Tartu Ülikool
Loodus-ja täppisteaduste valdkond
Tehnoloogiainstituut

Markus Erik Sügis

Jagatud juhtimise põhimõttel realiseeritud robotite kaugjuhtimissüsteem

Bakalaureusetöö (12 EAP)
Arvutitehnika eriala

Juhendajad:
Robootika kaasprofessor Karl Kruusamäe
Arun Kumar Singh

Tartu 2022

Resümee

Jagatud juhtimise põhimõttel realiseeritud robotite kaugjuhtimissüsteem

Viimase aastakümne jooksul on inimestele soovimatuid või liiga keerulisi ülesandeid hakatud järjest rohkem isetegutsevate robotite abil automatiseerima. Enamik tänapäeva robotisüsteeme pole võimelised täielikult autonoomseks opereerimiseks, põhjuseks pidevalt muutuv keskkond. Käesoleva bakalaureuse töö raames uuriti olemasolevaid võimalusi mobiilsete robotite juhtimiseks, loodi jagatud juhtimise põhimõttel ROS Navigation'il põhinev juhtimissüsteem ning testiti süsteemi kasutuskõlblikust ja jõudlust robotiplatvormil Robotont. Töö tulemusena valmis ROS-põhiste mobiilsete robotite juhtimissüsteem, mis on võimeline operaatori ning roboti koostööl robotit komplekssetes keskkondades ohutult navigeerima.

CERCS: T125 Automatiseerimine, robootika, control engineering, P176 Tehisintellekt, P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Märksõnad: robootika, kaugjuhtimine, jagatud juhtimine, ROS, ROS Navigation

Abstract

A continuous teleoperating system based on shared control concept

During the last decade, automation of tedious or unachievable tasks for humans has become increasingly popular by the usage of intelligent robot systems. Today, most robot systems are not capable of fully autonomous operating due to constantly changing environments. The aim of this thesis was to research existing means of operating mobile robots, designing a shared control teleoperating system using ROS Navigation and testing the usability and capability with the Robotont robotplatform. As a result of the thesis a teleoperating system for ROS-based mobile robots was designed, capable of safely navigating the robot in complex environments.

CERCS: T125 Automation, robotics, control engineering, P176 Artificial intelligence, P170 Computer science, numerical analysis, systems, control

Keywords: robotics, teleoperation, shared control, ROS, ROS Navigation

Sisukord

Resümee/Abstract **2**

Jooniste loetelu **5**

1 Sissejuhatus **6**

2 Kirjanduse ülevaade **7**

2.1 Telerobootika 7

2.1.1 Telerobootika põhimõte 7

2.1.2 Telerobootika rakendused 8

2.1.3 Telerobootika kitsaskohad 10

2.2 Poolautonoomne navigeerimine 11

2.2.1 Autonoomse roboti põhimõte 11

2.2.2 Poolautonoomse robootika rakendused 12

2.2.3 Autonoomse navigeerimise kitsaskohad 14

3 Töö eesmärk ja nõuded **16**

3.1 Töö eesmärk 16

3.2 Süsteeminõuded 16

3.2.1 Funktsionaalsed nõuded 16

3.2.2 Mitte-funktsionaalsed nõuded 16

4 Kasutatud töövahendid **17**

4.1 ROS 17

4.2 ROS Navigation 17

4.2.1 Asukohastamissüsteem 18

4.2.2 Liikumissüsteem 18

4.3 Robotont 20

5 Lahendus **21**

5.1 Süsteemi ülesehitus 21

5.1.1 Kaameramoodul 22

5.1.2 /depthimage_to_laserscan 22

5.1.3 Roboti asend 23

5.1.4 /slam_gmapping 23

5.1.5 /ps4_joy 23

5.1.6 /move_base_flex 23

5.1.7 /speed_controller_jsp 26
5.2 Kasutajaliides 27
5.2.1 JoystickPlannerViz liides 27
5.2.2 RViz liides 28

6 Tulemused ja analüüs **29**

7 Kokkuvõte **32**

Kasutatud kirjandus **33**

Lihtlitsents **36**

Jooniste loetelu

1 Näide telerobootika süsteemist. 7
2 R. C. Goertz'i poolt loodud kaugjuhitavad robotkäed. 8
3 V. Pruks jt loodud kasutajaliides ja virtuaalse kaardi loomisprotsess. 9
4 I. A. Sulistijono jt poolt loodud kullerrobot. 10
5 Erinevate kaugjuhitavate robotsüsteemide arhitektuuride kontseptsioonid. . . . 13
6 Võrdlus ilma kompensatsioonisüsteemita kuvatud pildist (vasakul) ja süsteemi genereeritud operatsiooni keskkonnast (paremal). 13
7 Näide tüüpiliselt ROS Navigation'i ülesehitusest. 18
8 Näide globaalsest kulukaardist. Värviliselt on tähistatud takistuste paisutused, kus kollane tähistab kõrgeimat ning hall madalaimat hinda. 19
9 Robotiplatvormi „Robotont" kolmikvaade. 20
10 Töö raames kujundatud süsteemi kontseptsioon. 21
11 Töö raames kujundatud süsteemi ROS sõlmede lihtsustatud diagramm. 22
12 Autori implementeeritud sõlme „JoystickPlanner" tarkvara loogika diagramm. . 24
13 Näide juhtpuldi kangi asendile vastavast rajast (roheline). 24
14 mbf_msgs/ExePath sõnumi struktuur. 25
15 Autori implementeeritud kasutajaliides süsteemi kasutamiseks. 27
16 Kujutis süsteemi testimisest loodud kasutajaliidesest tööriista *RViz* abil. 28
17 Võrdlus süsteemita (paremal) ning süsteemi kasutades (vasakul) roboti juhtimise vahel ühe takistuse näitel. Roheliselt on kujutatud globaalne ning punaselt lokaalne rada. 29
18 Süsteemi testimiseks loodud takistuste asetused. 30
19 Süsteemi testimise ajalised tulemused kolme takistuste asetuse korral. 30
20 Asetuste läbimise edukusprotsendid. 31

1 Sissejuhatus

Robootika on järjest populaarsemaks muutuv valdkond. Üha enam kohtab maailmas erinevaid ülesandeid täitvaid roboteid, näiteks majapidamistes [1], tööstuses [2] või transpordis [3]. Kuigi robotitel on lõpmata palju rakendusvaldkondi, jagavad need kõik ühist eesmärki – inimeste elu lihtsamaks muutmine. Robotid leiavad kasutust eelkõige selliste ülesannete automatiseerimisel, mida inimesed kas ei soovi või ei suuda täita või mille täitmine võib osutuda inimestele liiga ohtlikuks.

Robotsüsteemi definitsioonist tulenevalt on robot teatud määral autonoomne. Enamik robotisüsteeme pole aga tänasel päeval täielikult autonoomsed ning ei suuda kasulike ülesandeid täita pikkade ajaperioodide vältel [4]. Nimelt on reaalses maailmas lõpmata palju muutujaid, mille kõigiga pole üks tänapäeva robotitsüsteem veel võimeline arvestama.

Tehisintellekti kitsaskohtadest tulenevalt jagatakse robotite juhtimist inimese ehk operaatori ning roboti vahel. Selliste poolautonoomsete robotite kaugjuhtimise arhitektuur jaguneb üldiselt kaheks:

a) järelvalvega juhtimine (ingl *supervisory control*):
   operaator jagab robotile käsklusi ning saab robotilt kokkuvõtteid tehtud tööst;

b) jagatud juhtimine (ingl *shared control*):
   käsilolev ülesanne on jagatud mitmeteks alamülesanneteks, mille täitmine on jagatud roboti ja operaatori vahel.

Jagatud juhtimine leiab kasutust eelkõige operaatori töö lihtsustamiseks või korrigeerimiseks kriitiliste ülesannete täitmisel [5], näiteks kirurgias [6] või põlengute kustutamisel [7]. Ka navigatsioonis leiab jagatud juhtimine kasutust keskkondades, millele operaatoril on piiratud ligipääs [8], lihtsustades seeläbi roboti ohutut juhtimist.

Käesoleva bakalaureuse töö eesmärgiks on uurida olemasolevaid võimalusi mobiilse roboti kaugjuhtimiseks, disainida roboti kaugjuhtimissüsteem jagatud juhtimise põhimõttel ning hinnata selle kasutuskõlblikust ja jõudlust. Töös antakse ülevaade olemasolevatest roboti kaugjuhtimismetoodikatest ning kirjeldatakse valminud kaugjuhtimissüsteemi.

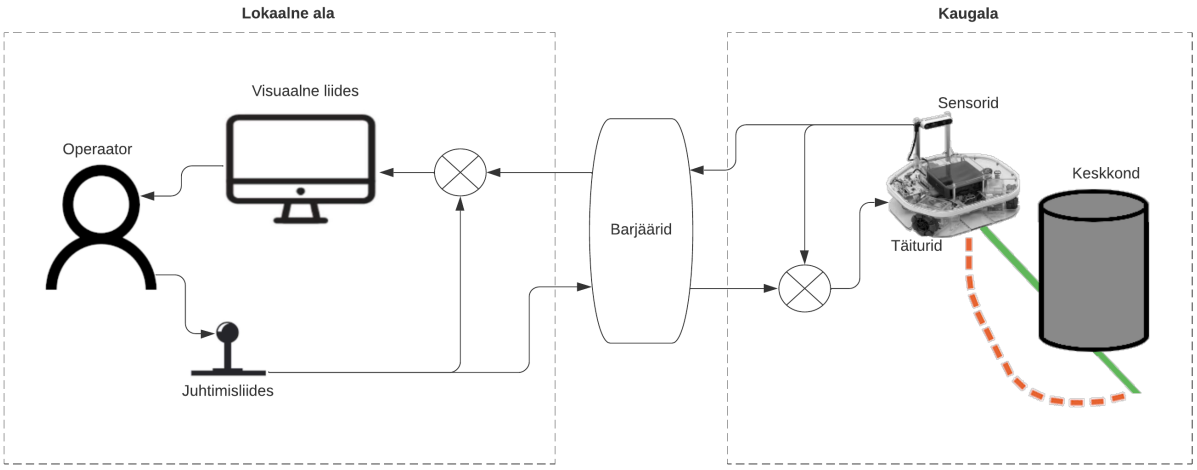
2 Kirjanduse ülevaade

2.1 Telerobootika

2.1.1 Telerobootika põhimõte

Telerobootika või kaugrobootika on robootika haru, mis käsitleb robotsüsteemide kaugjuhtimist ja haldamist inimoperaatori poolt. Operaatoril puudub roboti ümbruskonnale ligipääs ning ta annab robotile instruktsioone kaugjuhtimisliidese kaudu, omades piiratud ülevaadet roboti ümbruskonnast (näiteks videoülekanne robotile kinnitatud kaamerast või roboti andurite lugemid). Sellises inimene-juhtimistsüklis (ingl *human-in-the-loop*) süsteemis teostatakse kõik kõrgema taseme otsused ja planeerimine kasutaja poolt. Roboti ülesandeks on kasutaja poolt antud instruktsioonide täideviimiseks vajalike madalama taseme arvutuste tegemine ja mehaaniliste operatsioonide teostamine. [9].

