

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Brigite Kerge 213090IAIB

**HOLONOOMILISE AUTONOOMSE ROBOTI IASO
NAVIGATSIOONISÜSTEEM**

Bakalaureusetöö

Juhendaja: Gert Kanter
PhD

Tallinn 2024

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Brigitte Kerge

25.05.2024

Annotatsioon

Iaso on robot, mille eesmärk on mõõta patsientide jämemotoorikat ehk salvestada liikumist. Robot võtab palju ruumi, seega on vaja, et Iaso suudaks liikuda optimaalsesse salvestuskohta ning tagasi algpunkti, kus ei takista kabineti või labori kasutamist. Robotil on puudu navigatsioonisüsteem, millega autonoomselt ning holonoomiliselt sõita algpunktist salvestamiseks sobivasse kohta ning tagasi.

Käesoleva töö raames on valminud navigatsioonisüsteem, mille abil on Iaso võimeline sõitma sihtpunkti holonoomiliselt ja autonoomselt. Kasutuses oleva madala vertikaalsagedusega lidari tõttu liigub robot aeglaselt ning ebahühtlaselt. Üks võimalus sõitu parandada oleks lisada juurde kiirema sagedusega lidar, mille abil Iaso ei peaks pimesi ehk ümbrust nägemata navigeerima. Teine valik oleks muuta olemasolevat navigatsioonisüsteemi nii, et robot sõidaks pimesi pikema vahemaa.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 6 peatükki, 13 joonist, 2 tabelit.

Abstract

Navigation System of Holonomic Autonomous Robot Iaso

Iaso is a robot that was developed for measuring patients' gross motor skills, for example in patients who have Parkinson's disease to check for changes in their movement as the disease progresses.

The robot takes a lot of space and should be able to move to a place that leaves enough usable room when measurements are not being taken. Iaso should be able to move from the sides or corners of the room it is placed in, to a position where filming can be optimal and then back to a place where it does not block movement in the room. The robot currently has no navigation system that would allow it to move autonomously and holonomically to a given goal position.

The goal of this thesis was to create a navigation system that would enable Iaso to move to goal points autonomously. The developed system allows the robot to move in a holonomic and autonomous way to goal points on a map. The main flaw of this solution is that the robot waits after moving a short distance, because of the low lidar that Iaso has. This could be fixed by adding another faster lidar to Iaso or solving the issue of blind driving deeper inside the code.

The thesis is written in Estonian and is 27 pages long, including 6 chapters, 13 figures and 2 tables.

Lühendite ja mõistete sõnastik

Holonoomiline robot	Robot saab liikuda igas suunas
LiDAR	Laserskaneerimisseade (<i>Light Detection and Ranging</i>)
Odomeetria	Positsiooni arvutus rataste pöörlemise põhjal
ROS	Roboti operatsioonisüsteem (<i>Robot Operating System</i>)
SDF	Roboti kirjelduse faili formaat (<i>Simulation Description Format</i>)
SLAM	Samaaegne lokaliseerimine ja kaardistamine (<i>Simultaneous Localization and Mapping</i>)
Teisendus	ROS2-s roboti eri osasid siduv lüli koordinaatteljestikus (<i>transform</i>)
URDF	Roboti kirjelduse faili formaat (<i>Unified Robotics Description Format</i>)
XML	faili formaat(<i>Extensible Markup Language</i>)

Sisukord

1	Sissejuhatus	9
1.1	Taust	9
1.2	Probleem ja eesmärgid	9
1.3	Ülevaade tööst	10
2	Metoodika	11
2.1	Sensorid	11
2.2	Holonoomiline liikumine	12
2.3	Robot Operating System 2	13
2.4	Navigatsioon	13
2.4.1	Käitumispuud	13
2.4.2	Navigatsiooniserverid	13
2.5	Lokaliseerimine ja kaardistamine	15
2.6	Visualiseerimine	16
3	Tulemused	17
3.1	Sensorid	17
3.2	Liikumine	17
3.3	Simulatsioon	18
3.4	Navigatsioon	21
3.5	Lokaliseerimine ja kaardistamine	22
3.6	Komponentide vaheline suhtlus	22
4	Valideerimine	24
4.1	Madala vertikaalsagedusega lidar	24
4.2	Liikumine	26
5	Tulemuste analüüs	29
5.1	Lidar	29
5.2	Navigatsioon	29
5.3	Probleemid tarkvaradega	29
5.4	Diskussioon	31
5.5	Edasiarendus	31
6	Kokkuvõte	32

Kasutatud kirjandus	33
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	35

Jooniste loetelu

1	YRL3-20 lidar [6]	11
2	YRL3-20 lidari skaneerimise nurk pealt- ja küljevaatest [6]	12
3	Mecanum ratas [3]	12
4	Iaso käsi lidariga	18
5	Iaso (a) päris ja (b) simulatsioonis	19
6	Väike ruudukujuline ruum (a), keskmise suurusega ristküliku kujuga ruum (b) ja suur ruudukujuline ruum (c).	20
7	(a) RViz-is ette antud sihtpunkt ning planeeritud tee punktini ja (c) Iaso Gazebo simulatsioonis samal ajal ning (b) sihtpunkti jõudnud Iaso RViz-is ja (d) Gazebos	21
8	Kontseptuaalne komponentide vaheline suhtlus sihtpunkti saades robotiga	23
9	Lidari sageduse seos ajakulu ja vahemaaga	25
10	Aja, läbitud vahemaa ja pöördenurga suuruse seos	26
11	Sihtpunkti liikumisel üleliigne pöördenurk 0.57 Hz lidari ja 10 Hz lidari puhul	27
12	Sihtpunkti sõitmine	28
13	Roboti kiiruse muutumine ajas 10 Hz lidari ning 0.57 Hz lidariga	28
14	Nav2 komponendid sihtpunkti saades	30

Tabelite loetelu

1	Globaalsete planeerijate võrdlus [13]	14
2	Stsenaariumite kordused	24

1. Sissejuhatus

1.1 Taust

Roboti Iaso eesmärk on mõõta patsientide jämemotoorikat, nt Parkinsoni tõvega patsientide puhul liikumise muutuste hindamiseks haiguse kulgemisel. Roboti kaamerad salvestavad objektiivselt patsiendi motoorika hilisemaks analüüsiks. Kuna arstide kabinettides on ruum piiratud, siis on vaja, et robot suudaks autonoomselt sõita patsiendi liikumise salvestamiseks optimaalsesse kohta ja pärast tagasi nurka või seina äärde, kus see ei piira ruumi kasutamist. Seega peab Iaso sõitma algasendist lähteasendisse ning reguleerima end vastavalt sensorite andmetele.

1.2 Probleem ja eesmärgid

Hetkel puudub robotil tarkvaralahendus, mis võimaldaks autonoomset liikumist. Oodatav tulemus on toimiv navigatsioonisüsteem, mis võimaldab robotil autonoomselt liikuda soovitud sihtpunkti.

Kuna kasutatav lidar (laserskaneerimisseade) on madala vertikaalsagedusega, on suur osa liikumisest sõltuv odomeetriast (positsiooni arvutus rataste pöörlemise põhjal). Lidari puhul tähendab sagedus kui pika aja tagant on lidaril uus skaneering ümbrusest. Odomeetria muutub ajaga ebatäpsemaks, kuna see on arvatud rataste pöörlemise järgi ning libisemise korral on liigutud vahemaa erinev arvatud vahemaast. Lidari andmete abil saab tekkinud viga parandada, võrreldes odomeetria asukohta skaneeringutega. Aeglase lidariga tekib ka probleem dünaamiliste takistuste (näiteks inimene kõnnib robotile ette) vältimiseks skaneeringute vahel liikumisel. Seega on vaja, et robot liiguks muutunud ruumi esitult oodates võimalikult vähe, samas piisavalt, et jõuda sihtpunkti mõistliku ajaga ja planeeritud rajalt välja sõitmata.

Lõputöö eesmärk on luua Iasole navigatsioonisüsteem, mis töötab robotil olemasoleva madala vertikaalsagedusega 3D lidariga ning mida kasutades suudab Iaso liikuda sihtpunkti autonoomselt ning holonoomiliselt liikudes. Robot on holonoomiline kui see suudab sõita igas suunas.

1.3 Ülevaade tööst

Eelneva projekti raames on Iaso riistvara kokku pandud ja ka patsiendi liikumise salvestamine on töötav. Autonoomse navigatsioonisüsteemi lisamiseks robotile on vaja kõigepealt leida ideaalsete tingimustega töötav lahendus, mida saab teha simulatsioonis. Simulatsiooni kasutamise eelis päris robotil katsetamisele on efektiivsus. Riistvara peal koheselt katsetamine on tülikam, kuna katsetada saab ainult koha peal, riistvara peab olema olemas, keskkond peab sobiv olema ja võib tekkida ohtlike olukordi vigasest koodist. Simulatsioonis töötavat lahendust saab seejärel katsetada päris robotil.

Lõputöö koosneb metoodikast, tulemustest, valideerimisest ning tulemuste analüüsist. Töö tulemsena valmis navigatsioonisüsteem, millega Iaso liigub autonoomselt ning holonoomiliselt algpunktist sihtpunkti.

2. Metoodika

Roboti juhtimiseks ja komponentide vaheliseks suhtluseks on kasutusel raamistik nimega Robot Operating System 2 (ROS2) [1]. Roboti navigeerimiseks on kasutusel ROS-i navigatsioonisüsteemi Nav2 [2]. Roboti ohutuks liikumiseks ja ümbuse kolmemõõtmelise esituse loomiseks on robot varustatud 3D lidariga. Robotil on Mecanum rattad [3], mis võimaldavad robotil liikuda igas suunas. Virtuaalse roboti testimiseks simulatsioonis on kasutusel Ignition Gazebo simulaator [4] ja RViz2 [5].

