

Kinematska kalibracija mobilnog robota diferencijalne strukture koristeći OptiTrack sustav

Koprivnjak, Zvonimir

Undergraduate thesis / Završni 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:235:230204>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-04**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Zvonimir Koprivnjak

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

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

Student:

Zvonimir Koprivnjak

Zagreb, 2024.

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

Zahvaljujem se mentoru doc. dr. sc. Marku Švaci na usmjerenju i komentarima, te doktorandu i asistentu Branimiru Čaranu, mag. ing. mech. na stalnoj dostupnosti, danim komentarima, savjetima i usmjerenjima za vrijeme izrade ovog rada.

Zahvaljujem se također svojoj obitelji, prijateljima te kolegama na podršci tijekom studiranja i pisanja ovog rada.

Zvonimir Koprivnjak



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomске ispite
Povjerenstvo za završne i diplomске ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

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

ZAVRŠNI ZADATAK

Student: **Zvonimir Koprivnjak** JMBAG: **0035233316**

Naslov rada na hrvatskom jeziku: **Kinematska kalibracija mobilnog robota diferencijalne strukture koristeći OptiTrack sustav**

Naslov rada na engleskom jeziku: **Kinematic calibration of a differential drive mobile robot using the OptiTrack system**

Opis zadatka:

Mobilni robot diferencijalne kinematske strukture ASTRO - Autonomous System for Teaching Robotics razvijen je kao prototip niske razine u Regionalnom centru izvrsnosti za robotske tehnologije – CRTA. Robot je opremljen s dva pogonska motora s enkoderima, dubinskom kamerom, LIDAR senzorom i inicijalnom mjernom jedinicom. Izrađen je primjenom aditivnih tehnologija i laserskog rezanja te je zbog toga podložan nesavršenostima koje se javljaju prilikom izračuna brzine, pozicije i orijentacije. S obzirom na mogućnosti pogrešaka nastalih uslijed proizvodnje i sklapanja dijelova, u sklopu ovog završnog rada potrebno je:

- Istražiti metode kalibracije mobilnih robota diferencijalne kinematske strukture.
- Snimiti slijeđenje ravne, kvadratne i kružne putanje s nominalnim dimenzijama robota primjenom OptiTrack sustava.
- Primijeniti istražene kalibracijske metode i izračunati korekcijske parametre mobilnog robota.
- Primjenom korekcijskih parametara snimiti novo slijeđenje putanja te usporediti rezultate.

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

Zadatak zadan:

30. 11. 2023.

Zadatak zadao:

Doc. dr. sc. Marko Švaco

Datum predaje rada:

1. rok: 22. i 23. 2. 2024.
2. rok (izvanredni): 11. 7. 2024.
3. rok: 19. i 20. 9. 2024.

Predviđeni datumi obrane:

1. rok: 26. 2. – 1. 3. 2024.
2. rok (izvanredni): 15. 7. 2024.
3. rok: 23. 9. – 27. 9. 2024.

Predsjednik Povjerenstva:

Prof. dr. sc. Damir Godec

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA.....	V
POPIS OZNAKA	VI
SAŽETAK.....	VII
SUMMARY	VIII
1. UVOD.....	1
2. MOBILNI ROBOTI	2
2.1. Osnovni podsustavi mobilnih robota	4
2.2. Vrste kotača i konfiguracija mobilnih robota.....	5
2.3. Kinematski model diferencijalnog mobilnog robota.....	7
3. MOBILNI ROBOT ASTRO.....	10
3.1. Arhitektura mobilnog robota Astro	11
3.1.1. Upravljačka računala.....	11
3.1.2. Senzorski sustav robota.....	13
3.1.3. Mehaničke komponente robota	15
3.1.4. Napajanje robota	17
4. ROBOTSKI OPERATIVNI SUSTAV(ROS2)	19
4.1. Glavne značajke ROS-a	20
5. PRAĆENJE STVARNE PUTANJE MOBILNOG ROBOTA POMOĆU OPTITRACK SUSTAVA.....	22
5.1. OptiTrack	22
5.1.1. Motive	23
5.1.2. Povezivanje ROS2 sustava s OptiTrackom.....	24
6. KALIBRACIJA MOBILNOG ROBOTA	25
6.1. Svojstva <i>dead-reckoning</i> grešaka.....	25
6.2. Sustavne greške	27
6.3. Nesustavne greške	28
6.4. Kalibracijske metode.....	29
6.5. UMBmark metoda.....	29
6.5.1. Mjerenje sustavnih grešaka	29
6.5.2. Analiza sustavnih grešaka	32
6.5.2.1. Greška tipa A	32
6.5.2.2. Greška tipa B.....	33
6.5.2.3. Matematičke pretpostavke	34
6.5.3. Kompenzacija sustavnih grešaka mobilnog robota.....	38
6.6. Proširena UMBmark metoda.....	39
6.7. Pretpostavke i matematički izrazi za proširenu <i>UMBmark</i> metodu.....	39
7. PRIKAZ REZULTATA MJERENJA PRIJE KALIBRACIJE	42
7.1. Korekcijski faktori prema <i>UMBmark</i> metodi	47
7.2. Korekcijski parametri prema proširenoj <i>UMBmark</i> metodi.....	48

8. Validacija kalibracijskih metoda	49
8.1. Rezultati <i>UMBmark</i> metode.....	49
8.2. Rezultati proširene <i>UMBmark</i> metoda.....	54
8.3. Validacija metodi na ravnoj putanji	60
8.4. Validacija metodi na kružnoj putanji u smjeru kazaljke na satu.....	61
8.5. Validacija metodi na kružnoj putanji u suprotnom smjeru kazaljke na satu.....	62
9. ZAKLJUČAK.....	63
LITERATURA.....	64
PRILOZI.....	65

POPIS SLIKA

Slika 1. Industrijski robot	2
Slika 2. Mobilni robot	2
Slika 3. Primjena mobilnih robota [1][2]	3
Slika 4. Autonomni mobilni roboti za skladišta [3]	4
Slika 5. Konstrukcija kotača: a) fiksni kotač, b) kutni rotirajući kotač, c) rotirajući kotač, d) sferni kotač i e) švedski kotač [4]	5
Slika 6. Mobilni robot s diferencijalnim pogonom u X-Y ravnini[6]	7
Slika 7. CAD model mobilnog robota ASTRO	10
Slika 8. Mobilni robot ASTRO	10
Slika 9. Arhitektura mobilnog robota ASTRO	11
Slika 10. Mikrokontroler <i>Teensy 4.0</i> [7]	11
Slika 11. Mikrokontroler <i>Nvidia Jetson Nano Developer Kit-a</i> [8]	13
Slika 12. Kamera <i>Intel® RealSense™ Depth Camera D435</i> [9]	13
Slika 13. Lidarski senzor <i>Slamtec RPLIDAR A1</i> [10]	14
Slika 14. IMU-a <i>Adafruit Precision NXP 9-DOF</i> senzor [11]	15
Slika 15. Elektromotor modela <i>GB37Y3530-12V-251R</i> [12]	16
Slika 16. Driver <i>Polulu TB67H420FTG</i> [13]	17
Slika 17. Regulator napona <i>PQAE50-T</i> [14]	17
Slika 18. Regulator napona <i>PDQE20-Q24-S5-D</i> [15]	18
Slika 19. <i>Makita</i> litijum-ionska baterija [16]	18
Slika 20. Kompanije i fakulteti koji koriste ROS2	21
Slika 21. Prikaz pet od osam kamera unutar laboratorija CRTA-e	22
Slika 22. OptiTrack PrimeX13 kamera [18]	23
Slika 23. Motive programsko sučelje	23
Slika 24. Povezivanje OptiTrack s ROS2-om	24
Slika 25. Prikaz podataka koje sadrži tip poruke <i>geometry_msgs/PoseStamped</i>	24
Slika 26. Mobilni robot s osnovnim segmentima: drive motor (<i>hrv. pogonski motor</i>), incremental encoder (<i>hrv. inkrementalni enkoder</i>), bumper (<i>hrv. branik</i>), castor (<i>hrv. pomoćni kotačić</i>), drive wheels (<i>hrv. kotači</i>), centerpoint C (<i>hrv. središnja točka C</i>) [20]	25
Slika 27. Primjetno poništavanje sustavnih grešaka E_b i E_d prilikom kretnje u smjeru kazaljke na satu [20]	30
Slika 28. Primjetno zbrajanje sustavnih grešaka E_b i E_d prilikom kretnje u suprotnom smjeru kazaljke na satu [20]	30
Slika 29. Uobičajeni rezultati izvođenja UMBmark eksperimenta (kvadratne kretnje u oba smjera, CW i CCW) s nekalibriranim robotom [20]	31
Slika 30. Greška tipa A u smjeru kazaljke na satu i suprotno smjeru kazaljke na satu [20]	33
Slika 31. Greška tipa B u smjeru kazaljke na satu i suprotno smjeru kazaljke na satu [20]	34
Slika 32. Geometrijski odnosi za izračun polumjera skretanja R [20]	37
Slika 33. Prikaz greške završne pozicije i orijentacije robota s obzirom na tip greške A (a) u smjeru kazaljke na satu i tip greške B (b) smjer obrnuto kazaljke na satu [21]	39
Slika 34. Pogreške orijentacije uzrokovane tipovima grešaka A i B[21]	41
Slika 35. Mjerenja prije kalibracije u smjeru kazaljke na satu redom 1., 2., 3.	43
Slika 36. Mjerenja prije kalibracije u smjeru kazaljke na satu redom 4., 5.	44
Slika 37. Mjerenja prije kalibracije u suprotnom smjeru kazaljke na satu redom 1., 2.	45
Slika 38. Mjerenja prije kalibracije u smjeru kazaljke na satu redom 3., 4., 5.	46
Slika 39. Usporedba grešaka nekalibriranog robota pri kretnji u smjeru kazaljke na satu (CW) i suprotno kazaljke na satu (CCW)	47

Slika 40. 1. mjerenje nakon implementacije kalibracije <i>UMBmark metode</i> u smjeru kazaljke na satu.....	49
Slika 41. Mjerenja nakon implementacije kalibracije <i>UMBmark metode</i> u smjeru kazaljke na satu redom 2., 3., 4.	50
Slika 42. 5. mjerenje nakon implementacije kalibracije <i>UMBmark metode</i> u smjeru kazaljke na satu.....	51
Slika 43. Mjerenja nakon kalibracije <i>UMBmark metodom</i> u suprotnom smjeru kazaljke na satu redom 1., 2., 3.	52
Slika 44. Mjerenja nakon kalibracije <i>UMBmark metodom</i> u suprotnom smjeru kazaljke na satu redom 4., 5.	53
Slika 45. 1. mjerenje nakon implementacije kalibracije proširene <i>UMBmark metode</i> u smjeru kazaljke na satu.....	54
Slika 46. Mjerenja nakon implementacije kalibracije proširene <i>UMBmark metode</i> u smjeru kazaljke na satu redom 2., 3., 4.	55
Slika 47. Mjerenja nakon kalibracije proširenom <i>UMBmark metodom</i> u smjeru kazaljke na satu redom.....	56
Slika 48. Mjerenja nakon kalibracije proširenom <i>UMBmark metodom</i> u suprotnom smjeru kazaljke na satu redom	58
Slika 49. Usporedba greške završne pozicije robota prije kalibracije i nakon primjene dvije metode	59
Slika 50. Mjerenje nekalibriranog robota pri ravnoj kretnji.....	60
Slika 51. Mjerenje kalibriranog robota <i>UMBmark metodom</i> pri ravnoj kretnji	60
Slika 52. Mjerenje kalibriranog robota proširenom <i>UMBmark metodom</i> pri ravnoj kretnji	60
Slika 53. Mjerenje nekalibriranog robota pri kružnoj kretnji u smjeru kazaljke na satu	61
Slika 54. Mjerenje kalibriranog robota <i>UMBmark metodom</i> pri kružnoj kretnji u smjeru kazaljke na satu	61
Slika 55. Mjerenje kalibriranog robota proširenom <i>UMBmark metodom</i> pri kružnoj kretnji u smjeru kazaljke na satu.....	61
Slika 56. Mjerenje nekalibriranog robota pri kružnoj kretnji u suprotnom smjeru kazaljke na satu	62
Slika 57. Mjerenje kalibriranog robota <i>UMBmark metodom</i> pri kružnoj kretnji u suprotnom smjeru kazaljke na satu.....	62
Slika 58. Mjerenje kalibriranog robota proširenom <i>UMBmark metodom</i> pri kružnoj kretnji u suprotnom smjeru kazaljke na satu.....	62

POPIS TABLICA

Tablica 1. Prikaz mogućih konfiguracija mobilnih robota[5]	6
Tablica 2. Specifikacije upravljačke jedinice <i>Teensy 4.0</i> [7]	12
Tablica 3. Specifikacije upravljačkog računala <i>NVIDIA Jetson Nano Development kit</i> [8]	12
Tablica 4. Specifikacije <i>Intel® RealSense™ Depth Camera D435</i> kamere [9]	13
Tablica 5. Specifikacije <i>Slamtec RPLIDAR A1</i> senzora [10]	14
Tablica 6. Specifikacije <i>Adafruit Precision NXP 9-DOF</i> senzora [11]	15
Tablica 7. Parametri kotača i karakteristike mobilnog robota.....	15
Tablica 8. Tehničke specifikacije elektromotora modela <i>GB37Y3530-12V-251R</i> [12].....	16
Tablica 9. Kvantifikacija trošila i njihova potrošnja	18
Tablica 10. Tehničke specifikacije <i>OptiTrack Prime^x 13</i> kamere [18]	23
Tablica 11. Podaci o greškama nekalibriranog robota u smjeru kazaljke na satu za pet mjerenja	42
Tablica 12. Podaci o greškama nekalibriranog robota u smjeru kazaljke na satu.....	45
Tablica 13. Podaci o greškama kalibriranog robota UMBmark metodom u smjeru kazaljke na satu	49
Tablica 14. Podaci o greškama kalibriranog robota UMBmark metodom u suprotnom smjeru kazaljke na satu.....	51
Tablica 15. Podaci o greškama kalibriranog robota proširenom UMBmark metodom u smjeru kazaljke na satu.....	54
Tablica 16. Podaci o greškama kalibriranog robota proširenom UMBmark metodom u suprotnom smjeru kazaljke na satu.....	56

POPIS OZNAKA

Oznaka	Mjerna jedinica	Opis oznake
x	m	Pozicija mobilnog robota u smjeru osi X
y	m	Pozicija mobilnog robota u smjeru osi Y
θ	rad	Orijentacija mobilnog robota u X-Y ravnini
ω_l	rad/s	Kutna brzina lijevog kotača
ω_r	rad/s	Kutna brzina desnog kotača
ω	rad/s	Kutna brzina mobilnog robota u X-Y ravnini
r	m	Polumjer kotača robota
v_l	m/s	Linearna brzina lijevog kotača
v_r	m/s	Linearna brzina desnog kotača
v	m/s	Linearna brzina mobilnog robota
\dot{x}	m/s	Brzina mobilnog robota u smjeru osi X
\dot{y}	m/s	Brzina mobilnog robota u smjeru osi Y
$\dot{\theta}$	rad/s	Rotacijska brzina mobilnog robota oko osi Z
c_m	rad·m/ okretaju	Faktor pretvorbe koji pretvara impulse enkodera u linearno pomicanje kotača
D	m	Promjer kotača
C_e	-	Prijenosni omjer redukcije zupčanika
n	imp/okr	Rezolucija enkodera
N_i	-	Inkrementalna promjena impulsa motora
ΔU_i	m	Inkrementalni linearni pomak kotača
$\Delta \theta_i$	rad	Inkrementalna promjena orijentacije robota
b	m	Razmak između kotača mobilnog robota
E_b	-	Korekcijski parametar udaljenosti među kotačima
E_d	-	Korekcijski parametar promjera kotača
$x_{c.g.}$	m	Težište klastera odstupanja u smjeru osi X
$y_{c.g.}$	m	Težište klastera odstupanja u smjeru osi Y
$r_{c.g.}$	m	Apsolutna udaljenost težišta klastera odstupanja od ishodišta
$E_{max,syst}$	m	Mjera točnosti sustavnih grešaka
L	m	Duljina stranice kvadratne putanje
α	rad	Kut odstupanja
β	rad	Kut odstupanja
R	m	Polumjer kružnice koju robot opisuje prilikom gibanja ravno

SAŽETAK

Završni rad obrađuje zadatak implementacije kinematske kalibracije mobilnog robota ASTRO koji je razvijen kao prototip niske razine u Regionalnom centru izvrsnosti za robotske tehnologije-CRTA. Mobilni robot se upravlja s pomoću robotskog operativnog sustava (*eng. Robot Operating System – ROS2*), u kojem se stvaraju čvorovi koji generiraju reference putanje u obliku kvadrata, kruga i ravne linije. Primjenom *OptiTrack* sustava kamera za praćenje pozicija u prostoru pratit će se i bilježiti stvarna putanja robota, slijedno tome usporedit će se stvarna putanja s referentnom putanjom. Odabirom jedne od predloženih kinematskih kalibracijskih metoda, u ovom slučaju *UMBmark* metoda i proširena *UMBmark* metoda. Nakon implementacije kalibracijske metode robotu se ponovno zadaju reference kao na početku te se s pomoću *OptiTrack* sustava prati stvarna putanja robota. U završnom koraku grafički je prikazana usporedba rezultat nakon kretanja robota prije i poslije kinematske kalibracije mobilnog robota.

Ključne riječi: kinematska kalibracija, mobilni robot, robotski operativni sustav (ROS2), *OptiTrack*, diferencijalna kinematika

SUMMARY

The final paper addresses the task of implementing kinematic calibration of the ASTRO mobile robot, which was developed as a low-level prototype at the Regional Center of Excellence for Robotic Technology (CRTA). The mobile robot is controlled using the Robot Operating System (ROS2), where nodes are created to generate reference paths in the form of squares, circles, and straight lines. By employing the OptiTrack camera tracking system to monitor and record the actual robot path in space, the actual path will be compared with the reference path. One of the proposed kinematic calibration methods will be selected, in this case, the *UMBmark* method and the extended *UMBmark* method. After implementing the calibration method, references are re-assigned to the robot as at the beginning, and the actual robot path is monitored using the OptiTrack system. In the final step, a graphical comparison is shown between the results before and after the kinematic calibration of the mobile robot.

Keywords: kinematic calibration, mobile robot, Robot Operating System (ROS2), OptiTrack, differential kinematics

1. UVOD

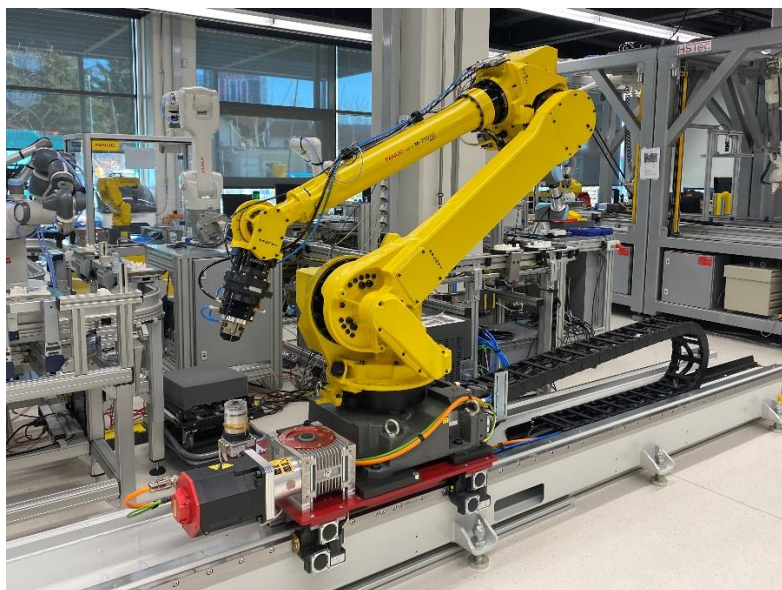
Robotika općenito, a posebno mobilna robotika u posljednjih deset godina doživljava snažan rast u svim područjima djelovanja kao što su proizvodnja, distribucija, vojna industrija, medicina, ispitivanje kvalitete, održavanje te mnogi drugi. Razlog tome je lakša i brža proizvodnja pojedinih dijelova, te svakodnevna nova znanja u područjima kao što su strojarstvo, elektrotehnika i računarstvo koja su objedinjena upravo u robotici. Treba naglasiti da takav snažan rast doprinosi potrebi za automatizacijom raznih poslova koji su monotoni ili opasni po zdravlje i život čovjeka, odnosno doprinosi većoj ponovljivost u poslovima koji zahtijevaju veliku preciznost.

Ovaj rad se fokusira na kinematsku kalibraciju mobilnog robota koja je iznimno važna za osiguravanje sigurne i točne kretnje mobilnog robota po prostoru. Da bi se to postiglo potrebno je osigurati što točniju estimaciju odometrije. Odometrija je postupak procjene položaja mobilnog robota u odnosu na početnu točku na temelju kretanja kotača, u osnovi integrirajući učinak brzine kotača. Budući da je senzor za praćenje položaja kotača dostupan na svim mobilnom robotima, estimacija odometrije je jednostavna i jeftina. Međutim postoji mogućnost u nakupljanju pogrešaka prilikom procjene, stoga treba obratiti veliku pozornost na sustavne i nesustavne pogreške. Iz tog razloga, u ovom radu obrađena je kalibracija mobilnog robota s obzirom na sustavne greške kao što su nejednoliki promjer kotača i dimenzijska netočnost između kotača. Za upravljanje mobilnog robota te implementaciju kalibracije koristi se ROS2, dok se za praćenje stvarne putanje koristiti vizijski sustav OptiTrack koji će biti objašnjen u nastavku. S obzirom na to da ROS2 nije neovisan operativni sustav, važno je odabrati osnovni operativni sustav na kojem će se instalirati. Odlučeno je koristiti Linux Ubuntu kao temeljnu platformu za upotrebu ROS2-a.

2. MOBILNI ROBOTI

Za razliku od industrijski robota koji se nalaze u stacioniranom položaju, mobilne robote ističe upravo ta ne stacioniranost odnosno mobilnost. Mobilnost im je omogućena s pomoću integracije pogonskih sustava (motora i aktuatora), sustava koji omogućuje dohvaćanje informacija iz radne okoline (senzori i kamere) te upravljačkog sustava (mikrokontrolera i računala) koji je zadužen za primanje informacija i slanje informacija prema pogonskom sustavu.

Razlika između mobilnih i industrijskih robota prikazana je na slikama 1. i 2.



Slika 1. Industrijski robot



Slika 2. Mobilni robot

U skupinu mobilnih robota osim klasičnih mobilnih robota koje možemo podijeliti prema načinu pogona na autonomne i teleoperacijske, tu spadaju i humanoidni roboti, roboti kućni pomagači, dronovi, autonomne podmornice i mnogi drugi prikazani za slici 3. Zaključujemo da je za takvu primjenu potrebna široka paleta znanja unutar područja mehanike, kinematike, dinamike, računalnog vida, elektrotehnike, programiranja i mnogih drugih.



Slika 3. Primjena mobilnih robota [1][2]

Svakako treba istaknuti činjenicu da se najznačajniji razvoj i primjena mobilne robotike događaju u industriji, gdje su potrebe za mobilnim robotima u konstantnom porastu. U postrojenjima i skladištima, mobilne platforme često se koriste za manipulaciju paketima, distribuciju proizvoda te kao podrška industrijskim robotima u rukovanju predmetima slika 4. Iako mobilni roboti u industriji koriste inteligenciju za percepciju i snalaženje u prostoru, valja napomenuti da su industrijska postrojenja često organizirana i uređena, što rezultira manjim zahtjevima za složenijom inteligencijom. Takva primjena ima snažan utjecaj na smanjenje potrebe za radom ljudi.



Slika 4. Autonomni mobilni roboti za skladišta [3]

2.1. Osnovni podsustavi mobilnih robota

Mobilni robot je kompleksan sklop koji se sastoji od nekoliko podsustava koji u slučaju ispravne međusobne interakcije i suradnje mogu omogućiti ispravnu i kvalitetnu kretnju prilikom izvođenja određenih zadataka. Osnovan dio cijelog mobilnog robota je konstrukcija robota na kojoj su smješteni podsustavi te se u sklopu konstrukcije ti podsustavi štite od vanjskih utjecaja.

Prvi sustav koji predstavlja osnovu za autonomnost mobilnog robota je upravljački sustav. Svrha tog sustava je održavanje konstantne komunikacije s ostalim sustavima te između njih međusobno. Glavni element ovog sustava je mikrokontroler, koji prikuplja podatke iz okoline, obrađuje ih i upravlja aktuatorima prema određenim referencama. Da bi se moglo upravljati senzorima, aktuatorima i kamerama uz glavni mikrokontroler nalaze se i drugi mikrokontroleri. Odnosno možemo podijeliti upravljački sustav na upravljanje visoke razine (nadređeno upravljačko računalo) u kojega spada glavni mikrokontroler u kojem se nalaze algoritmi za razne funkcije i upravljanje niske razine (podređeno upravljačko računalo) u koji spadaju ostali mikrokontroleri koji su zaduženi za senzore i aktuatora. Jezici u kojima se programiraju kodovi za upravljačke sustave su najčešće C++ ili Python.