Kontseptuaalselt jaguneb kaugjuhitava roboti süsteem kaheks alaks: lokaalne ehk kohalik ala ning kaugala (joonis 1). Kohaliku alasse kuuluvad operaator ning roboti kaugjuhtimiseks vajalikud komponendid, näiteks audio-visuaalne liides, juhtkangid või klaviatuur. Kaugalasse kuulub robot koos kõigi sensorite ja täiturite. Alad ei pea olema üksteisest tingimata eraldatud. Operaator ning robot võivad asuda üksteisega samas ruumis, kuid see ei mõjuta nende alade eristamist. Barjääride nimetatakse kõiki faktoreid, mis takistavad või piiravad operaatori ligipääsu robotile ning roboti ümbruskonnale. Süsteemi selliseks ülesehituseks rakendab telerobootika mitmeid teisi robootika valdkondi. [9].



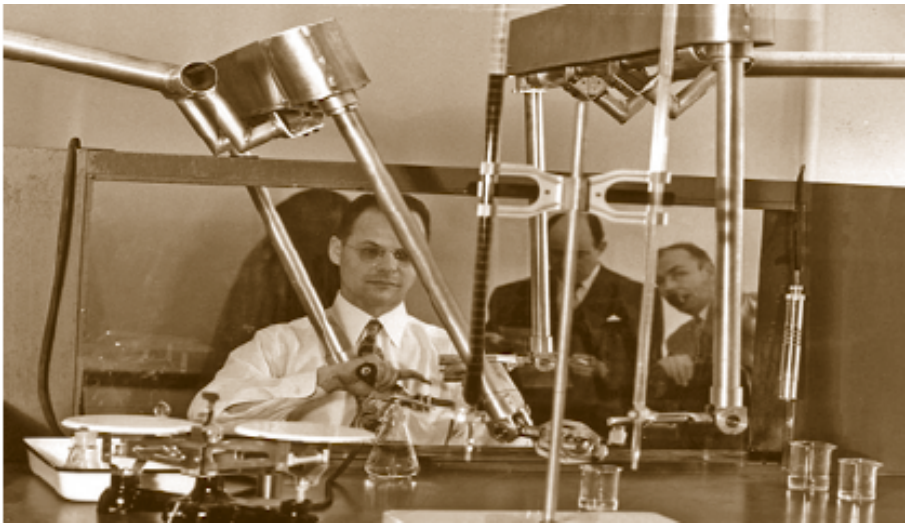
Joonis 1. Näide telerobootika süsteemist.

Enamike telerobootiliste süsteemide ülesehitusse kuulub mingil määral roboti otsene juhtimine kasutaja poolt, kas juhtkangi või muu samalaadse seadme abil. Sellest tulenevalt võib juhtseadet vaadelda kui lokaalses alas paiknevat robotsüsteemi, mis juhib teist robotit kaugalas, andes kogu süsteemile peremees-alluv (ingl *master-slave*) topoloogilise ülesehituse, kus juhtseade on peremees ning opereeritav robot alluv. Otsese juhitavuse tagamiseks programmeeritakse alluv robot järgima peremehe käsklusi, mida omakorda kontrollib kasutaja. Selline kasutaja ja peremehe vaheline suhtlus on inimese-roboti vahelise suhtluse vorm. [9].

Peremees-alluv topoloogia peamiseks eesmärgiks on kaugkohalolu (ingl *telepresence*) võimaldamine operaatorile. Kaugkohalolu võimaldab operaatoril interakteeruda ja tajuda kaugpiirkonda nagu viibiks operaator ise seal – peremees-alluv süsteem on piirkondade vaheliseks suhtlusmeediumiks. Mida vähem on operaator teadlik enda eemalviibimisest ehk piirkondade vahelistest barjääridest, seda läbinähtavam süsteem on. Täielik läbinähtavus on kaugkohalolu ülim eesmärk. [9].

2.1.2 Telerobootika rakendused

Kaugjuhitavaid roboteid kasutatakse eelkõige operaatoritele kättesaamatutes või eluohtlikes keskkondades, näiteks õnnetuspaikades [10], vee all [11] või kosmoses [12]. Eluohtlikes keskkondades kriitiliste ülesannete täitmisel on esmatähtis võimalikult realistlik kaugkohalolu [9].



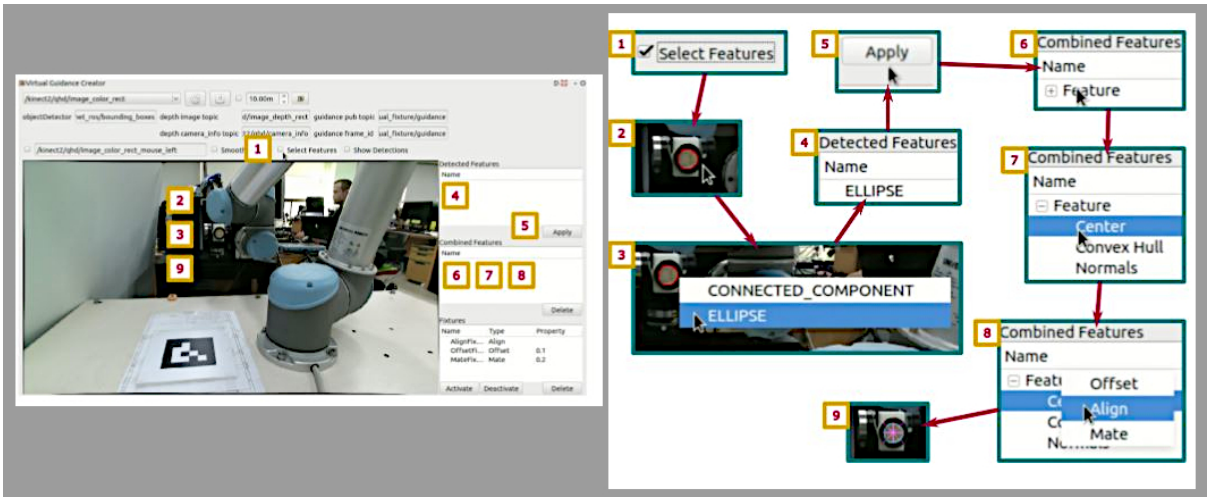
Joonis 2. R. C. Goertz'i poolt loodud kaugjuhitavad robotkäed.

Ajalooliselt üheks esimeseks telerobotsüsteemiks on R. C. Goertz'i ja W. M. Thompsoni loodud

radioaktiivsete materjalidega töötamiseks mõeldud robotmanipulaatorid (joonis 2 [13]), mille täiturite liikumisi erinevatel telgedel juhiti lülitite jadaga [14].

Lahendusel puudus füüsiline tagasiside operaatorile, mis muutis süsteemi kasutamise aeglaseks ja kohmakaks. Seetõttu kujundas Goertz peremees-alluv robotite paaridel põhineva uuendatud süsteemi, mis võimaldas operaatoril juhtliidesega imiteerida inimesele omaseid käeliigutusi ning võimaldas süsteemil läbi jõudude ja vibratsiooni operaatorile realistliku tagasisidet pakkuda. [9].

V. Pruks jt on oma uurimustöös loonud tuumajaamas inimestele ohtliku radiatsiooni tasemega keskkonnas töötavale robotile interaktiivse kaugjuhtimisliidese [8]. Lahendus võimaldab operaatoril kahedimensionaalse pildi alusel arvutihiire abil roboti ümbruskonda kaardistada. Esmalt kuvatakse kasutajaliideses videoülekanne robotil paiknevast Kinect kaamerast. Operaator liigub hiirega punkti, kus süsteem on automaatselt mingi kontuuri olemasolu tuvastanud. Kontuuriks võib olla näiteks serv või sirgjooneline segment. Operaator saab kontuuri kaardile märkida (joonis 3 [8]).

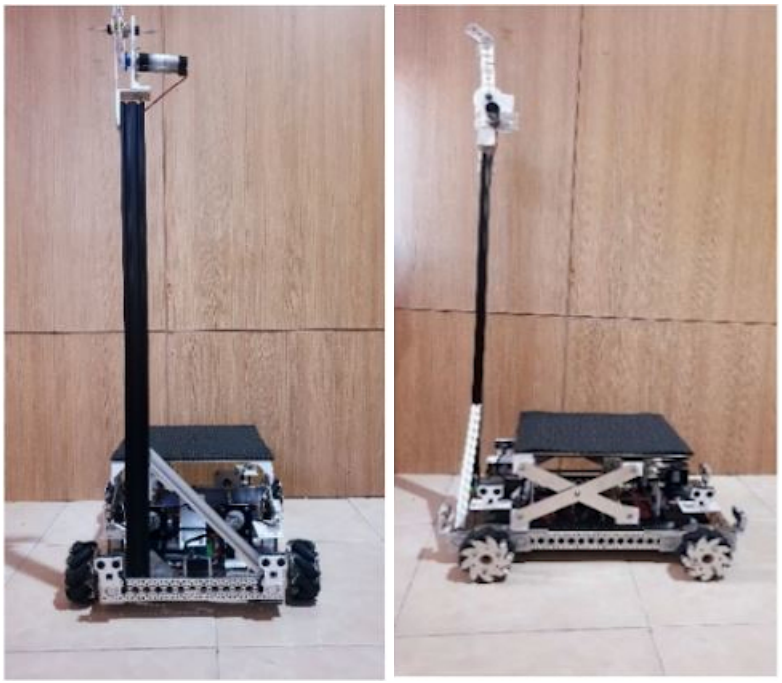


Joonis 3. V. Pruks jt loodud kasutajaliides ja virtuaalse kaardi loomisprotsess.