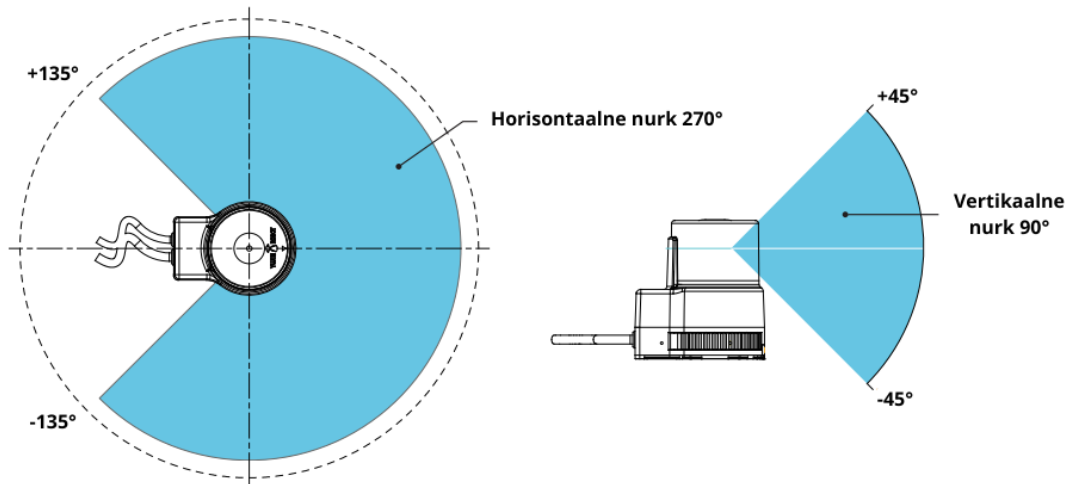
2.1 Sensorid

Iasol on varasematest etappidest olemas 3D lidar (laserskaneerimisseade), mis laserkiirega kauguste mõõtmise tulemusena moodustab seadme ümbusest kolmemõõtmelise esituse. 3D lidariga saab esituse ruumist madalama vertikaalsagedusega kui 2D lidariga, kuna laserkiir skaneerib horisontaalset vahemikku üles-alla. Saadud 3D esitus ruumist sisaldab rohkem andmepunkte kui 2D esitus, mille abil on võimalik eri kõrgustel asuvate takistuste tuvastamine. Kasutusel on YUJIN YRL3-20 (joonis 1) 3D lidar, mille vertikaalne skaneerimissagedus on 0.57 Hz ehk ligikaudu 1.75 sekundi vahega. Lidari vaatenurk on horisontaalselt 270°(mõlemale poole 135°), vertikaalne nurk on 90°(üles ja alla 45°) (joonis 2) ning ulatus on 20 meetrit.



Joonis 1. YRL3-20 lidar [6]

Odomeetria on võte, mille abil on võimalik robotil oma positsiooni arvutada pöördkoodritelt saadud rataste pöörlemise põhjal. Odomeetria probleem seisneb selles, et mida rohkem robot liigub, seda ebatäpsemaks see muutub. See juhtub kuna sensorid võivad olla ebatäpsed ning rattad võivad libiseda või kinni jääda, mis põhjustab erinevusi läbitud



Joonis 2. YRL3-20 lidari skaneerimise nurk pealt- ja küljevaatest [6]

vahemaa ja rataste liikumisest arvutatud vahemaa vahel. Iaso kasutab odomeetriat roboti ligikaudse asukoha hindamiseks ja parandab seda asukohta kasutades lidari andmeid, mis võimaldavad täpsemalt mõõta ümbritsevaid objekte ja nende kaugust ning muutusi võrreldes eelmise skaneeringuga.

2.2 Holonoomiline liikumine

Iasol on Mecanum rattad (joonis 3), kuna nendega saab liikuda holonoomiliselt. See tähendab, et robot saab liikuda igas suunas. Mecanum ratas koosneb 45° nurga all kinnitatud rullikutest. Kui kasutatakse nelja Mecanum ratast koos, saab rataste pöörlemissuunda ja -kiirust muutes on võimalik kokku saada liikumisvektori igas suunas [3]. Seega on võimalik liikuda paremale, vasakule, diagonaalselt ja on ka võimalik koha peal ümber pöörata. Kuna Iaso on mõeldud piiratud ruumiga kabinetis liikumiseks, on igas suunas liikumise võime oluline.



Joonis 3. Mecanum ratas [3]

Lisaks Mecanum ratastele on holonoomilised ka omnidireksionaalsed rattad, mille puhul rullikud on paigutatud risti ratta pöörlemise suunaga ja rattad on roboti külge kinnitatud nurkadesse risti roboti diagonaaliga. Mecanum rataste eelis omnirataste ees on parem veojõud ning lihtsam ülesseadmine. Projektis on kasutusel Mecanum rattad lisaks mainitud eelistele ka kuna Iaso riistvara on valminud eelnevate etappidega.

2.3 Robot Operating System 2

Robot Operating System 2 (ROS2) on avatud lähtekoodiga teekide ja tööriistade komplekt. ROS2 on uuendatud versioon ROS1-st, mille viimane versioon on kasutatav kuni 2025 aastani, seega on soovitatav kasutada ROS2-te [7]. ROS-i abil on võimalik panna roboti erinevad komponendid koos töötama. Iaso kasutab ROS jaotust (distribution) 2022 Humble Hawksbill [8], mis hetkel on kõige pikema toetusajaga versioon.

2.4 Navigatsioon

Navigation2 (Nav2) on ROS2 jaoks loodud navigatsiooni raamistik, mis võimaldab luua autonoomseid süsteeme, pakkudes dünaamilist tee planeerimist, takistuste vältimist ning arvutades sihtpunktini liikumiseks vajalikke mootorite kiiruseid [9]. Nav2-e alternatiivina saaks kasutada Nav1-te, mis on ROS1-e navigatsioonisüsteem. Selle puhul on vaja kasutada ka ROS1-te ROS2-ega siduvat silda, mis lisab ebavajalikku keerukust. Võimalik on ka terve süsteem ise teha, mis aga lõputöö ajaressursiga ei ole mõistlik. Veel on olemas Autoware Universe [10], mis on loodud isejuhtivatele sõidukitele, kuid sellel on palju liigseid osasid, mis antud projekti puhul vajalikud pole. Seega on Nav2 kõige mõistlikum ning kompaktsem valik Iaso navigatsioonisüsteemi tegemisel.

2.4.1 Käitumispuud

Käitumispuud (Behaviour trees) loovad hierarhilisi struktuure, mis hõlmavad erinevaid navigatsioonikäitumisi, näiteks takistuste vältimist, sihtpunkti otsimist ja dokkimist. See võimaldab robotil paindlikult reageerida erinevatele keskkonnatingimustele ja ülesannetele [11], [12].

2.4.2 Navigatsiooniserverid

Planeerijad ja kontrollerid on navigatsiooni põhilised osad. Taastamiskäitumise abil on võimalik reageerida dünaamilistele takistustele ning teelt kaldumisele. Siluja peamine eesmärk on planeeritud teekonna sujuvamaks ning sirgemaks muutmine [11].

Globaalse raja planeerija

Globaalse raja planeerija (Planner) peamine ülesanne on arvutada välja kehtiv ja optimaalne tee algsest asukohast sihtpunktini [11]. Nav2 dokumentatsiooni [11] kohaselt on olemasolevad planeerijad NavFn, Theta Star, Smac 2D, Smac Hybrid-A* ja Smac State Lattice. Antud projekti puhul on oluline, et robot liiguks holonoomiliselt ning kuna robot on kandiline, on ka oluline, et planeerija suudaks arvestada roboti mitteringikujulise kujuga.

Nendele tingimustele vastav planeerija on Smac State Lattice planeerija, mis toetab kandilise kujuga holonoomilisi roboteid. Lisaks on see üks kahest kõige kiiremast ning lühima planeeritud teega planeerijast (tabel 1). Nav2 autorid soovivad kandilise holonoomilise roboti jaoks kasutada Smac State Lattice globaalset planeerijat [13].

Projektis jäi kasutusse NavFn planeerija, kuna Smac State Lattice korrektselt töötamiseks vajalike parameetrite määramise ajakulu oli suur ning NavFn planeerijaga oli planeeritud tee sobiv. Kuigi NavFn ei ole loodud kandilistele robotile, on seda võimalik kasutada muutes roboti jalajälg roboti kuju järgi.

Tabel 1. Globaalsete planeerijate võrdlus [13]

Planeerija	Planeerimisaeg (ms)	Tee pikkus (m)	Mitteringikujuline	Omni
NavFn	61.02	52.25	-	+
Lazy Theta*-P	94.42	50.28	-	+
Smac 2D-A*	88.82	49.65	-	+
Smac Hybrid-A*	38.77	50.78	+	-
Smac State Lattice	39.40	50.51	+	+

Lokaalse trajektoori planeerija

Lokaalse trajektoori planeerija (Controller) peamine ülesanne on arvutada positsiooni järgi juhtsignaal, mille abil robot järgib globaalselt arvutatud teed või täidab väiksemat ülesannet (nt dokkimine) [11]. Dokumentatsiooni [11] põhjal on olemasolevad planeerijad DWB Controller (eelkäija on Dynamic Window Approach), Timed Elastic Band kontrolleri (TEB), Regulated Pure Pursuit, Model Predictive Path Integral kontrolleri (MPPI), Rotation Shim kontrolleri ja Graceful kontrolleri. Nav2 autorid soovivad kandilise holonoomilise roboti jaoks kasutada MPPI lokaalset planeerijat [13], mis on ka lõputöös valitud lokaalne planeerija. Katsetatud sai ka DWB Controller, millega robot ei liikunud optimaalselt.

