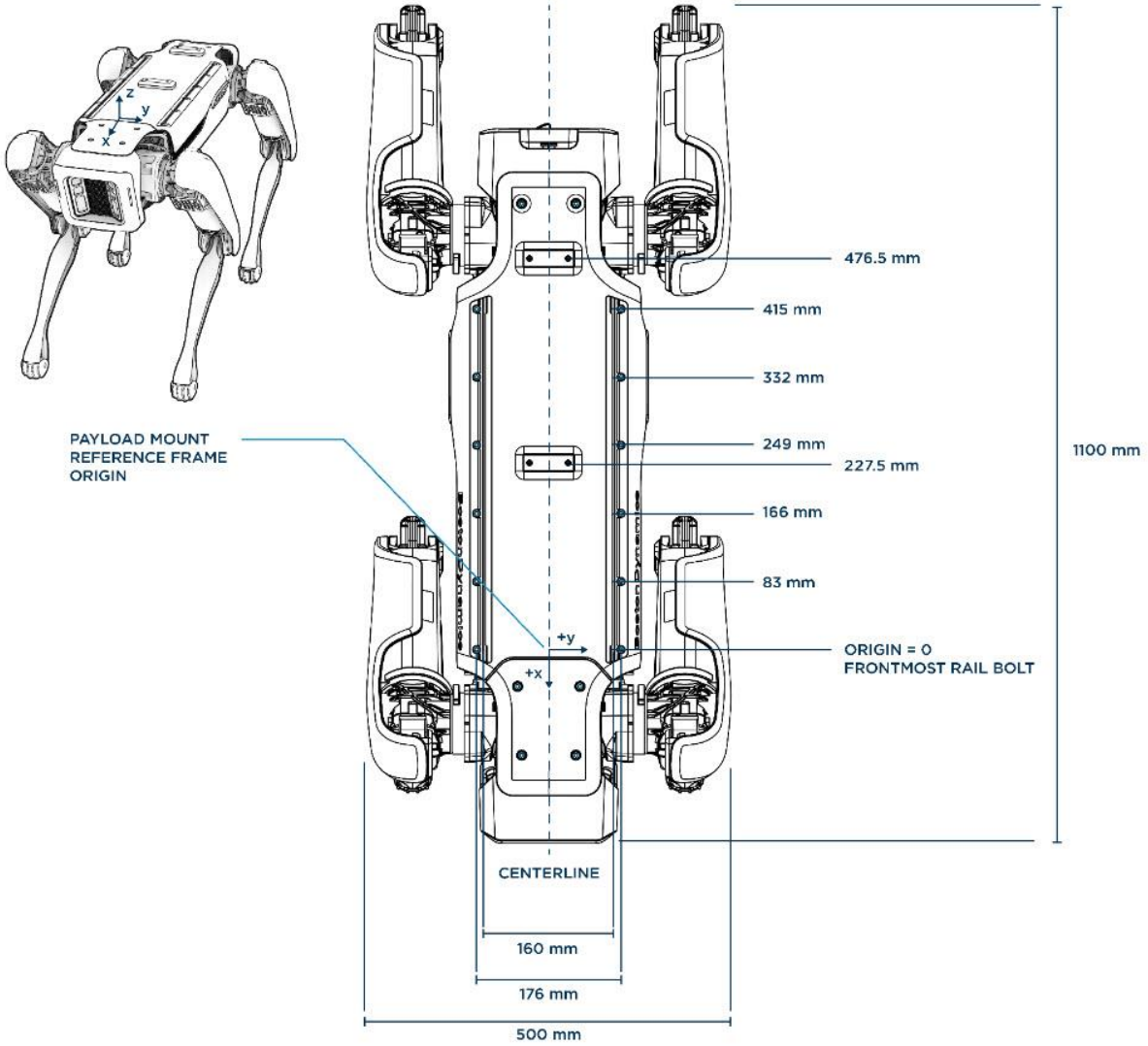
[85] https://youtu.be/H3HxzD9agio

[86] https://youtu.be/TBGFqyuLiZw

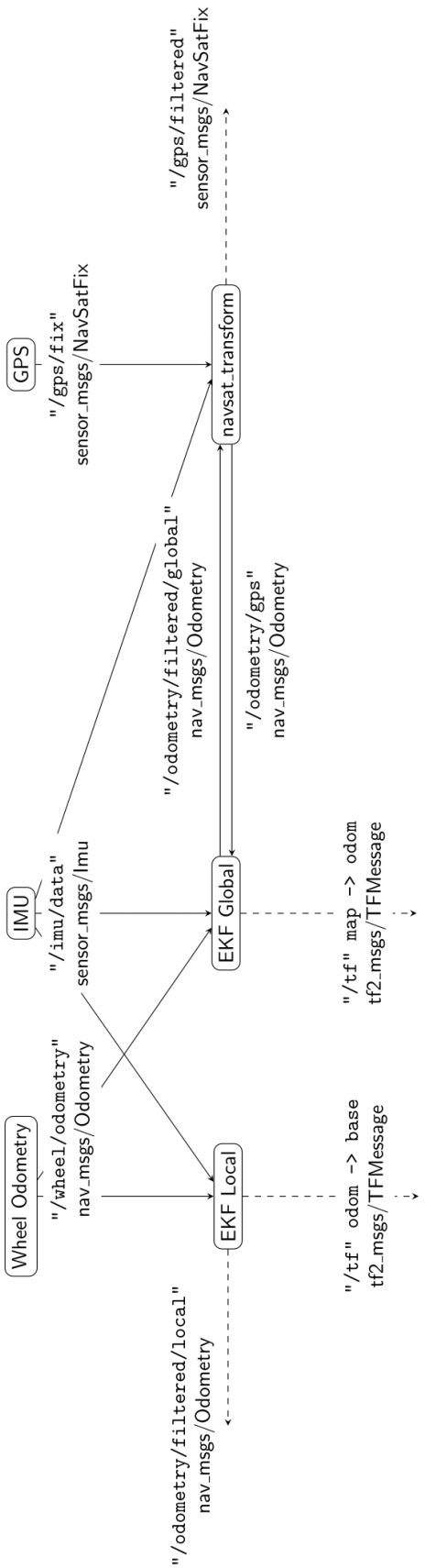
[87] https://youtu.be/WR-_aAT592o

PRILOZI

I. Prikaz međusobnog položaja vodilica za montiranje dodataka i baze sustava za dodatke [45]



II. Struktura tema pri globalnoj lokalizaciji [60]



III. Magnetska deklinacija lokacije Fakulteta strojarstva i brodogradnje u Zagrebu [63]

Declination

Date	2024-01-18
Latitude	45° 47' 44" N
Longitude	15° 58' 14" E
Elevation	0.0 km GPS
Model Used	WMM-2020
Declination	4° 53' E changing by 0° 9' E per year
Uncertainty	0° 22'