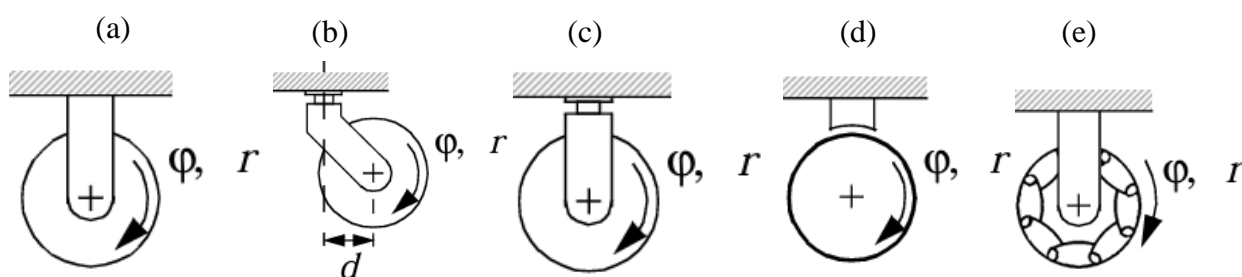
Sljedeći sustav koji treba istaknuti za pravilan rad je senzorski sustav. Zadužen je za prikupljanje informacija iz okoline i pretvaranje u referentni strujni ili naponski signal koji će mikrokontroleru

biti razumljiviji i moći će ga obraditi. Važno je odabrati kvalitetan senzorski sustav koji je bit pouzdan i koji će davati realne podatke te tako omogućiti percepciju radne okoline. Odabir senzora ovisi o potrebi za dobivanjem kvalitetnih podataka, a primjeri uključuju senzore udaljenosti, temperature, brzine, akceleracije, žiroskope, enkodere i druge

Treći ključni sustav koji se koristi u izradi robota naziva se aktuatorski sustav. Ovaj sustav odgovoran je za sva gibanja koja izvodi mobilni robot, bilo da se radi o kretanju kroz prostor ili manipulaciji predmetima u okolini. Najčešće instalirani aktuatori na mobilnim robotima su elektromotori istosmjerne struje, koji se mogu pojaviti s ili bez četkica. Ponekad se mogu pronaći i mobilni roboti s koračnim motorima, iako su oni manje uobičajeni. Elektromotori na mobilnim robotima često su opremljeni enkoderima čije je osnovna funkcija mjerenje kuta zakreta motora. Enkoderi omogućuju procjenu pozicije, brzine i akceleracije mobilnog robota. Detaljan opis arhitekture mobilnog robota ASTRO navedena je u nastavku.

2.2. Vrste kotača i konfiguracija mobilnih robota

Gibanje robota omogućava se tako da se prijenos gibanja s aktuatorskog sustava preko kotača prenosi na podlogu. Važno je istaknuti da s minimalno tri kotača robot dobiva stabilnost potrebnu za gibanje. Korištenje više od četiri pogonska kotača je moguće, ali nije poželjno zato što dolazi do povećanja kompleksnosti kinematskog modela mobilnog robota. Povećanjem kompleksnosti kinematskog modela dolazi do otežavanja sinteze između planera trajektorija gibanja i regulatora koji su zaduženi da upravljaju motorima. Na slici 5. prikazani su osnovni načini konstrukcije kotača.

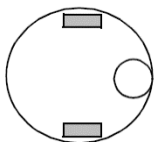
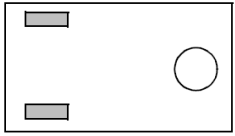
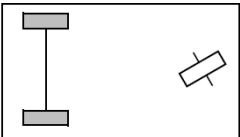
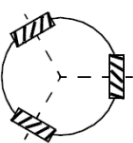
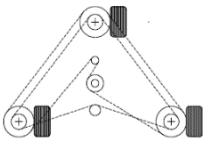
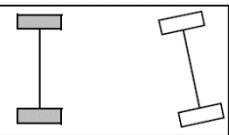
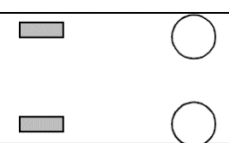
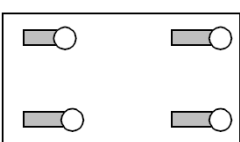
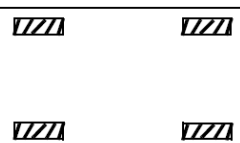
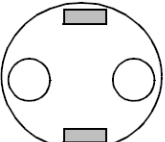


Slika 5. Konstrukcija kotača: a) fiksni kotač, b) kutni rotirajući kotač, c) rotirajući kotač, d) sferni kotač i e) švedski kotač [4]

Konfiguracija robota koja se najčešće koristi je s dva fiksna kotača koji su pogonjeni zasebno svaki svojim elektromotorom i jednim pasivnim rotirajućim kotačem.

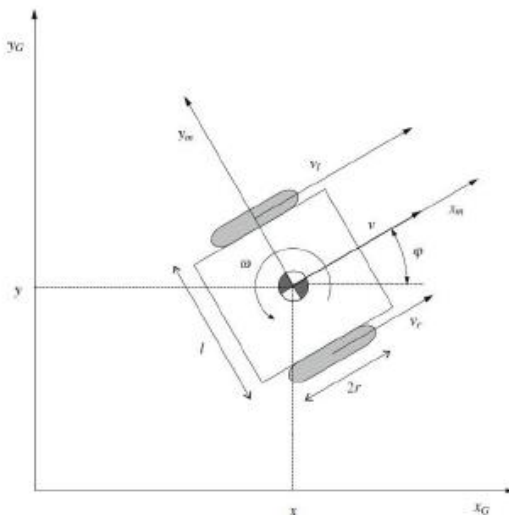
S obzirom na potrebe korištenja mobilnih robota postoje također ostale konfiguracije mobilnih robota koje su prikazane u tablici 1.

Tablica 1. Prikaz mogućih konfiguracija mobilnih robota[5]

Broj kotača	Prikaz konfiguracije	Opis
3		Diferencijalni pogon s dva fiksna centrirana pogonska kotača i trećim sfernim kotačem.
		Dva neovisna fiksna pogonska kotača odostraga/sprijeda i treći sferni kotač sprijeda/odostraga za stabilnost
		Dva povezana fiksna pogonska kotača odostraga i treći rotirajući kotač za upravljanje
		Tri pogonska švedska ili sferna kotača raspoređena u trokutu, omogućeno gibanje u svim smjerovima.
		Tri sinkronizirana fiksna pogonska kotača, orijentacijom se ne može upravljati
4		Dva povezana fiksna pogonska kotača odostraga i dva povezana rotirajuća kotača za upravljanje (Ackermannovo upravljanje)
		Dva fiksna pogonska kotača odostraga/sprijeda i dva sferna kotača sprijeda/odostraga
		Četiri pogonska kutno rotirajuća kotača
		Četiri švedska pogonska višesmjerna kotača
		Dva pogonska diferencijalna kotača s dodatna dva sferna kotača koji osiguravaju stabilnost

2.3. Kinematski model diferencijalnog mobilnog robota

Da bi opisali matematički ponašanje mobilnog robota u nekoj okolini koristi se kinematski model robota. Robot korišten u ovom radu pokreće se pogonom zasnovanom na diferencijalnom principu, takav pogon karakteriziraju dva fiksna kotača pričvršćena na aktuatoru gdje je svakom kotaču moguće zadati kutnu brzinu okretanja te četiri pasivno rotirajuća kotača na kuglama koji imaju funkciju održavanja horizontalnog položaja mobilnog robota. Slika 6. prikazuje diferencijalnog mobilnog robota u X-Y ravnini.



Slika 6. Mobilni robot s diferencijalnim pogonom u X-Y ravnini[6]

Uzimanjem u obzir određenih ograničenja i pretpostavki omogućujemo ispravno sprovođenje postavljanja i korištenja kinematskog modela:

- Gibanje mobilnog robota je planarno, tj. u horizontalnoj ravnini
- Mobilni robot se promatra kao kruto tijelo na kotačima
- Nema proklizavanja kotača, odnosno kontakt između mobilnog robota i podloge je idealan.

Sa slike 5. može se uočiti da je stacionirani globalni koordinatni sustav označen s x_G i y_G , dok su lokalne pomične koordinate mobilnog robota dane s x_m i y_m , gdje x_m predstavlja smjer kretanja mobilnog robota. Stanje mobilnog robota opisano je koordinatama x i y koje predstavljaju udaljenost lokalnog pomičnog koordinatnog sustava u odnosu na globalni koordinatni sustav te kutom zakreta θ koji predstavlja zakret lokalnog pomičnog koordinatnog sustava u odnosu na globalni koordinatni sustav, tj. možemo to shvatiti kao rotaciju oko osi Z.

Stanje mobilnog robota predstavlja vektor zapisan u izrazu (1).

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad (1)$$

Kao što je spomenuto, na robotu se nalaze dva aktuatora na koje su pričvršćeni kotači. Ako se kotači okreću u identičnom brzinom i u istom smjeru, robot će se kretati pravocrtno, nadalje u slučaju okretanja kotača u suprotnim smjerovima, robot izvršiti rotaciju oko svoje osi. Centar rotacije mobilnog robota može biti bilo gdje duž pravca koji prolazi kroz oba kotača, odnosno kada brzine kotača nisu međusobno jednake robot se neće okretati oko svoje osi već po nekoj kružnici. Utjecaj na linearnu brzinu mobilnog robota u smjeru osi x_m možemo prikazati s pomoću obodnih brzina lijevog i desnog kotača zapisano izrazom (2).

$$v = \frac{v_r + v_l}{2} \quad (2)$$

Gdje je:

- v_l – obodna brzina lijevog kotača
- v_r – obodna brzina desnog kotača
- v – linearna brzina mobilnog robota.

Izrazi (3) i (4) prikazuju međusoban odnos koji povezuje linearnu i obodnu brzinu lijevog i desnog kotača:

$$v_r = \omega_r r \quad (3)$$

$$v_l = \omega_l r \quad (4)$$

Gdje je:

- r – radijus kotača mobilnog robota

Izrazi (2) i (3) pružaju relativnu brzinu mobilnog robota u odnosu na lokalni koordinatni sustav, definiran osima x_m i y_m . Da bismo odredili brzinu robota u odnosu na stacionarni (globalni) koordinatni sustav, koristit ćemo matricu rotacije T , koja je definirana izrazom (5):

$$\mathbf{T} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Transformacija iz lokalnog i globalnog koordinatnog sustava, za translacijsku brzinu v dobivena je primjenom matrice transformacije i prikazana izrazom (6):

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v \cos\theta \\ v \sin\theta \\ 0 \end{bmatrix} \quad (5)$$

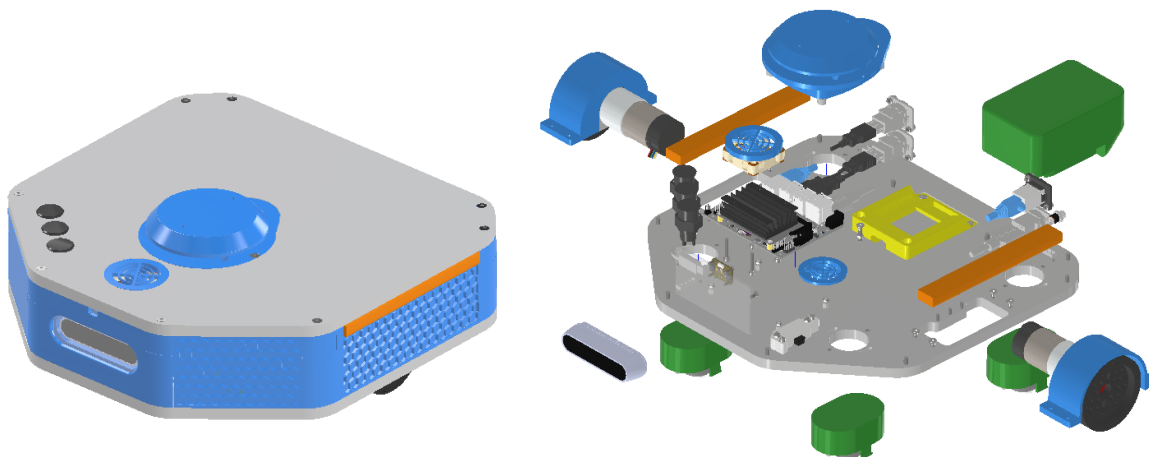
S obzirom na to da su rotacijske brzine u pomičnom lokalnom i stacioniranom globalnom koordinatnom sustavu jednake transformaciju rotacijske brzine nije potrebno provoditi. Krajnji izraz za translacijsku i rotacijsku brzinu u stacioniranom globalnom koordinatnom sustavu glasi:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos\theta \\ v \sin\theta \\ \omega \end{bmatrix} \quad (6)$$

Kinematski model mobilnog robota ukazuje na to da se radi o pod-upravljanom sustavu. Upravljanje mobilnim robotom ostvaruje se s pomoću dviju upravljačkih varijabli, odnosno kutnih brzina ω_x i ω_y , koje izravno utječu na tri varijable stanja: linearnu brzinu duž osi x , linearnu brzinu duž osi y te rotacijsku brzinu ω . Ovaj podatak dodatno ukazuje na zahtjevnost algoritama upravljanja brzinama, a time i na složenost algoritama za upravljanje pozicijom mobilnog robota.

3. MOBILNI ROBOT ASTRO

Mobilni robot ASTRO razvijen je u CRTA-i, robot ima diferencijalnu konfiguraciju robota koja je opisana u prethodnom poglavlju. Iz tog razloga da bi omogućili kretanje u radnoj okolini mobilni robot koristi dva zasebno upravljana istosmjerna motora s kotačima za prijenos gibanja na podlogu te dodatna četiri kotača koji imaju svrhu zadržavanja stabilnosti robota prilikom kretanja. Na slici 7. prikazan je CAD model te na slici 8. stvarni izgled korištenog mobilnog robota.



Slika 7. CAD model mobilnog robota ASTRO

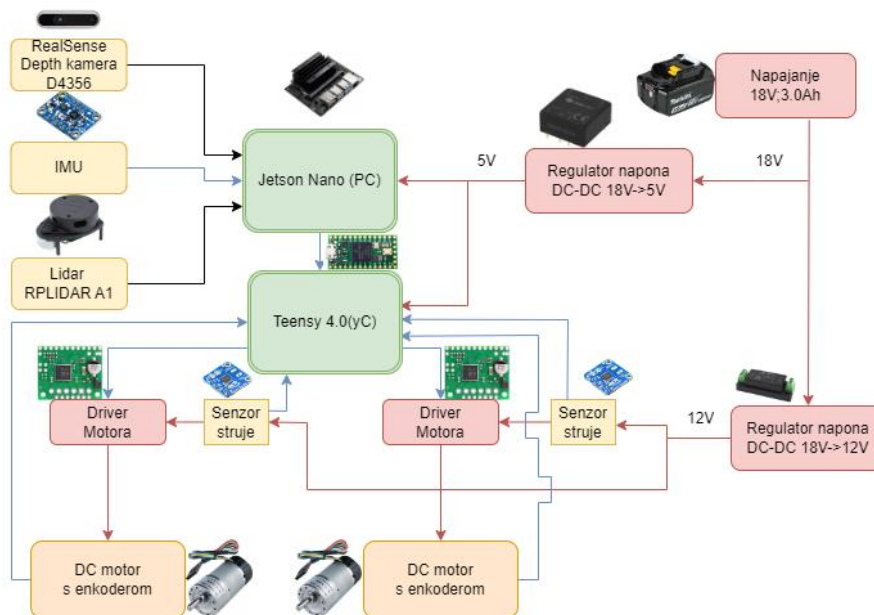


Slika 8. Mobilni robot ASTRO

Cilj razvoja mobilnog robota je radi primjene u edukativne svrhe, studenti će moći raditi na realnim problemima i te ispitati sve mogućnosti koje ovaj mobilni robot nudi. Ovaj robot je osnova i predstavlja izvrstan početka za savladavanje znanja o mobilni robotima te primjene ROS2-a kao operacijskog sustava za upravljanje robotom. U sljedećem poglavlju detaljno će se opisati arhitektura mobilnog robota.

3.1. Arhitektura mobilnog robota Astro

Na slici 9. prikazana je detaljna arhitektura mobilnog robota ASTRO. Crvenim strelicama prikazan je tok energije od izvora napajanja te sve do istosmjernih motora te upravljačkih računala. Plavim strelicama prikazan je tok signala, a crnim strelicama prikazan je tok signala sa senzora vanjskih stanja prema nadređenom upravljačkom računalu.



Slika 9. Arhitektura mobilnog robota ASTRO

3.1.1. Upravljačka računala

Za upravljanje robotom na najnižoj razini koristi se podređeno upravljačko računalo *Teensy 4.0*. Performanse samog kontrolera su daleko iznad sličnih dostupnih kontrolera zato što sadrži snažan mikroprocesor, te mnogo mogućnosti koje će biti navedene u tablici. Kontroler podržava razne komunikacijske protokole kao što su I2C, I2S, CAN, SDIO, SPI itd. Na slici 10. prikazan je primijenjeni mikrokontroler.



Slika 10. Mikrokontroler *Teensy 4.0*[7]

Komunikacija između podređenog i nadređenog upravljačkog računala ostvaruje se preko USB *port-a* i serijske komunikacije.

U nastavku je priložena tablica specifikacija mikrokontrolera *Teensy 4.0*.

Tablica 2. Specifikacije upravljačke jedinice *Teensy 4.0* [7]

Mikroprocesor:	ARM Cortex-M7 at 600 MHz
RAM:	1024kB
FLASH-ROM:	2024kB
Broj kanala za prijenos podataka između kontrolera i memorije	32 DMA kanala
Broj PWN izlaza	31
Broj digitalnih ulaza/izlaza	40
Broj analognih ulaza	14

Dodatna prednost je što se upravljački programi mogu razvijati s pomoću Arduino IDE sučelja, što omogućuje korisnicima da koriste poznato i korisniku lako shvatljivo okruženje za razvoj softvera. Također dolazi s već stvorenim knjižnicama koje značajno olakšavaju rad s raznim hardverskim komponentama, čineći ga idealnim izborom za brzi razvoj prototipa i implementaciju projekata.

Uz mikrokontroler, mobilni robot je opremljen nadređenim upravljački računalom *Nvidia Jetson Nano*. Upravo to računalo sadrži *Linux* operativni sustav na kojemu je instaliran robotski operativni sustav (ROS2) te omogućuje implementiranje složenih algoritama na mobilnom robotu kao što su mapiranje, lokalizacija i navigacija u prostoru. Ističe ga mogućnost bežičnog povezivanja i *bluetooth* komunikacija. U tablici 3. navedeni su sve specifikacije odabranog upravljačkog računala.

Tablica 3. Specifikacije upravljačkog računala *NVIDIA Jetson Nano Development kit* [8]

Procesor	128-core Maxwell + Quad-core ARM A57 @ 1.43 GHz
RAM memorija	4 GB 64-bit LPDDR4 25.6 GB/s
Memorija	micro SD
Video enkodiranje	4K @ 30 4x 1080p @ 30 9x 720p @ 30 (H.264/H.265)
Video dekodiranje	4K @ 60 2x 4K @ 30 8x 1080p @ 30 18x 720p @ 30 (H.264/H.265)
Kamera	2x MIPI CSI-2 DPHY lanes
Ethernet komunikacija	Gigabit Ethernet, M.2 Key E
Zaslon	HDMI i DisplayPort
USB	4x USB 3.0, USB 2.0 Micro-B
Komunikacijski protokoli	GPIO, I2C, I2S, SPI, UART
Napajanje	5V



Slika 11. Mikrokontroler Nvidia Jetson Nano Developer Kit-a [8]

3.1.2. Senzorski sustav robota

ASTRO na sebi ima dubinsku kameru *Intel® RealSense™ Depth Camera D435* koja koristi stereo viziju za izračunavanje dubine. Sastoji se od para dubinskih senzora, RGB senzora i infracrvenog projektora. Idealna je korištenje u projektima kojima se želi dodati mogućnost percepcije dubine. Kamera dolazi s moćnim vizualnim procesorom koji koristi tehnologiju procesa od 28 nanometara (nm) i podržava do 5 MIPI Camera Serial Interface 2 trake za izračunavanje dubinskih slika u stvarnom vremenu. Na slici 12. prikazan je izgled kamera.



Slika 12. Kamera Intel® RealSense™ Depth Camera D435 [9]

U nastavku je dana tablica 4. koja sadrži dodatne specifikacije o kameri.

Tablica 4. Specifikacije Intel® RealSense™ Depth Camera D435 kamere [9]

Idealni raspon:	0,3 m do 3 m
Minimalna udaljenost dubine (Min-Z) na maksimalnoj razlučivosti:	Oko 28 cm
Točnost dubine:	Manje od 2% na 2 m
Rezolucija izlaznog toka dubine:	Do 1280 × 720
Stopa okvira izlaznog toka dubine:	Do 90 fps
Rezolucija okvira RGB slike:	1920 × 1080
Stopa okvira RGB slike:	30 fps
Procesor za obradu vizije:	Intel RealSense Vision Processor D4
Dimenzije kamere (Duljina × Dubina × Visina):	90 mm × 25 mm × 25 mm
Komunikacija:	USB-C 3.1

Da bi se zadovoljili zahtjevi za mapiranjem prostora i kasnijom lokalizacijom, na robota je dodan lidar senzor *Slamtec RPLIDAR A1*. Lidar je vrsta senzora koji koristi infracrveni laserski snop za precizno i brzo mjerenje udaljenosti. Laserski snop je montiran na elektromotor koji rotira, skenirajući okolinu u svom vidnom polju. Senzor se nalazi približno u središnjem dijelu robota te ima zakret od 360°. Na slici 13. prikazan je senzor.



Slika 13. Lidarski senzor *Slamtec RPLIDAR A1* [10]

U tablici 5. nalaze se sve važne specifikacije laserskog skenera.

Tablica 5. Specifikacije *Slamtec RPLIDAR A1* senzora [10]

Kut skeniranja	0°-360°
Mogućnost mjerenja	0,15m-12m
Kutna rezolucija	1°
Frekvencija uzorkovanja	8000Hz
Frekvencija skeniranja	5,5Hz
Valna duljina lasera	785nm
Komunikacija	UART/USB
Napajanje	5V
Masa	170g

Posljednji senzor instaliran na robota je IMU (*eng. inertial measurement unit*) *Adafruit Precision NXP 9-DOF*. IMU se sastoji od dva senzora od kojih *FXOS8700* predstavlja 3-osni akcelerometar i magnetometar te *FXAS21002* koji predstavlja 3-osni žiroskop. Na slici 14. je prikazan senzor, te su u tablici 6. navedene sve važne specifikacije.



Slika 14. IMU-a Adafruit Precision NXP 9-DOF senzor [11]

Tablica 6. Specifikacije Adafruit Precision NXP 9-DOF senzora [11]

Napajanje	2-3,6V
Apsolutna orijentacija	Eulerov vektor ili kvaternioni
Kutna brzina prilagodljiva	$\pm 250/500/1000/2000^\circ/s$ u sve 3 osi
Akceleracija prilagodljiva	$\pm 2\text{ g}/\pm 4\text{ g}/\pm 8\text{ g}$ u sve 3 osi
Magnetsko polje	$\pm 1200\ \mu\text{T}$ u sve 3 osi
Izlazna frekvencija podataka	do 800Hz

3.1.3. Mehaničke komponente robota

Kao što je već ranije spomenuto mobilni robot za kretnju u prostoru koristi dva zasebno upravljana elektromotora s povratnom vezom. Globalne karakteristike mobilnog robota navedeni su u tablici 7.

Tablica 7. Parametri kotača i karakteristike mobilnog robota

Broj pogonskih kotača	2
Vrsta kotača	Polimerni kotač s gumenom oblogom
Promjer kotača [mm]	72
Širina kotača [mm]	32
Razmak kotača [mm]	290
Masa	4,7kg
Radijus skretanja [mm]	0
Radijus rotacije [mm]	425
Gabaritne mjere robota [mm]	325x337x120
Maksimalna translacijska brzina [m/s]	0,75
Maksimalna rotacijska brzina [rad/s]	2,0

Uz karakteristike mobilnog robota ASTRO, priložena je tablica 8. u kojoj su navedene sve važne specifikacije elektromotora.