Taastamiskäitumine

Taastamiskäitumine (Recovery behaviour) on vajalik liikumisel tekkivate probleemide vältimiseks. Nende eesmärk on tegeleda süsteemi tundmatute või tõrketingimustega ja

nendega iseseisvalt toime tulla. Sellised olukorrad on näiteks kinni jäämine ja ootamatud takistused. [11]

Siluja

Kuna planeerija otsitava tee optimaalsuse kriteeriumid on tavaliselt tegelikkusega võrreldes vähendatud, on sageli kasulik tee täiendamine. Sel eesmärgil on Nav2 olemas silujad (Smoother), mis tavaliselt vastutavad tee ebatasasuste vähendamise ja järskude pöörete tasandamise eest, aga ka kauguse suurendamise eest takistustest. [11]

2.5 Lokaliseerimine ja kaardistamine

Samaaegne lokaliseerimine ja kaardistamine (SLAM) on meetod korrakaardi moodustamiseks ja roboti lokaliseerimiseks. SLAM-i algoritmid lubavad sõidukil kaardistada teadmata keskkonda. Saadud kaarti saab kasutada tee planeerimiseks ja takistuste vältimiseks [14].

Varasemalt eksisteerivatest laser SLAM algoritmidest, mis töötavad ROS-iga, on GMapping, Karto, Cartographer ja Hector. Nendest vähesed suudavad kaardistada suuri ruume ja veel vähemad suudavad seda teha reaajas tänapäevaste robotite süsteemides leiduvate protsessoritega. Ainuke, mis suudab, on Cartographer, mida aga pole uuendatud 2021. aastast [15]. Slam Toolbox loodi just nende probleemide lahendamiseks. Slam Toolbox moodustab 2D kaardi ruumist kui ühendab laserilt saadud andmepunktid ning odomeetria ja roboti vahelise lüli koordinaatteljestiku teisenduse (transform) [16].

Kuigi Iasol olev lidar on 3D lidar, ei ole mõistlik Iaso puhul kasutada 3D SLAM-i, kuna 3D SLAM-i algoritmid on arvutusmahukad ja vajavad võimsamat riistvara kui Iasol on. Seega on projektis kasutusel Slam Toolbox.

Slam Toolbox vajab töötamiseks teisendust odomeetria ja roboti vahel ning selle saamiseks kasutab Iaso Extended Kalman Filter (EKF) implementatsiooni Robot Localization [17] EKF sõlm-e. See kasutab omnidirektsionaalset liikumismudelit, et ennustada roboti asukohta ja korrigeerib seda prognoositud hinnangut sensorite abil [18]. Iaso puhul kasutab EKF asukoha ennustuse parandamiseks rataste pöörlemisest arvatud odomeetria ja juhtsignaali antud kiiruseid.

2.6 Visualiseerimine

Ignition Gazebo, mis lõputöö raames on kasutuses Iaso simuleerimiseks, on tarkvara, mis töötab koos ROS2-ga. See võimaldab simuleerida füüsikalisi jõudusid ning luua detailseid 3D mudeleid keskkondadest, kus robot tegutseda saab. See on oluline vahend robotite testimiseks ja arendamiseks enne nende pärismaailma rakendamist.

RViz2 on teine oluline tööriist, mida kasutatakse robotite arendamisel ja testimisel. See on 3D visualiseerija, mis on loodud töötamiseks ROS raamistikuga. RViz võimaldab visualiseerida roboti sensorite andmeid, ruumi kaarti ning roboti asukohta kaardil. RVizi abil saab näha, mida robot "näeb" ja "mõistab" oma keskkonnast. See on oluline vahend robotite testimiseks, aidates tuvastada ja lahendada probleeme robotite tajumises ja navigeerimises.

Roboti visualiseerimiseks nii RVizis kui ka Gazebos kasutatakse XML formaadiga URDF (Unified Robot Description Format) faili, kus saab siduda ära roboti erinevad osad ja määrata nendele dünaamilisi omadusi, näiteks massi ja inertsmomente, ning visuaalseid omadusi. Kuigi Gazebo toetab ka URDF faile, on selle peamine failitüüp robotite kirjeldamiseks SDF (Simulation Description Format). Projektis on kasutusel URDF failid, kuna nendega töötab ROS2-ga paremini kui SDF korral. Gazebo võimaldab maailma kujundamist roboti liikumise ja keskkonna simuleerimiseks, kasutades SDF faile.

3. Tulemused

Lõputöö tulemuseks on töötav navigatsioonisüsteem, mille abil robot liigub kaardi peal sihtpunktiks määratud punkti ning tagasi ruumi nurka.

3.1 Sensorid

Päris robotil on kasutuses madala vertikaalsagedusega (0.57 Hz) 3D lidar (YUJIN YRL3-20) seega saab Iaso uuendatud esituse ruumist ligikaudu iga 1.7 sekundi tagant. Sellise sagedusega on navigeerimine keeruline, sest robot peab liikuma pimesi ning sel juhul võib robot liikuda planeeritud rajalt välja või ka sõita otsa dünaamiline takistusele. Juhul kui ilma uue ruumi esituseta sõitmise ajal peaks miski liikuma roboti ette, tekib oht, et robot ei jää enne seisma, kui on hilja.

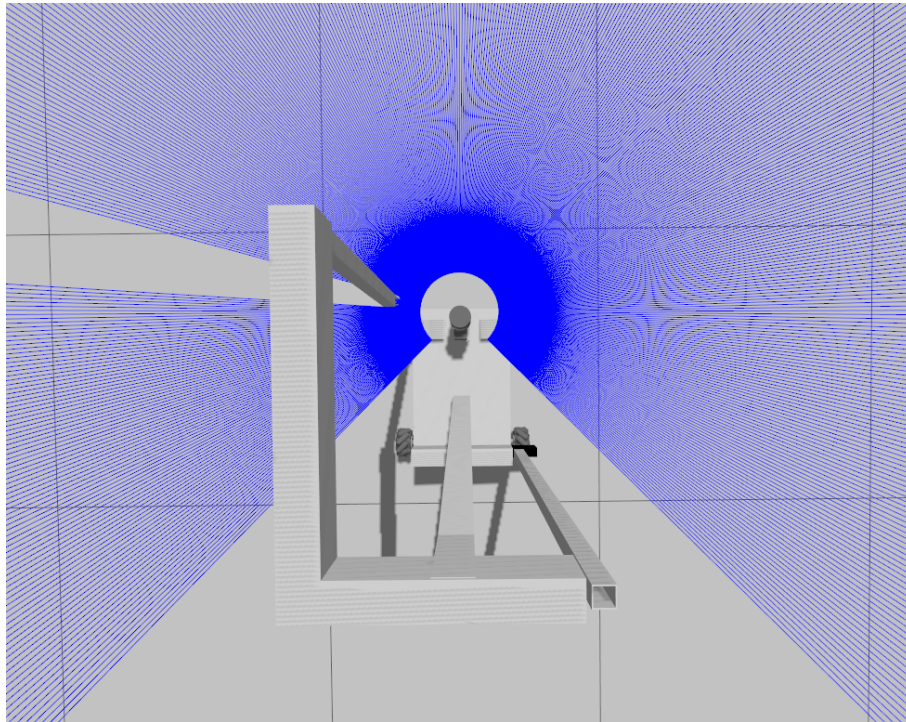
Lokaliseerimiseks kasutab Iaso SLAM-i, mis võrdleb lidarilt saadud skaneeringut eelnevatega ja leiab muutuste põhjal uue asukoha. Liiga suure erinevusega skaneeringute korral on SLAM-il raskem asukohta õigesti määrata, seega aeglase lidari korral on oluline liikuda piisavalt väike vahemaa, et SLAM töötaks kuid ka piisavalt pikk maa, et jõuda sihtpunktini mõistliku ajaga. Simulatsioonis sõidab Iaso lidari tõttu aeglaselt ja ebasujuvalt, kuid siiski jõuab õigesse kohta.

3.2 Liikumine

Simulaatoris liigub Iaso holonoomiliselt, kuid mitte füüsikapõhiselt. Iaso Mecanum rataste simulatsioonis liikuma saamisega oli probleeme ratastele õigete füüsikaliste omaduste lisamisega, kuna Gazebo kasutab SDF failitüüpi ja ROS2 URDF tüüpi. Mecanum rataste puhul on oluline, et rataste liikumisel tekiks kokku õige jõusuund, URDF formaadist SDF formaadile Gazebo poolt konverteerimisega jääb aga üks vajalik realistliku füüsika väli puudu. Töötavate SDF failide järgi oli võimalik leida parameeter, mis muutis Mecanum rattad õigesti toimivaks, kuid sellist parameetrit ega võrdväärset pole URDF formaadis olemas.

Antud projekti juhul oli olulisem, et robot saab kasutada ROS2 meetodeid, kui et rattad robotit simulatsioonis õigesti liiguvad. Seega Ignition Gazebo-s ringi liikudes Iaso rattad ei pöörle, vaid uus asukoht on arvutatud juhtsignaaliga antud rataste kiiruste järgi lõputöös valminud kontrolleri abiga.

Kuna Iasol on kaamerategega käed, mis mitte kasutuses olles on all asendis, on need lidari skaneeritavas alas ees. Joonisel 4 on näha sinisega lidari kiired. Vasakul, kus käsi on all, on näha vahemikku, kus laserkiir pörkab tagasi käe juures. See põhjustab probleeme navigeerimises, kuna robotile tundub see kui lähedal asuv takistus, mida peab vältima. Vea parandamiseks peab enne lidari andmete komponentile jagamist eemaldada selles vahemikus olevad andmed. Simulatsioonis on võimalik probleem lahendada objektilt kokkupõrkemudeli eemaldamisega, muutes objekti lidarile nähtamatuks.