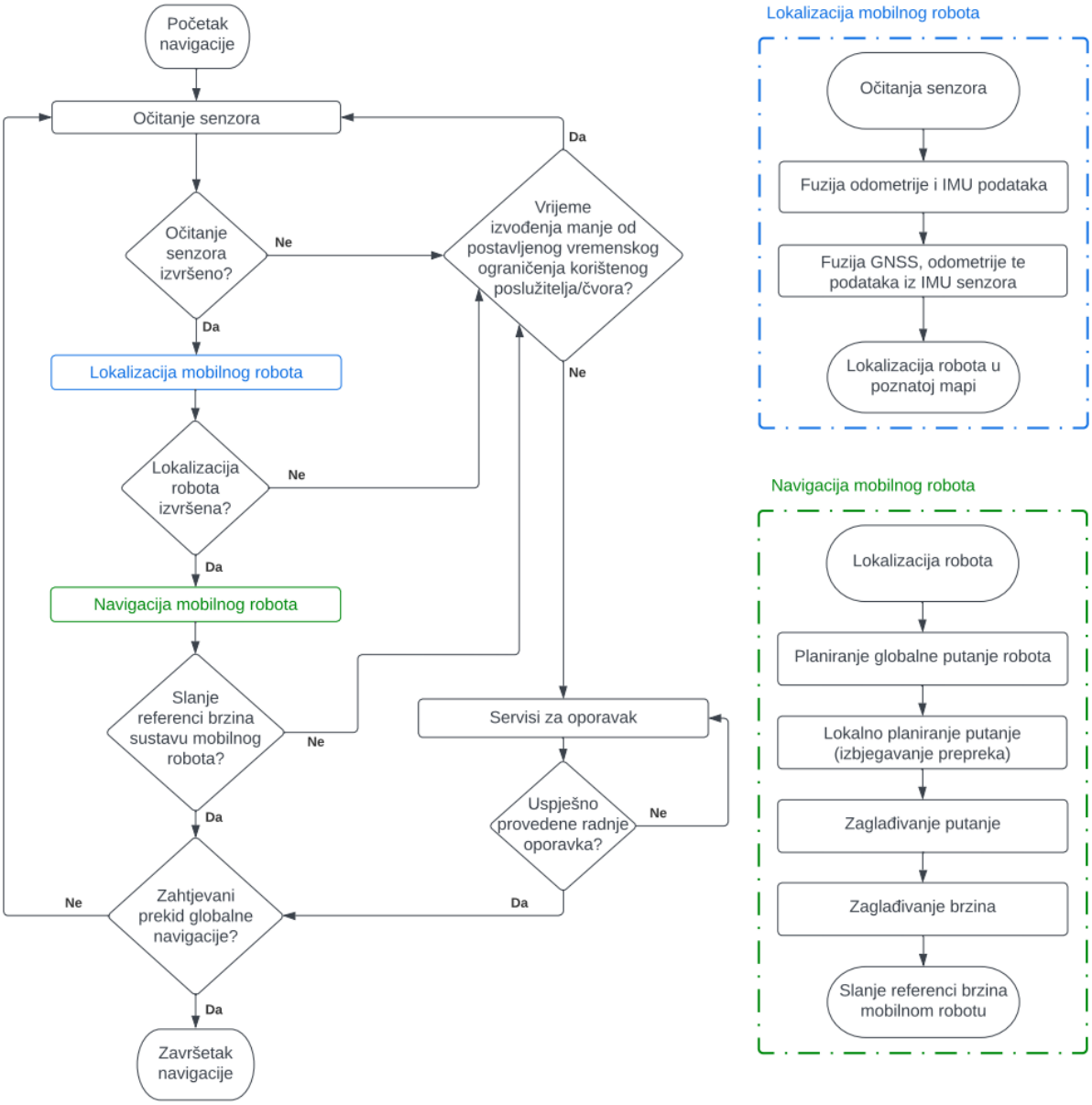
The compass shows the bearing of the local magnetic field relative to geographic north.

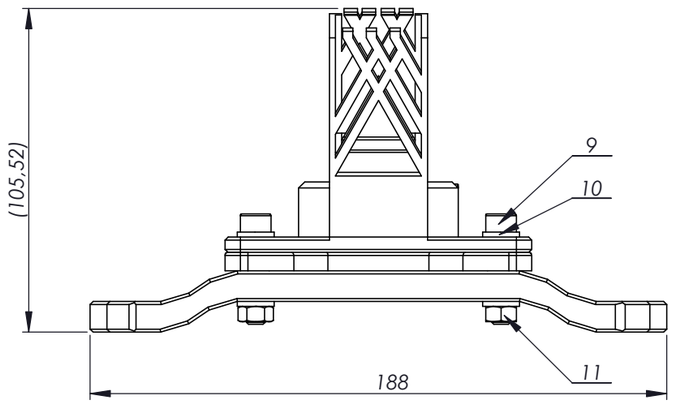
Magnetic declination is the angle between true north and the horizontal trace of the local magnetic field. In general, the present day field models such as the IGRF and World Magnetic Model (WMM) are accurate to within 30 minutes of arc for the declination. However, local anomalies exceeding 10 degrees, although rare, do exist.