Tablica 8. Tehničke specifikacije elektromotora modela GB37Y3530-12V-251R [12]

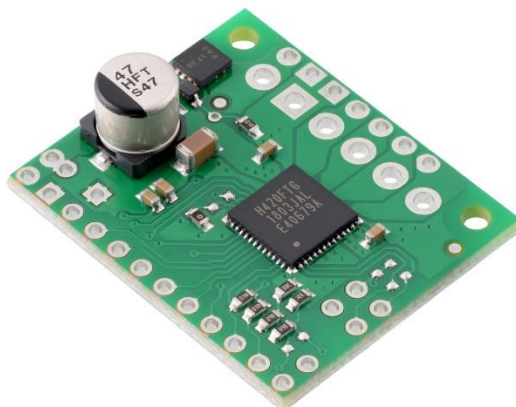
Radni napon	12V
Dimenzije [mm]	ϕ 37 x 86
Prijenosni omjer	43,8:1
Maksimalni potezni moment	1,76 Nm
Maksimalna struja	7A
Struja bez tereta	350mA
Minimalni napon rotacije bez tereta	1V
Brzina vrtnje bez tereta	251 min ⁻¹
Masa motora	205g
Tip enkodera	Magnetski
Rezolucija enkodera	16 impulsa po okretaju
Napon enkodera	5V



Slika 15. Elektromotor modela GB37Y3530-12V-251R [12]

Da bi se istosmjernim motorom moglo upravljati potreban je tranzistorski pretvarač, poznat i kao *driver*, upravo taj sklop ima ključnu ulogu u isporuci energije prema motoru na temelju nisko-naponskog signala s upravljačkog računala. Obično se nisko-naponski upravljački signali šalju korištenjem konvencionalnih komunikacijskih protokola, ali u ovom slučaju komunikacija se ostvaruje putem pulsno-širinske modulacije (PWM signala). Glavna dio drivera je tranzistorski H-

most koji, iz istosmjernog izvora napona, ovisno o kombinaciji upaljenih tranzistora, isporučuje željeni napon na kontakte motora. Na slici 16. nalazi se prikaz opisanog tranzistorskog pretvarača.



Slika 16. Driver Pololu TB67H420FTG [13]

Ovaj odabrani driver može se primijeniti za upravljanje istosmjernim kolektorskim motorima koji imaju napajanje u rasponu od 10 V do 47 V. Također, ovaj driver je dvokanalni, što znači da sadrži dva nezavisna H-mosta. Režim rada i smjer napona definiraju se putem dva odvojena signala na kontaktima pretvarača.

3.1.4. Napajanje robota

Kao izvor napajanja koristi se litij-ionska baterija proizvođača *Makita* napona 18 V te kapaciteta baterije 3 Ah. Baterija je u prosjeku dovoljna za nekoliko sati korištenja robota. Kako bi ostvarili radni napon od 12 V kojeg koriste motori te radni napon od 5 V kojeg koriste podređeno i nadređeno upravljačko računalo potrebno je prilagoditi izlazni napon baterije s pomoću dva regulatora napona. Regulator napon koji daje na izlazu 12 V je *PQAE50-T* proizvođača *CUI-INC* te ga ističe visoka efikasnost od 92 %, regulator napona koji daje na izlazu 5 V je *PDQE20-Q24-S5-D* također proizvođača *CUI-INC* njegova efikasnost je 90 %. Na slikama 17. i 18. nalaze se navedeni regulatori.



Slika 17. Regulator napona PQAE50-T [14]



Slika 18. Regulator napona PDQE20-Q24-S5-D [15]



Slika 19. Makita litijum-ionska baterija [16]

U tablici 9. navedena su sav trošila i njihova prosječna potrošnja te ukupna prosječna potrošnja samog robota.

Tablica 9. Kvantifikacija trošila i njihova potrošnja

Komponenta	Potrošnja
<i>NVIDIA Jetson Nano</i>	20W
<i>Teensy 4.0</i>	0,5W
<i>Intel® RealSense™ Depth Camera D435</i>	0,55W
<i>Slamtec RPLIDAR A1</i>	0,5W
IMU-a <i>Adafruit Precision NXP 9-DOF</i>	0,33W
2 x Elektromotor modela	24W
Prosječna ukupna potrošnja	45,88W

4. ROBOTSKI OPERATIVNI SUSTAV(ROS2)

U sklopu ovog rada korišten je ROS2 Humble, te će se sve što je u nastavku navedeno odnositi upravo na ROS2. To je verzija koja je izašla u svibnju 2022. godine te će kako je planirano nova ažuriranja za ovu verziju dolaziti do svibnja 2027. godine nakon čega neće više imati službenu podršku.

Robot Operating System (ROS) je operativni sustav razvijen s ciljem ubrzanja i standardizacije razvoja robotskih sustava te se trenutno smatra glavnim razvojnim sučeljem za robotske aplikacije. ROS2 je besplatan i dostupan softver koji iza sebe ima snažnu zajednicu istraživača i znanstvenika iz raznih područja, koji njegove mnogobrojne gotove biblioteke i alate mogu koristiti, razvijati i prenositi.

Početak ROS-a datira iz 2007. godine na Sveučilištu Stanford, na kojem je student Morgan Quigley započeo projekt robotskog sustava nazvanog Switchyard. Ovaj sustav bio je namijenjen podršci projektu STAIR (Stanford Artificial Intelligence Robot) i bio je dostupan svima. U početku, razvoj ROS-a bio je decentraliziran i odvijao se u različitim laboratorijima za robotiku. Iako je to na prvi pogled izgledalo nepregledno i neobično, jer nije postojao centralizirani pristup razvoju, kasnije se ispostavilo da je ta decentralizacija bila ključna za uspjeh ROS-a. Glavna značajka ROS-a bila je omogućavanje svakoj skupini istraživača da razvije vlastiti repozitorij, podijeli ga s drugim korisnicima putem servera, ali i zadrži potpunu kontrolu i vlasništvo nad svojim kodom. To je omogućilo suradnju i dijeljenje znanja i resursa među različitim istraživačkim timovima diljem svijeta, potičući inovacije i razvoj robotskih tehnologija.

Kroz godine, ROS je postao standardni alat u području robotske istraživačke zajednice. Njegova otvorena arhitektura omogućava istraživačima i inženjerima da brzo razvijaju, testiraju i implementiraju različite algoritme i sustave na raznolikim robotskim platformama. Ova fleksibilnost i prilagodljivost čine ROS ključnim alatom u različitim područjima kao što su autonomna vožnja, manipulacija objekata, istraživanje okoliša i mnogi drugi.

Jedna od ključnih značajki ROS-a je njegova sposobnost da podržava različite kombinacije hardvera. To znači da možete koristiti ROS na raznim robotskim platformama, sensorima i aktuatorima bez obzira na njihove specifičnosti. Odnosno što se tiče mogućnosti programiranja, ROS napravljen kao višejezičan, tj. neovisan o programskom jeziku, ali najčešće se koriste C++ i Python. Programiranje u C++ omogućava brzu izvedbu i često se koristi za razvoj algoritama koji zahtijevaju visoku računalnu snagu. Python, s druge strane, pruža jednostavnost i brzinu razvoja, što ga čini popularnim izborom za prototipiranje i brzo testiranje ideja. ROS omogućava

integraciju koda napisanog u oba jezika odnosno robotske aplikacije koje imaju čvorove napisane ili u C++ ili u Python programskom jeziku mogu neometano funkcionirati i komunicirati preko tema, servisa i akcija neovisno o jeziku. Ovaj interakcijski model je omogućen korištenjem IDL-a (Interface Definition Language), što je jezik za definiranje sučelja. IDL opisuje strukturu i format poruka koje se razmjenjuju između različitih modula u ROS-u. Koristeći kratke tekstualne datoteke, IDL precizno opisuje karakteristike i oblik poruka koje se prenose između različitih dijelova sustava.

Generatori koda zatim koriste ove IDL datoteke kao ulaz i automatski generiraju odgovarajući kod za pojedine programske jezike. To znači da se na temelju tih tekstualnih opisa poruka generira odgovarajući kod u željenom programskom jeziku. Ovaj proces omogućava lako prevođenje poruka iz ROS sustava u različite programske jezike, što omogućava fleksibilnost i prilagodljivost u razvoju robotskih aplikacija.

4.1. Glavne značajke ROS-a

Neke od glavnih značajki ROS-a koje su ključne za njegov ispravan rad i funkcionalnost, te su obavezne za implementaciju u svim ROS projektima su:

- Paketi (*engl. Packages*)
- Čvorovi (*engl. Nodes*)
- Teme (*engl. Topics*)
- Poruke (*engl. Messages*)
- Servisi (*engl. Services*).

Svaka od značajki će biti opisana u nekoliko rečenica dok se detaljne informacije o svakoj mogu pronaći na službenim stranicama ROS-a[17].

Paketi (*engl. Packages*) su najvažniji dio ROS-a te se predstavljaju kao osnovna građevna jedinica svakog projekta, objedinjujući čvorove, ROS knjižnice, *dataset*-ove i konfiguracijske datoteke prema načelu "Zlatokosa (*engl. goldilocks*)" - dovoljno veliki da sadrže potrebne funkcionalnosti, ali ne previše komplicirani kako bi se olakšala upotreba s drugim softverom, omogućujući prilagodbu za različite sustave i korisnike.

Čvorovi (*engl. Nodes*) predstavljaju srž ROS-a s obzirom na procese koji se izvode u njima te mogućnost međusobne komunikacije između pojedinih čvorova. Kao što je navedeno, čvorovi imaju mogućnost pisanja kodova u programskim jezicima Python i C++ s pomoću knjižica *rospy* i *roscpp*. Velika prednost korištenja čvorova je ta što prilikom ne funkcioniranja nekog od

segmenta robota ne moramo prepravljati cijeli programski kod već čvor koji je zadužen za funkcionalnost tog segmenta.

Teme (*engl. Topics*) služe za jednostavnu komunikaciju između čvorova. Ako čvor objavljuje poruku na neku temu onda je on objavljivač (*engl. Publisher*), odnosno ako se čvor pretplati na temu te prima poruke koje se nalaze na temi, on postaje pretplatnik (*engl. Subscriber*).

Poruke (*engl. Messages*) su jednostavne podatkovne strukture koje se sastoje od tipova podataka poput cijelih brojeva, logičkih vrijednosti i matrica. Definiiraju se u obliku datoteka koje određuju strukturu i format podataka za razmjenu između različitih procesa unutar ROS sustava,

Servisi (*engl. Services*) su dio paketa koji služe za interakciju između ROS čvorova koji funkcionira na principu zahtjev-odgovor odnosno pruža dvosmjernu komunikaciju nalik modelu *client-server*. Za razliku od tema na odgovarajući zahtjev samo jedan čvor može pružiti uslugu odgovora. Na slici 20. prikazani su samo neke u nizu kompanija i fakulteta koji koriste ROS2.



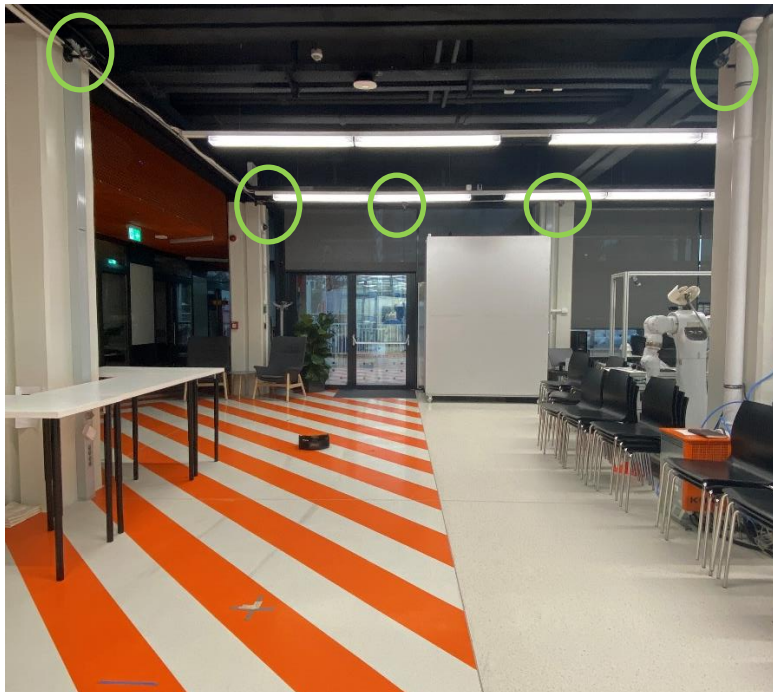
Slika 20. Kompanije i fakulteti koji koriste ROS2

5. PRAĆENJE STVARNE PUTANJE MOBILNOG ROBOTA POMOĆU OPTITRACK SUSTAVA

S obzirom na to da je cilj rada ispraviti kinematske pogreške prilikom gibanja, potrebno je prije primjene određenog algoritma pratiti odstupanja od referentnih putanja. Praćenje pokreta (*engl. Motion capture*), kao disciplina, obuhvaća snimanje elemenata u trodimenzionalnom prostoru te virtualnu rekonstrukciju tog prostora iz tih elemenata. OptiTrack se često koristi u istraživanjima kako bi precizno lokalizirao objekte bez internog mehanizma za lokalizaciju. Ova tehnologija ima raznolike primjene u praćenju stvarnih tijela, od sportskih i medicinskih istraživanja do interakcije s virtualnim okruženjima. Posebno zanimljiva primjena je u videoigrama, gdje se koristi za praćenje stvarnih kretanja kao osnove za animaciju likova.

5.1. OptiTrack

Laboratorij CRTA-e opremljen je sustavom od osam kamera s izvorom infracrvenog zračenja koje svojim ispravnom postavom imaju značajan utjecaj na kvalitetu prikupljenih podataka. Na slici 21. je prikazan i označen dio kamera koje se nalaze u sklopu laboratorija.



Slika 21. Prikaz pet od osam kamera unutar laboratorija CRTA-e

OptiTrack ima široku paletu kamera za korištenje, dok su u sklopu sustava unutar laboratorija koriste *OptiTrack Prime^x13* koje se ističu s velikom brzinom i praćenjem u prostorima srednje veličine te su preporučene od strane OptiTracka za primjenu u robotici. Specifikacije i prikaz kamere navedeni su u nastavku u tablici 9.

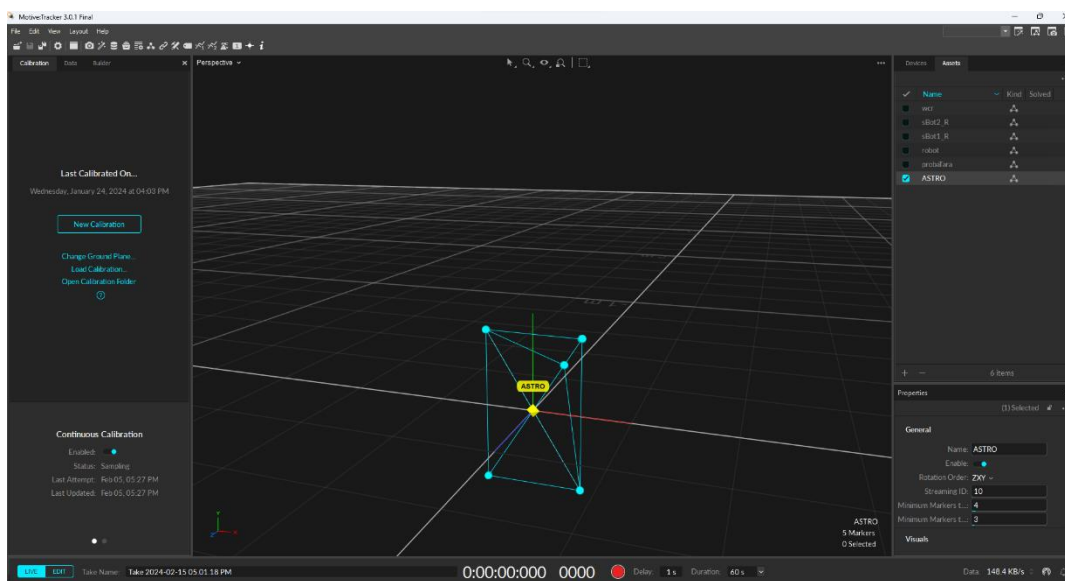
Tablica 10. Tehničke specifikacije OptiTrack Prime^x 13 kamere [18]

Rezolucija	piksela	1280x1024
Brzina osvježavanja	FPS	30-240
Točnost	mm	+/-0,20
Preciznost nakon kalibracije	mm	0,892
Domet (pasivni markeri)	m	16
Domet (aktivni markeri)	m	25

**Slika 22. OptiTrack PrimeX13 kamera [18]**

5.1.1. Motive

Motive softver ističe jednostavna postavka kroz suradnju s OptiTrack kamerama kako bi pratio i zabilježio pokrete malih retroreflektirajućih markera pričvršćenih u ovom slučaju na mobilnog robota. Izračunava preciznu poziciju markera u prostoru tijekom vremena snimanja.

**Slika 23. Motive programsko sučelje**

5.1.2. Povezivanje ROS2 sustava s OptiTrackom

OptiTrack snima poziciju markera koji se nalaza na robotu te pomoći njih stvara kruto tijelo (ridge body) u čije se središte postavlja koordinatni sustav prilikom svakog pokretanja robota. Da bi se podaci s OptiTracka-a mogli kasnije obraditi, OptiTrack mora biti povezan s ROS-om. Komunikacija između sustava OptiTrack i ROS2-a ostvaruje se korištenjem VRPN paketa. Detaljan opis VRPN-a može se pronaći na [19]. Nakon instalacije VRPN-a, stvara se paket u osobnom radnom prostoru nazvan "optitrack", u kojem se stvara .launch datoteka "client.launch.yaml" preko koje se pozivaju čvorovi koji omogućuju komunikaciju između OptiTrack-a i ROS2-a. Važno je poznavati IP adresu na kojoj je nalazi OptiTrack te VRPN server port. Na priloženoj slici prikazano je povezivanje ROS2-a i OptiTracka u ovom radu.

```
zvonimir@zvonimir-virtual-machine:~$ ros2 launch vrpn_mocap client.launch.yaml server:=192.168.0.103 port:=3883
[INFO] [launch]: All log files can be found below /home/zvonimir/.ros/log/2024-02-02-21-46-25-478182-zvonimir-virtual-machine-3582
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [client_node-1]: process started with pid [3604]
[client_node-1] check_vrpn_cookie(): VRPN Note: minor version number doesn't match: (prefer 'vrpn: ver. 07.35', got 'vrpn: ver. 07.33 0'). This is not normally a problem.
[client_node-1] [INFO] [1706906787.400770304] [vrpn_mocap.vrpn_mocap_client_node]: Created new tracker ASTRO
[client_node-1] [INFO] [1706906802.639924189] [vrpn_mocap.vrpn_mocap_client_node]: Creating sensor 0
```

Slika 24. Povezivanje OptiTrack s ROS2-om

Nakon povezivanja stvara se tema (topic) /astro_optitrack na koju se frekvencijom od 60 Hz objavljuju podaci tipa poruke *geometry_msgs/PoseStamped*. Slika 25. prikazuje podatke unutar teme /astro_optitrack

```
std_msgs/Header header
  builtin_interfaces/Time stamp
    int32 sec
    uint32 nanosec
  string frame_id
Pose pose
  Point position
    float64 x
    float64 y
    float64 z
  Quaternion orientation
    float64 x 0
    float64 y 0
    float64 z 0
    float64 w 1
```

Slika 25. Prikaz podataka koje sadrži tip poruke *geometry_msgs/PoseStamped*

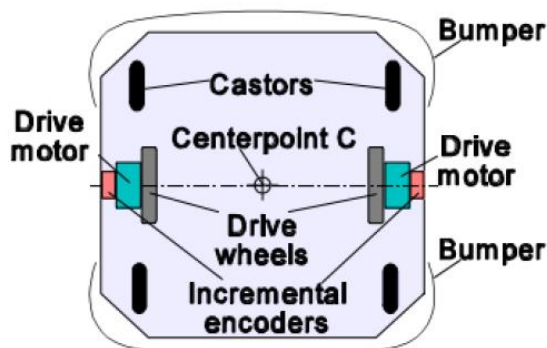
6. KALIBRACIJA MOBILNOG ROBOTA

Primarna primjena mobilnih robota obuhvaća sposobnost navigacije unutar prostora. Optimizacija takve navigacije zahtijeva preciznu estimaciju pozicije robota. U tu svrhu, primjenjuju se matematički modeli koji se oslanjaju na enkodere motora kako bi relativno estimirali položaj robota u odnosu na početnu koordinatnu referencu. S obzirom na ograničenosti matematičkih modela u opisu stvarnog sustava, nužno je provesti mjerenja specifičnih parametara na samom robotu kako bi se osigurala što veća preciznost estimiranih podataka. Kada imamo estimaciju pozicije robota na temelju podataka iz enkodera koja se još naziva odoometrija robota, cilj je poklopiti tu estimaciju sa stvarnom pozicijom robota u radnoj okolini. Upravo u tom dijelu dolazi do pojave odstupanja između odoometrije robota i stvarne pozicije robota zbog sustavnih i nesustavnih grešaka. Sustavne i nesustavne greške bit će objašnjene u nekom od sljedeći poglavlja.

6.1. Svojstva *dead-reckoning* grešaka

U ovom poglavlju detaljno je objašnjeno relativno pozicioniranje robota koje se temelji na mrtvom računanju (*engl. dead-reckoning*), odnosno praćenjem podataka iz enkodera te izračunom pomaka u odnosu na početni položaj robota. *Dead-reckoning* je jednostavno, relativno jeftino i primjenjivo u realnom vremenu.

Slika 26. prikazuje mobilnog robota s diferencijalnom konfiguracijom koja je objašnjena u jednom od prethodnih poglavlja. Na robotu se nalaze dva inkrementalna enkodera zasebno za svaki pogonski elektromotor koji broje okretaje motora, a samim time i okretaje kotača. Primjenom jednostavnih matematičkih izraza moguće je izračunati poziciju mobilnog robota s obzirom na poznatu startnu poziciju iz koje kreće.



Slika 26. Mobilni robot s osnovnim segmentima: drive motor (*hrv. pogonski motor*), incremental encoder (*hrv. inkrementalni enkoder*), bumper (*hrv. branik*), castor (*hrv. pomoćni kotačić*), drive wheels (*hrv. kotači*), centerpoint C (*hrv. središnja točka C*) [20]

Uvodi se pretpostavka da na nekom intervalu uzorkovanja pogonskih motora, desni i lijevi enkoder pokazuju porast inkremenata impulsa za desni kotač N_R te za lijevi kotač N_L . Uvodi se također nova oznaka prema izrazu:

$$c_m = \frac{\pi D_n}{nC_e} \quad (7)$$

gdje je:

- c_m -faktor pretvorbe koji pretvara impulse enkodera u linearno pomicanje kotača
- D_n -nazivni promjer kotača [mm]
- n -rezolucija enkodera [imp/okr]
- C_e -prijenosni omjer redukcijuskog zupčanika između motora (gdje je enkoder pričvršćen) i pogonskog kotača.

Sada kada imamo koeficijent koji povezuje karakteristike od enkodera prema kotaču (7) možemo izračunati inkrementalne pomak za desni kotač (ΔU_R) te lijevi kotač (ΔU_L) prema izrazu (8):

$$\Delta U_{R/L,i} = c_m N_{R/L,i} \quad (8)$$

odnosno primjenom inkrementalnih pomaka oba kotača iz izraza (8) možemo odrediti inkrementalni linearni pomak samog robota tj. središnje točke robota (slika 26. oznaka C) prema izrazu (9), te inkrementalnu promjenu orijentacije robota prema izrazu (10):

$$\Delta U_i = \frac{\Delta U_{R,i} + \Delta U_{L,i}}{2} \quad (9)$$

$$\Delta \theta_i = \frac{\Delta U_{R,i} - \Delta U_{L,i}}{b} \quad (10)$$

gdje je koeficijent b definiran kao idealna udaljenost između dodirnih točaka između kotača i podloge.

Nova relativna orijentacija robota θ_i može se odrediti prema izrazu:

$$\theta_i = \theta_{i-1} + \Delta \theta_i \quad (11)$$

odnosno relativna pozicija središnje točke robota (slika 26. oznaka C) izračunava se prema izrazu:

$$x_i = x_{i-1} + \Delta U_i \cos \theta_i \quad (12)$$

$$y_i = y_{i-1} + \Delta U_i \sin \theta_i \quad (13)$$

Kako se može vidjeti iz jednadžbi (9) do (13), *dead reckoning* se temelji na jednostavnim jednadžbama koje se mogu lako implementirati i koje koriste podatke iz jeftinih inkrementalnih enkodera montiranih na motor. Međutim, *dead reckoning* se temelji na pretpostavci da se broj okretaja kotača može pretvoriti u linearno pomicanje u odnosu na pod. Ova pretpostavka ima ograničenu valjanost. Jedan primjer nemogućnosti primjene ovih izraza bilo bi klizanje kotača odnosno ako bi jedan kotač klizao, recimo, na mrlji od ulja, tada bi povezani enkoder registrirao okretaje kotača, iako ti okretaji ne bi odgovarali linearnom pomicanju kotača.

Osim ovog ekstremnog slučaja potpunog klizanja, postoji nekoliko drugih, suptilnijih razloga za netočnosti u prevođenju očitavanja s enkodera u linearno kretanje. Ovakve vrste grešaka uklapaju se u jednu od dvije kategorije:

- sustavne greške
- nesustavne greške.

Koje će biti objašnjene u sljedećem poglavlju.