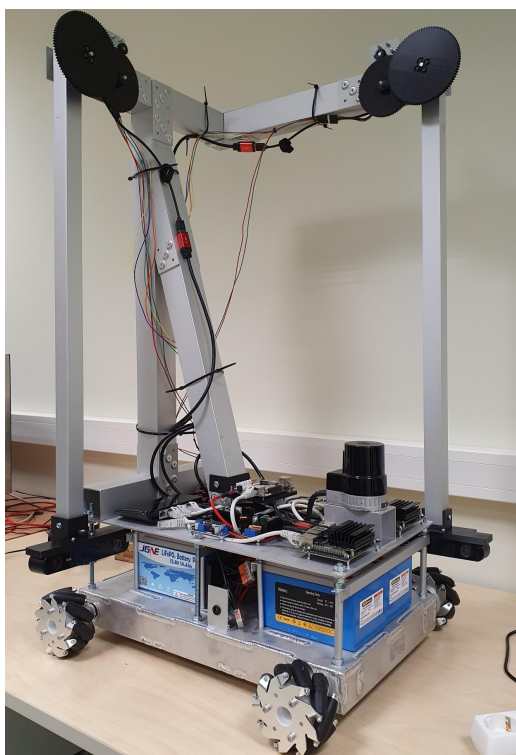
Joonis 4. Iaso käsi lidariga

3.3 Simulatsioon

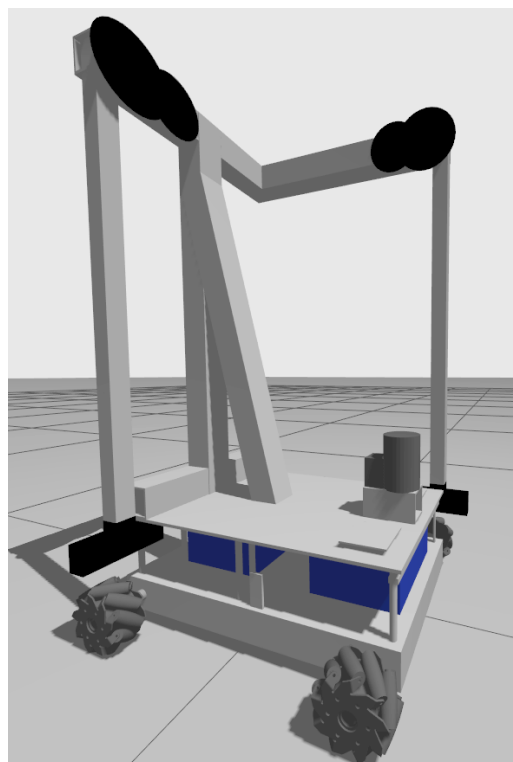
Simulatsiooni kasutamisel on palju eeliseid, näiteks lihtsus ja efektiivsus. Simulatsioonis saab katsetada roboti tehnilist lahendust ideaalsete tingimustega ning erinevates keskkondades, peale mida on ohutum ja aega säästvam päris robotil tarkvara käivitada. Lõputöö raames, kus eesmärgiks on töötav navigatsioonisüsteem Iasole, mis peab liikuma piiratud alaga ruumis, oli vaja teha roboti mudel ning ruum ka simulatsioonis.

Roboti mudeli töötamiseks simulatsioonis on Iasost loodud URDF fail, kus on kirjeldatud roboti lülid ning nende füüsilised omadused ja seosed teiste lülidega. Ignition Gazebo kasutamiseks on lõputööga valminud roboti 3D mudel, kus on lülidele pandud modelleeritud kujud. Peamiselt on Iaso mudeli eri osad kujundatud päris roboti järgi kasutades 3D modelleerimise tarkvara Blender [19]. Rataste ja võllide mudelid on saadud Nexus 4WD Mecanum Simulator projektist [20]. Pildil 5a on näha päris Iaso võrdluses simulatsiooni

mudeliga 5b.



(a)



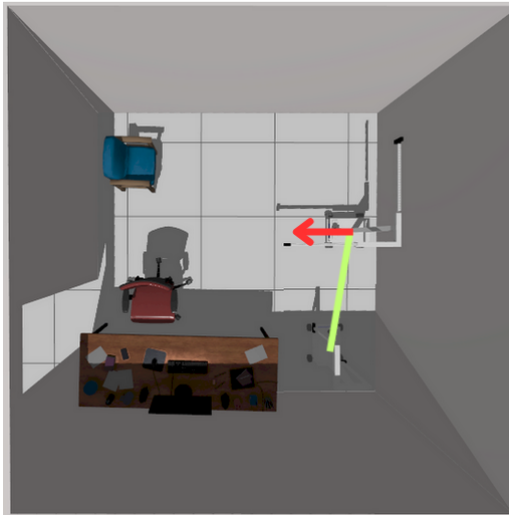
(b)

Joonis 5. Iaso (a) päris ja (b) simulatsioonis

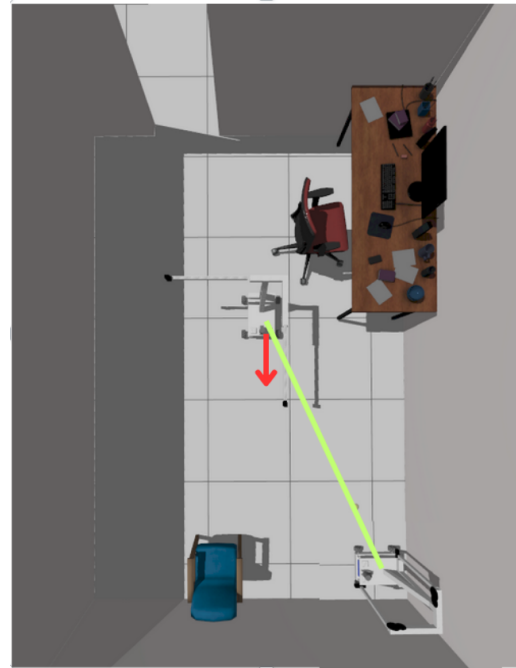
Lisaks roboti mudelile saab gazebos luua erinevaid keskkondi ning nendes roboti liigutada. Lõputöös on tehtud kolm erineva suuruse ja paigutusega ruumi. Nendest väikseim (joonis 6a) on mõõtmetega 4x4 meetrit, keskmine (joonis 6b) 4x6 meetrit ning suurim (joonis 6c) 7x7 meetrit. Joonistel 6 on Iaso algpunktis ning sihtpunktis, roheline joon näitab trajektoori ning punane nool näitab sihtpunkti suunda.

Üks eelis simulatsioonis katsetamisel on, et roboti käitumist saab katsetada erinevates ruumides ja erinevate keskkonnatingimustega, mis teeb kogu protsessi lihtsamaks, kiiremaks ja ka ohutumaks kui päris robotil kohe katsetamine. Simulatsiooni tööle pannes on võimalik valida ruum, milles virtuaalne Iaso mudel liikuma hakkab.

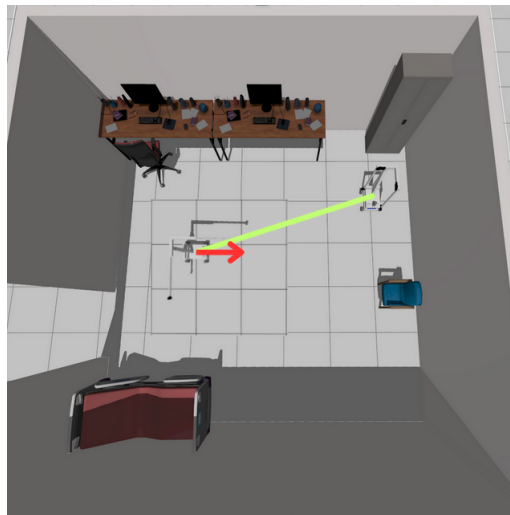
Kuigi Ignition Gazebo on veel pikeima toetusega versioon Gazebost, on sellega koos töötavat tarkvara keeruline leida. Probleeme tekitas paar aastat tagasi tehtud nime muutus, mille tõttu näiteid Ignition Gazeboga töötavatest lahendustest on vähe.