Inimtunnetusliku komponendi integreerimine kaardistusprotsessi annab väga täpse lõpptulemuse. Loodud kaardile on operaatoril võimalik joonestada radu, mida järgides on robotit kergem kaugjuhtida. Süsteemi testimisel mutrite pingutamise (ingl *peg-in-hole*) ülesandega, vähendas juhtimissüsteemi kasutamine keskmist tööaega $\sim 5,9$ sekundi võrra. [8].

Meditsiinivaldkonnas riskivad nakkushaigustega patsientide eest hoolitsevad meditsiinitöötajad pidevalt oma tervise ning kohati ka eluga, mille vältimiseks on hakatud rakendama kaug-

kohalolu põhimõtet. I. A. Sulistijono jt on oma uurimustöös kujundanud kaugjuhitava toidu ja ravimite kuller-roboti (joonis 4 [15]) nakkushaigustega patsientide eest hoolitsemiseks.



(a) Roboti eestvaade (b) Roboti külgvaade

Joonis 4. I. A. Sulistijono jt poolt loodud kullerrobot.

Roboti ülesehitus lubab robotil holonoomselt ruumis ringi liikuda. Madala-taseme juhtsüsteemi ülesandeks on ainult lokaalne planeerimine ja dünaamiliste takistuste vältimine. Robotit on võimalik juhtida täielikult ka juhul, kui roboti autonoomsusfunktsionaalsus peaks töötamise lõpetama. [15].

2.1.3 Telerobootika kitsaskohad

Robotite kaugjuhtimise oluliseks aspektiks on stabiilne ja usaldusväärne juhtmevaba ühendus operaatori ning roboti vahel, mis on ka üks teleopereerimise nõrgemaid lülisid. Tihti pole stabiilsed võrguühendused operatsioonipaigas pidevaks kaugjuhtimiseks piisavalt kättesaadavad, mille tõttu ei saa operaator keskenduda ainult sõiduki juhtimisele, vaid peab keskenduma ka võrguühenduse stabiilsuse kontrollile ning sellest tulenevatele probleemidele.

R. Parasuraman jt on oma uurimustöös välja pakkunud uue kaugjuhtimisliidese, millel on raadiosignaali saabumissuuna ning tugevuse määramisfunktionaalsus, mille implementeerimine peaks tõstma süsteemi tõhusust. [16].

Samas on B. Doroodgar, Y. Liu ja G. Nejat otsustanud probleemile läheneda teisiti, pakkudes välja poolautonoomse, operaatorist vähem sõltuva süsteemi, jagades töökoormust operaatori ning sõiduki vahel. Väljapakutud süsteem on masinõppiv, parandades süsteemi sooritusvõimet ning jõudlust operatsioonipaigas kogutud andmete ja kogemuste põhjal. [10].

Telerobotsüsteem, mille kaugala asub näiteks kosmosejaamas ning lokaalne ala Maal, kannatab andmevahetuse ribalaiuse kitsuse tõttu, mille tulemusena kõrge eraldusvõimega videoülekanne kaugkohalolu tarbeks võib muutuda aeglaseks. Videopildi madal kaadrite arv sekundis võib kaasa tuua operaatori poolseid vigu, mis võivad põhjustada kahjustusi nii kaugjuhitavatele kui ka ülesande kontekstis käsitletavatele süsteemidele. L. Stark jt on oma uurimustöös väljatöötanud pilditöötlusalgoritmi, mille eesmärgiks on vähendada igas uuendustsüklis tehtavate arvutuste mahtu, filtreerides pildile vaid esmatähtsad parameetrid, näiteks ülesande täitmiseks vajalike objektide kontuurid, mis kujutavad roboti mudeli uuendamist ja korrigeerimist ning informatsiooni roboti töökeskkonna kohta. [12].

2.2 Poolautonoomne navigeerimine

2.2.1 Autonoomse roboti põhimõte

Autonoomne robot on intelligentne robotisüsteem, mis on ilma pideva otsese juhtimiseta võimeline reaalse maailma ülesandeid täitma [4]. Enamik robotisüsteeme pole tänasel päeval täielikult autonoomsed ning pole võimelised pidevalt muutuvas keskkonnas pikkade ajaperioodide vältel kasulikke ülesandeid täitma [4]. Robotite võime olla täielikult autonoomsed igal ajal ja igas olukorras, vaatamata kasutuses olevatest automatiseeritud autodest või lennukitest, on müüt [5].

Autonoomse roboti juhtimine koosneb mitmest tasemest. Kõrgema taseme juhtimise alla kuulub üldiselt autonoomse robotsüsteemi puhul mingi Asimovi seaduste implementatsioon, mille kohaselt ei tohi robot teha liiga inimesele, peab kuuletuma inimesele (kui see pole vastuolus esimese seadusega) ning ei tohi teha liiga teisele robotile (kui see pole vastuolus esimese või teise seadusega). Madalaimal juhtimise tasemel toimub roboti täiturite (mobiilse roboti puhul näiteks mootorite) juhtimine ja stabiliseerimine. Tase kõrgemal juhitakse robotit teisi roboteid ning keskkonnas paiknevaid takistusi vältima, hoides madalaima taseme juhtimist stabiilsena.

Autonoomse roboti arhitektuuri kuulub eeldatavalt ka kindlate ülesannete komplekt, mida süsteem on võimeline iseseisvalt täitma, vältides samaaegselt keskkonnas olevaid takistusi ning hoides liikumise stabiilsena. Selliseid paralleelseid juhtimisarhitektuure nimetatakse ka käitumispõhisteks arhitektuurideks. [4].

Robotsüsteemid pole enamasti täielikult autonoomsed ning vajavad ülesannete täitmiseks käsklusi ja juhiseid operaatorilt või kasutajalt. Selliseid süsteeme võib nimetada poolautonoomseteks. Telerobootika vaatepunktist kategoriseeritakse poolautonoomsed robotsüsteemid kaheks peamiseks arhitektuuriks, mille kontseptsioone praktikas tihti liidetakse. [9].

Järelvalvega juhtimine (ingl *Supervisory control*) on analoogne ülemuse tööülesannete jagamisega inimalluvale. Ülevaataja ehk operaator jagab inimalluvale ehk robotile tööülesandeid ning saab alluvalt vastu kokkuvõtte tehtud tööst. Täpseid instruktsioone, kuidas ülesannet täita operaator robotile ei anna. Ülesannet täidab robot autonoomselt. Operaatori ülesanneteks on järelvalvega juhtimise korral telerobootilise süsteemi järelvalvamine ning kõrgema taseme otsuste langetamine. [9].

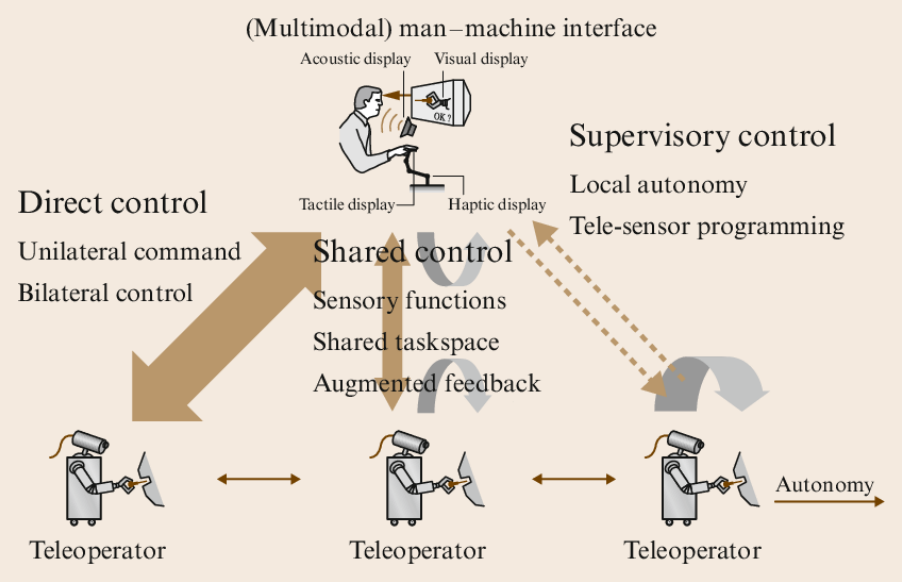
Jagatud juhtimine (ingl *Shared control*) on kaugjuhitavates süsteemides enamasti ülesandest orienteeritud. Käsilolev ülesanne on jagatud alamülesanneteks, mille täitmine on jagatud operaatori ning robotsüsteemi autonoomse komponendi vahel [9]. Näiteks kaasaegsetesse autodesse paigaldatud püsikiiruse hoidjad lubavad auto juhtimise ülesannet autojuhil ehk operaatoril kergemaks muuta, kontrollides autonoomselt seatud kiiruse saavutamiseks vajalikke mootori pöördeid [17].

Autonoomse kaugjuhtimise kontseptsioone kirjeldab joonis 5 [18].

2.2.2 Poolautonoomse robootika rakendused

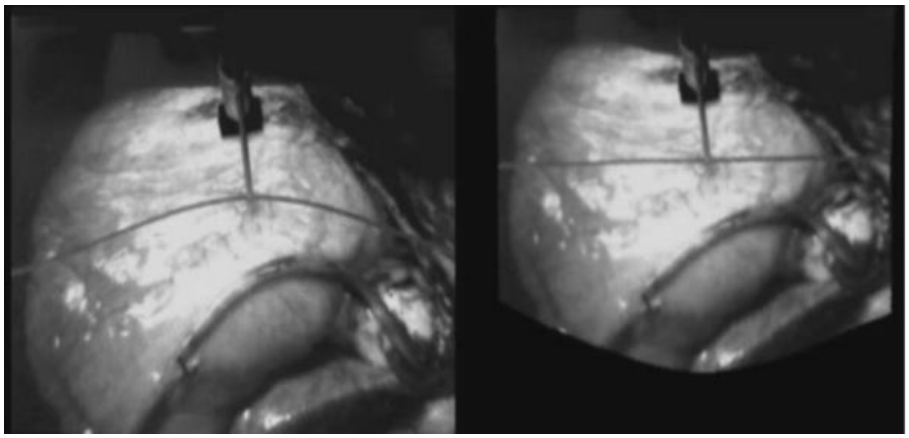
Poolautonoomseid või jagatud juhtimisega robotsüsteeme rakendatakse eelkõige operaatori töö lihtsustamiseks või korrigeerimiseks [5]. Näiteks võib operaator vajada aeg-kriitilise ülesande täitmisel robotsüsteemi abi [9] või nõuab ülesanne üliinimlikku täpsust, näiteks kirurgilistel protseduuridel [6]. Poolautonoomne navigeerimine leiab laialdast kasutust arenenud juhiabi süsteemides, mille autonoomne süsteem juhti varieeruva intensiivsusega pidevalt abistab [19].