6.2. Sustavne greške

Sustavne greške su posebno zahtjevne jer se neprestano gomilaju. Odnosno u unutarnjim radnim uvjetima s glatkim površinama sustavne greške utječu više na greške estimacije odometrije od nesustavnih grešaka. Svakako treba napomenuti da prilikom hrapavih podloga, nesustavne greške ipak češće dominiraju. Sustavne greške uzrokovane su zbog:

- nejednakosti promjera kotača
- prosjeka oba promjera kotača koji se razlikuje od nominalnog promjera
- neusklađenosti kotača
- nesigurnosti u razmaku između kotača
- limitirane rezolucije enkodera
- limitiranog uzorkovanja enkodera.

Sustavne greške su najčešće uzrokovane nesavršenošću prilikom konstruiranja ili mehaničkom izradom mobilnog robota. U dugom razvoju mobilnih robota ustanovljeno je na mobilne robote s diferencijalnim pogonom najčešći izvori sustavnih grešaka nejednakost u promjeru kotača i nesigurnost u razmaku između kotača, iz tog razloga se upravo ta dva tipa grešaka obrađuju u ovome radu.

- 1) **Nejednaki promjer kotača.** Većina mobilnih robota koristi gumene kotače da bi se poboljšalo trenje prilikom kontakta s podlogom. Takvi kotači su izrazito teški za proizvesti da budu jednakog promjera. Nadalje, gumeni kotači su podložni kompresiji koja se javlja prilikom asimetričnog opterećenja robota. Bilo koji od ovih uzroka može znatno utjecati na greške odometrije.
- 2) **Nesigurnost u razmaku između kotača.** Razmak između kotača definiran je s udaljenošću točaka kontakta dva pogonska kotača te podloge. Razmak osi kotača mora biti poznata kako bi se izračunao broj impulsa diferencijalnog enkodera koji odgovaraju određenom kutu rotacije mobilnog robota. Nesigurnost u efektivnoj dužini osi kotača uzrokovana je činjenicom da gumeni kotači dodiruju podlogu ne u jednoj točki, već u određenoj površini. Rezultirajuća nesigurnost u vezi s efektivnom razmakom između osi može biti reda veličine 1 % kod nekih komercijalno dostupnih robota.

Olakšavajuća okolnost je ta da se na sustavne greške može utjecati. Iz tog razloga definirani su izrazi za greške:

- 1) Greška nejednakosti u promjeru kotača računa se prema izrazu:

$$E_d = \frac{D_R}{D_L} \quad (14)$$

gdje su E_d koji daje informaciju o omjeru veličina promjera lijevog i desnog kotača, pri čemu idealan omjer iznosi 1, D_R promjer desnog kotača i D_L promjer lijevog kotača.

- 2) Greška nesigurnosti razmaka između kotača računa se prema izrazu:

$$E_b = \frac{b_{stvarni}}{b_{nazivni}} \quad (15)$$

gdje su E_b koji daje informaciju o omjeru veličina udaljenosti između kotača, pri čemu idealni omjer iznosi 1, $b_{stvarni}$ stvarni razmak između kotača i $b_{nazivni}$ nominalni razmak između kotača.

6.3. Nesustavne greške

Nesustavne greške uzrokovane su uslijed nepredvidive okoline unutar koje se robot nalazi i obavlja svoju kretnju. Nesustavne greške uzrokovane su zbog:

- kretnje po neravnom podlozi

- kretanje preko nenadanih objekata koji se nalaze na podlozi
- proklizavanja kotača do čega dolazi zbog:
 - skliske podloge
 - prevelikog ubrzanja
 - naglog skretanja
 - vanjskih sila (interakcija s drugim objektima)
 - unutarnjih sila
 - nemogućnosti kontakta kotača robota i podloge.

Nesustavne greške stvaraju veliki problem prilikom pozicioniranja jer njihov utjecaj toliko nepredvidiv da se njihovo djelovanje teško može izbjeći.

6.4. Kalibracijske metode

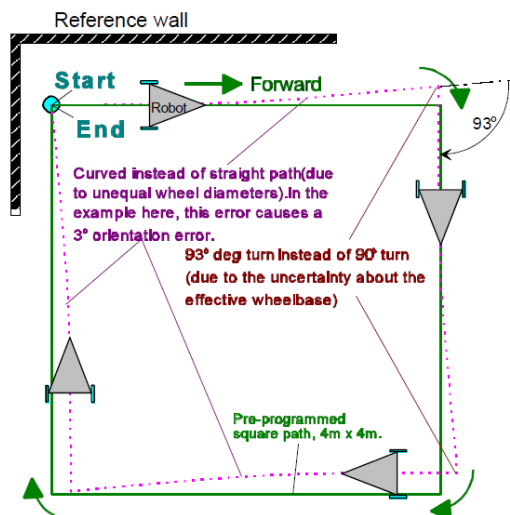
U ovom radu implementirane su dvije metode za ispravljanje sustavnih grešaka, prva metoda je najčešće korištena metoda za ispravljanje sustavnih grešaka kod robota s diferencijalnom kinematikom pod nazivom „*University of Michigan Benchmark (UMBmark)*“ [20] koju su predložili Borenstein, J., Feng, L. i suradnici 1994. godine, te druga metoda koja je proširenje prve metode te je objavljena u članku pod nazivom „*Accurate Calibration of Two Wheel Differential Mobile Robots by Using Experimental Heading Errors*“ [21] koju su predložili Changbae J., Woojin C. i *Member IEEE* 2012. godine. Važno je napomenuti da su obje metode isključivo za primjenu na robotima s diferencijalnom konfiguracijom. Slijedi detaljan opis obje metode.

6.5. UMBmark metoda

6.5.1. Mjerenje sustavnih grešaka

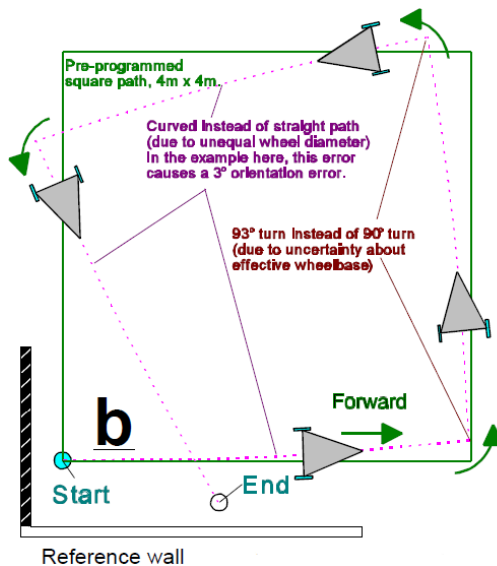
Djelovanje sustavnih grešaka može se ispraviti na način da ih je prvo potrebno mjeriti. U [20] je predstavljen normiran način mjerenja pod nazivom *The bi-directional square path experiment: „UMBMark“* (hrv. *eksperiment dvosmjerne kvadratne trajektorije: „UMBMark“*).

Eksperiment se provodi na način da se robotu zadaje putanja u obliku kvadratna dimenzija 4 m x 4 m, koja obuhvaća kretanje u smjeru kazaljke na satu i suprotno od smjera kazaljke na satu. Razlog provođenja kretanja u oba smjera prikazan je na slikama 27. i 28., iz slike 27. može se zaključiti da se djelovanje sustavnih grešaka u specifičnim situacijama može kompenzirati te iz tog razloga postoji mogućnost da se robot vrati u istu početnu poziciju iz koje je krenuo.



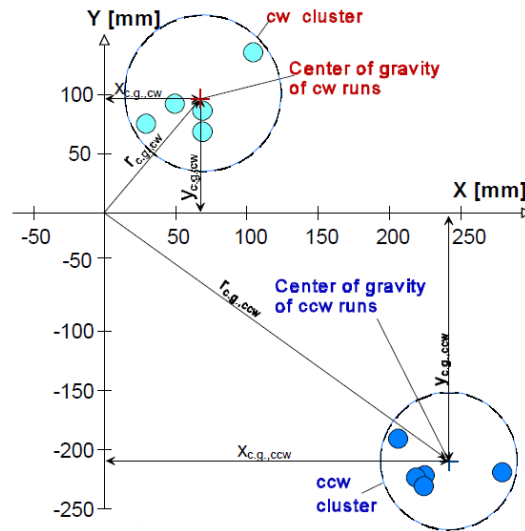
Slika 27. Primjetno poništavanje sustavnih grešaka E_b i E_d prilikom kretnje u smjeru kazaljke na satu [20]

Odnosno ako se kao što je na slici 28. prikazano robot pusti u suprotnom smjeru jasno je vidljiv utjecaj sustavnih grešaka. To se događa jer se dvije dominantne sustavne greške, koje se mogu nadoknaditi jedna drugom kada se izvodi kretnja u samo u jednom smjeru, tj. zbrajaju se i povećavaju ukupnu grešku kada se izvode u suprotnom smjeru.



Slika 28. Primjetno zbrajanje sustavnih grešaka E_b i E_d prilikom kretnje u suprotnom smjeru kazaljke na satu [20]

Da bi se prikazao tipičan primjer eksperimenta, korišteni su podaci iz [21]. U primjeru je robot programiran da prati kvadratnu putanju 4 m x 4 m, počevši iz pozicije (0,0). Krajnje pozicije za svaku od deset kretnji prikazane su na slici 29. od čega je pet kretnji u smjeru kazaljke na satu (*cw*) odnosno preostalih pet u suprotnom smjeru kazaljke na satu (*ccw*).



Slika 29. Uobičajeni rezultati izvođenja UMBmark eksperimenta (kvadratne kretnje u oba smjera, CW i CCW) s nekalibriranim robotom [20]

Sa slike 29. vidljivo je da se krajnje pozicije nakon kretnji u oba smjera mogu grupirati u dva klastera, od koji je svaki klaster za jedan smjer kretnje robota. Raspršenost unutar klastera pojavljuje se zbog utjecaja nesustavnih grešaka koje su ranije objašnjene u poglavlju 6.3. Također iz slike se može zaključiti da je utjecaj sustavnih grešaka znatno veći od nesustavnih (udaljenost klastera za oba smjera je znatno veća od udaljenosti krajnjih točaka unutar samog klastera).

S ciljem minimiziranja utjecaja nesustavnih grešaka, predlaže se određivanje centra težišta svakog klastera koji će biti reprezentativan za sustavne greške u smjeru kazaljke na satu i suprotnom smjeru kazaljke na satu:

$$x_{c.g.,cw/ccw} = \frac{1}{n} \sum_{i=1}^n \varepsilon x_{i,cw/ccw} \quad (16)$$

$$y_{c.g.,cw/ccw} = \frac{1}{n} \sum_{i=1}^n \varepsilon y_{i,cw/ccw} \quad (17)$$

gdje je n ukupan broj kretnji robota u svakom smjeru, a ostali parametri se računaju prema izrazu:

$$\varepsilon x = x_{abs} - x_{calc} \quad (18)$$

$$\varepsilon y = y_{abs} - y_{calc} \quad (19)$$

gdje su:

- ϵ_x, ϵ_y - greške u smjeru x i y osi zbog sustavnih i nesustavnih grešaka
- x_{abs}, y_{abs} -apsolutna pozicija robota u smjeru x i y osi
- x_{calc}, y_{calc} -pozicija robota u smjeru x i y osi dobivena iz odometrije.

Apsolutna udaljenost težišta oba klastera od ishodišta računa se prema izrazu:

$$r_{c.g.,cw/ccw} = \sqrt{(x_{c.g.,cw/ccw})^2 + (y_{c.g.,cw/ccw})^2} \quad (20)$$

Na kraju, definiramo veću vrijednost između $r_{c.g.,cw}$ i $r_{c.g.,ccw}$ kao mjeru točnost za sustavne pogreške.

$$E_{max,syst} = \max(r_{c.g.,cw}, r_{c.g.,ccw}) \quad (21)$$

Razlog zašto se ne koristi prosjek dvaju centara težišta $r_{c.g.,cw}$ i $r_{c.g.,ccw}$ je taj što se u praktičnim primjenama treba brinuti o najvećoj mogućoj greški. Treba napomenuti da u ovoj metodi konačna greška orijentacije nije eksplicitno uzeta u obzir u izrazu za $E_{max,syst}$. To je zato što su sve sustavne greške orijentacije implicitno obuhvaćene konačnim greškama položaja. Drugim riječima, budući da kvadratna staza ima stranice fiksne duljine, sustavne greške orijentacije direktno se prevode u pogreške položaja, te će to biti prikazano u daljnjoj analizi.

6.5.2. Analiza sustavnih grešaka

UMBMark eksperiment ističe se zato što proizlazi iz njegove sposobnosti postizanja značajnih poboljšanja u odometriji robota kroz relativno jednostavna mjerenja i analitičke jednadžbe. Prije formulacije matematičkog modela pogreške, nužno je precizno definirati dva tipa grešaka, tip A i tip B.

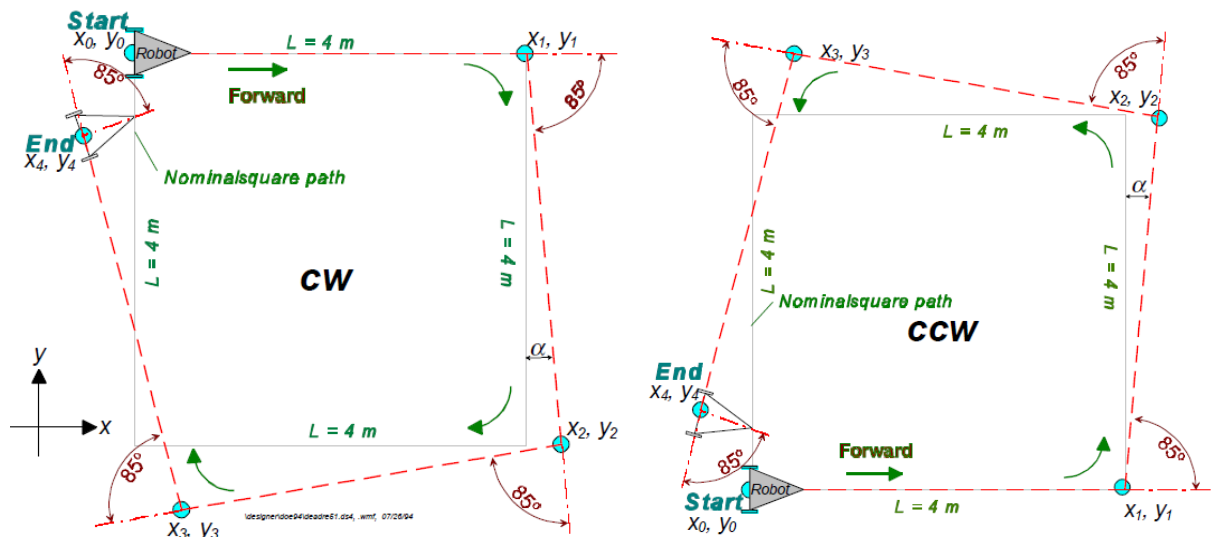
6.5.2.1. Greška tipa A

Greška tipa A je definirana kao greška orijentacije koja djeluje na ukupan iznos rotacije robota na način da ju povećava ili smanjuje prilikom izvođenja putanje u obliku kvadrata tijekom obilaska u smjeru kazaljke na satu i u suprotnom smjeru kazaljke na satu. Ovaj tip greške nastaje zbog lošeg omjera stvarnog razmaka između kotača i nominalnog razmaka između kotača, taj omjer je

definiran koeficijentom E_b te je ranije objašnjen. Ako se uvede pretpostavka da je stvarni kut zakreta robota manji od nazivnog za oba smjera obilaznja kvadratne putanje, može se napisati:

$$|\theta_{ukupni,cw}| < |\theta_{nazivni,cw}| \text{ i } |\theta_{ukupni,ccw}| < |\theta_{nazivni,ccw}|. \quad (22)$$

Slika 30. prikazuje grešku tipa A prilikom obilaska kvadratne putanje u smjeru kazaljke na satu i u suprotnom smjeru kazaljke na satu.



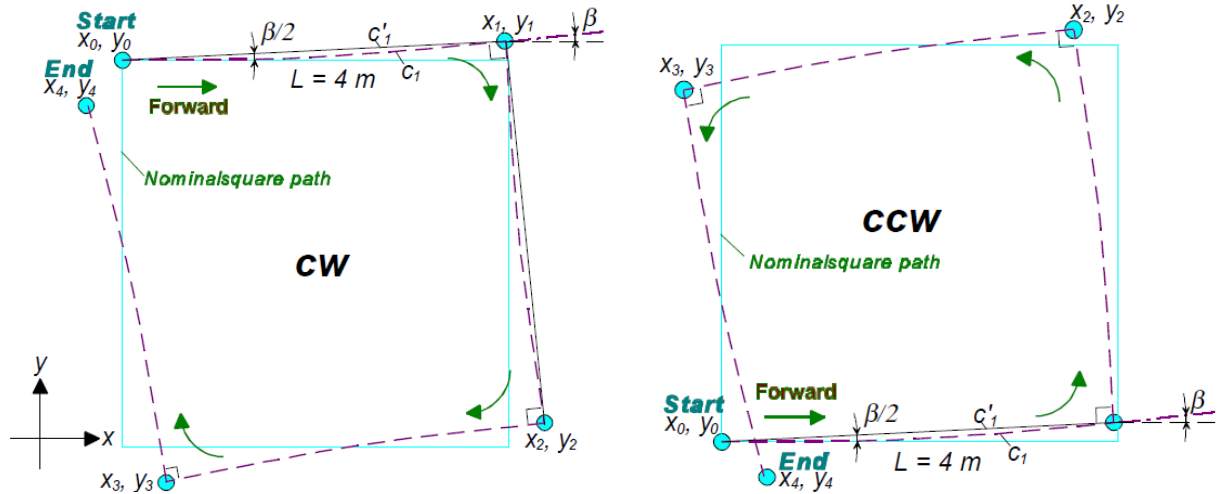
Slika 30. Greška tipa A u smjeru kazaljke na satu i suprotno smjeru kazaljke na satu [20]

6.5.2.2. Greška tipa B

Greška tipa B je definirana kao greška orijentacije koja djeluje na ukupan iznos rotacije robota na način da ju povećava ili smanjuje u jednom smjeru obilaska kvadratne putanje te ju suprotno tome smanjuje ili povećava u suprotnom smjeru obilaska kvadratne putanje. Ovaj tip greške nastaje zbog nejednakosti promjera kotača, taj nejednakost je definirana koeficijentom E_d te je ranije objašnjena. Tako se za jedan i drugi smjer može zapisati:

$$|\theta_{ukupni,cw}| < |\theta_{nazivni,cw}| \text{ i } |\theta_{ukupni,ccw}| > |\theta_{nazivni,ccw}|, \quad (22)$$

i obrnuto. Slika 31. prikazuje grešku tipa B prilikom obilaska kvadratne putanje u smjeru kazaljke na satu i u suprotnom smjeru kazaljke na satu.



Slika 31. Greška tipa B u smjeru kazaljke na satu i suprotno smjeru kazaljke na satu [20]

6.5.2.3. Matematičke pretpostavke

Da bismo pojednostavili matematički model grešaka nakon definiranja modela, uvodimo određene pretpostavke. Prva pretpostavka je da je početna pozicija robota (x_0, y_0) u ishodištu $(0, 0)$. Osim toga, primijenjene su aproksimacije za male kutove grešaka proizašle iz praćenja kvadratne putanje. Te aproksimacije su valjane za male kutove grešaka generirane izvođenjem kvadratne putanje u oba smjera, što dodatno pojednostavljuje matematički model grešaka. Važno je naglasiti da kada dolazi do pojave širokog raspona sustavnih grešaka odometrije takva pretpostavka nije baš prikladna.

$$L \sin \gamma \approx L \gamma$$

$$L \sin 2\gamma \approx 2L \gamma \quad (23)$$

$$L \sin 3\gamma \approx 3L \gamma$$

$$L \cos \gamma \approx L \gamma$$

$$L \cos 2\gamma \approx L \quad (24)$$

$$L \cos 3\gamma \approx L$$

gdje su:

- L - duljina svake stranice kvadratne putanje
- γ - svaka inkrementalna greška uzrokovana E_d ili E_b .

Sljedi prikaz matematičkog model grešaka tipa A i B, sve oznake prilikom određivanja ovih izraza nalaze se na slikama (30.) i (31.)

Za grešku tipa A u smjeru kazaljke na satu prema slici mogu se izvesti sljedeći izrazi:

$$\begin{aligned}x_1 &= x_0 + L \\y_1 &= y_0\end{aligned}\tag{25}$$

$$\begin{aligned}x_2 &= x_1 + L \sin \alpha \approx L + L\alpha \\y_2 &= y_1 - L \cos \alpha \approx -L\end{aligned}\tag{26}$$

$$\begin{aligned}x_3 &= x_2 - L \cos 2\alpha \approx L\alpha \\y_3 &= y_2 - L \sin 2\alpha \approx -L - 2L\alpha\end{aligned}\tag{27}$$

$$\begin{aligned}x_4 &= x_3 - L \sin 3\alpha \approx -2L\alpha \\y_4 &= y_3 - L \cos 3\alpha \approx -2L\alpha\end{aligned}\tag{28}$$

Za grešku tipa A u smjeru kazaljke na satu prema slici mogu se izvesti sljedeći izrazi:

$$\begin{aligned}x_1 &= x_0 + L \\y_1 &= y_0\end{aligned}\tag{29}$$

$$\begin{aligned}x_2 &= x_1 + L \sin \alpha \approx L + L\alpha \\y_2 &= y_1 + L \cos \alpha \approx L\end{aligned}\tag{30}$$

$$\begin{aligned}x_3 &= x_2 - L \cos 2\alpha \approx L\alpha \\y_3 &= y_2 + L \sin 2\alpha \approx L + 2L\alpha\end{aligned}\tag{31}$$

$$\begin{aligned}x_4 &= x_3 - L \sin 3\alpha \approx -2L\alpha \\y_4 &= y_3 - L \cos 3\alpha \approx 2L\alpha\end{aligned}\tag{32}$$

Greška tipa B zbog nejednakosti promjera kotača uzrokuje krivudavu kretnju robota te se slijedno tome stvara greška orijentacije β na slici. Analizirajući sliku može se uočiti linija c'_1 koja spaja stvarne točke robota te ima nagib od $\frac{\beta}{2}$ zato što je ta linija paralelna s tangentom u središtu luka c_1 .

Za grešku tipa B u smjeru kazaljke na satu prema slici mogu se izvesti sljedeći izrazi:

$$\begin{aligned}x_1 &= x_0 + L \cos \frac{\beta}{2} \approx L \\y_1 &= y_0 + L \sin \frac{\beta}{2} \approx L \frac{\beta}{2}\end{aligned}\tag{33}$$

$$\begin{aligned}
 x_2 &= x_1 + L \sin \frac{3\beta}{2} \approx L + 3L \frac{\beta}{2} \\
 y_2 &= y_1 - L \cos \frac{3\beta}{2} \approx L \frac{\beta}{2} - L
 \end{aligned}
 \tag{34}$$

$$\begin{aligned}
 x_3 &= x_2 - L \cos \frac{5\beta}{2} \approx 3L \frac{\beta}{2} \\
 y_3 &= y_2 - L \sin \frac{5\beta}{2} \approx -L(2\beta + 1)
 \end{aligned}
 \tag{35}$$

$$\begin{aligned}
 x_4 &= x_3 - L \sin \frac{7\beta}{2} \approx -2L\beta \\
 y_4 &= y_3 - L \cos \frac{7\beta}{2} \approx -2L\beta
 \end{aligned}
 \tag{36}$$

Za grešku tipa B u smjeru kazaljke na satu prema slici mogu se izvesti sljedeći izrazi:

$$\begin{aligned}
 x_1 &= x_0 + L \cos \frac{\beta}{2} \approx L \\
 y_1 &= y_0 + L \sin \frac{\beta}{2} \approx L \frac{\beta}{2}
 \end{aligned}
 \tag{37}$$

$$\begin{aligned}
 x_2 &= x_1 - L \sin \frac{3\beta}{2} \approx L - 3L \frac{\beta}{2} \\
 y_2 &= y_1 + L \cos \frac{3\beta}{2} \approx L \frac{\beta}{2} + L
 \end{aligned}
 \tag{38}$$

$$\begin{aligned}
 x_3 &= x_2 - L \cos \frac{5\beta}{2} \approx -3L \frac{\beta}{2} \\
 y_3 &= y_2 - L \sin \frac{5\beta}{2} \approx -L(2\beta - 1)
 \end{aligned}
 \tag{39}$$

$$\begin{aligned}
 x_4 &= x_3 + L \sin \frac{7\beta}{2} \approx 2L\beta \\
 y_4 &= y_3 - L \cos \frac{7\beta}{2} \approx -2L\beta
 \end{aligned}
 \tag{40}$$

Superponiranjem grešaka tipa A i B za kretanje robota u smjeru osi x za gibanje u smjeru kazaljke na satu i suprotno kazaljke na satu vrijede izrazi:

$$x_{c.g.,cw} = -2L(\alpha + \beta)
 \tag{41}$$

$$x_{c.g.,ccw} = -2L(\alpha - \beta) \quad (42)$$

oduzimanjem (42) od (41)

$$-4L\beta = x_{c.g.,cw} - x_{c.g.,ccw} \quad (43)$$

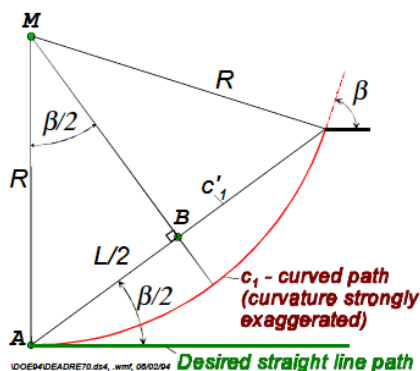
odnosno u stupnjevim β iznosi:

$$\beta = \frac{x_{c.g.,cw} - x_{c.g.,ccw}}{-4L} \cdot \frac{180^\circ}{\pi} \quad (44)$$

Također, sličan izraz vrijedi i za pomake po y osi:

$$\beta = \frac{y_{c.g.,cw} + y_{c.g.,ccw}}{-4L} \cdot \frac{180^\circ}{\pi} \quad (45)$$

Iz dobivenih izraza možemo izračunati polumjer krivulje R na slici, koristeći jednostavne geometrijske odnose s trokutom ABM s slike .