(a)



(b)



(c)

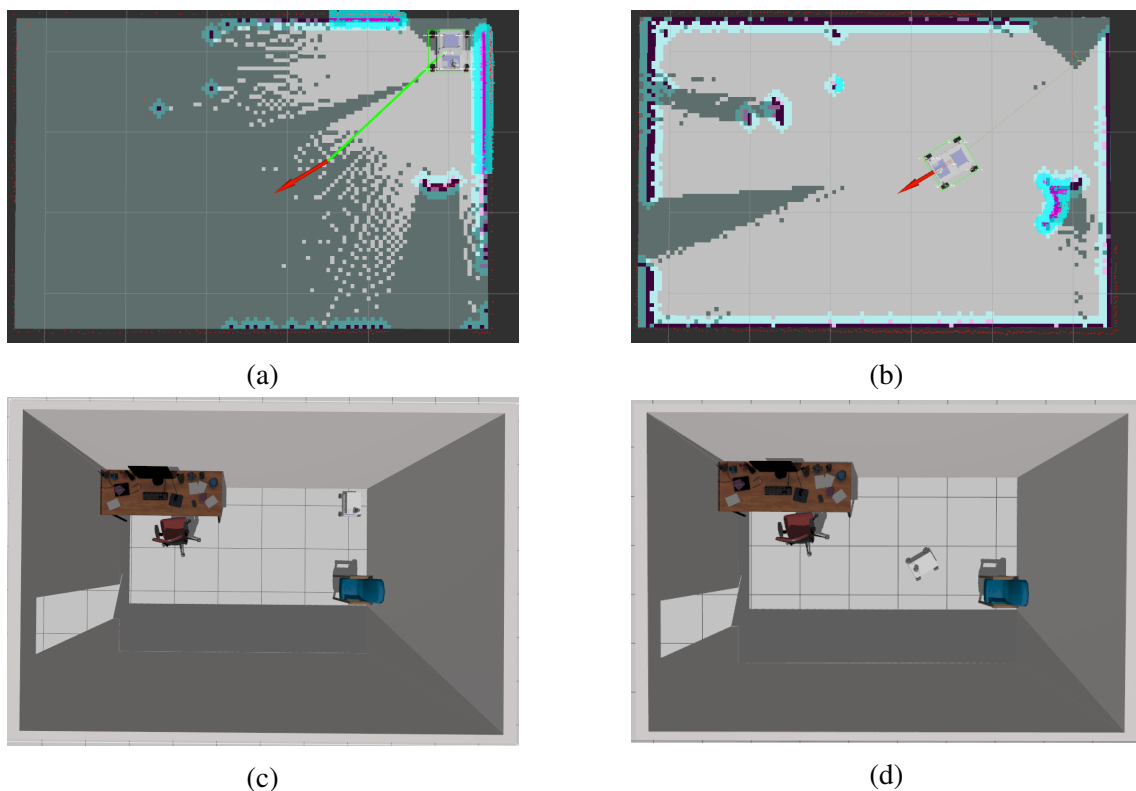
Joonis 6. Väike ruudukujuline ruum (a), keskmise suurusega ristküliku kujuga ruum (b) ja suur ruudukujuline ruum (c).

3.4 Navigatsioon

Nav2-e lisamisel Iasole sai proovitud erinevaid navigatsiooni algoritme, nende hulgas Smac State Lattice ja NavFn planeerijad ning DWB ja MPPI kontrollid. Algselt oli kasutusel DWB koos NavFn-ga, mis ei andnud soovitud tulemust, kuid DWB MPPI vastu vahetades oli planeeritud rada korrektne ning robot sõidab sihtpunkt.

Soovitav kombinatsioon loojate poolt [13] holonoomilisele kandilisele robotile on MPPI ja Smac State Lattice. Kuid peale pikka parameetrite muutmist, ei saanud lõputöö raames tööle Smac State Lattice nii hästi kui töötas NavFn. Seega on valminud robotil navigeerimiseks kasutusel MPPI ja NavFn.

Joonisel 7a on näha Iaso ruumis 6b, kus roheline joon on planeeritud rada ning punane nool on sihtpunkt koos suunaga. Aktiivselt nähtavad seinad ja takistused on märgitud lillaga, mille ümber on sinine joon. Pildil 7c on Iaso enne sihtpunkti saamist Ignition Gazeno simulatsioonis. Joonisel 7b on näha, kuidas Iaso on saanud sihtpunkti RViz-is ja joonisel 7d samal ajal Ignition Gazebo-s. Mõlemal joonisel on helehall ala tuvastatud vaba ruum ning tumedam ala kaardistamata osa.



Joonis 7. (a) RViz-is ette antud sihtpunkt ning planeeritud tee punktini ja (c) Iaso Gazebo simulatsioonis samal ajal ning (b) sihtpunkti jõudnud Iaso RViz-is ja (d) Gazebos

3.5 Lokaliseerimine ja kaardistamine

Slam Toolbox täpsustab odomeetriast arvatud positsiooni lidariga võrreldes samal ajal ka eelnevaid skaneeringuid. Slam Toolbox parandab asukoha ainult lidarist andmete saamisel, seega töötab see sama sagedusega, mis lidar ise. Iaso aeglase lidari tõttu on roboti liikumine ebasujuv ning aeglane, kuna kiiremini või pikemalt sõites on kahju oht suurem kui kasu.

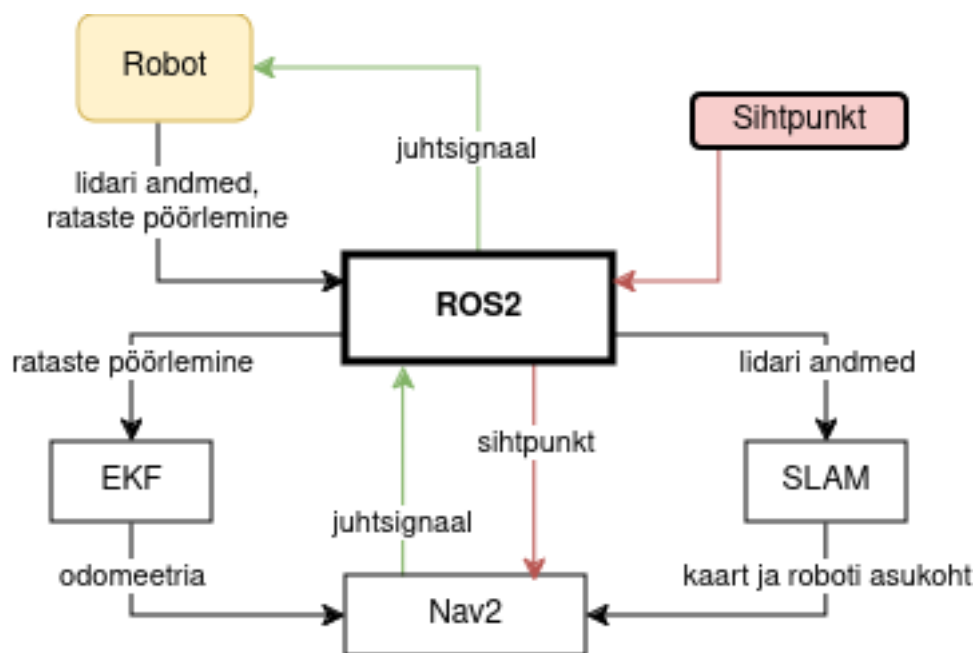
Lõputöös tehtud navigatsioonisüsteem arvestab lidari madala vertikaalsagedusega ning Iaso jääb peale lühikese vahemaa liikumist ootama uut skaneeringut. Siiski leiab Iaso enda asukoha kaardil ja ka kaardistab toa õigest. Probleeme võib tekkida kui robotile käsitsi anda suuremad kiirused kui seda teeb navigatsioonisüsteem, kuna siis on SLAM-il raske kaasas püsida ning seinad võivad algse kaardiga võrreldes paigast liikuda.

Kuna lidari tulemust oodates ei uuenda SLAM ka kaardi ning odomeetria vahelist teistendust koordinaatteljestikus, sai lõputöös proovitud ka teisenduse käsitsi uuendamine, mis aga ei teinud roboti liikuvust paremaks.

3.6 Komponentide vaheline suhtlus

Selleks, et Iaso jõuaks algpunktist sihtpunkti toimub suhtlus kõigi komponentide vahel. ROS, mis on peamiselt andmete vahendamise kiht, saab robotilt lidari andmed ja rataste pöörlemise. EKF sõlm arvutab odomeetria rataste pöörlemise põhjal. ROS-i kaudu jõuavad lidari andmed Slam Toolbox-ile, mis kaardistab skaneeringu ning odomeetria ühendamisega ruumi ja leiab roboti asukoha kaardil.

Saadess uue sihtpunkti, leiab Nav2 eelnevalt seadistatud algoritmi(de)ga globaalse tee punktini ja hakkab robotile saatma kiiruseid, mis on vajalikud planeeritud rada mööda sõitmiseks. Juhtsignaali järgi liigub robot edasi mootoritele sobiliku kiiruse määrates ja seega muutub ka odomeetria ning roboti asukoht SLAM-i abil. Kui robot kaldub teest kõrvale või rajale tekib takistus, toimub ka planeerimine uuesti ja rada võib muutuda (joonis 8).



Joonis 8. Kontseptuaalne komponentide vaheline suhtlus sihtpunkti saades robotiga

4. Valideerimine

Tulemuste valideerimiseks sai kirjutatud Pythoni skript, mis andis navigatsioonile erinevad sihtpunktid erinevates ruumides ning salvestas tulemused. Lidari sageduse mõju navigatsiooni efektiivsusele (tabelis 2 stsenaarium nr 1) hindamiseks salvestas skript läbitud vahemaa ning ajakulu. Katsetatud sai 4 punkti jõudmine, kuhu lidari sagedustega 0.57 Hz, 0.8 Hz, 1 Hz, 2Hz ja 10 Hz robotid sõitsid 5 korda ehk kokku 100 sõitu.

Sihtpunkti jõudmiseks kuluva aja ja vahemaa ning orientatsiooni seoste (tabelis 2 stsenaarium nr 2) mõõtmiseks salvestus lisaks veel algse suuna ja lõppsuuna vahe. 0.57 Hz lidariga robot sõitis kolme erinevasse punkti kolme erineva suunaga (0° , 90° , 180°) ja igasse 15 korda ehk kokku 135 sõitu.

Roboti liikumise efektiivsuse (tabelis 2 stsenaarium nr 3) hindamiseks salvestus ka roboti pööramine algusest lõpuni, ehk kui palju Iaso suund muutus liikumise jooksul ning kui suur vahe oli vajaliku pöördnurga ja tegeliku pööramise vahel. Robot sõitis kolme eri punkti, igas 8 eri suunaga (45° vahega) ja igasse 10 korda ehk kokku 240 sõitu.

Roboti liikumiskiiruse muutumise (tabelis 2 stsenaarium nr 4) hindamiseks ajas salvestus ka roboti kiirus algusest lõpuni. Robot sõitis 0.57 Hz lidariga ning 10 Hz lidariga ühte punkti ühe korra.