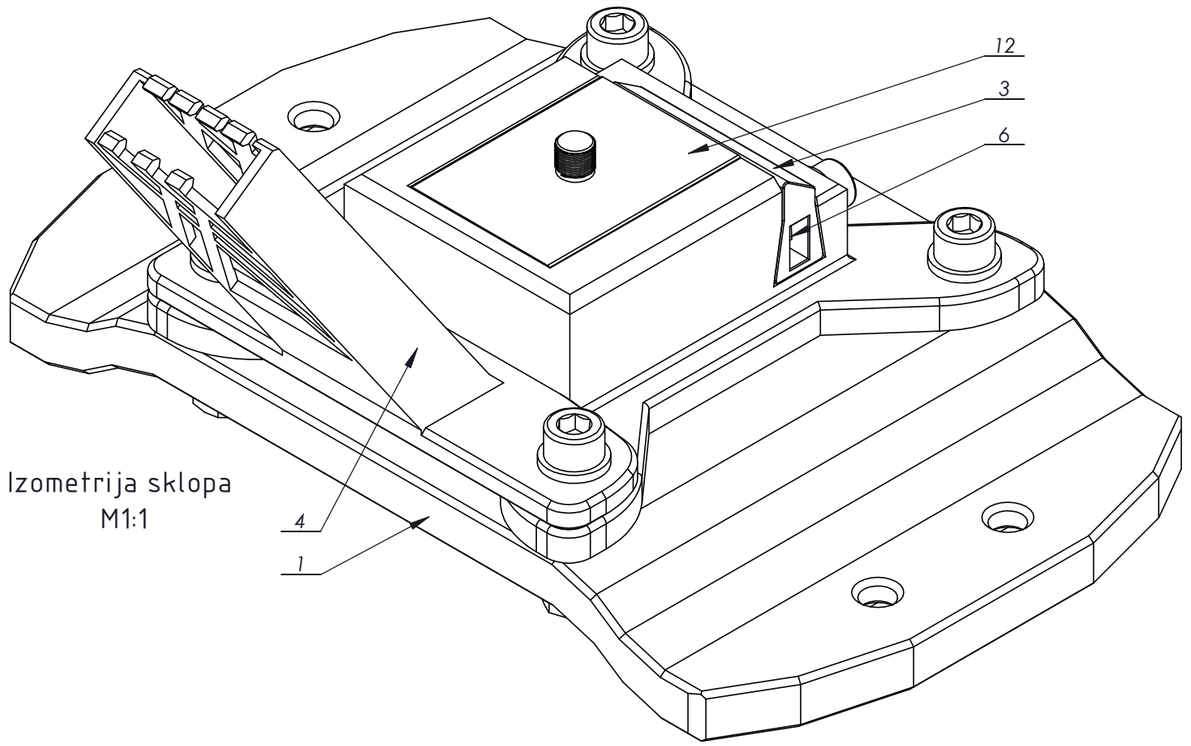
Document created: 2024-01-18 20:31 UTC

Help: How to interpret results Questions: geomag.models@noaa.gov

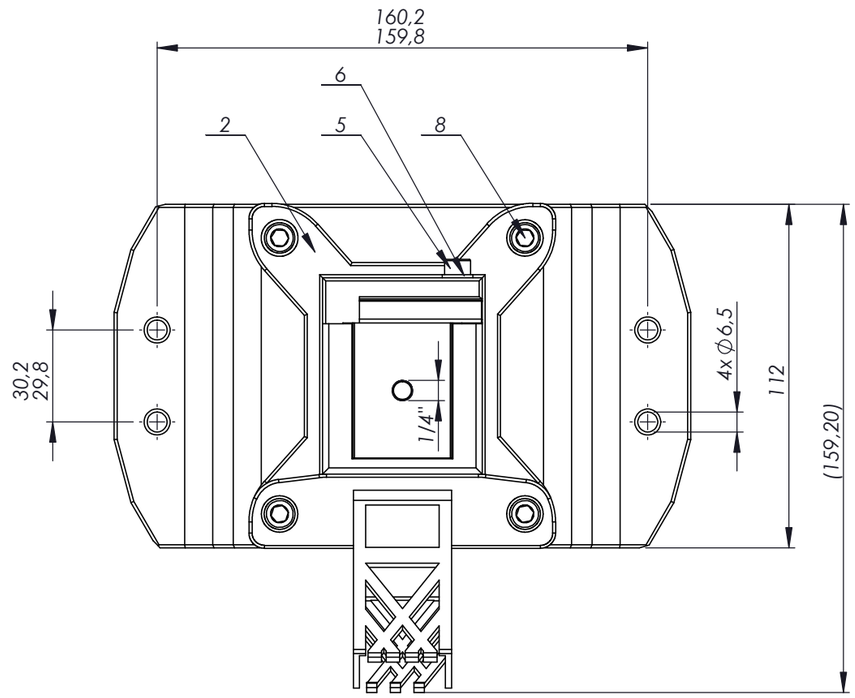
IV. Dijagram toka konceptualnog prikaza lokalizacije i globalne navigacije mobilnog robota







Izometrija sklopa
M1:1



Naponema: GNSS prijamnik se montira na priključak za brzo otpuštanje putem 1/4" vijka (12)