Slika 32. Geometrijski odnosi za izračun polumjera skretanja R [20]

$$R = \frac{L/2}{\sin(\frac{\beta}{2})} \quad (46)$$

Pomoću izraza za polumjer jednostavno se dobiva izraz za omjer desnog i lijevog kotača E_d :

$$E_d = \frac{D_R}{D_L} = \frac{R + \frac{b_{nazivni}}{2}}{R - \frac{b_{nazivni}}{2}} \quad (47)$$

Analogno određivanju kuta β , kut α može se izračunati zbrajanjem relevantnih jednadžbi (40) i (41):

$$-4\alpha L = x_{c.g.,cw} + x_{c.g.,ccw} \quad (48)$$

odnosno u stupnjevim α iznosi:

$$\alpha = \frac{x_{c.g.,cw} + x_{c.g.,ccw}}{-4L} \cdot \frac{180^\circ}{\pi} \quad (49)$$

Također, sličan izraz vrijedi i za pomake po y osi:

$$\alpha = \frac{y_{c.g.,cw} - y_{c.g.,ccw}}{-4L} \cdot \frac{180^\circ}{\pi} \quad (50)$$

Kada imamo izračunat kut α , možemo odrediti grešku u omjeru stvarnog i nominalnog razmaka između kotača E_b . S obzirom na to da je razmak među kotačima b izravno proporcionalan stvarnoj rotaciji, koristeći jednadžbu (10), možemo je zapisati:

$$\frac{b_{stvarni}}{90^\circ} = \frac{b_{nazivni}}{90^\circ - \alpha} \quad (51)$$

Pravilnim sređivanjem izraza (51) možemo izračunati vrijednost E_b :

$$E_b = \frac{b_{stvarni}}{b_{nazivni}} = \frac{90^\circ}{90^\circ - \alpha} \quad (52)$$

6.5.3. Kompenzacija sustavnih grešaka mobilnog robota

Kada se kvantitativne vrijednosti omjera E_b i E_d izračunaju, ispravljanje i anuliranje sustavni grešaka programski je jednostavno. Može se lako promijeniti udaljenost između kotača b u diferencijalnom kontroleru mobilnog robota s pomoću izraza (52).

Međutim, kompenzacija promjera kotača predstavlja veći izazov. Prilikom primjene kompenzacijskog faktora, važno je osigurati da se srednji promjer kotača D_a održava konstantnim. Inače, morala bi se izvršiti i kalibracija srednjeg promjera kotača. Zahtjev da srednji promjer ostane konstantan može se promatrati kao ograničenje prema izrazu:

$$D_a = \frac{D_R + D_L}{2} \quad (53)$$

S obzirom da izraz (14) daje korekcijski parametar za nejednakosti promjera kotača koji se određuje s pomoću izraza (47) dobivamo jednadžbu s dvije nepoznanice, te se promjeri desnog i lijevog kotača računaju prema izrazima:

$$D_R = \frac{2}{E_d + 1} D_a \quad (54)$$

$$D_L = \frac{2}{(1/E_d) + 1} D_a \quad (55)$$

Nadalje mogu se definirati dva korekcijska faktora:

$$c_R = \frac{2}{E_d + 1} \quad (56)$$

$$c_L = \frac{2}{(1/E_d) + 1} \quad (57)$$

Koji se mogu implementirati u unutar izraza za estimaciju pozicije prema *dead-reckoning* algoritmu.

$$\Delta U_{R/L,i} = c_{R/L} c_m N_{R/L,i} \quad (58)$$

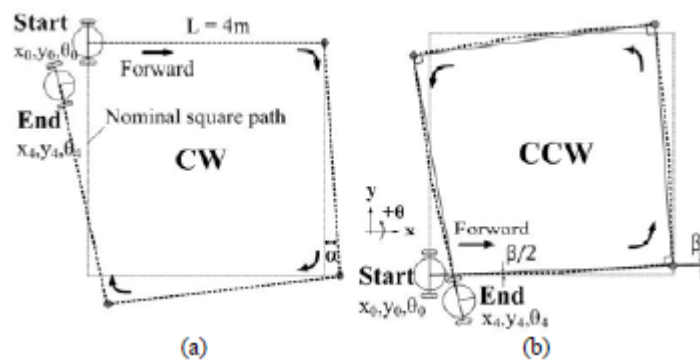
Implementacijom ovih parametara ispravljene su obje sustavne greške mobilnog robota.

6.6. Proširena UMBmark metoda

Ova metoda kao što je navedeno objavljena u [21] predstavlja proširenje s obzirom na postojeću *UMBmark*. U ovoj metodi da bi se izbjeglo korištenje aproksimacije za male kuteve ne koriste se podatci o grešci krajnje pozicije robota već su novi izrazi za kalibraciju vođeni s obzirom na grešku završne orijentacije robota koja se pojavljuje zbog grešaka tipa A i B (objašnjenih u jednom od prethodnih poglavlja) pri obilasku kvadratne putanje mobilnog robota u smjeru kazaljke na satu i u suprotnom smjeru kazaljke na satu isto kao kod *UMBmark*. Odnosno s obzirom na toda *UMBmark* metoda ne uzima u obzir združeni efekt greške koji daju greška međuosovinskog razmaka i greške u promjeru kotača već ih smatra potpuno neovisnim u ovoj metodi je to uzeto u obzir.

6.7. Pretpostavke i matematički izrazi za proširenu *UMBmark* metodu

Metoda je vrlo jednostavna i praktična jer je potrebno izmjeriti samo završnu orijentaciju robota.



Slika 33. Prikaz greške završne pozicije i orijentacije robota s obzirom na tip greške A (a) u smjeru kazaljke na satu i tip greške B (b) smjer obrnuto kazaljke na satu [21]

Slika 33. prikazuje završnu poziciju robota s obzirom na tip greške A (a) u smjeru kazaljke na satu i tip greške B (b) smjer suprotnom kazaljke na satu. Greška orijentacije završne pozicije je jednostavno izračunata primjenom kuteva α i β koje dobivamo s obzirom na tip greške. α i β prikazani su na slici 33.

$$\begin{aligned} \text{Greška tipa A: } \theta_{4,CW} &= 4\alpha, \theta_{4,CCW} = -4\alpha \\ \text{Greška tipa A: } \theta_{4,CW} &= 4\beta, \theta_{4,CCW} = 4\beta \end{aligned} \quad (59)$$

gdje su:

- $\theta_{4,CW}$ -greška orijentacije pri obilasku kvadratne putanje u smjeru kazaljke na satu
- $\theta_{4,CCW}$ - greška orijentacije pri obilasku kvadratne putanje u suprotnom smjeru kazaljke na satu.

greška orijentacije računa prema izrazu:

$$\epsilon_{\theta_{4,CW/CCW}} = \theta_{abs} - \theta_{calc} \quad (60)$$

gdje su:

- θ_{abs} -apsolutna orijentacija robota u x i y ravnini
- θ_{calc} -orijentacije robota u x i y ravnini dobivena iz odometrije.

Konačna greška orijentacije $\theta_{CW}, \theta_{CCW}$ u završnoj poziciji mobilnog robota u smjeru kazaljke na satu i suprotno smjeru kazaljke na satu dobiva se superponiranjem grešaka tipa A i tipa B prema izrazima:

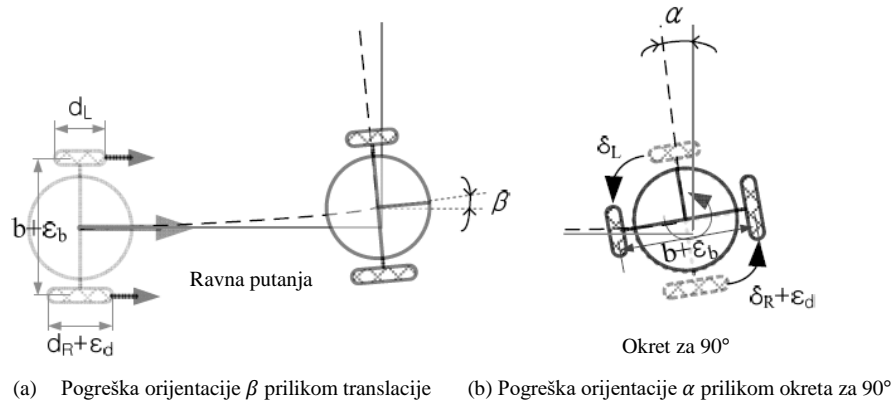
$$\begin{aligned} \theta_{CW} : \quad 4\alpha + 4\beta &= \theta_{c.g.,CW} \\ \theta_{CCW} : \quad -4\alpha + 4\beta &= \theta_{c.g.,CCW} \end{aligned} \quad (61)$$

Iz izraza (61) slijede izrazi za kut α_{ϵ_b} koji se javlja zbog pogreške pri definiranju međuosovinskog razmaka i kut β_{ϵ_d} koji se javlja zbog nejednakosti promjera kotača:

$$\alpha_{\epsilon_b} = \frac{\theta_{c.g.,CW} - \theta_{c.g.,CCW}}{8} \quad (62)$$

$$\beta_{\epsilon_d} = \frac{\theta_{c.g.,CW} + \theta_{c.g.,CCW}}{8} \quad (63)$$

Greške orijentacije α_{ϵ_b} i β_{ϵ_d} procjenjuju se bez aproksimacije za male kuteve kao kod UMBmark metode, iz tog se zaključuje da se korekcijski parametri mogu točnije odrediti.



Slika 34. Pogreške orijentacije uzrokovane tipovima grešaka A i B[21]

Sa slike 34. vidljivo je da se greška orijentacije javlja zbog zajedničkog djelovanja tipa grešaka A i B prilikom ravne kretnje i okreta za devedeset stupnjeva prilikom obilaska u suprotnom smjeru kazaljke na satu. Rezultantna greška orijentacije robota je definirana kao:

$$\theta_{\text{greška, okret za } 90^\circ}: \alpha = \alpha_{\varepsilon_b} + \alpha_{\varepsilon_d} \quad (64)$$

$$\theta_{\text{greška, ravna putanja}}: \beta = \beta_{\varepsilon_b} + \beta_{\varepsilon_d}$$

Sa slike 34. vidljivo je da se greška orijentacije α_{ε_d} koja se javlja zbog nejednakosti promjera kotača može izračunati prema izrazu:

$$\alpha_{\varepsilon_d} = \frac{\pi \cdot b_{\text{nazivni}}}{4L} \cdot \frac{\theta_{c.g.,CW} + \theta_{c.g.,CCW}}{8} \quad (65)$$

Iz čega slijedi da se parametar greške nesigurnosti razmaka između kotača računa prema izrazu:

$$E_b = \frac{b_a}{b_n} = \frac{90^\circ}{90^\circ - (\alpha_{\varepsilon_b} + \alpha_{\varepsilon_d})} \quad (66)$$

Robot putuje zakrivljenom putanjom s radijusom zakreta R prilikom ravne putanje kako je prikazano na slici 34.(a). Greška orijentacije β_{ε_d} zbog nejednakosti promjera kotača pogođena je pogreškom definiranja međuosovinskog razmaka kao što je prikazano u izrazu (64). Greška orijentacije β_{ε_b} uzrokovana je pogreškom definiranja međuosovinskog razmaka. Kako bi se izračunao kut β_{ε_b} , stvarni međuosovinski razmak b_{stvarni} izvodi se koristeći b_{nazivni} i E_b iz izraza (9). Prema izrazu (7), parametar pogreške zbog nejednakosti promjera kotača E_d može se izvesti prema izrazu:

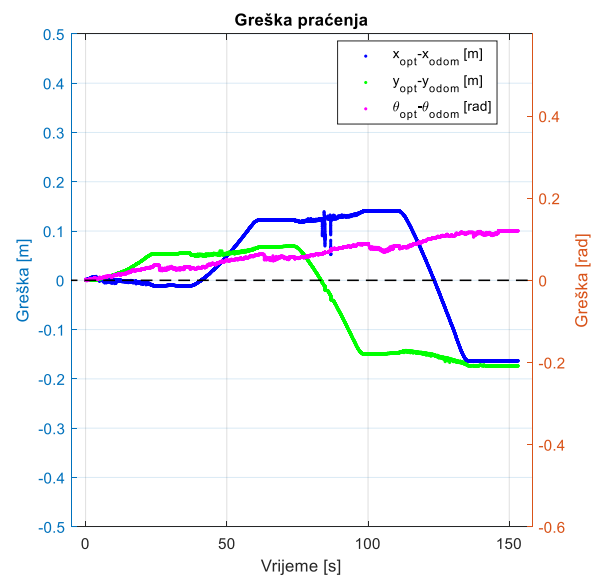
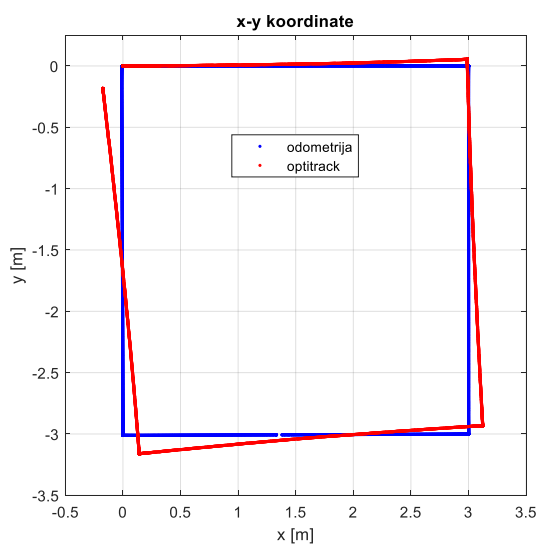
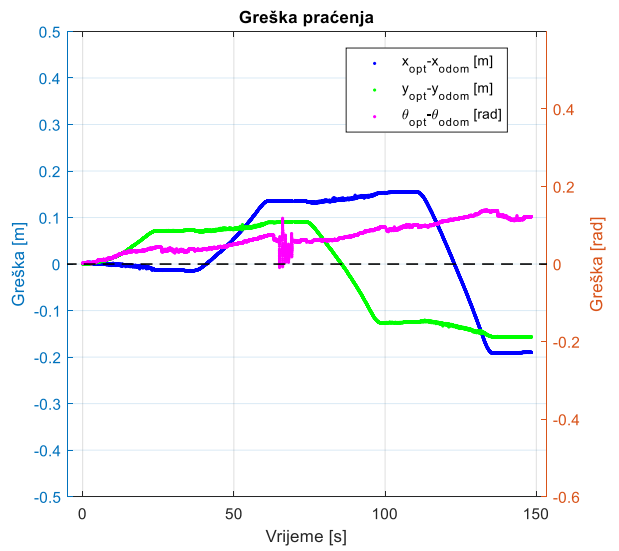
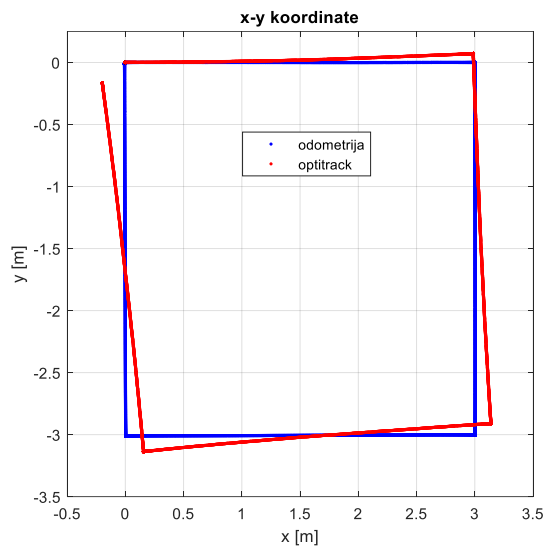
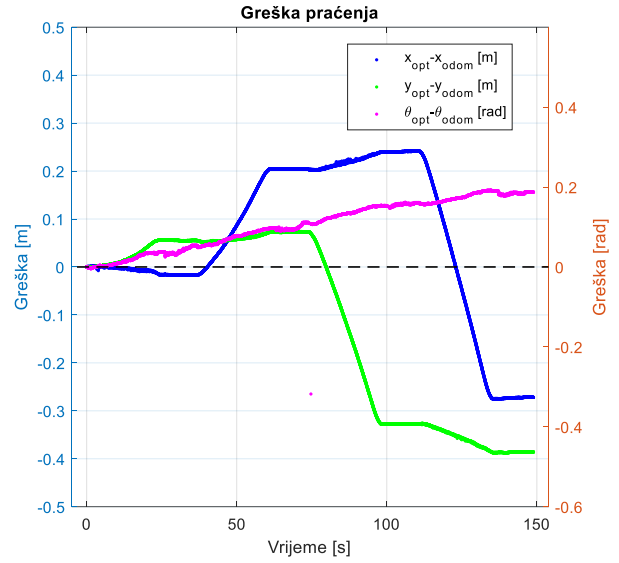
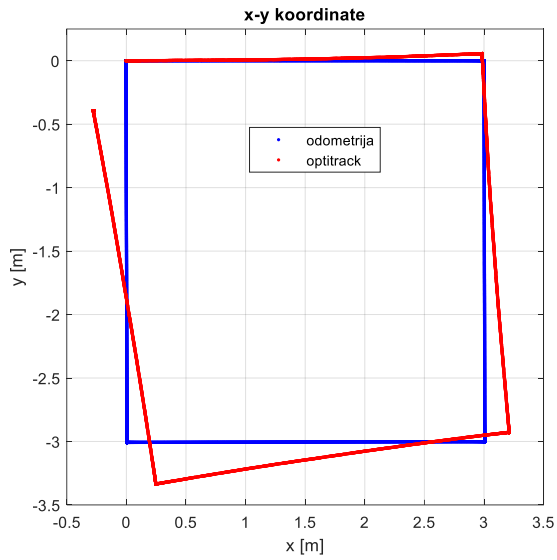
$$E_d = \frac{R + \frac{b_{\text{stvarni}}}{2}}{R - \frac{b_{\text{stvarni}}}{2}} = \frac{\frac{L/2}{\sin(\beta_{\varepsilon_d}/2)} + \frac{(b_{\text{nazivni}} \times E_b)}{2}}{\frac{L/2}{\sin(\beta_{\varepsilon_d}/2)} - \frac{(b_{\text{nazivni}} \times E_b)}{2}} \quad (67)$$

7. PRIKAZ REZULTATA MJERENJA PRIJE KALIBRACIJE

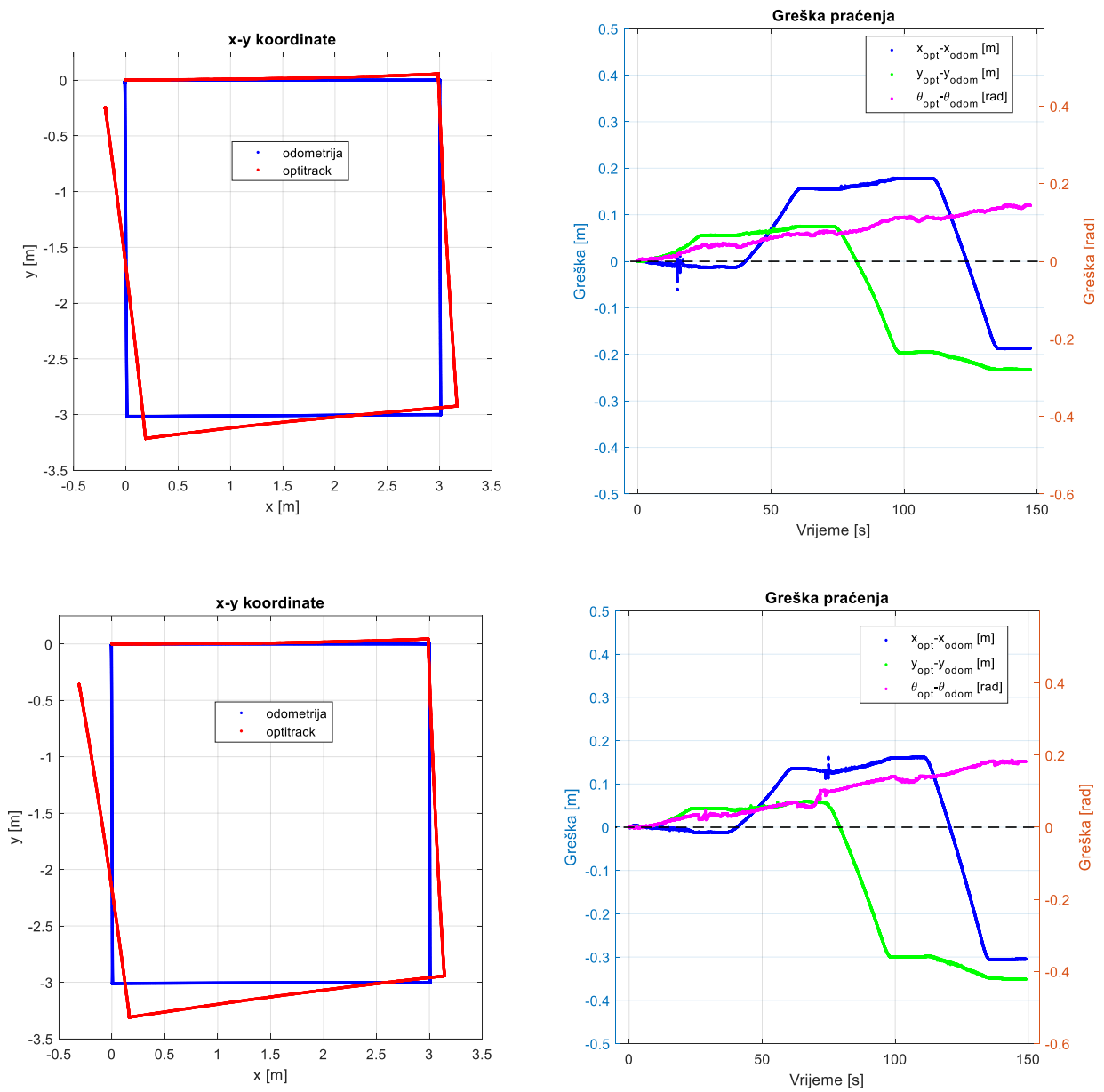
Da bi se mogla provesti kalibracija prema kalibracijskim metodama koje su navedene u prethodnom poglavlju potrebno je provesti mjerenja s nekalibriranim robotom. Kao što je navedeno, potrebno je provesti pet kretnji robota po kvadratnoj putanji u smjeru kazaljke na satu i u suprotno smjeru kazaljke na satu. Mobilni robot kreće iz ishodišta koordinatnog sustava te mu je zbog ograničenosti prostora mobilnom robotu zadana je kvadratna putanja 3 m x 3 m. S obzirom na to da kalibracijske metode imaju fokus na završnoj poziciji i orijentaciji robota, u tablicama 10. i 11. navedene su greške u smjeru osi x i y, apsolutna greška te greška završne orijentacije. Ispod svake tablice priloženi su detaljni grafovi mjerenja za svako snimanje u kojima su prikazane kretnje robota na temelju odometrije i stvarne putanje te su prikazane greške u smjeru x-osi, y-osi i same greške orijentacije robota. Greška se računa na temelju stvarne pozicije robota koja je određena OptiTrack sustavom i odometrije robota.