Igal sõidul oli ka salvestatud, kas robot jõudis sihtpunkti. Kõik katsete sõidud lõppesid õiges punktis.

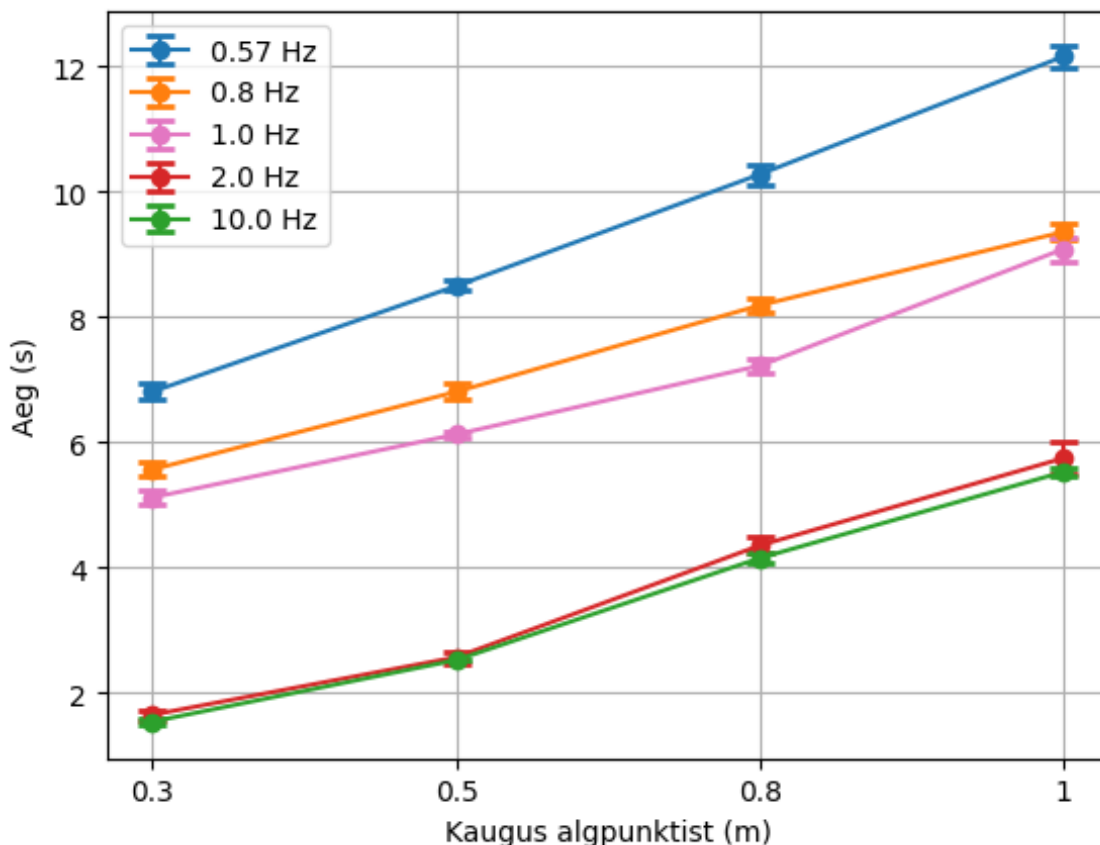
Tabel 2. Stsenaariumite kordused

Stsenaarium	Lidareid	Sihtpunkte	Nurkasid punktis	Korduste arv	Kokku
1	5	4	1	5	100
2	1	3	3	15	135
3	2	3	8	10	240
4	2	1	1	1	2

4.1 Madala vertikaalsagedusega lidar

Lõputööga valminud navigatsioonisüsteemiga, mille eesmärk on töötada madala vertikaalsagedusega lidariga, on Iaso liikumine ebasujuv ning aeglane. Üks lahendus oleks lisada Iasole suurema sagedusega lidar. Joonisel 9 on tulemused erinevate sagedustega lidaritega

navigeerimisest, kus lidariga vahemaa läbimiseks kuluva aja näitamiseks on punktid keskmiste tulemustega ning kõige suuremad ja väiksemad ajakulud näha vearibana punkti kohal ja all. Jooned ühendavad vahemaade läbimiseks kulunud keskmised ajad vastavalt lidari sagedusele.



Joonis 9. Lidari sageduse seos ajakulu ja vahemaaga

Jooniselt on näha, et 0,57 Hz lidari puhul on ajakulu kordades suurem kiiremate lidaritega võrreldes. Kõigest 0.43 Hz kiirema lidariga sõidaks iaso juba paar sekundit kiiremini. Lisaks on näha, et juba 2 Hz puhul on roboti kiirus ligikaudu võrdne 10 Hz lidariga, mis tuleneb sellest, et robot suudab madala vertikaalsagedusega lidariga roboti liikumiseks loodud navigatsioonisüsteemiga sõita sujuvalt ilma pausideta juba 2 Hz lidariga.

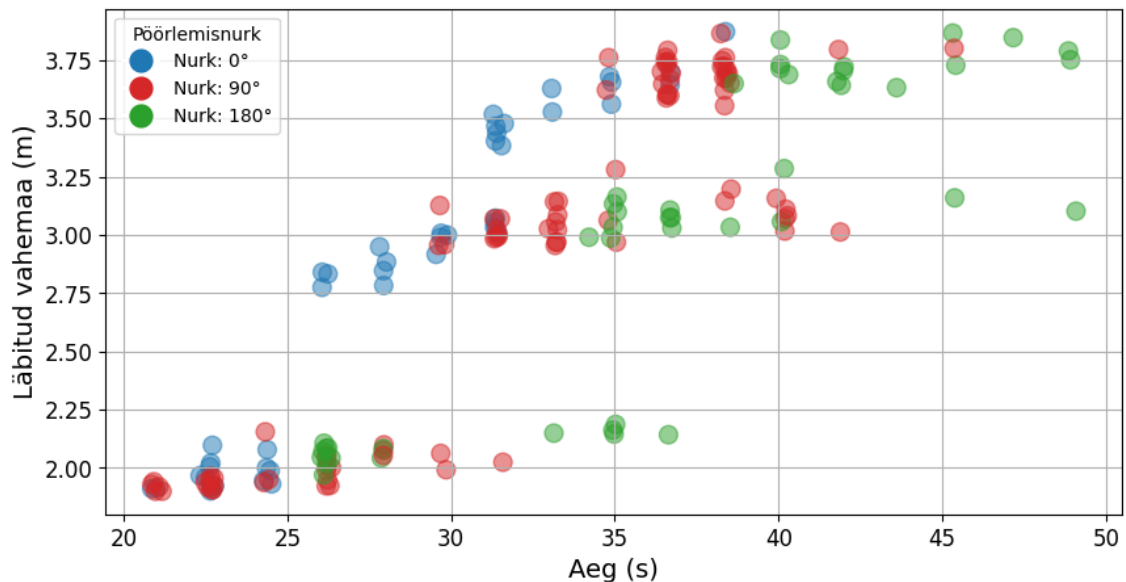
Süsteemi tõttu on kiirema lidariga liikumisel kiirusepiirang sama, mis aeglase puhul, ehk mõne muu kiirema lidariga sõitmiseks loodud süsteemiga, oleks 2 Hz ja 10 Hz lidariga liikumise vahe eeldatavasti suurem.

Kuigi katsetatud on erinevaid sagedusi 0.57 Hz ja 10 Hz vahel, siis sellise sagedusega lidareid tegelikult olemas pole, need on lisatud lihtsalt kiiruse vahemike näitamiseks. Kui navigatsioonisüsteem ei tööta hästi lõputöö skoobist välja jäänud päris robotiga katsetades on valikuks lisada Iasole kiirem 2D lidar, mille abil sujuvamalt ja kiiremini navigeerida.

Teiseks võimalikuks lahenduseks oleks 3D lidarist ja odomeetriast edasi sõltudes muuta Nav2 olemasolevate komponentide sisu osaliselt või tervenisti ise kirjutada ja katsetada, kas elementide muutmine parandab roboti efektiivsust. See lahendus on aga keerulisem ning aeganõudvam ja jäi lõputöö skoobist välja.

4.2 Liikumine

Joonisel 10 on näha roboti läbitud vahemaa ja ajakulu suhe pöördenurgaga. Graafikul on kolme värvi punktid, mis on määratud algpunkti ja sihtpunkti orientatsiooni vahega. See tähendab iga värv näitab, kui palju oleks pidanud robot pöörama võrreldes algse suunaga, et jõuda sihtpunkti õige suunaga. Joonisel on näha, et suurema nurga puhul on ajakulu suurem. Joonisel on kolme punkti kolme orientatsiooniga sõitmise tulemus ehk vahemaa läbimine peaks olema samasse punkti sõites sarnane, kuid variatsioon punkti jõudmiseks läbitud vahemaal on ligikaudu 50 cm.