Inimese elutähtsate organite läheduses opereerimisel on alati risk kirurgi inimliku vea tõttu pat-



Joonis 5. Erinevate kaugjuhitavate robotsüsteemide arhitektuuride kontseptsioonid.

sienti vigastada või lausa tema elu ohtu seada. G. P. Moustris jt on oma uurimustöös loonud jagatud juhtimisel põhineva robotsüsteemi abistamaks kirurgi inimese südame läheduses opereerimisel [6]. Väljapakutud süsteem kompenseerib südame löökidest tingitud liikumise pilditöötlusalgoritmide abil, kuvades kirurgile südamest staatilise pildi. Selle süsteemiga loodetakse vähendada potensiaalseid kirurgi segavaid faktoreid, mis võivad mõjutada operatsiooni edukust. [6]. Süsteemi testimisel juhtis kogenud kirurg meditsiinilist operatsioonirobotit, üritades järgida simuleeritud keskkonnas tuksuvale südamele joonistatud joont (joonis 6 [6]). Pakutud kompensatsioonisüsteemi kasutamine parendas keskmist kõrvalekallet joonest $41,26\%$ [6].



Joonis 6. Võrdlus ilma kompensatsioonisüsteemita kuvatud pildist (vasakul) ja süsteemi genereeritud operatsiooni keskkonnast (paremal).

Jagatud juhtimise rakendamine sõidukite või robotite navigeerimisel aitab vähendada juhi põh-

justatud õnnetuste sagedust, muuta sõitu reisijatele mugavamaks ning vähendada juhi töökoormust [19]. Mootorsõidukite puhul on autonoomseteks komponentideks enamasti rajajärgimissüsteem, kohanduv püsikiirusehoidja või rajast kõrvalekaldumise hoiatussüsteem. Arenenumates süsteemides võib sõiduk olla võimeline ka järelvalvega juhtimiseks [19], st sõitma järelvalve all autonoomselt punktist A punkti B.

B. Zhang jt on oma uurimustöös kujundanud jagatud juhtimise printsiibil toimiva poolautonoomse ratastooli [20]. Süsteemi eesmärgiks on vähendada tooli operaatori koormust ning muuta tooliga liikumine ohutumaks. Süsteem on varustatud 12 ultraheli anduri, RGB-D kaamera ning laseranduri ehk lidariga. Nende andurite abil toimub keskkonna kaardistamine ohutuks liiklemiseks. Kasutades DWA (*Dynamic Window Approach*) algoritmi on tool võimeline aitama operaatoril rahvarohketes keskkondades staatilisi ning dünaamilisi takistusi vältida. Kõrgema taseme juhtimise eest vastutab operaator [20].

Autonoomsed robotid leiavad üha enam kasutust ka inimestele liiga ohtlikes olukordades, näiteks põlengutes [7]. S. V. Maddukuri jt on oma uurimustöös kujundanud täielikult autonoomse ja jagatud juhtimisega tuletõrjeroboti [7]. Täielikult autonoomses režiimis tuvastab robot põlengu ning asub iseseisvalt tuld kustutama. Robot on võimeline teada oleva ruumipaigutuse korral takistusi tuvastama ning vältima. Jagatud juhtimise režiimis jagab robotile käsklusi operaator, kasutades häälkäsklusi või puutetundlikku kasutajaliidest. Põlengu tuvastamisel teavitab robot ohust kõiki tööl olevaid isikuid. Robot on ühenduses hoones olevate suitsuanduritega, mis annavad robotile teada puhkenud tulekahjust. [7].

2.2.3 Autonoomse navigeerimise kitsaskohad

Autonoomne robotsüsteem saab ümbritseva kohta teavet läbi andurite, ent andurite arvu piiravad füüsiline ruum roboti kerel, kandevõime või pardaarvuti võimekus. Viimane mõjutab ka roboti arvutusvõimsust ning andmetöötluskiirust [21]. Kaugjuhitavate mobiilsete robotite arvutusvõimsuse tõstmiseks on üha enam hakatud robootikas rakendama pilvearvutust [22].

Pilverobootikat kasutab iga andmete või programmide käitamiseks pilve infrastruktuuri rakendav roboti-või automaatikasüsteem [23]. Pilve kasutamine annab robotitele teoreetiliselt piiramatu arvutusvõimsuse ja talletusruumi, pakkudes võimaluse kaugarvutis teostada ressurssinõudvaid ülesandeid, näiteks esemete, kõne või mustrite tuvastus [22].

Ühiskonnaga interakteeruvate autonoomsete robotite puhul on alati kitsaskohaks olnud eetiliste otsuste langetamine [24]. Süsteemiinsenerid seisavad vastamisi mitmete moraalsete dilemmadega, sest just nemad peavad määrama sõiduki reageerimisviisi ohuolukordades. Näiteks situatsioonis, kus sõiduk peab valima autos viibivate inimeste või jalakäijate potensiaalse vigastamise vahel [25]. Võimalikeks valikuteks on võimalikult väheste inimeste vigastamine, autos olevate reisijate kaitsmine või kõrvaliste isikute kaitsmine. Inimese käitumist sellises olukorras on võimatu ennustada, kuid autonoomse sõiduki puhul teevad selle otsuse sõiduki pardaarvuti programmeerinud tehnikud [25].

Probleemiks on ka metslooma ootamatu ilmumine teele. Kuidas peaks auto reageerima, kui looma päästmiseks on tarvilik äkkpidurdamine, mis halbade ilmastikuolude või sõidukile tagant läheneva teise sõiduki korral võib ohtustada autos viibijaid. Juhul kui kõrvalrajal liiklevad teised autod võib ka loomast kõrvalepõikamine tähendada reisijate ohtu seadmist. Samuti on seatud küsimuse alla autonoomse sõiduki seaduskuulekuse eetilisus olukorras, kus looma elu päästmiseks vajaliku manöövri sooritamiseks peaks sõiduk liiklusseadust rikkuma. [24].

3 Töö eesmärk ja nõuded

Käesolev peatükk annab ülevaate töö eesmärgist ja süsteemile seatud funktsionaalsetest ning mitte-funktsionaalsetest nõuetest.

3.1 Töö eesmärk

Töö eesmärgiks oli uurida olemasolevaid võimalusi mobiilse roboti kaugjuhtimiseks, kirjeldatud süsteemi disainimine jagatud juhtimise põhimõttel ning selle kasutuskõlblikuse ja jõudluse hindamine.

3.2 Süsteeminõuded

3.2.1 Funktsionaalsed nõuded

1. Operaatoril on juhtkangi abil võimalik robotit jagatult juhtida.
2. Robot järgib kasutaja poolt ette antud radu ning väldib autonoomselt rajal olevaid takistusi.
3. Süsteemi on võimalik kasutada kõigi diferentsiaal-teljeliste ROS' i põhiste mobiilsete robotite juhtimiseks.
4. Süsteem kuvab visuaalsel kasutajaliidesel kasutaja poolt antud liikumissuuniseid ning takistusi vältivat sõidutrajektoori.

3.2.2 Mitte-funktsionaalsed nõuded

1. Robot on võimeline autonoomselt takistusi vältima kuni $0,8\frac{m}{s}$ lineaarse liikumiskiiruse juures.
2. Süsteemi ülesehitus võimaldab juhtkangina kasutada DS3 ning DS4 pulte.
3. Loodud kaugjuhtimisssüteem toetab ROS Melodic ning ROS Noetic väljalaskeid.
4. Ühildub ROS Navigation teegiga.

4 Kasutatud töövahendid

4.1 ROS

ROS (*Robot Operating System*) on avatud lähtekoodiga alam-operatsioonisüsteem, mille peamiseks kasutusalaks on robootikavaldkond [26]. ROS implementeerib mitmeid operatsioonisüsteemile omaseid funktsionaalsusi, millest olulisemad on riistvara abstraktsioon ning madala taseme seadmete haldamine. Riistvara abstraktsiooni tõttu on võimalik ROS'i jooksutada kõigil toetatud Unixi-põhistel operatsioonisüsteemidel töötavatel arvutitel, mis on ka üheks põhjuseks, mille tõttu ROS laialdast kasutust leiab [26].

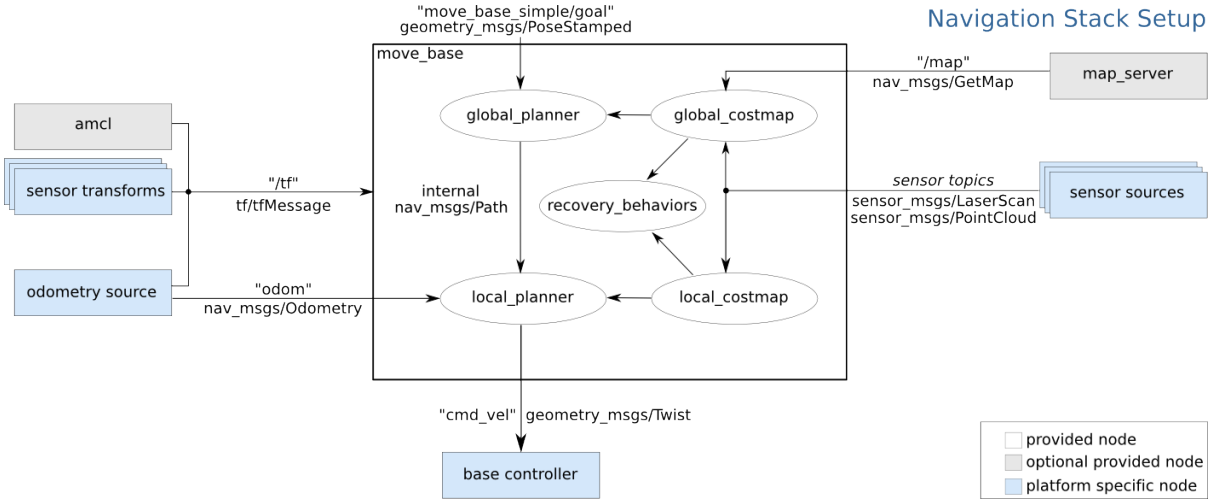
ROS pole traditsioonilises mõttes operatsioonisüsteem, sest ROS ei võimalda sarnast protsesside- ja ajahaldust nagu näiteks Windows või Linux, vaid lisab heterogeensele võõrustaja-operatsioonisüsteemile struktuureeritud suhtluskihi. [26].