Tablica 11. Podaci o greškama nekalibriranog robota u smjeru kazaljke na satu za pet mjerenja

Br. snimanja	Greška u smjeru osi x[mm]	Greška u smjeru osi y[mm]	Euklidska greška u smjeru x i y[mm]	Greška orijentacije[rad]
1.	-271,98	-385,85	472,07	0,1874
2.	-189,93	-156,64	246,19	0,1222
3.	-163,22	-173,55	238,24	0,1206
4.	-186,73	-232,51	298,21	0,1439
5.	-304,81	-351,47	465,23	0,1821
RMS	229,99	276,09	359,33	0,1539



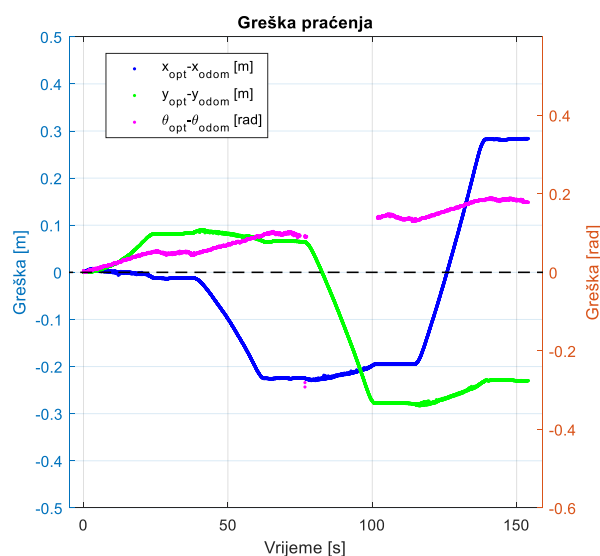
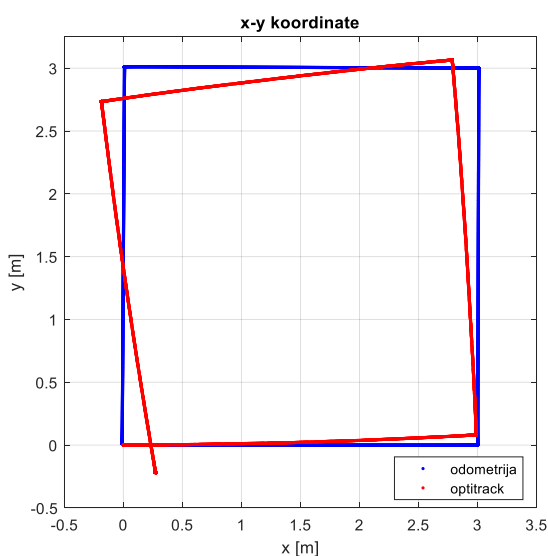
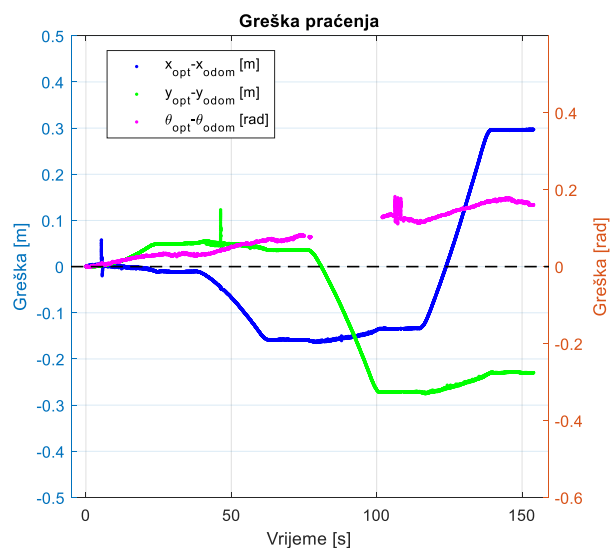
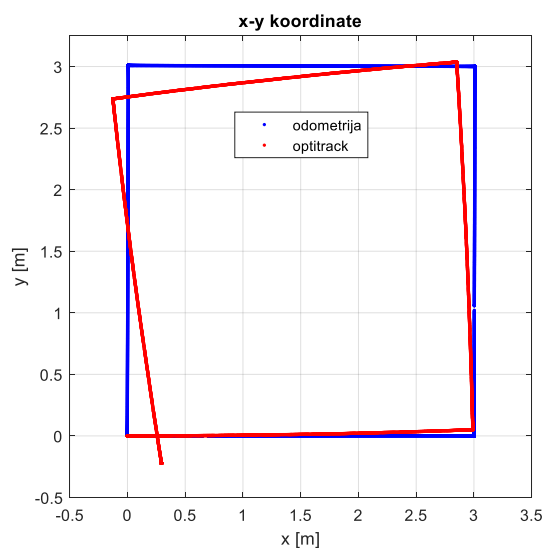
Slika 35. Mjerenja prije kalibracije u smjeru kazaljke na satu redom 1., 2., 3.



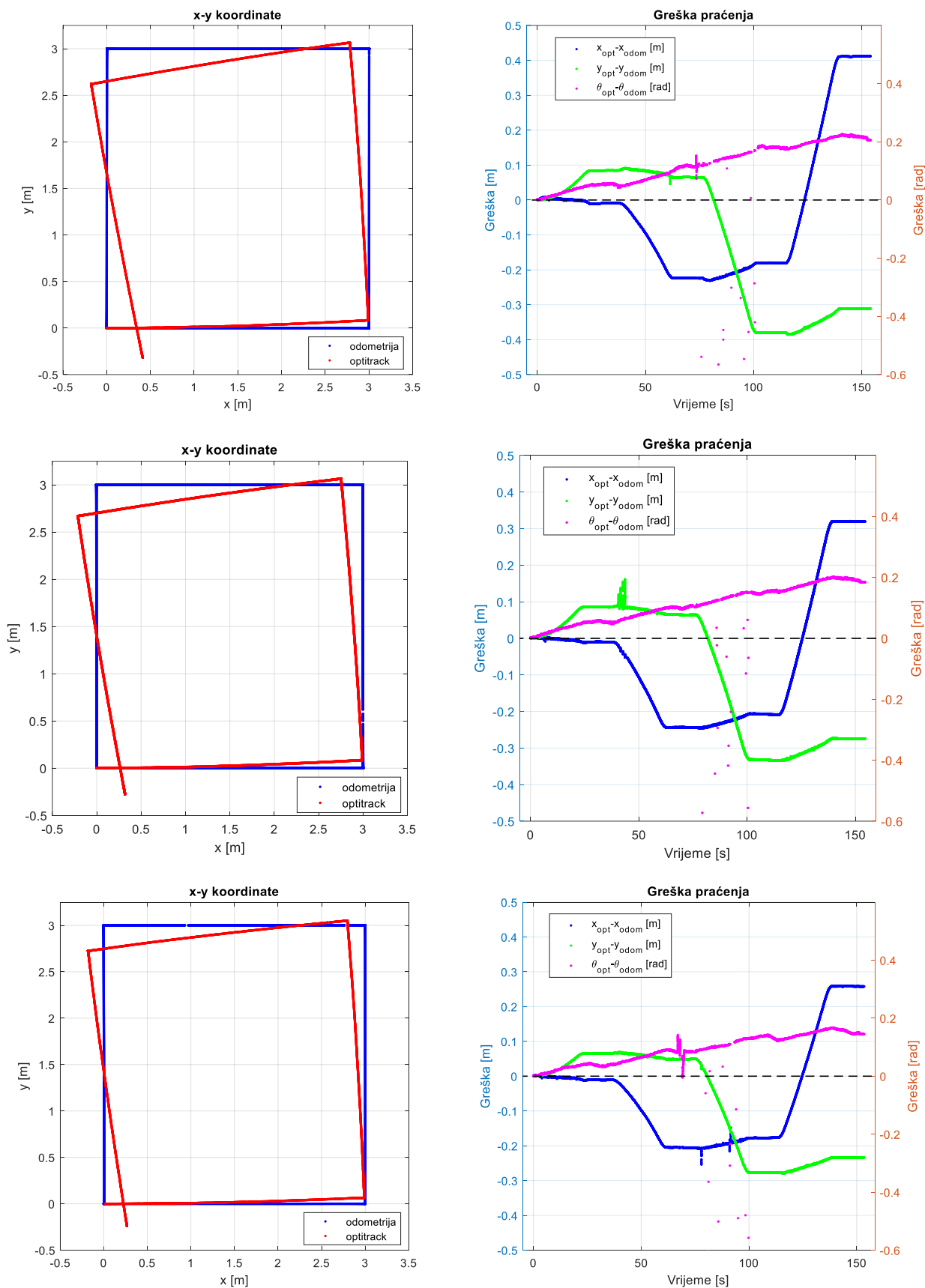
Slika 36. Mjerenja prije kalibracije u smjeru kazaljke na satu redom 4., 5.

Tablica 12. Podaci o greškama nekalibriranog robota u smjeru kazaljke na satu

Br. snimanja	Greška u smjeru osi x[mm]	Greška u smjeru osi y[mm]	Euklidska greška u smjeru x i y[mm]	Greška orijentacije[rad]
1.	296,89	-229,72	375,39	0,1603
2.	283,94	-229,84	365,30	0,1782
3.	412,08	-311,36	516,49	0,2058
4.	319,38	-274,06	420,85	0,1842
5.	257,95	-233,58	347,98	0,1448
RMS	318,47	257,76	409,71	0,1759

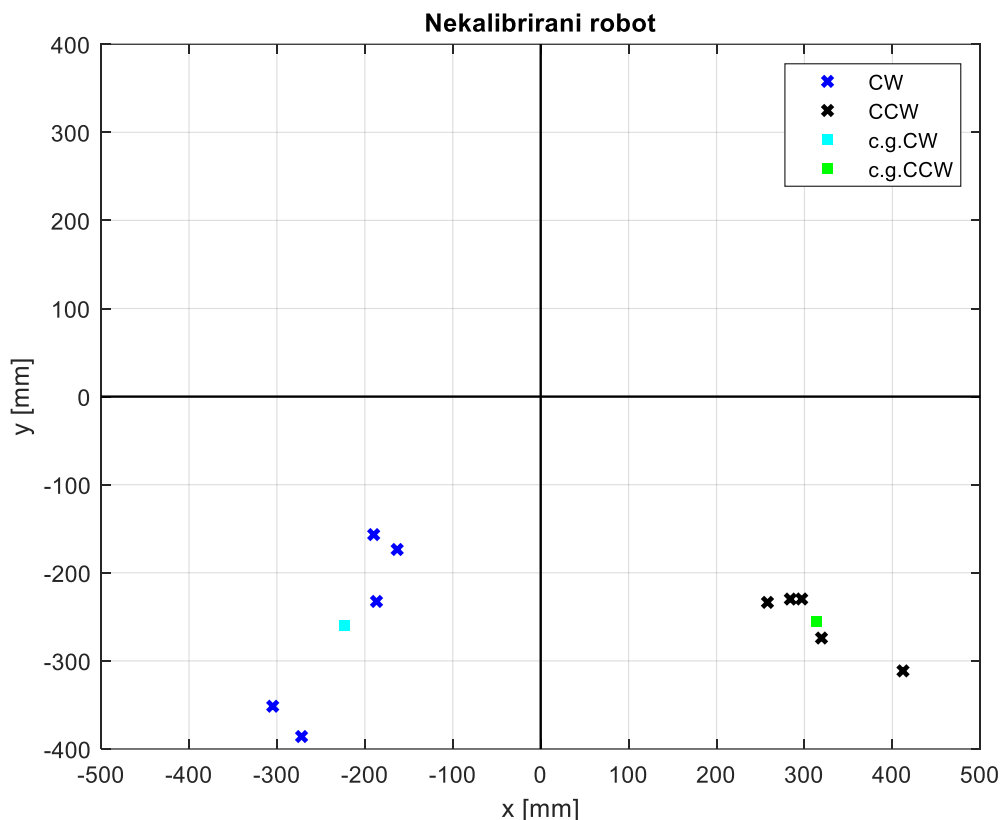


Slika 37. Mjerenja prije kalibracije u suprotnom smjeru kazaljke na satu redom 1., 2.



Slika 38. Mjerenja prije kalibracije u smjeru kazaljke na satu redom 3., 4., 5.

Iz grafova se može zaključiti da mobilni robot ASTRO ima osjetan utjecaj sustavnih grešaka koje su dodatno prikazane na slici 39., također iz razloga što zbog utjecaja nesustavnih grešaka dolazi do rasipanje rezultata ucrtane su srednje vrijednosti klastera za svaki smjer gibanja robota koje se računaju kao srednje vrijednosti grešaka u smjeru x i y osi.



Slika 39. Usporedba grešaka nekalkibriranog robota pri kretnji u smjeru kazaljke na satu (CW) i suprotno kazaljke na satu (CCW)

7.1. Korekcijski faktori prema *UMBmark* metodi

Na temelju *UMBmark* metode koja je objašnjena u jednom od prethodnih poglavlja izračunati su korekcijski parametri.

Koordinate težišta klastera te ujedno srednje vrijednosti grešaka:

- u smjeru kazaljke na satu: $x_{c.g.,CW} = -223,34\text{mm}$ (68)

$$y_{c.g.,CW} = -260,00\text{mm}$$

- u suprotnom smjeru kazaljke na satu: $x_{c.g.,CCW} = 314,05\text{mm}$ (69)

$$y_{c.g.,CCW} = -255,71\text{mm}$$

Uvrštavanjem vrijednosti iz izraza (68) i (69) u izraze (45) i (49) dobiveni su kutevi α i β :

$$\alpha = -0,4331^\circ \quad (70)$$

$$\beta = 2,4623^\circ \quad (71)$$

Korekcijski faktor za ispravljanje greške pri nesigurnosti razmaka između kotača E_b izračunat uvrštavanjem izraza (70) u izraz (52):

$$E_b = 0,99521 \quad (72)$$

Korekcijski faktor za ispravljanje greške pri nejednakosti promjera kotača E_d , izračunat je primjenom (71) i (47):

$$E_d = 1,00416 \quad (73)$$

7.2. Korekcijski parametri prema proširenoj *UMBmark* metodi

Kao i kod *UMBmark* u proširenoj *UMBmark* metodi također je potrebno pronaći težište klastera, ali u ovom slučaju težište klastera greške orijentacije:

- u smjeru kazaljke na satu: $\theta_{c.g.,CW} = 0,15123\text{rad} \quad (74)$

- u suprotnom smjeru kazaljke na satu: $\theta_{c.g.,CCW} = 0,1747\text{rad} \quad (75)$

Uvrštavanjem vrijednosti iz izraza (74) i (75) u izraze (62) i (63) dobiveni su kutevi α_{ε_b} i β_{ε_d} :

$$\alpha_{\varepsilon_b} = -0,1678^\circ \quad (76)$$

$$\beta_{\varepsilon_d} = 2,334^\circ \quad (77)$$

Uvrštavanjem vrijednosti u izraz (65) dobiven je kut α_{ε_d} :

$$\alpha_{\varepsilon_d} = 0,177206^\circ \quad (78)$$

Korekcijski faktor za ispravljanje greške pri nesigurnosti razmaka između kotača E_b izračunat uvrštavanjem vrijednosti iz izraza (76) i (78) u izraz (60):

$$E_b = 1,0001 \quad (79)$$

Korekcijski faktor za ispravljanje greške pri nejednakosti promjera kotača E_d , izračunat je uvrštavanjem vrijednosti iz izraza (77) i (79) u izraz (67):

$$E_d = 1,00395 \quad (80)$$

8. Validacija kalibracijskih metoda

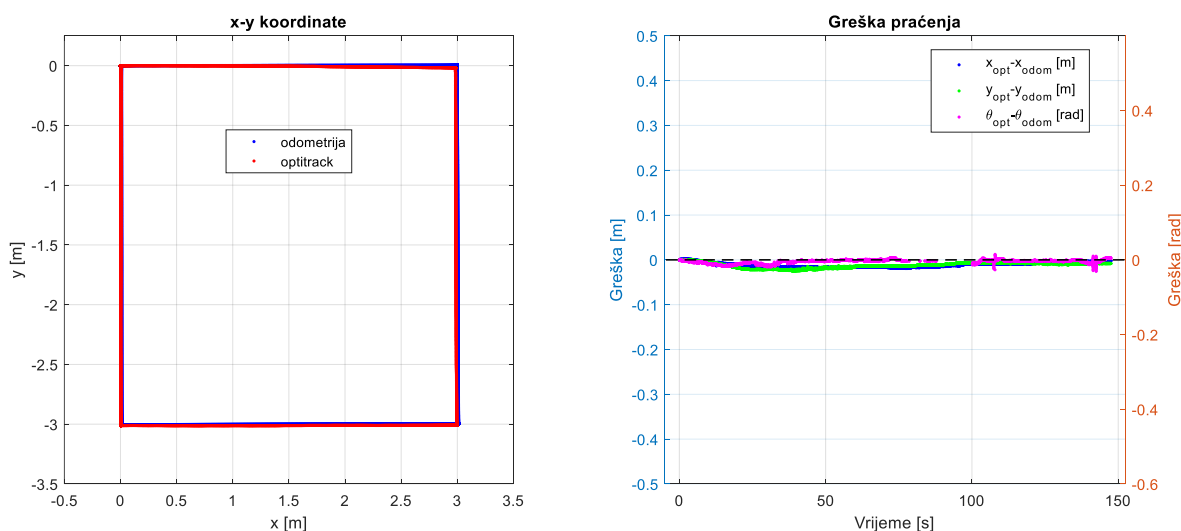
U ovom poglavlju prikazani su rezultati mjerenja s primijenjenim kalibracijskim parametrima.

8.1. Rezultati UMBmark metode

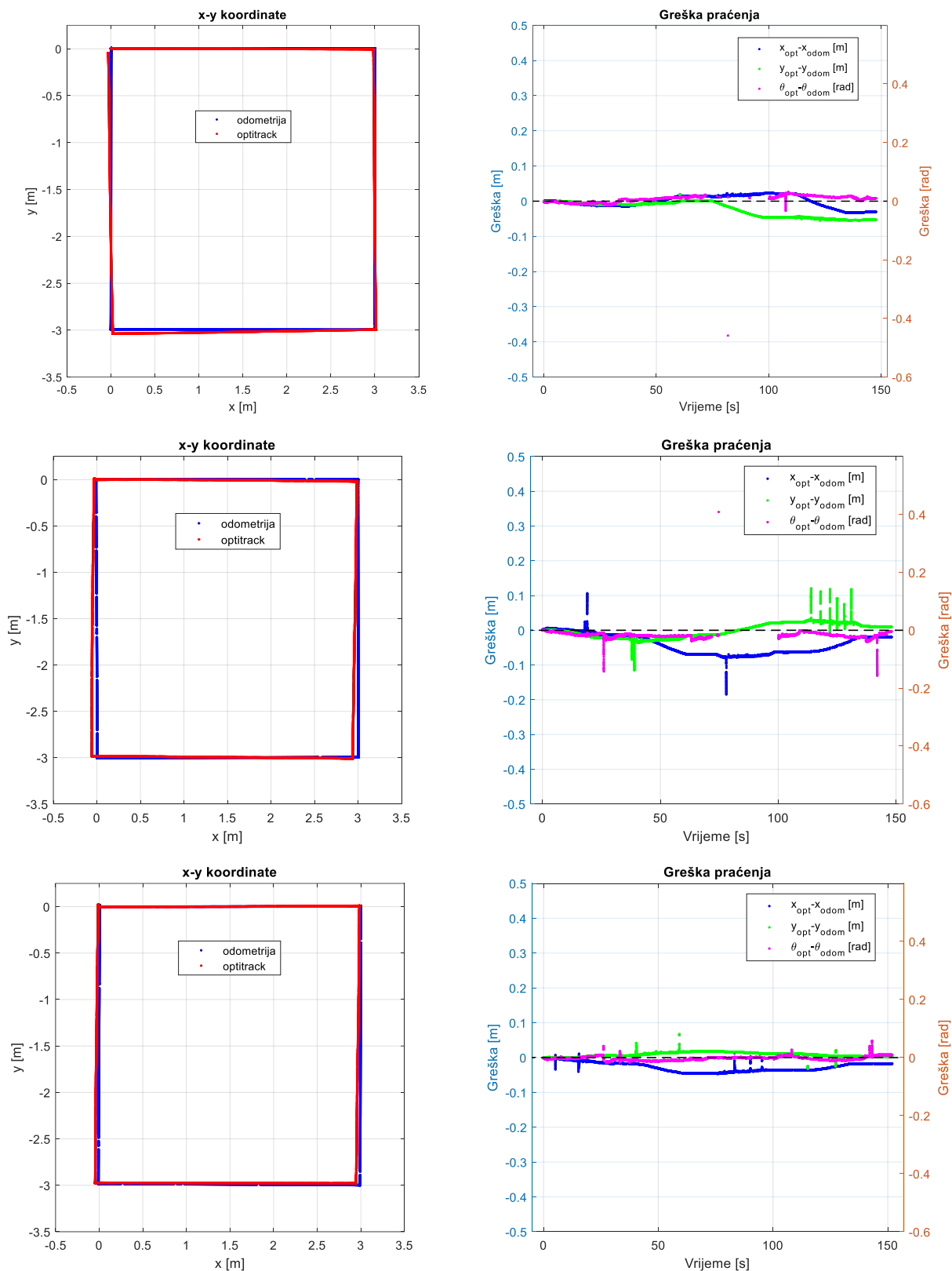
U tablicama 12. i 13. navedene su greške u smjeru osi x i y te završne greške orijentacije. Ispod svake tablice priloženi su detaljni grafovi mjerenja za svako snimanje u kojima su prikazane kretanje robota na temelju odometrije i stvarne putanje te su prikazane greške u smjeru x-osi, y-osi i same greške orijentacije robota tijekom gibanja. Greška se računa na temelju stvarne pozicije robota koja je određena OptiTrack sustavom i odometrije robota.

Tablica 13. Podaci o greškama kalibriranog robota UMBmark metodom u smjeru kazaljke na satu

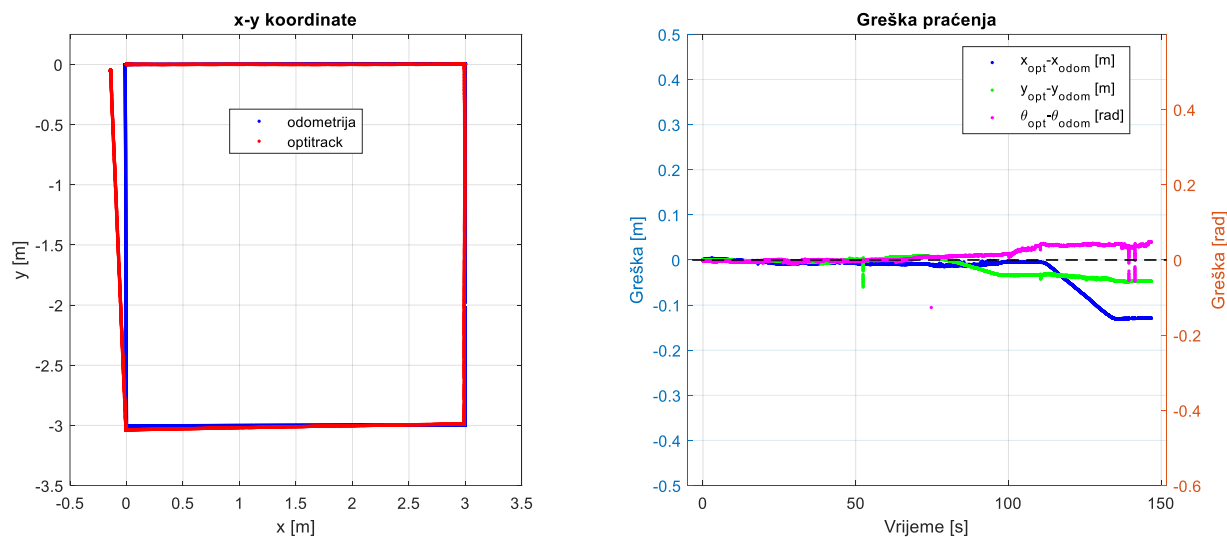
Br. snimanja	Greška u smjeru osi x[mm]	Greška u smjeru osi y[mm]	Euklidska greška u smjeru x i y[mm]	Greška orijentacije[rad]
1.	-1,94	-8,12	8,35	0,0047
2.	-30,37	-52,88	60,98	0,0101
3.	-19,58	9,44	21,74	-0,0036
4.	-17,70	2,57	17,89	0,0094
5.	-128,89	-46,97	137,18	0,0478
RMS	60,39	32,14	68,41	0,0224



Slika 40. 1. mjerenje nakon implementacije kalibracije UMBmark metode u smjeru kazaljke na satu



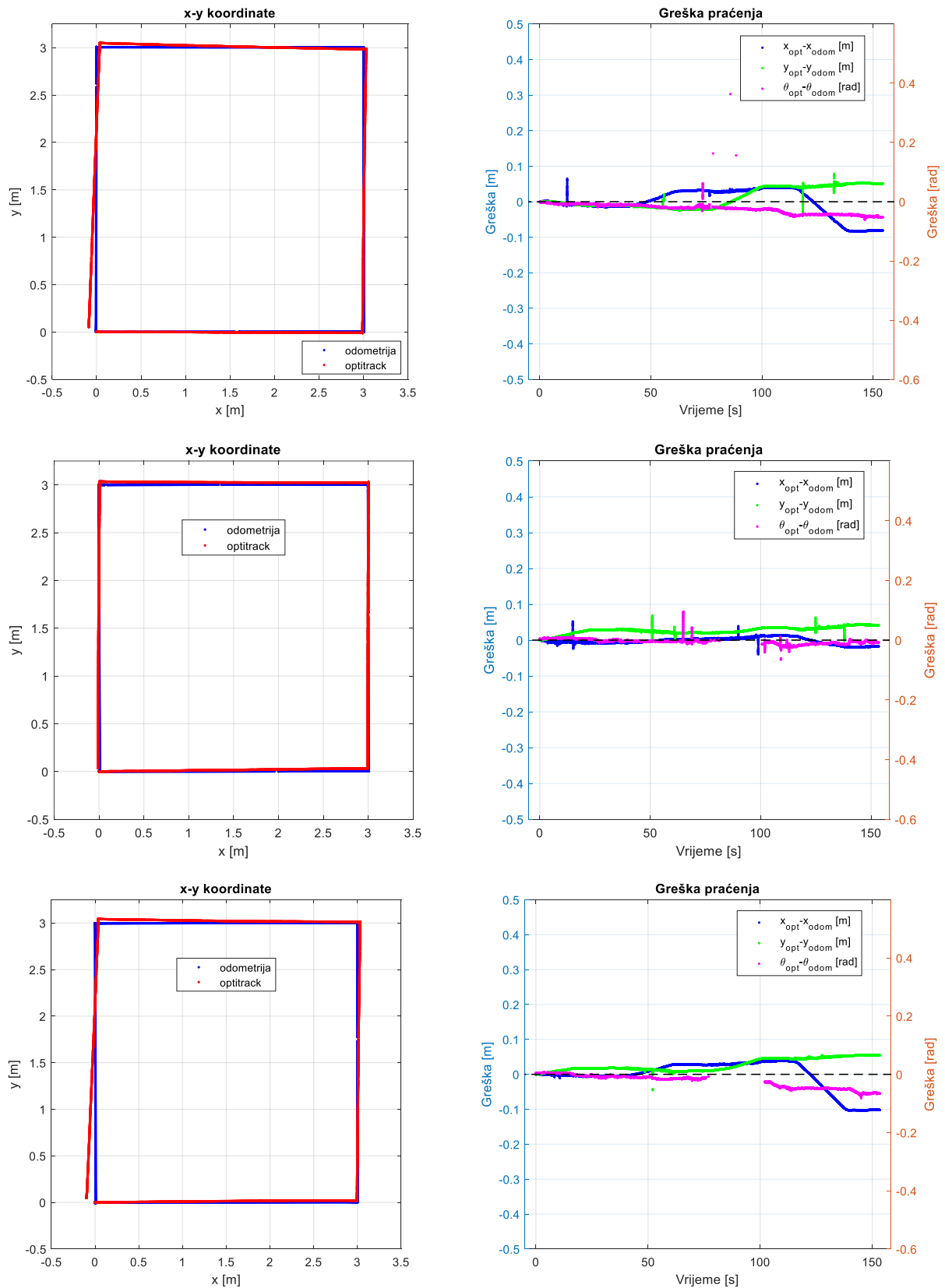
Slika 41. Mjerenja nakon implementacije kalibracije *UMBmark metode* u smjeru kazaljke na satu redom 2., 3., 4.