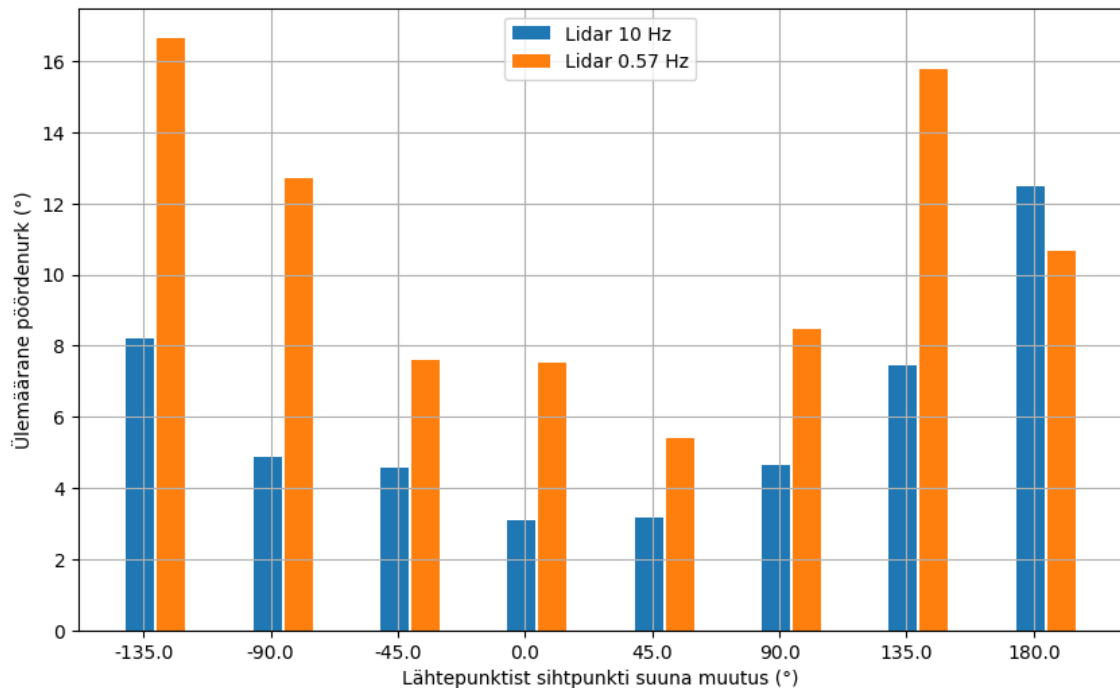


Joonis 10. Aja, läbitud vahemaa ja pöördenurga suuruse seos

Joonisel 11 on sihtpunkti liikumisel pööramise vahe minimaalse pööramisega 0.57 Hz lidariga ja 10 Hz lidariga roboti puhul. Efektiivse holonoomilise liikumise puhul robot sihtpunkti liikumise ajal ei tee üleliigseid pöördeid ehk pööramise kogus tee peale peaks olema vaid algse orientatsiooni ja sihtpunkti orientatsiooni vahe.

Joonisel on kaheksa erineva orientatsiooniga kolme erineva asukohaga punkti sõitmise jooksul roboti kogu pööramine. Joonise tegemiseks on võetud kõik orientatsiooni vahega punkti sõitmisel toimunud pööramise keskmine, millest on maha lahutatud minimaalne pööramise kogus. Joonisel on näha kui suur on minimaalse ja sooritatud pööramise vahe 0.57 Hz ja 10 Hz lidari puhul.

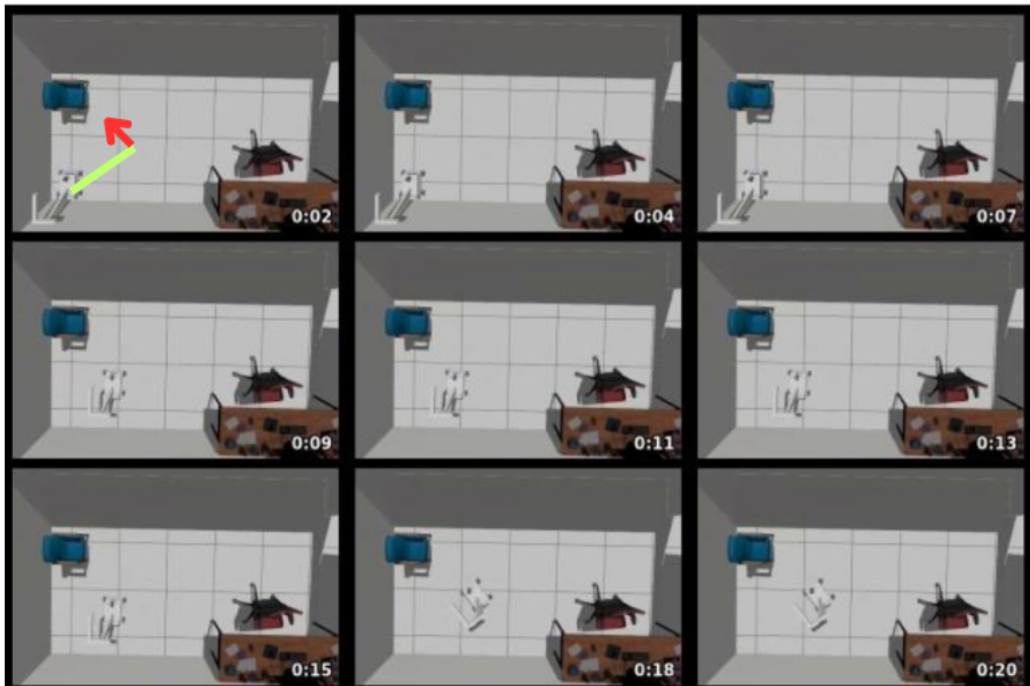
Saab järeldada, et algsuunast rohkem pööramist vajavad sihtpunktid tekitavad rohkem üleliigset pööramist kui väiksemad suunad. Lisaks on näha, et 0.57 Hz lidari puhul on üleliigset pööramist keskmiselt 2 korda rohkem kui 10 Hz lidariga. See tekib kuna Iaso pöörab punkti kohale jõudes ka suuna õigeaks, kuid madala vertikaalsagedusega lidari puhul on vaja rohkem suunda mitu korda parandada pimesi liikumise tõttu, ehk roboti pöörlemiskiirus on lidariga kindlamaks töötamiseks liiga suur.



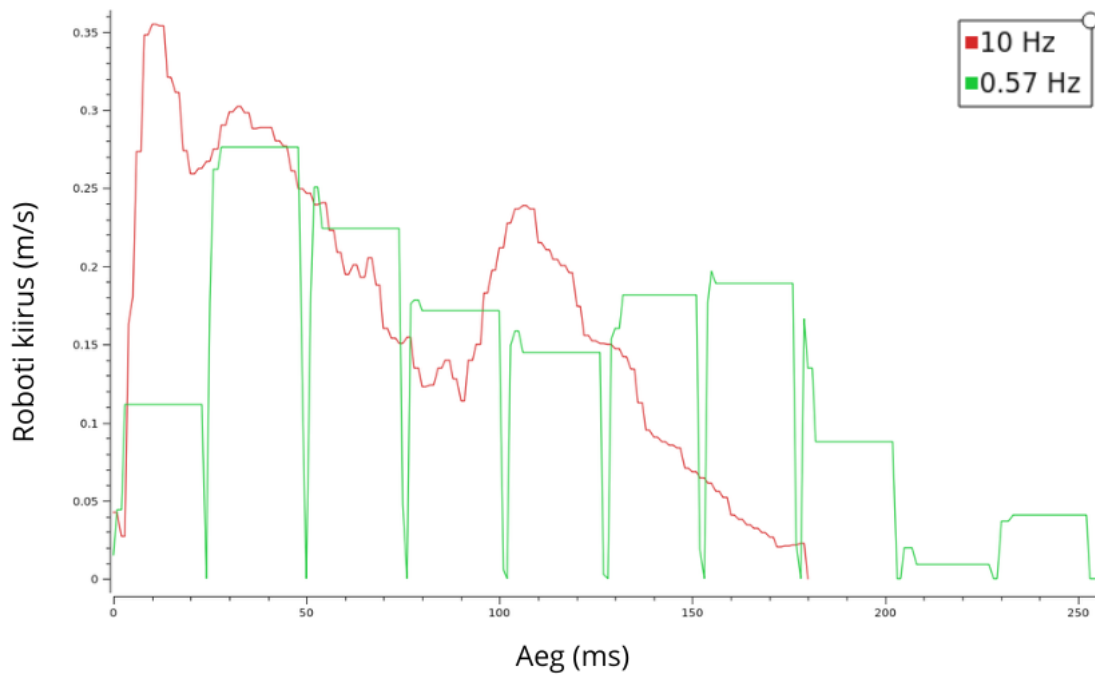
Joonis 11. Sihtpunkti liikumisel üleliigne pöördenurk 0.57 Hz lidari ja 10 Hz lidari puhul

Joonisel 12 on näha liikumine algpunktist ühte sihtpunkti. Pildid liiguvad vasakult paremale alustades esimesest reast ning lõpetades kolmanda reaga. Esimesel pildil on näha roheline joonena planeeritud tee ning punase noolega soovitud suund.

Joonisel 13 on rohelisega näha 0.57 Hz lidariga ning punasega 10 Hz lidariga kiiruse muutumine. 0.57 Hz lidariga on näha, et kiirus püsib pikemalt täpselt sama kuid iga kahe sekundi tagant läheb kiirus hetkeks nulli. 10 Hz lidariga muutub kiirus pidevalt kuid ühtlaselt ning sujuvalt. Mõlemad robotid sõitsid joonisel 12 näidatud punkti ning on näha, et aeglasema lidariga võttis kohale jõudmine kauem aega.



Joonis 12. Sihtpunkti sõitmine



Joonis 13. Roboti kiiruse muutumine ajas 10 Hz lidari ning 0.57 Hz lidariga

5. Tulemuste analüüs

5.1 Lidar

Kuigi Iaso liigub õigesse sihtpunkti ka madala vertikaalsagedusega 3D lidariga, on kasulik arvestada ka alternatiividega, peamiselt kuna loodud navigatsioonisüsteemiga on Iaso aeglane ning sõidustiil on ebahühtlane. Valideerimisest selgus, et kiirema lidariga on võimalik Iaso liikumist teha sujuvamaks ning kiiremaks. Kuigi kiirema lidariga on liikumine efektiivsem, ei tohiks sellega asendada olemasolevat 3D lidarit, kuna punkti jõudes peab robot käed liigutama all asendist üles, ning sel juhul on ohutuse tagamiseks vaja teada, et käte liikumise tee on tühi.