ROS on ülesehituselt protsesside ehk sõlmede (ingl *nodes*) raamistik. Sõlmed on protsessid, mis teostavad arvutusi ning suhtlevad omavahel range tüüpmääratlusega sõnumite (ingl *message*) vahendusel. Sõlmed kuulutavad sõnumeid ettenähtud rubriikides (ingl *topics*), millest teised sõlmed sõnumeid tellivad (ingl *subscribe*). Rubriigis võib olla mitmeid kuulutajaid ja tellijaid. Selline suhtlusskeem toetab mitmekesist sõlmede arendust, näiteks võivad sõlmede lähtekoodid olla arendatud erinevatel programmeerimisraamistikel. Mitmekesisus, eelkõige robootika uurimus-ja arendusvaldkondades väljatöötatud lahenduste jagamine ning taaskasutamine analoogsetes projektides, on ka ROS'i peamine eesmärk. [27].

4.2 ROS Navigation

ROS Navigation on tarkvarateekide ja töövahendite komplekt, mis võimaldab ROS'i põhisel robotil liikuda takistusi vältides sihtmärgini [28]. Sisendina saab määrata roboti sihtmärgi (nt move_base_simple/goal joonisel 7 [29]) ning väljundina kuulutatakse rubriigis cmd_vel sõnumeid, mille põhjal robot liikuma hakkab. ROS Navigation liidab moodulite poolt kaardistatud kulukaartide (ingl *costmap*), andurite lugemite, asukohapõhise positsiooneerimise ja odomeetria informatsiooni, et arvutada ohutu teekond sihtmärgini (joonis 7) [28]. Samuti on ROS Navigation'i ülesandeks arvutatud teekonna järgimine ja liikumiskäskluste arvutamine ning täitu-

ritele edastamine. Navigatsiooni kuhi on üheks enimkasutatavaks ROS-i komponendiks, mida enamik mobiilseid roboteid kasutab [30].



Joonis 7. Näide tüüpiliselt ROS Navigation'i ülesehitusest.

ROS Navigation võib koosneda paljudest teekidest, kuid iga navigatsiooni kuhja ülesehituse kuulub asukohastamissüsteem ning liikumissüsteem.

4.2.1 Asukohastamissüsteem

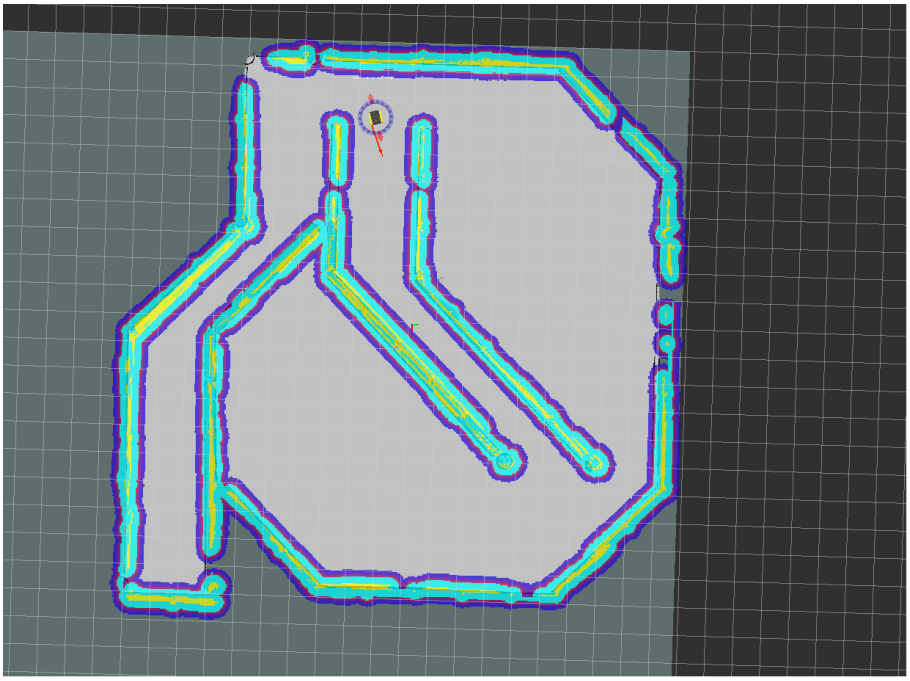
Asukohastamis- või lokaliseerimisüsteemi peamiseks ülesandeks on roboti keskkonna kaardistamine ning roboti oleku hindamine ja asukohastamine kaardistatud keskkonnas.

Töö raames loodud süsteemis kasutati ROS'i gmapping kimpu, mis rakendab SLAM (*Simultaneous Localization and Mapping*) algoritmi tundmatu keskkonna kaardistamiseks roboti külge kinnitatud laseranduri ning odomeetria lugemite abil, samaaegselt robotit selle sama kaardi põhjal asukohastades [31]. Kimp gmapping rakendab OpenSLAM tehnoloogiat, mis on Rao-Blackwellized osakeste filtreerimisalgoritm laseranduri lugemite põhjal võrkkaartide loomiseks [32]. Asukohastamissüsteemi hinnangut edastatakse roboti navigeerimisega tegelevale sõlmele.

4.2.2 Liikumissüsteem

Liikumissüsteemi põhiliseks ülesandeks on arvutada sihtpunktini jõudmiseks vajalikud liikumiskäsklused, mis edastatakse robotile. Töö raames loodud süsteemis kasutati liikumissüstee-

mina ROS'i *move_base_flex* kimpu [33], mis on omakorda kimbul move_base [34] põhinev edasiarendatud versioon. Kõigi move_base põhiste kimpude üheks alamkomponendiks on kulukaartide põhises navigeerimises globaalse ja lokaalse kulukaardi haldamine (joonis 7).



Joonis 8. Näide globaalsest kulukaardist. Värviliselt on tähistatud takistuste paisutused, kus kollane tähistab kõrgeimat ning hall madalaimat hinda.

Globaalne kulukaart (joonis 8 [35]) on roboti andurite lugemite põhjal loodud laialdane keskkonna kujutus. Globaalse kulukaardi põhjal arvutatakse globaalne teekond sihtkohani, mis võtab arvesse teadaolevaid takistusi ja maastikueripärasid. Globaalset kulukaarti kirjutatakse üle vaid juhul kui robot peaks navigeerimisel kuhugi kinni jääma või teekondade arvutamine peaks ebaõnnestuma. [36].

Lokaalne kulukaart on roboti lähiümbruse kujutus, mida uuendatakse sagedamini kui globaalset kaarti. Lokaalsel kulukaardil märgitakse kõik roboti lähiümbruses paiknevad takistused, mille tõttu pole võimalik globaalset teekonda rangelt järgida. [36].

Loodud kulukaarte kasutavad globaalne ja lokaalne plaanur (joonis 7). Globaalse plaanuri eesmärgiks on arvutada globaalsest kulukaardist lähtudes teekond lähtepunktist sihtpunkti, võttes arvesse näiteks staatilist, eeldefineeritud kaarti ja/või eelnevate sihtmärkideni navigeerides kaardistatud takistusi.

Lokaalse kulukaardi põhjal arvutatakse globaalse teekonna järgimist võimaldav lokaalne tee-

kond. Kui globaalne teekond on ettearvutatud kogu tee ulatuses lähtekohast sihtkohta, siis lokaalne teekond võib olla näiteks 2 meetri pikkune teekond, mille eesmärgiks on robot staatilistest ja dünaamilistest takistustest ohutult mööda juhtida, võttes arvesse arvutatud globaalset teekonda. Arvutatud lokaalse raja põhjal arvutatakse raja täitmiseks vajalikud liikumiskäsklused.

4.3 Robotont

Robotont (joonis 9 [37]) on avatud lähtekoodiga omnisuunaliseks liikumiseks võimeline robotiplatvorm. Robotondi peamiseks eesmärgiks on anda robotihuvilistele õppuritele reaalne kogemus raamistiku ROS kasutamises [38].

Joonis 9. Robotiplatvormi „Robotont“ kolmikvaade.

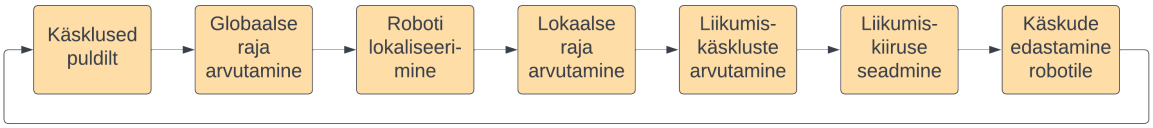
Platvormil Robotont on kasutusel Intel Realsense D435 sügavuskaamera [38], mis on võimeline vaateväljas olevate objektide kauguseid määrama kuni 10 meetri kaugusele [39]. Robotondil on täielik ROS operatsioonisüsteemi tugi [38], lisaks ka eelseadistatud ROS Navigation tugi [40].

5 Lahendus

Käesolev peatükk annab ülevaate süsteemi kontseptsioonist ja ülesehitusest, kirjeldatud töövahendite rakendustest ning süsteemi kasutamiseks mõeldud kasutajaliidestest.