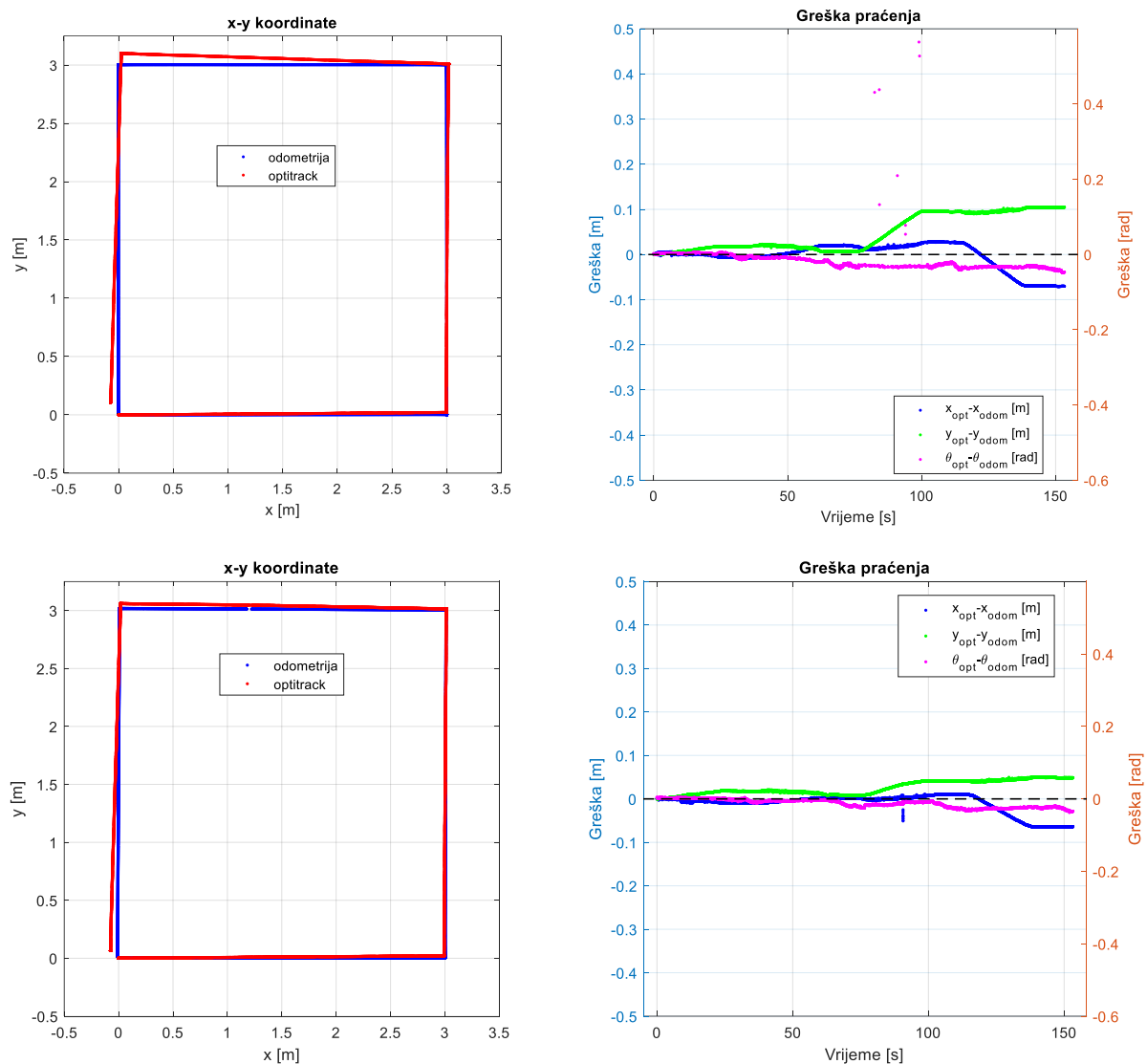
Slika 42. 5. mjerenje nakon implementacije kalibracije *UMBmark metode* u smjeru kazaljke na satu

Tablica 14. Podaci o greškama kalibriranog robota *UMBmark* metodom u suprotnom smjeru kazaljke na satu

Br. snimanja	Greška u smjeru osi x[mm]	Greška u smjeru osi y[mm]	Euklidska greška u smjeru x i y[mm]	Greška orijentacije[rad]
1.	-80,88	51,13	95,69	0,0523
2.	-17,34	42,01	45,45	0,0076
3.	-102,11	54,15	115,58	0,066
4.	-70,35	104,66	126,11	0,0471
5.	-63,89	48,67	80,31	0,0343
RMS	72,52	64,24	96,88	0,0459



Slika 43. Mjerenja nakon kalibracije UMBmark metodom u suprotnom smjeru kazaljke na satu redom 1., 2., 3.



Slika 44. Mjerenja nakon kalibracije UMBmark metodom u suprotnom smjeru kazaljke na satu redom 4., 5.

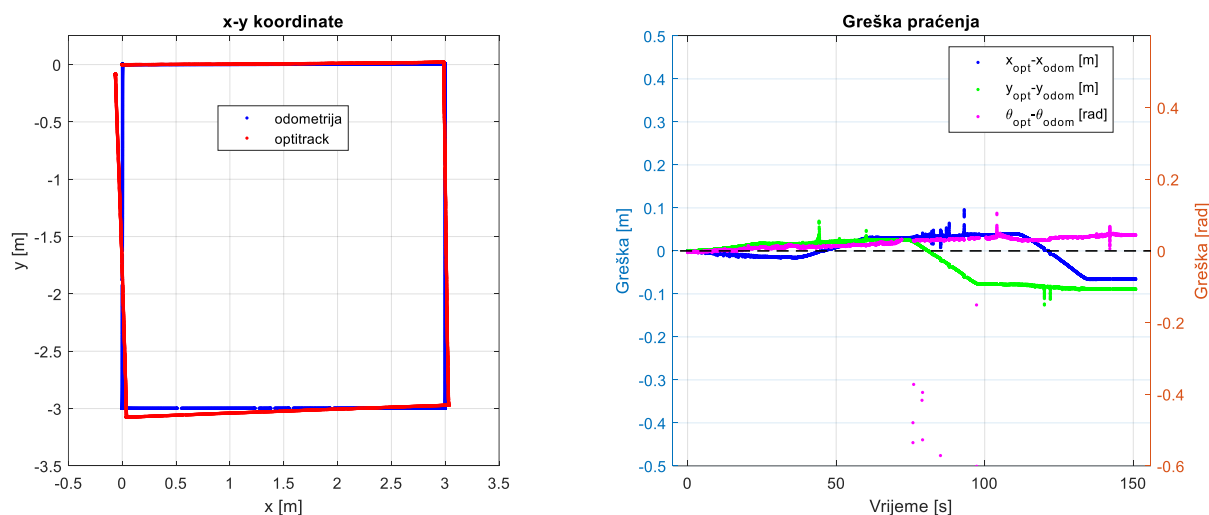
Iz grafova se može zaključiti da je ova kalibracijska metoda dala znatno bolje rezultate u odnosu mjerenja s nekalibriranim robotom, također kao i u slučaju s nekalibriranim robotom zbog utjecaja nesustavnih grešaka dolazi do rasipanje rezultata. Da bi dobili još bolju estimaciju odometrije potrebno je provesti još nekoliko puta kalibraciju.

8.2. Rezultati proširene *UMBmark* metoda

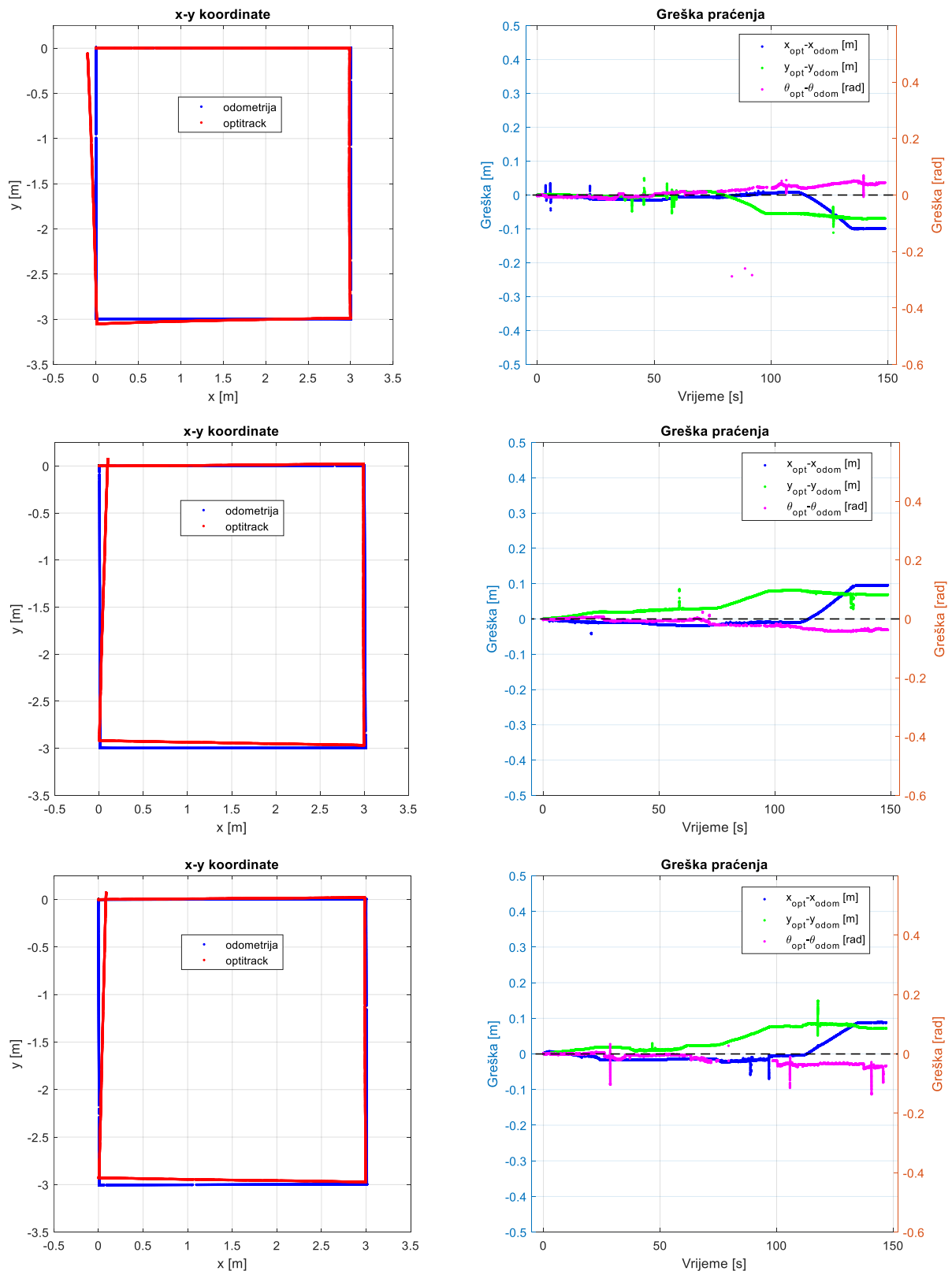
U tablicama 14. i 15. navedene su greške u smjeru osi x i y te završne greške orijentacije. Ispod svake tablice priloženi su detaljni grafovi mjerenja za svako snimanje u kojima su prikazane kretnje robota na temelju odometrije i stvarne putanje te su prikazane greške u smjeru x-osi, y-osi i same greške orijentacije robota tijekom gibanja. Greška se računa na temelju stvarne pozicije robota koja je određena OptiTrack sustavom i odometrije robota.

Tablica 15. Podaci o greškama kalibriranog robota proširenom *UMBmark* metodom u smjeru kazaljke na satu

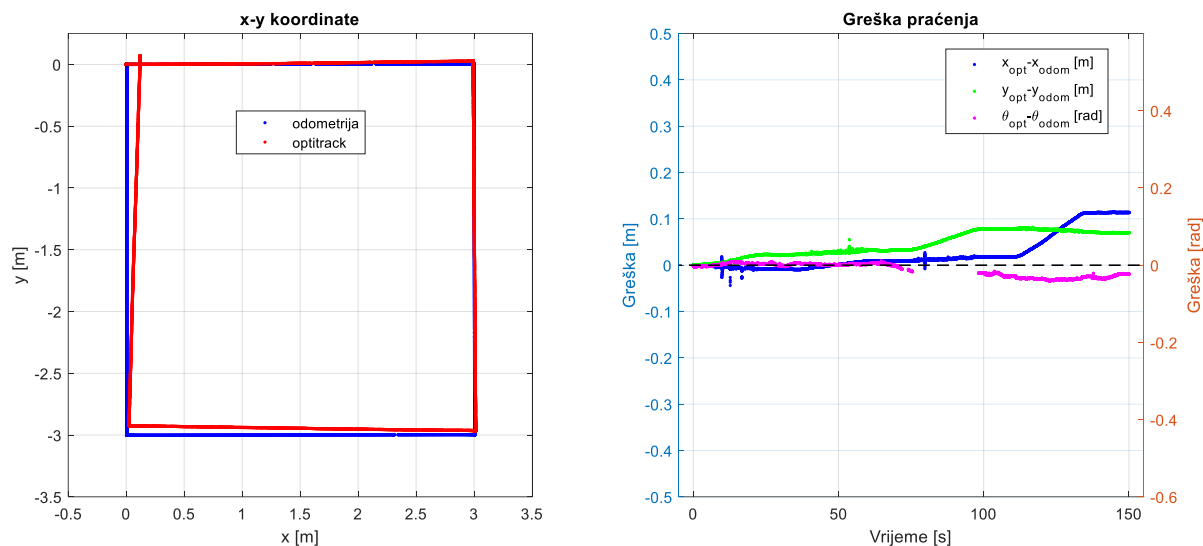
Br. snimanja	Greška u smjeru osi x[mm]	Greška u smjeru osi y[mm]	Euklidska greška u smjeru x i y[mm]	Greška orijentacije[rad]
1.	-65,49	-89,00	110,49	0,0441
2.	-99,01	-69,33	120,87	0,0442
3.	95,39	68,58	117,48	-0,0373
4.	87,79	71,94	113,51	-0,0403
5.	113,73	70,33	133,72	-0,023
RMS	93,63	74,23	119,48	0,0386



Slika 45. 1. mjerenje nakon implementacije kalibracije proširene *UMBmark* metode u smjeru kazaljke na satu



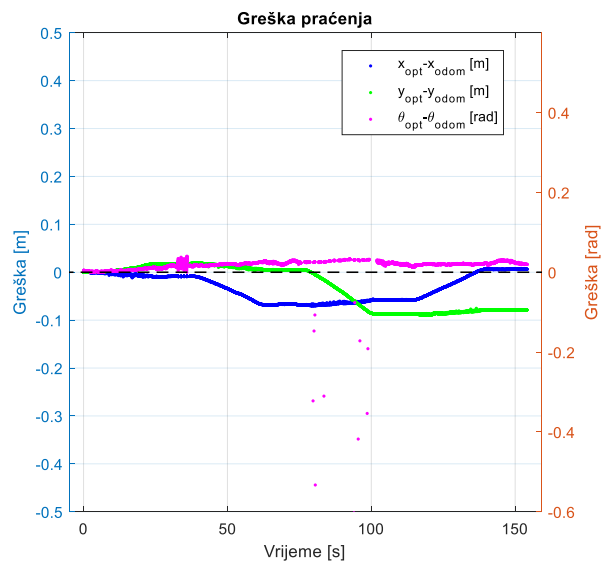
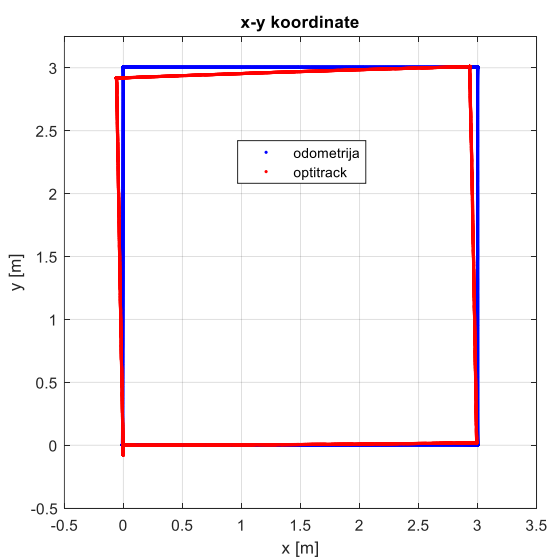
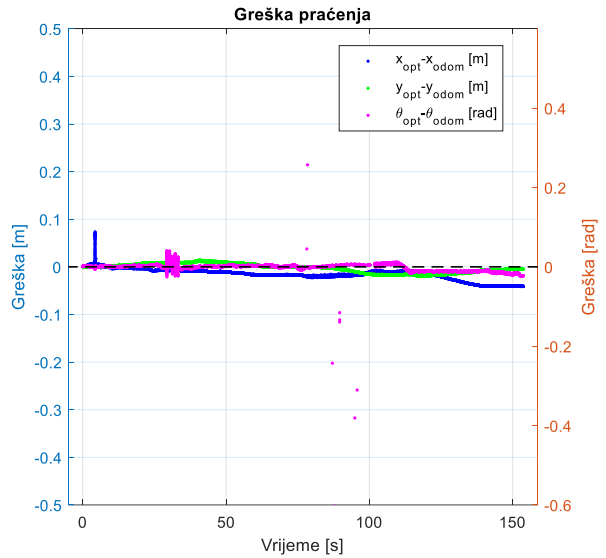
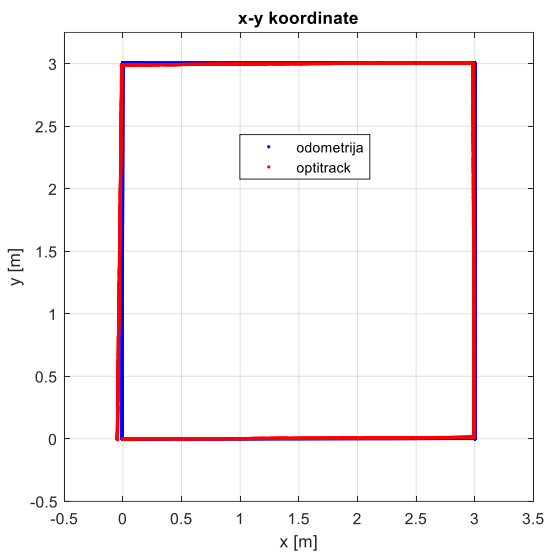
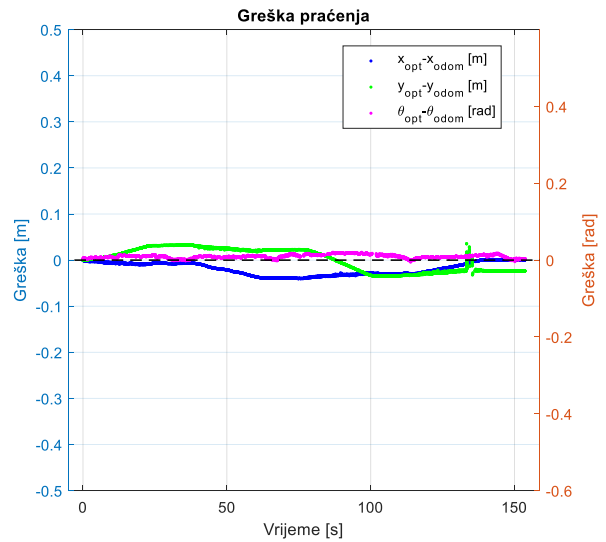
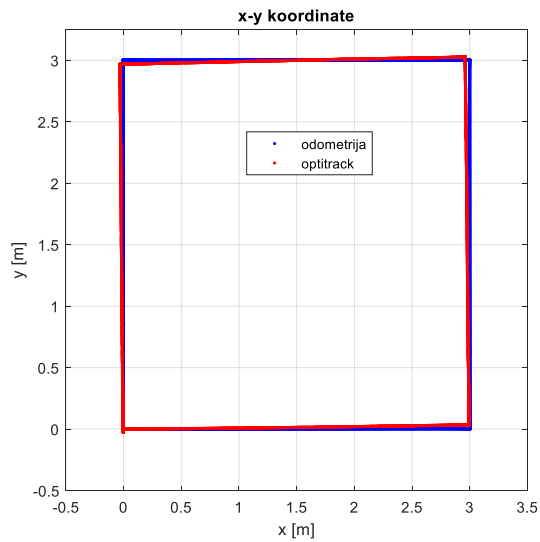
Slika 46. Mjerenja nakon implementacije kalibracije proširene *UMBmark metode* u smjeru kazaljke na satu redom 2., 3., 4.