Poz.	Naziv dijela	Kom.	Crtez broj Norma	Materijal	Sirove dimenzije Proizvodac	Masa
12	Priklj. za brzo otpuštanje predmeta s 1/4" navojem	1	DIN 4503-1			
11	Podlozna plocica Ø6	8	DIN 125	S235		
10	Imbus vijak M6x25	2	DIN 912	Čelik 8.8, cinčano		
9	Matica M6	4	DIN 934	Čelik 8, cinčano		
8	Imbus vijak M5x20	2	DIN 912	Čelik 8.8, cinčano		
7	Podlozna plocica Ø5	1	DIN 125	S235		
6	Matica M5	1	DIN 934	Čelik 8, cinčano		
5	Imbus vijak M5x20	1	DIN 912	Čelik 8.8, cinčano		
4	Drzac antene za prijenos korekcijskih podataka	1	NGNSS-00-04	PLA		
3	Osigurac prikljucka GNSS prijamnika	1	NGNSS-00-03	PLA		
2	Nosac prikljucka GNSS prijamnika	1	NGNSS-00-02	PLA		
1	Ploca za montazu na vodilice za dodatke Spot robota	1	NGNSS-00-01	PLA		

Broj naziva - code

	Datum	Ime i prezime	Potpis
Projektirao	05.01.2024.	Filip Miklauzic	
Razradio	05.01.2024.	Filip Miklauzic	
Crtao	05.01.2024.	Filip Miklauzic	
Pregledao			
Mentor			

FSB Zagreb
Studij strojarstva

ISO - tolerancije

Objekt: GNSS sustav

Objekt broj:

R. N. broj:

Napomena:

Smjer: Mehatronika i robotika

Kopija

Materijal:

Masa:

DIPLOMSKI RAD

Mjerilo originala
M1:2

Naziv: Nosac GNSS prijamnika i antene za prijenos korekcijskih podataka

Pozicija:

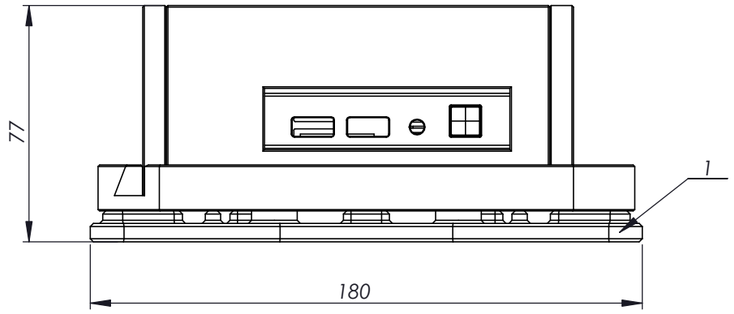
Format: A3

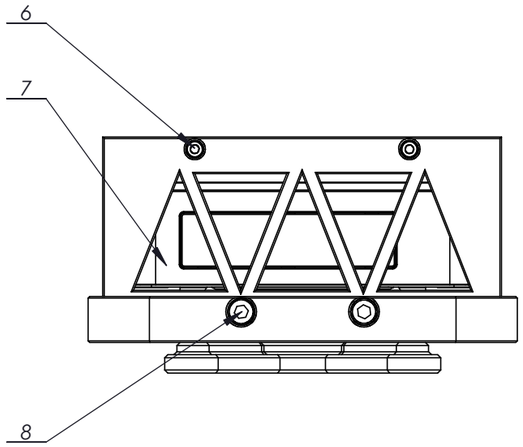
Listova: 1

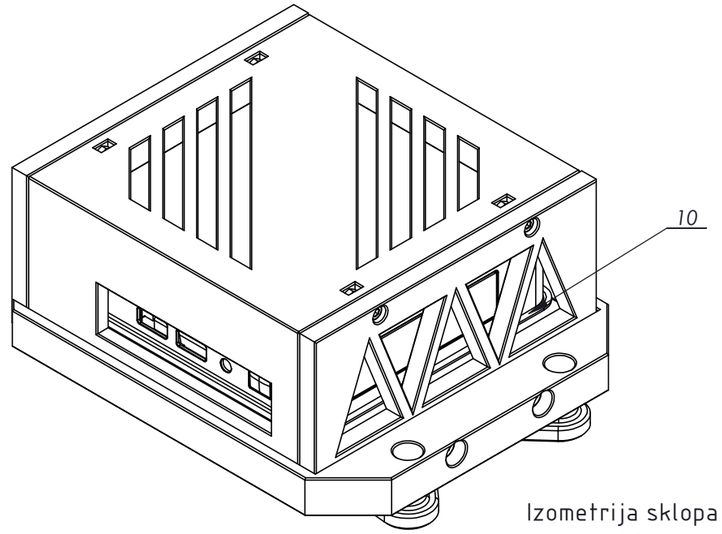
Crtež broj: NGNSS-00-00

List: 1

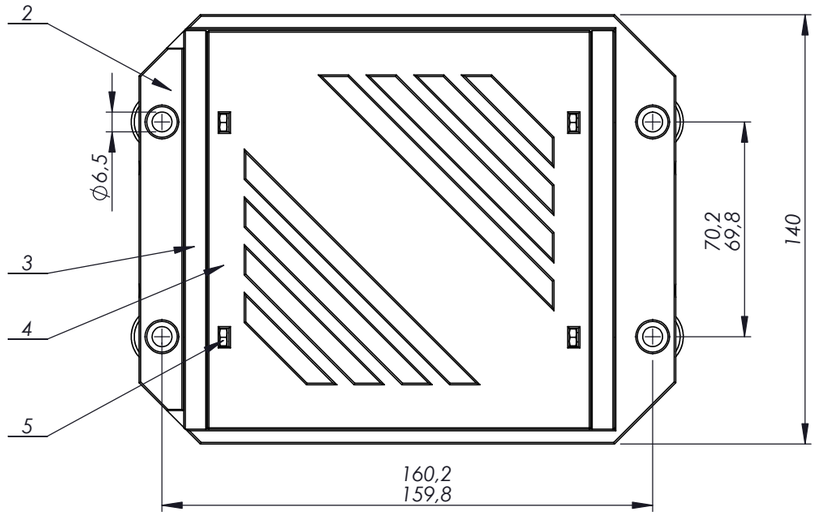
Design by CADLab







Izometrija sklopa
M1:2



Poz.	Naziv dijela	Kom.	Crtez broj Norma	Materijal	Sirove dimenzije Proizvodac	Masa
11	Imbus vijak s upustenom glavom	4	DIN 7991	Čelik 8.8, cinčano		
10	VESA nosac 100x100	1		S235		
9	Matica M5	4	DIN 934	Čelik 8, cinčano		
8	Imbus vijak M5x30	4	DIN 912	Čelik 8.8, cinčano		
7	Intel Nuc 13 racunalo	1				
6	Imbus vijak M3x16	8	DIN 912	Čelik 8.8, cinčano		
5	Matica M3	8	DIN 934	Čelik 8, cinčano		
4	Gornji pokrov baze	1	NNUC-00-04	PLA		
3	Bocni pokrov baze	2	NNUC-00-03	PLA		
2	Baza nosaca Nuc racunala	1	NNUC-00-02	PETG		
1	Distanca nosaca Nuc racunala	1	NNUC-00-01	PETG		

		Datum	Ime i prezime	Potpis
Broj naziva - code	Projektirao	10.01.2024.	Filip Miklauzic	
	Razradio	10.01.2024.	Filip Miklauzic	
	Crtao	10.01.2024.	Filip Miklauzic	
	Pregledao			
	Mentor			

FSB Zagreb
Studij strojarstva

ISO - tolerancije

Objekt: Nuc racunalo

Objekt broj:

R. N. broj:

Napomena:

Smjer: Mehatronika i robotika

Kopija

Materijal:

Masa:

DIPLOMSKI RAD

Naziv: Nosac Intel Nuc 13 Pro racunala

Pozicija:

Format: A3

Mjerilo originala M1:2

Listova: 1

Crtež broj: NNUC-00-00

List: 1

Napomena: VESA nosac je montiran na bazu nosaca (2) putem 2 M3x16 vijka s upustenom glavom DIN 7991 (11).

Napomena: Nuc računalo (7) je postavljeno na VESA nosac (10) putem 4 M3x16 vijka DIN 912 (6) i 4 matice M3 DIN 934 (5).

Napomena: Bocni pokrov baze (3) povezan je s bazom nosaca (2) putem 4 imbus vijka M5x30 DIN 912 (8) i 4 matice M5 DIN 934 (9).

Design by CADLab