5.1 Süsteemi ülesehitus

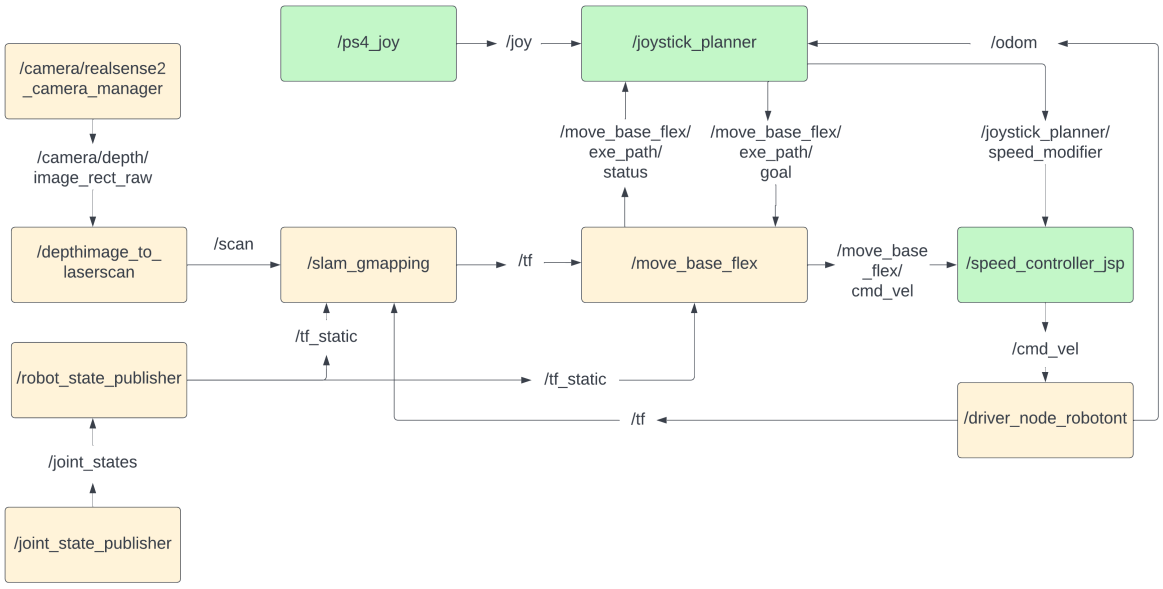
Kasutaja soovib robotit eemalt juhtida keskkonnas, millest kasutajal on piiratud ülevaade. Sellest tulenevalt on roboti liigutamisel suurendatud oht kahjustada robotit või ümbritsevat keskkonda. Süsteemi rakendades saab kasutaja jagada juhtimist robotiga. Kasutaja juhib robotit kasutades pulti, kuid roboti pardal olevad sensorid jälgivad takistusi ning väldivad kokkupõrkeid.



Joonis 10. Töö raames kujundatud süsteemi kontseptsioon.

Süsteem on ülesehitatud manuaalse ning täielikult autonoomse juhtimise ühendamisel (joonis 10). Kasutaja ehk operaatori poolt puldiga antud käskluste põhjal arvutatakse pidevalt globaalset rada, mida edastatakse liikumissüsteemile. Robot asukohastatakse ning lokaalne plaanur arvutab teel olevate takistustega arvestades globaalset rada järgiva lokaalse raja. Seejärel arvutab liikumissüsteem lokaalse raja jälgimiseks vajalikud liikumiskäsud. Roboti liikumiskiirus seatakse vastavusse kasutaja sisendiga ning lõpuks edastatakse käsklused robotile.

Kirjeldatav süsteem on ülesehitatud ROS Navigation teegile. Süsteemi täpsemat ülesehitust moodustavad ROS'i sõlmed on kujutatud joonisel 11. Joonisel on rohelisena kujutatud lokaalses alas ning kollasena kaugalas töötavaid sõlmi. Sõlmi ühendavate joonte vahel on kirjeldatud rubriigid, mille vahendusel sõlmed omavahel suhtlevad.



Joonis 11. Töö raames kujundatud süsteemi ROS sõlmede lihtsustatud diagramm.

5.1.1 Kaameramoodul

Joonisel 11 on kaameramooduli sõlmed tähistatud /camera/realsense2_camera_manager, kaameramooduli alamsõlmesid pole joonisel kujutatud, sest need ei täpsusta relevantselt süsteemi ülesehitust. Kaameramooduli ülesandeks antud süsteemis on käituda laserandurina roboti ümbruskonna kaardistamiseks. Vastava Intel D400 seeria kaamera ning ROS'i vaheliseks suhtluseks kasutati Inteli loodud Intel RealSense ROS Wrapperit [41]. Intel D435 salvestab kaadreid eraldusvõimega 1920x1080 [39], mis tõttu on kaugjuhitava süsteemi korral andmevahetuse ribalaiuse piirangutest tulenevalt tarvilik kasutada kaamera poolt salvestatavate kaadrite suuruse, eraldusvõime ja kaadrite arvu sekundis piiravat komponenti. Alade vahelise andmevahetuse mahu vähendamiseks kasutati d435_throttle sõlme Intel RealSense ROS Wrapper kimbust [41].

5.1.2 /depthimage_to_laserscan

Kaamera poolt hõivatav 3D sügavuspilt on vajalik teisendada asukohastamissüsteemile arusaadavale 2D kujule. Selleks kasutati depthimage_to_laserscan sõlme [42], mis teisendab sügavuskaamera väljastatava sügavuspildi (sensor_msgs/Image [43]) kahemõõtmeliseks laserskanniks (sensor_msgs/LaserScan [44]).

5.1.3 Roboti asend

Roboti asukohastamiseks maailmaruumis on vajalik roboti asendi ning kere oleku pidev uuendamine. Roboti kere asendi ja oleku uuendamiseks on kasutusel joint_state_publisher sõlm [45]. Kasutatud sõlme ülesandeks on roboti liigeste oleku edastamine [45] robot_state_publisher sõlmele, mille peamiseks ülesandeks on roboti oleku kuulutamine tf2 puule [46], kust ülejäänud süsteem roboti olekut jälgida saab. Joonisel 11 on sõlmed tähistatud vastavalt /joint_state_publisher ning /robot_state_publisher.

5.1.4 /slam_gmapping

Asukohastamissüsteemina on loodud süsteemis kasutusel SLAM-printsiibil põhinev sõlm *gmapping* [31]. Asukohastamissüsteemi sisenditeks on sügavuspilti teisendava komponendi poolt loodud sensor_msgs/LaserScan sõnumid ning roboti kere olek robot_state_publisher sõlmelt. Sisendite põhjal toimub robotsüsteemi samaaegne asukohastamine ning ümbruskonna kaardistamine.

5.1.5 /ps4_joy

Juhtpuldi ning ROS'i vaheliseks suhtluseks kasutati ROS'i Linux'ile mõeldud sõlme Joy [47]. Sõlm loeb juhtpuldi nuppude olekuid ning avaldab need sensor_msgs/Joy formaadis [47].

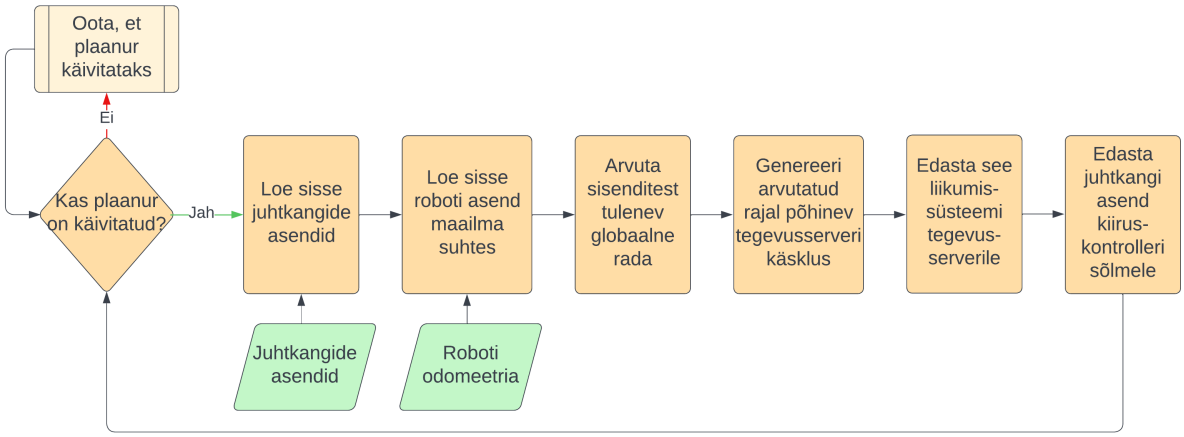
5.1.6 /move_base_flex

Liikumismoodulina kasutati süsteemi loomisel kimpu move_base_flex [33] kulukaartide põhises navigeerimise seadistuses. Move_base_flex pakub võrreldes klassikalise move_base kimbuga lisaks paindlikumale planeerimisele, taastavatele tegevustele ja juhtimisega seotud lahendustele [33] ka võimekamat toiminguserverit (ingl *action server*). Võrreldes move_base kimbuga, mis võimaldab liikumissüsteemile edastada vaid ühe punkti põhiseid navigatsiooni sihtmärke, toetab move_base_flex liikumissüsteemi väliselt loodud eeldefineeritud globaalsete radade järgimist. Move_base_flex võimaldab teistel sõlmedel edastada globaalseid radu mbf_msgs/ExePath sõnumitena läbi tegevusserveri [33].

Liikumismoodul on üles seatud vastavalt joonisel 7 toodud ROS Navigation tüüpülesehitusele.

a) **Globaalne plaanur**

Globaalse plaanurina on implementeeriti JoystickPlanner sõlm, mis võtab sisendiks operaatori juhtpuldi nuppude olekud ja roboti odomeetria ning väljastab sisendite põhjal arvutatud fikseeritud pikkusega globaalse raja.



Joonis 12. Autori implementeeritud sõlme „JoystickPlanner“ tarkvara loogika diagramm.

Arvutatud rada edastatakse pidevalt liikumismoodulile. JoystickPlanner sõlm toetab Sony Playstation süsteemile mõeldud DS3 ning DS4 pulte, kuid vastava nuppude kaardistuse lisamisel sõlme aluskoodi on sõlmega võimalik kasutada ükskõik millist sarnase ülesehitusega pulti. Vasakpoolse juhtkangiga määrab operaator roboti liikumissuuna ning kiiruse. Parempoolse kangiga on võimalik muuta projitseeritava raja nurkkiirust, võimaldades robotile ette anda kurviline rada.