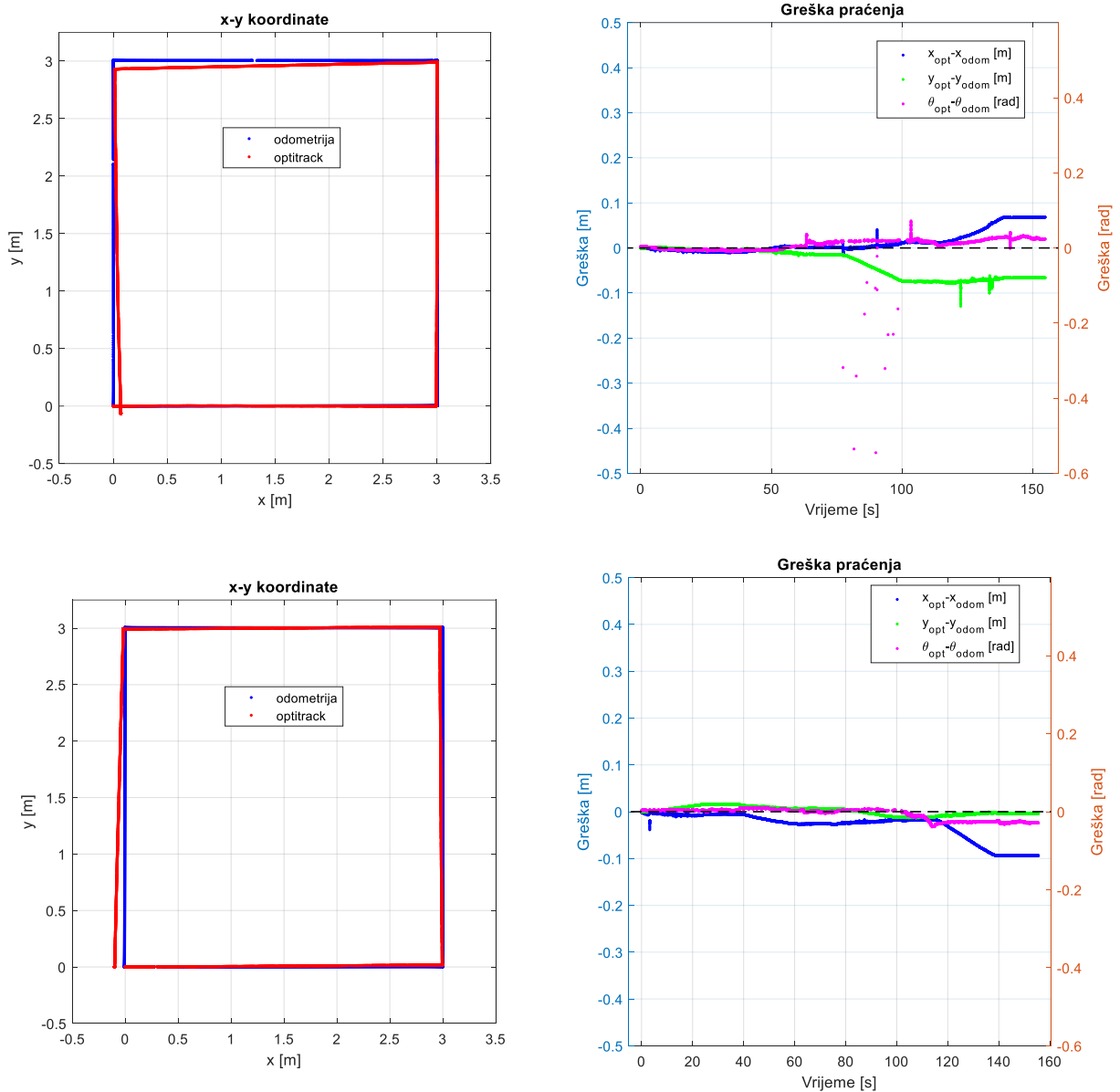


Slika 47. Mjerenja nakon kalibracije proširenom UMBmark metodom u smjeru kazaljke na satu redom

Tablica 16. Podaci o greškama kalibriranog robota proširenom UMBmark metodom u suprotnom smjeru kazaljke na satu

Br. snimanja	Greška u smjeru osi x[mm]	Greška u smjeru osi y[mm]	Euklidska greška u smjeru x i y[mm]	Greška orijentacije[rad]
1.	0,43	-23,53	23,54	0,0036
2.	-41,08	-4,92	41,38	-0,0226
3.	6,22	-78,73	78,98	0,0196
4.	68,49	-66,28	95,31	0,0239
5.	-93,73	-4,02	93,81	-0,0283
RMS	55,14	47,30	72,65	0,0214

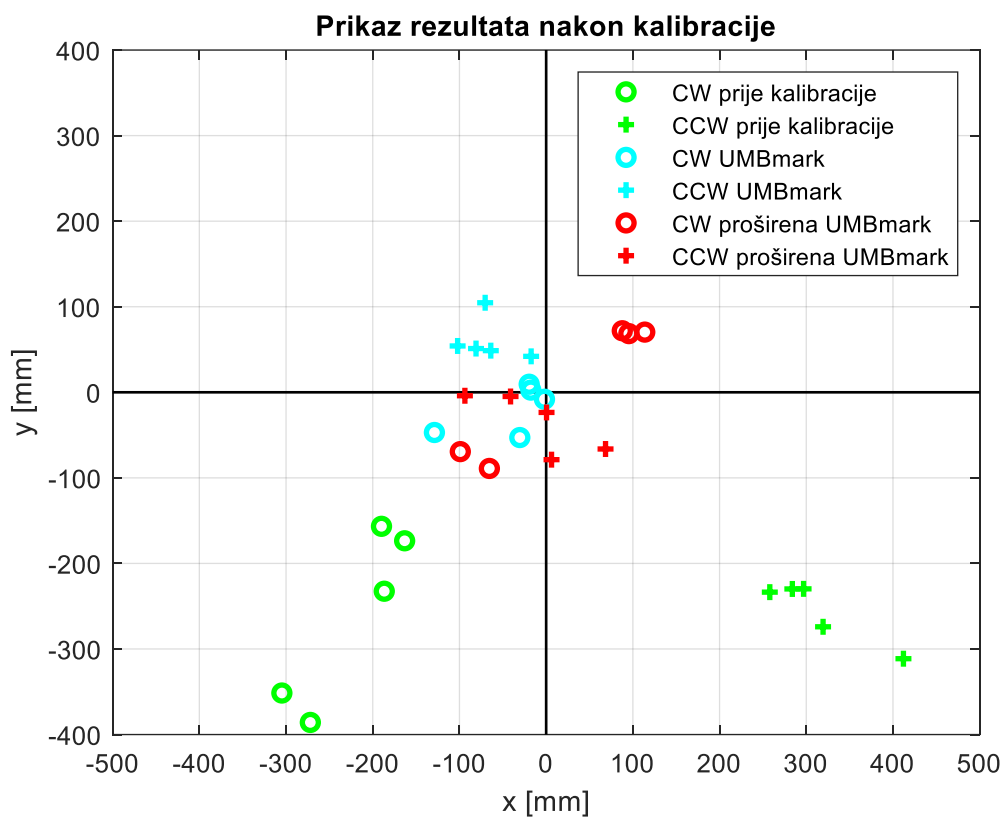




Slika 48. Mjerenja nakon kalibracije proširenom UMBmark metodom u suprotnom smjeru kazaljke na satu redom

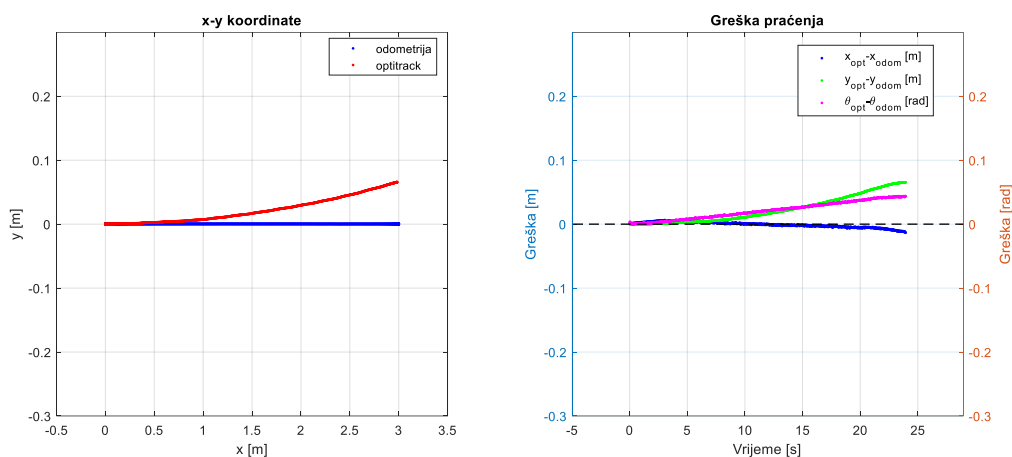
Proširena *UMBmark* metoda također je dala bolje rezultate u odnosu na nekalibriranog robota, također kao i u slučaju s nekalibriranim robotom zbog utjecaja nesustavnih grešaka dolazi do rasipanje rezultata. Za usporedbu s standardnom *UMBmark* metodom može se zaključiti da metoda nije dala bolje rezultate.

Na slici 49. prikazan je usporedba rezultata mjerenja za sve slučajeve. Vidljivo je da prvobitna *UMBmark* metoda pruža manje rasipanje završnih pozicija u odnosu na proširenu kalibracijsku metodu.

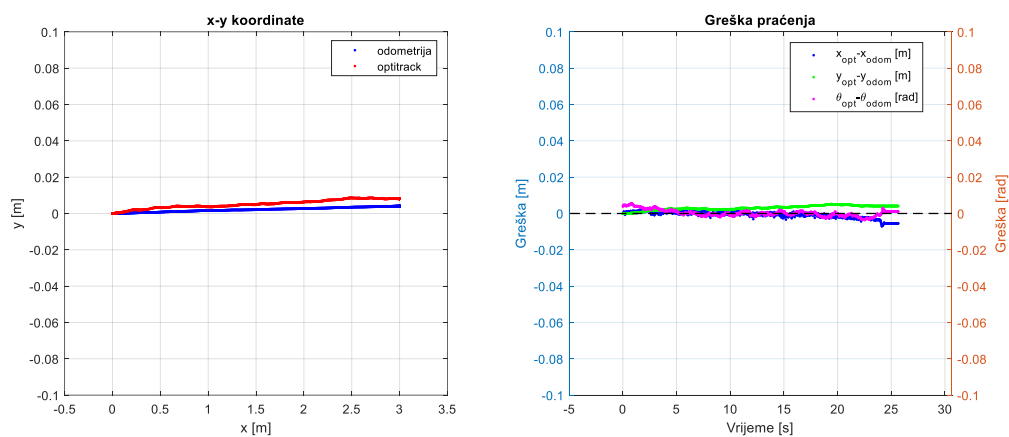


Slika 49. Usporedba greške završne pozicije robota prije kalibracije i nakon primjene dvije metode

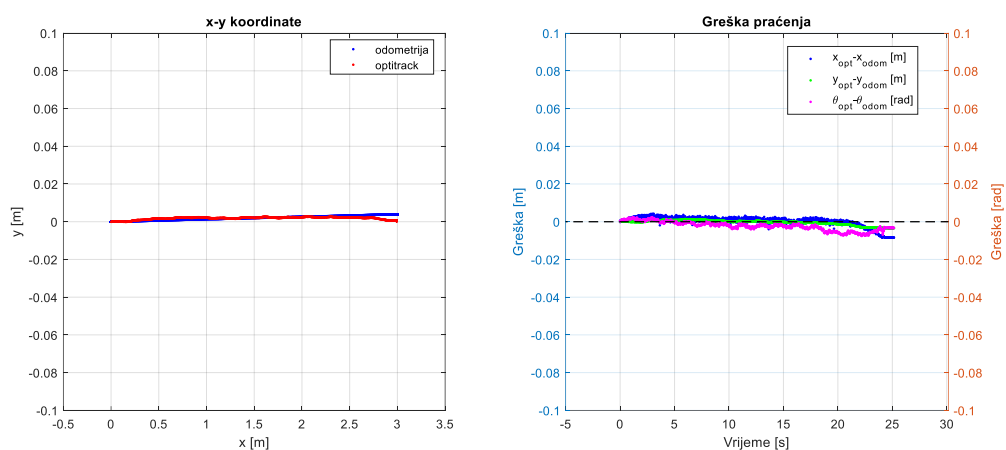
8.3. Validacija metodi na ravnoj putanji



Slika 50. Mjerenje nekalibriranog robota pri ravnoj kretnji



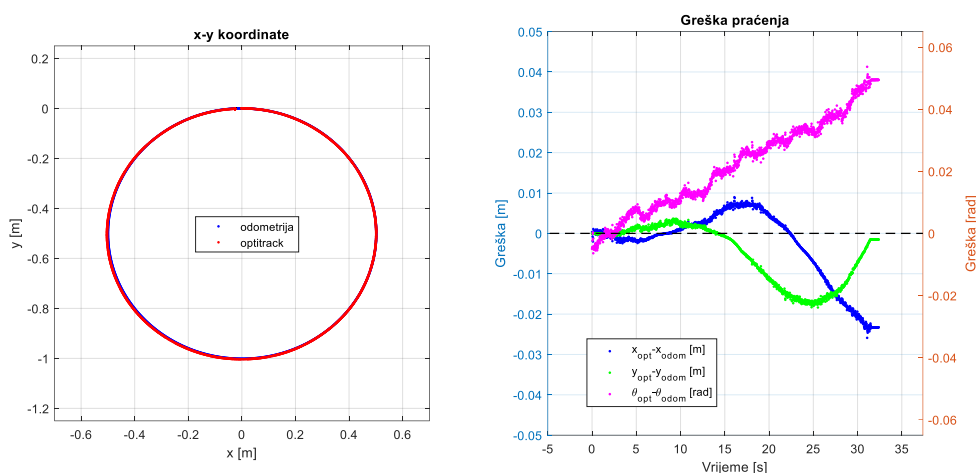
Slika 51. Mjerenje kalibriranog robota *UMBmark* metodom pri ravnoj kretnji



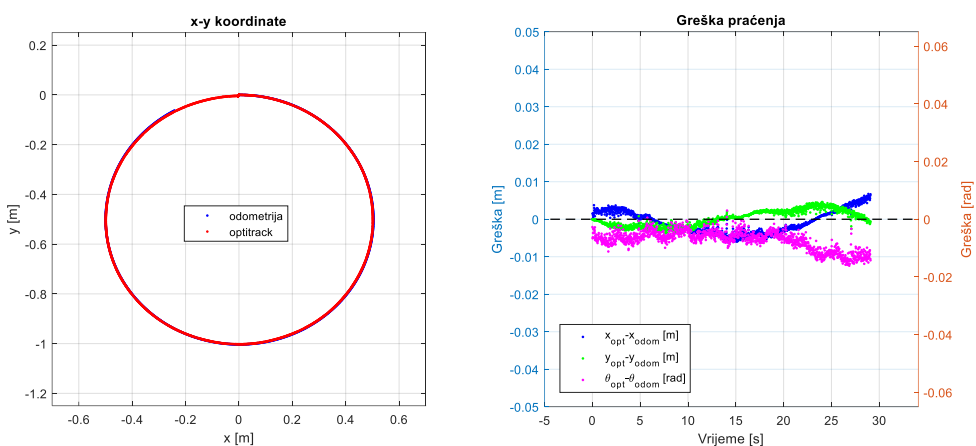
Slika 52. Mjerenje kalibriranog robota proširenom *UMBmark* metodom pri ravnoj kretnji

Iz priloženih grafova mjerenja zaključuje se da su obje kalibracijske metode dale znatno bolje rezultate u odnosi na nekalibriranog robota.

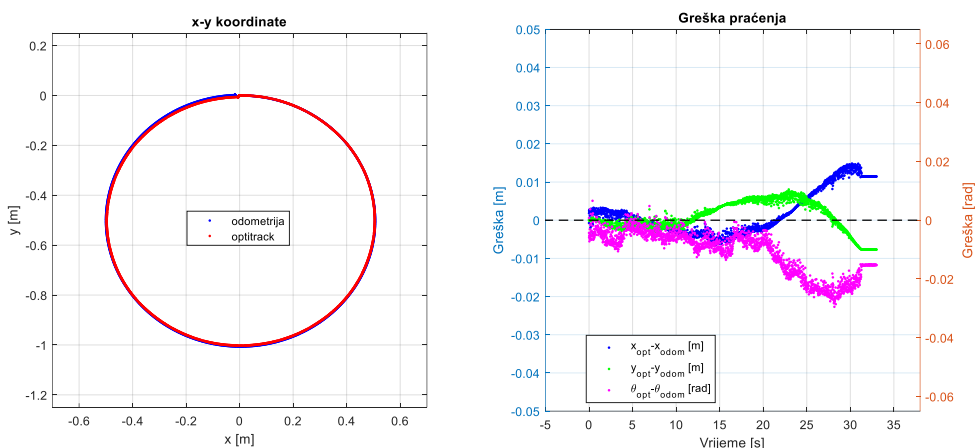
8.4. Validacija metodi na kružnoj putanji u smjeru kazaljke na satu



Slika 53. Mjerenje nekalibriranog robota pri kružnoj kretnji u smjeru kazaljke na satu



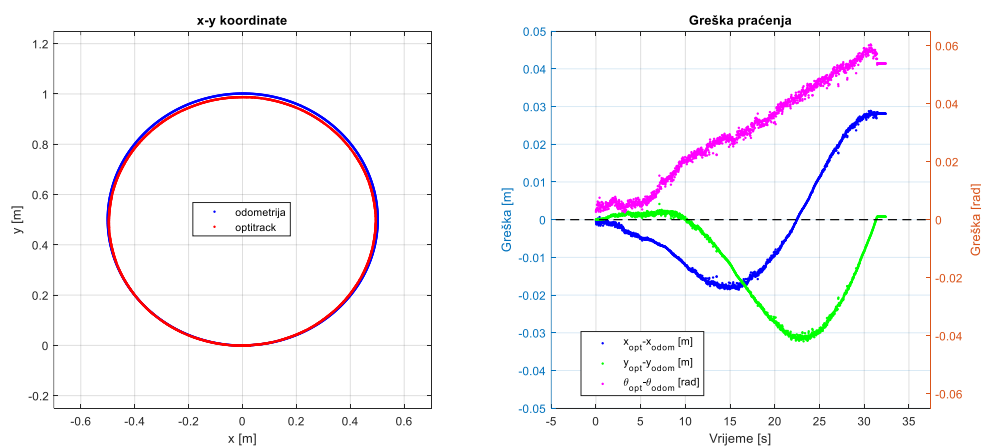
Slika 54. Mjerenje kalibriranog robota *UMBmark* metodom pri kružnoj kretnji u smjeru kazaljke na satu



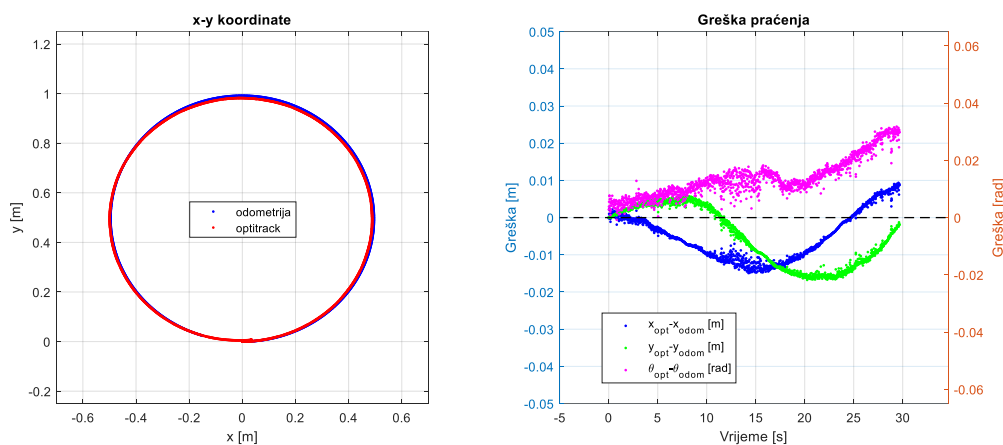
Slika 55. Mjerenje kalibriranog robota proširenom *UMBmark* metodom pri kružnoj kretnji u smjeru kazaljke na satu

Iz priloženih grafova mjerenja zaključuje se da su obje kalibracijske metode dale znatno bolje rezultate u odnosi na nekalibriranog robota.

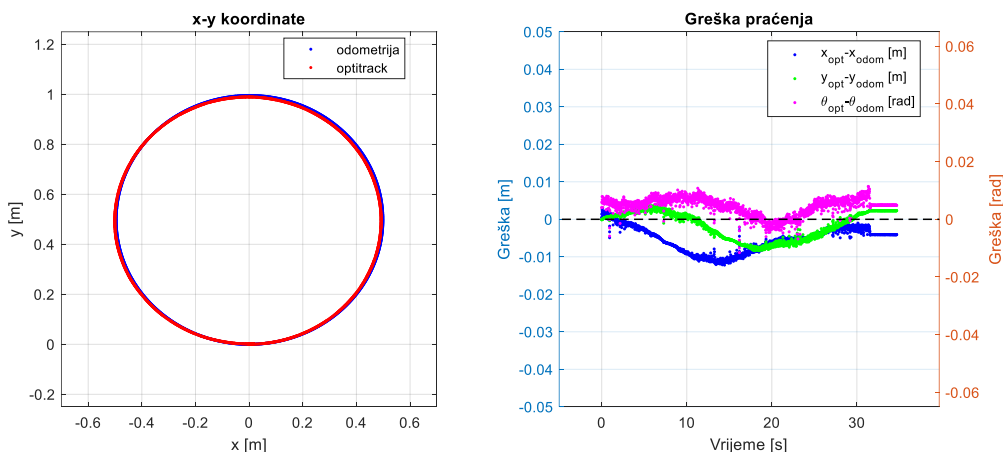
8.5. Validacija metodi na kružnoj putanji u suprotnom smjeru kazaljke na satu



Slika 56. Mjerenje nekalibriranog robota pri kružnoj kretnji u suprotnom smjeru kazaljke na satu



Slika 57. Mjerenje kalibriranog robota *UMBmark* metodom pri kružnoj kretnji u suprotnom smjeru kazaljke na satu



Slika 58. Mjerenje kalibriranog robota proširenom *UMBmark* metodom pri kružnoj kretnji u suprotnom smjeru kazaljke na satu

Iz priloženih grafova mjerenja zaključuje se da su obje kalibracijske metode dale znatno bolje rezultate u odnosi na nekalibriranog robota.

9. ZAKLJUČAK

Zaključak ovog završnog rada naglašava važnost kinematske kalibracije mobilnog robota za osiguravanje preciznih i sigurnih pokreta u prostoru. Kroz analizu i implementaciju *UMBmark* metode i proširene *UMBmark* metode, pokazano je kako pravilna kalibracija može značajno poboljšati estimaciju odometrije i smanjiti utjecaj sustavnih grešaka u kretanju robota. Korištenje ROS2 operativnog sustava pruža fleksibilnost i mogućnost za razvoj sofisticiranih algoritama upravljanja i kalibracije, dok OptiTrack sustav omogućuje precizno praćenje stvarne putanje robota u prostoru.

Kroz implementaciju kalibracijskih metoda, u ovom slučaju *UMBmark* metode, demonstrirana je praktična primjena teorijskih koncepta na stvarnom sustavu mobilnog robota ASTRO. Nakon provedene kalibracije i usporedbe stvarne i referentne putanje, vidljivo je poboljšanje u točnosti kretanja, potvrđujući važnost kalibracije u postizanju željenih performansi mobilnog robota.

U zaključku se također ističe važnost kontinuiranog istraživanja i razvoja u području robotike kako bi se omogućilo daljnje unaprjeđenje tehnologija i algoritama za preciznije i pouzdanije kretanje mobilnih robota.

LITERATURA

- [1] <https://www.space.com/scientists-test-submarines-saturn-moon-enceladus>, 25.12.2023.
- [2] https://store.dji.com/hr/event/valentine-sale-2024?from=store_homepage, 25.12.2023.
- [3] <https://mobilerobotguide.com>, 25.12.2023.
- [4] Siegwart R., EPFL, Nourbakhsh I. R. –Autonomous Mobile Robots; 2004
- [5] Siegwart R., Nourbakhsh I. R. – Introduction to Autonomous Mobile Robots; 2004
- [6] Mihel M, Bajd T, Ude A, Lenarčič J, Stanovnik A, Munih M, Rejc J, Šlajpah S. Robotics. Secnod Edition. Ljubljana: Springer; 2019.
- [7] <https://www.pjrc.com/store/teensy40.html>, 8.2.2024.
- [8] <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>, 8.2.2024.
- [9] <https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d435.html>, 8.2.2024.
- [10] <https://www.slamtec.ai/product/slamtec-rplidar-a1/>, 8.2.2024.
- [11] <https://www.adafruit.com/product/3463>, 8.2.2024.
- [12] <https://www.dfrobot.com/product-634.html>, 9.2.2024.
- [13] <https://www.pololu.com/product/2999/specs>, 9.2.2024.
- [14] <https://www.cui.com/product/dc-dc-converters/isolated/pqae50-t-series>, 9.2.2024.
- [15] <https://www.digikey.com/en/products/detail/cui-inc/PDQE20-Q24-S5-D/10230156>, 9.2.2024.
- [16] <https://www.makspower.hr/>, 9.2.2024.
- [17] <https://docs.ros.org/en/humble/index.html>, 10.11.2023.
- [18] <https://optitrack.com/>, 15.1.2024.
- [19] https://index.ros.org/r/vrpn_mocap/, 15.1.2024.
- [20] Borenstein, J., Feng, L.: UMBmark – A Method for Measuring, Comparing and Correcting Dead-reckoning Errors in Mobile Robots.
- [21] Changbae Jung and Woojin Chung, *Member, IEEE*: Accurate Calibration of Two Wheel Differential Mobile Robots by Using Experimental Heading Errors

PRILOZI

I. Programski kodovi-

https://github.com/ZvonimirKoprivnjak/Kalibracija_diferencijalnog_mobilnog_robota.git