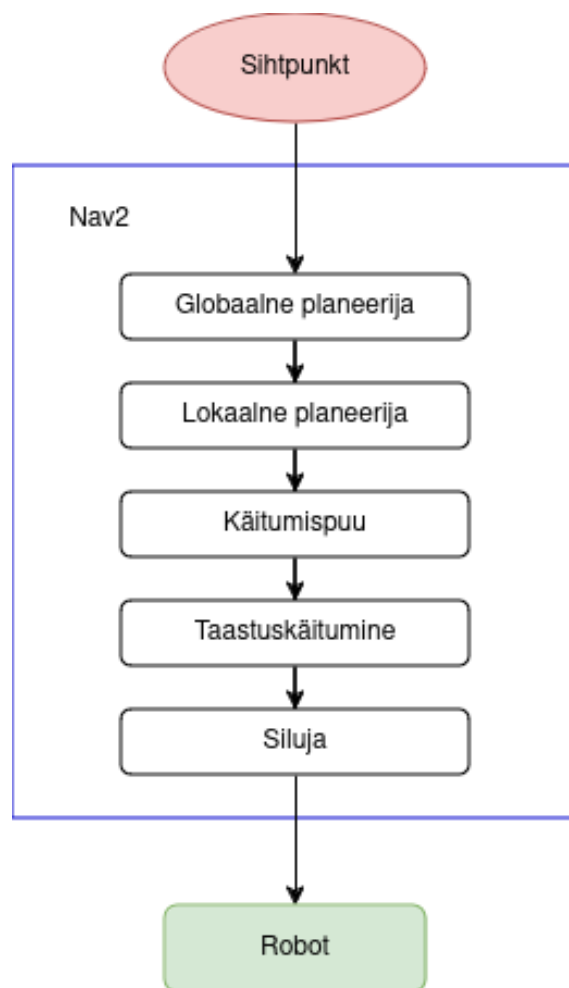
5.2 Navigatsioon

Joonisel 14 on näha Nav2-e toimimine. Käitumispuu Globaalne planeerija leiab tee punktini, lokaalne planeerija saadab robotile kiirused globaalse tee järgimiseks. Juhul kui robot on kinni, lõksus või ära eksinud, käivitub taaskäitumine. Siluja muudab glbaalse tee sujuvamaks ning paremaks.

5.3 Probleemid tarkvaradega

Kuigi suure osa kasutatud tarkvarade koos kasutamiseks on olemas tugi, siis ikkagi tuli lõputöö käigus ette mitu takistust eri osade koos töötamisel. Näiteks oli probleem Ignition Gazebo liikumise füüsika tööle saamisega URDF failiga, kus pole vajalikku parameetrit. Probleeme tekitas ka Ignition Gazebo vähene dokumentatsioon, kuna vahepeal on toimunud nimevahetus tagasi Gazebo peale.

Lisaks oli probleeme navigatsiooni eri osade parameetrite väärtuste määramisel, kuna nende kirjeldused dokumentatsioonis on lühikesed ja tihti nõuavad sügavat arusaamist kogu süsteemist. Ka esitatud näidete puhul oli probleemiks, et kõik on diferentsiaal sõiduks mõeldud või on kasutusel erinev versioon tekidest, mis ei tööta koos valitud tarkvaraga.



Joonis 14. Nav2 komponendid sihtpunkti saades

5.4 Diskussioon

Projekti alguses kulus suur osa ajast simulatsioonis korrektse füüsikaga sõitmise peale, mida tagantjärele mõeldes oleks pidanud kasutama pigem lõputöö eesmärgi, navigatsioonisüsteemi, arendamiseks. Veel oli ajakulu suurem, kuna kogemus robotikas enne lõputöö tegemist oli väga väike ehk oli palju uusi teemasid, millest nüüd rohkem teadlik olles kulgeks vähem oluliste probleemide lahendamine kiiremini. Nende hulka kuulub roboti mudelile URDF faili loomine ning navigatsiooniparameetrite muutmine.

Uuesti tehes kindlasti ka peaks panema rohkem rõhku kogu süsteemi põhjalikumaks arusaamiseks, kuna osa ajakulust oleks ka saanud vältida, kui oleks alguses dokumentatsioon ning juhendeid põhjalikumalt uurinud.

5.5 Edasiarendus

Projektile saaks edasiarendusena muuta navigatsiooni komponente nii, et lidari ootamise asemel Iaso sõidaks pimesi, kuid see vajab suuremat ajaressurssi, mis antud lõputöö skoobist välja jäi. Kindlam ja kiirem lahendus paremaks navigatsiooniks oleks siiski suurema sagedusega lidari lisamine. Kindlasti oleks kasu navigatsioonisüsteemi edasisel põhjalikul katsetamisel.

Veel oleks võimalik edasi katsetada ka päris robotil ning siduda ära olemasoleva tarkvaraga ja ka tegeleda kaamera käte üles liigutamisega.

6. Kokkuvõte

Lõputöö peamine eesmärk oli luua navigatsioonisüsteem, mis töötab Iasol olemasoleva madala vertikaalsagedusega 3D lidariga. Lõputööga valminud lahendusega, mis kasutab Nav2-te, suudab Iaso simulatsioonis autonoomselt liikuda soovitud punkti. Algselt plaanitud päris robotil katsetamine jäi lõputöö mahust kahjuks välja.

Kuigi peamine eesmärk sai täidetud, on lahendusel siiski puuduseid. Aeglase lidariga töötamiseks loodud navigatsioonisüsteemiga liigub Iaso järjest väikse vahemaa ning jääb ootama uut skanneeringut põhjustades ebasujuva liikumise. Probleem tekib kuna lõputöös kasutatud SLAM ei uuenda asukohta enne uue skanneeringu saamist.

Lõputööga sai kinnitatud eeldus, et kiirema lidariga on võimalik ka kiiremini ja täpsemalt liikuda. Seega aeglase lidari probleemi üheks lahenduseks oleks lisada Iasole kiirem 2D lidar, mille abil SLAM töötaks paremini ning robot liiguks pausideta. Teine lahendus oleks põhjalikumalt uurida olemasolevaid tehnoloogiaid ning probleemi kohtades või tervenisti neid muuta, kuid selline lahendus nõuab rohkem aega.

Kasutatud kirjandus

- [1] Steven Macenski et al. “Robot Operating System 2: Design, architecture, and uses in the wild”. In: *Science Robotics* (2022).
- [2] Steven Macenski et al. “The Marathon 2: A Navigation System”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020.
- [3] Ecam Eurobot. *Mecanum wheels*. [Accessed: 2023-10-24]. URL: <https://ecam-eurobot.github.io/Tutorials/mechanical/mecanum.html>.
- [4] *Gazebo*. [Accessed: 2023-10-27]. URL: <https://gazebo.org/home>.
- [5] *RVIZ*. [Accessed: 2024-05-01]. URL: <http://wiki.ros.org/rviz>.
- [6] Yujin Robot Co., Ltd. *YRL User Manual*. Version YRL_UM_KR_2103_V1.3. 2020. URL: <https://device.report/m/8e35b289a64c64f079d8405b04b990df6ec3debd24a6835994b253abb6959e15>.
- [7] The Robotics Back-End. [Accessed: 2024-05-11]. URL: <https://roboticsbackend.com/ros1-vs-ros2-practical-overview/>.
- [8] *ROS2 Humble*. [Accessed: 2024-05-02]. 2024. URL: <https://docs.ros.org/en/humble/index.html>.
- [9] *Nav2*. [Accessed: 2024-05-02]. 2023. URL: <https://navigation.ros.org/>.
- [10] [Accessed: 2024-05-11]. URL: <https://autowarefoundation.github.io/autoware.universe/main/>.
- [11] *Navigation Concepts*. [Accessed: 2024-05-02]. 2023. URL: <https://navigation.ros.org/concepts/index.html>.
- [12] [Accessed: 2024-05-13]. URL: <https://www.behaviortree.dev/>.
- [13] S. Macenski et al. “From the desks of ROS maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2”. In: *Robotics and Autonomous Systems* (2023).
- [14] *SLAM*. [Accessed: 2023-10-27]. URL: <https://www.mathworks.com/discovery/slam.html>.
- [15] Steve Macenski and Ivona Jambrecic. “SLAM Toolbox: SLAM for the dynamic world”. In: *Journal of Open Source Software* 6.61 (2021), p. 2783. DOI: 10.21105/joss.02783. URL: <https://doi.org/10.21105/joss.02783>.

- [16] *Slam Toolbox*. [Accessed: 2024-05-02]. 2021. URL: http://wiki.ros.org/slam_toolbox.
- [17] *Robot Localization wiki*. [Accessed: 2024-05-02]. 2016. URL: http://docs.ros.org/en/melodic/api/robot_localization/html/index.html.
- [18] *State Estimation Nodes*. [Accessed: 2024-05-02]. 2016. URL: http://docs.ros.org/en/kinetic/api/robot_localization/html/state_estimation_nodes.html.
- [19] [Accessed: 2024-05-11]. URL: <https://www.blender.org>.
- [20] Robin Baran. *Nexus 4WD Mecanum Simulator*. [Accessed: 2024-05-02]. URL: https://github.com/RBinsonB/nexus_4wd_mecanum_simulator/tree/e26640c600050b3da95312885d6a6e4de0786ff6/nexus_4wd_mecanum_description/meshes.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Brigitte Kerge

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Holonoomilise autonoomse roboti Iaso navigatsioonisüsteem”, mille juhendaja on Gert Kanter
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

25.05.2024

¹Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.