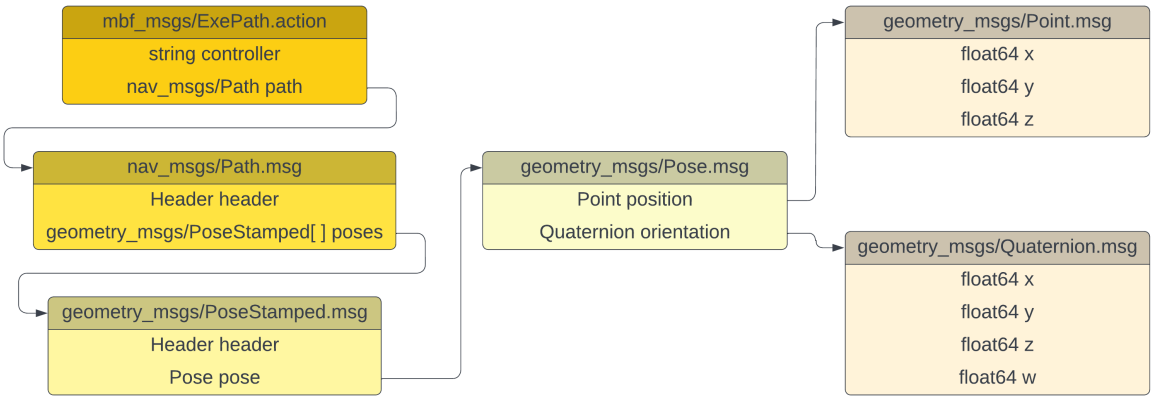
(a) Juhtkangi asend puldil (b) Asendi järgi arvutatud rada

Joonis 13. Näide juhtpuldi kangi asendile vastavast rajast (roheline).

Lisaks juhtkangide sisenditele kasutatakse radade genereerimisel roboti lengerdusnurka vä-

lismaailma suhtes, et juhtkangi asend oleks alati vastavuses roboti esiotsaga (joonis 13).

Radu projitseeritakse kahemõõtmelisele tasandile, mis on kohakuti maapinnaga. mbf_msgs/ExePath sõnumites tähistavad järgitavaid radu nav_msgs/Path sõnumid, mille moodustab massiiv geometry_msgs/PoseStamped sõnumitest. geometry_msgs/PoseStamped sõnumid koosnevad omakorda päisest ning geometry_msgs/Pose sõnumitest. Iga geometry_msgs/Pose sõnum tähistab ühte rada läbivat punkti. Iga selline punkt koosneb kolmest ruumikoordinaadist ning asendit tähistavast kvaternionist. Roboti asendit kirjeldavate kvaternioni koordinaatide väärtused x, y, z võib seada nulliks, sest need ei mõjuta antud implementatsiooni kontekstis raja järgimist. Asendi kvaternioni w-väärtus on seatud iga punkti puhul võrdseks 1-ga. Sõnumite täpsemaid struktuure illustreerib joonis 14.



Joonis 14. mbf_msgs/ExePath sõnumi struktuur.

Kahemõõtmelisest projektsioonist tulenevalt on raja Z-koordinaadi väärtuseks alati 0. Raja paiknemist tasandil kirjeldavad positsiooni P_x ning P_y koordinaadid, mille arvutamiseks kasutatakse valemeid (1) ja (2)

$$P_x = C_x + \cos(\psi + J_{Rx} \cdot t) \cdot t \tag{1}$$

$$P_y = C_y + \sin(\psi + J_{Rx} \cdot t) \cdot t, \tag{2}$$

kus C_x ja C_y tähistavad vastavalt roboti asukohta raja arvutamise hetkel, ψ tähistab raja suuna lengerdusnurka radiaanides roboti esiosa suhtes, J_{Rx} tähistab juhtpuldi parempoolse kangi asendit X-telje suhtes vahemikus operaatori poolt seatud vahemikus ning t tähistab aega.

Lengerdusnurk ψ määrab raja nurga roboti esiosa suhtes. Antud implementatsioonis on võimalik robotit suunata $\pm$90-kraadi roboti otsesihist. Raja lengerdusnurga ψ arvutamiseks on

kasutatud valemit (3)

$$\psi = \Psi + J_{Lx} \cdot \frac{\pi}{2}, \tag{3}$$

kus Ψ tähistab roboti lengerdusnurka välismaailma suhtes ning J_{Lx} tähistab juhtpuldi vasakpoolse kangi asendit X-telje suhtes vahemikus $[-1, 1]$ ühikut.

Projitseeritavad rajad on 2-meetrise fikseeritud pikkusega ning genereeritud radu edastatakse liikumissüsteemile sagedusega 20 Hz. Vasakpoolse kangi asend Y-telje suhtes edastatakse kiiruse kontrolleri moodulile (joonisel 11 „/joystick_planner/speed_modifier“).

b) **Lokaalne plaanur**

Lokaalse plaanurina on süsteemis kasutusel TEB meetodit rakendav teb_local_planner sõlm [48]. Antud sõlme otsustati kasutada plaanuri arvutatavate trajektooride sujuvuse ning kaardistamata aladele planeerimise võimekuse tõttu, kuid süsteem toetab kõiki base_local_planner põhiseid plaanureid. Süsteemi loomisel testiti lokaalse plaanurina ka dwa_local_planner sõlme, kuid teb_local_planner sõlme kasutamisel oli süsteemil parem sihtmärgini jõudmise edukusprotsent ning kontrolltsükkel suutis joosta suurema sagedusega.

Kasutatud plaanuri süsteemikohasele häälestamisele läheneti loogilise katse-eksitusmeetodiga ning kasutatud parameetrid on leitavad töö raames loodud Github repositooriumist [49].

c) **Kulukaartide lisand**

Liikumissüsteemi ülesehituses kasutati roboti ümbruskonna kulukaartide loomiseks kimbu move_base_flex lisandina costmap_2d kimpu [36].

d) **Taastavad tegevused**

Liikumissüsteemi taastavateks tegevusteks seati roboti tõrkumise puhul kulukaartide puhastamise clear_costmap_recovery/ClearCostmapRecovery ning ümber oma keskpunkti pööramise rotate_recovery/RotateRecovery tegevused.

5.1.7 /speed_controller_jsp

Kiiruskontrollerina implementeeriti speed_controller_jsp sõlm, mis vastavalt juhtpuldi vasakpoolse kangi asendile Y-telje suhtes skaleerib roboti liikumiskiirust operaatori eelnevalt defineeritud vahemikus (vaikimisi $0,1 - 0,5 \frac{m}{s}$). Antud moodul on vahekomponendiks roboti kontrolleri (robotont_driver/driver_node) ja liikumissüsteemi (move_base_flex) vahel. Liikumis-

süsteem edastab arvutatud geometry_msgs/Twist sõnumi kiiruskontrollerile, mis skaleerib käsklu-se lineaarset kiirust vastavalt kangi asendile.

Roboti lineaarse liikumiskiiruse seos juhtkangi asendiga on esitatav suhtena, mida esitab valem (4)

$$S_{mod} = (S_{max} - S_{min}) \cdot J_{Ly} + S_{min}, \tag{4}$$

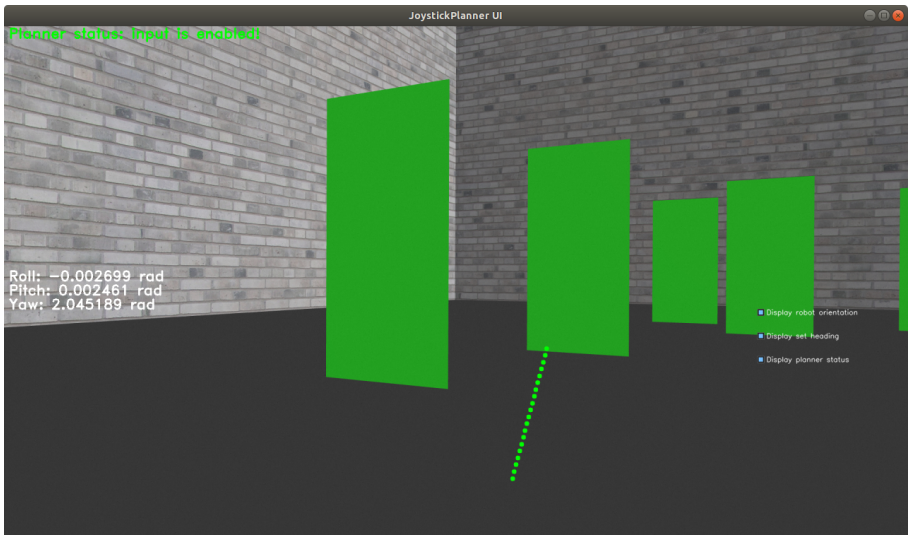
kus S_{max} tähistab roboti maksimaalset liikumiskiirust $\frac{m}{s}$, J_{Ly} tähistab puldi vasakpoolse kangi asendit Y-telje suhtes ning S_{min} tähistab roboti minimaalset liikumiskiirust $\frac{m}{s}$. Modifitseeritud liikumiskäsklused edastab kiiruskontrolleri moodul roboti täituritele.

5.2 Kasutajaliides

Töö raames loodi süsteemi kasutamiseks kaks kasutajaliidese varianti.

5.2.1 JoystickPlannerViz liides

JoystickPlannerViz on autori poolt implementeeritud ROS'i sõlm. Sõlme peamiseks eesmärgiks on kuvada kaamera poolt jäädvustatud kaadreid ekraanil, kuvades samaaegselt erinevat juhtimist abistavat informatsiooni (joonis 15).

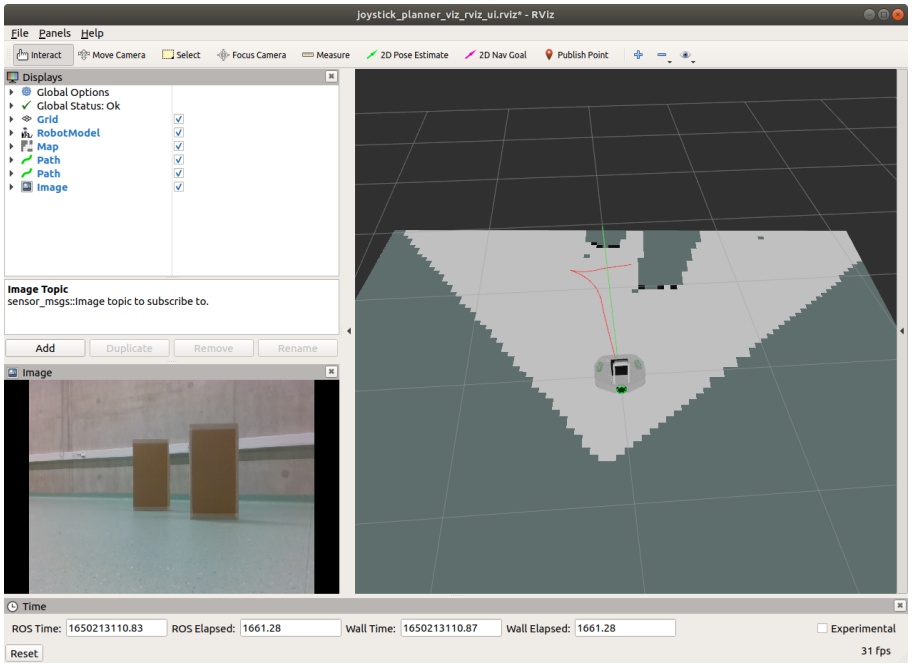


Joonis 15. Autori implementeeritud kasutajaliides süsteemi kasutamiseks.

Lisaks kaamerapildile kuvatakse ekraanil Euler'i nurkadena roboti asendit ruumis, informat-

siooni plaanuri oleku kohta ning juhtpuldi kangide asendite põhjal arvutatud roboti sihti. Kõiki täiendavaid parameetreid on jõudluse parendamiseks võimalik sisse-välja lülitada. Kaadrite kuvamiseks kasutati sõlme implementeerimisel teeki OpenCV [50]. Kaadrite kuvamiseks teegiga OpenCV oli vajalik esmalt kaamera draiverite poolt edastatavad ROS'i sensor_msgs/Image sõnumid teisendada OpenCV poolt toetatud formaati. Selleks kasutati sõlme implementeerimisel cv_bridge teeki [51]. Kasutajaliidesele informatsiooni kuvamise reguleerimise funktsionaalsuse lisamiseks kasutati OpenCV teegile mõeldud lisandit cvui [52].

5.2.2 RViz liides

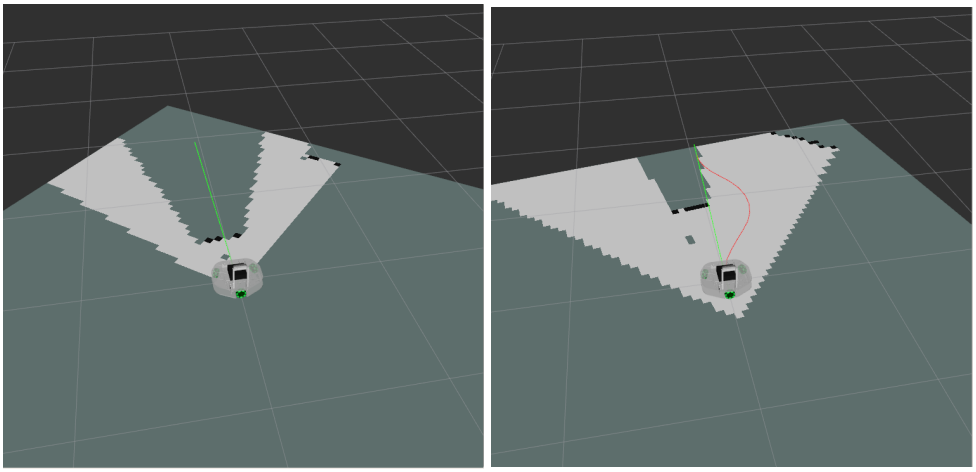


Joonis 16. Kujutis süsteemi testimisest loodud kasutajaliidesest tööriista *RViz* abil.

Süsteemi arendamiseks ja testimiseks loodi ka ROS'i 3D visualiseerimistööriista RViz [53] põhine navigeerimist lihtsustav kasutajaliides. Tööriista abil kuvatakse kasutajale loodud lokaalne kulukaart, millele on paigutatud roboti mudel. Roboti mudeli ning kulukaardi suhtes joonistatakse kaardile kasutaja poolt edastatav globaalne ja lokaalse plaanuri poolt arvutatud lokaalne rada. Virtuaalse keskkonna paremaks seostamiseks reaalse maailmaga kuvatakse kasutajale ka roboti kaamera poolt jäädvustatud kaadreid (joonis 16). Joonisel 16 on rohelise joonena kujutatud globaalset ning punaselt plaanuri poolt arvutatud lokaalset rada.

6 Tulemused ja analüüs

Lõputöö raames kujundatud süsteemi arendamine oli edukas ning süsteem vastab peatükis 3 püstitatud nõuetele. Töö tulemusena arendati diferentsiaal-teljelistele ROS raamistikul ülesehitatud robotitele pidev jagatud juhtimisega juhtimissüsteem. Süsteemi kasutamiseks loodi kaks erinevat kasutajaliidest. Lahendus on vormistatud erinevateks ROS kimpudeks. ROS Navigation *launch* failid ning parameetrid, süsteemi testimiseks loodud Gazebo simulatsioonikeskkonnad ning liikumissüsteemi sõlmed on leitavad GitHub'i koodihoidlast [49].



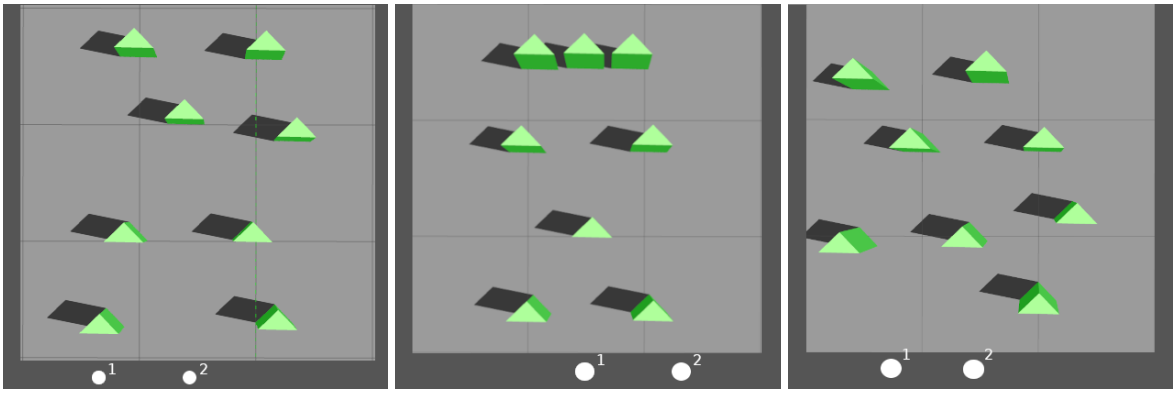
(a) Ilma süsteemita roboti juhtimine (b) Süsteemi abil roboti juhtimine

Joonis 17. Võrdlus süsteemita (paremal) ning süsteemi kasutades (vasakul) roboti juhtimise vahel ühe takistuse näitel. Roheliselt on kujutatud globaalne ning punaselt lokaalne rada.

Süsteemi peamiseks eesmärgiks oli jagada roboti juhtimist operaatori ning roboti enda vahel. Operaator reguleerib roboti suunda ning kiirust kasutades juhtpuldid olevaid kange ja robot väldib autonoomselt teel olevaid takistusi. Võrreldes roboti juhtimist ilma süsteemi ja süsteemi kasutades, sõidaks robot ilma süsteemi kasutamata takistusele otsa. Joonisel 17(a) kujutatud roheline joon illustreerib kasutaja poolt etteantud sõidutrajektoori ning on selgelt näha, et robot sõidaks takistusele otsa. Süsteemi kasutades, mida demonstreerib joonis 17(b), leiab robot kasutaja poolt antud sõidutrajektoorile vastava ohutu tee ümber teel oleva takistuse.

Süsteemi töövõimet ja jõudlust testiti kolme erineva asetusega takistuste labürindiga (joonis 18). Kõik asetused koosnesid kaheksast takistusest, mis olid jaotatud 3x3 m alale.

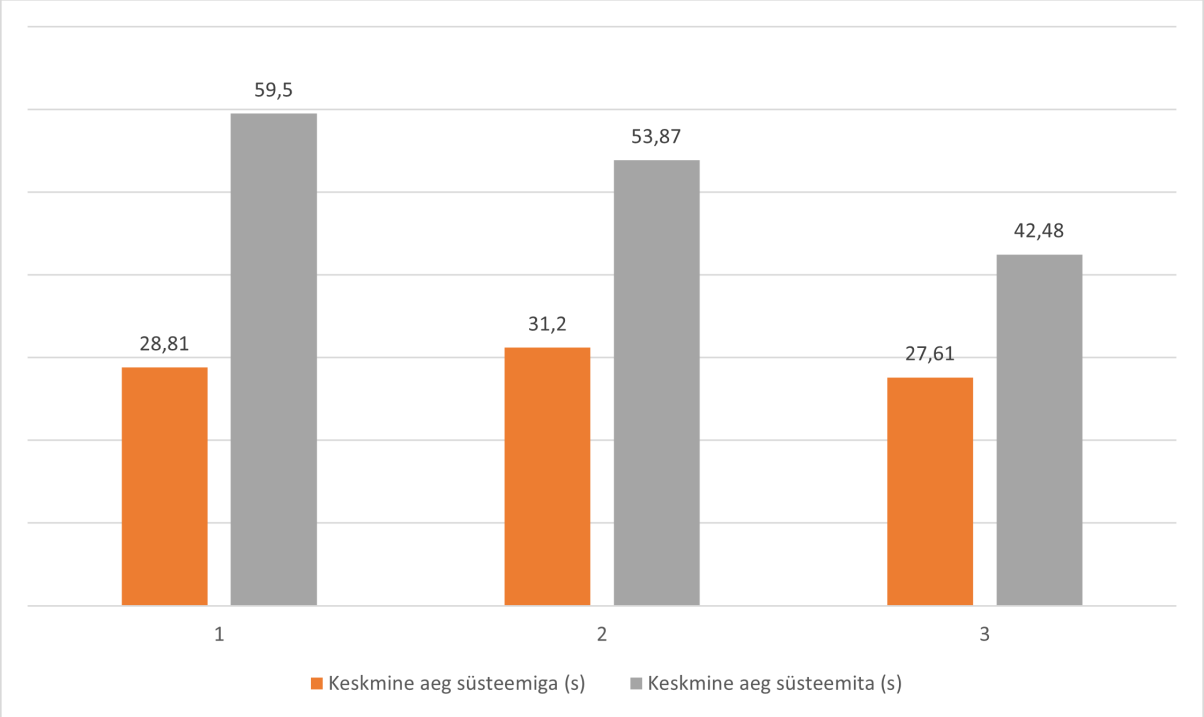
Iga asetuse korral viidi katsed läbi kahe alguspunktiga, alustades igast punktist viiel korral (joonisel 18 valged ringid). Katse loeti lõppenuks, kui robot oli asetuse edukalt läbinud ehk



(a) Esimene takistuste asetus (b) Teine takistuste asetus (c) Kolmas takistuste asetus

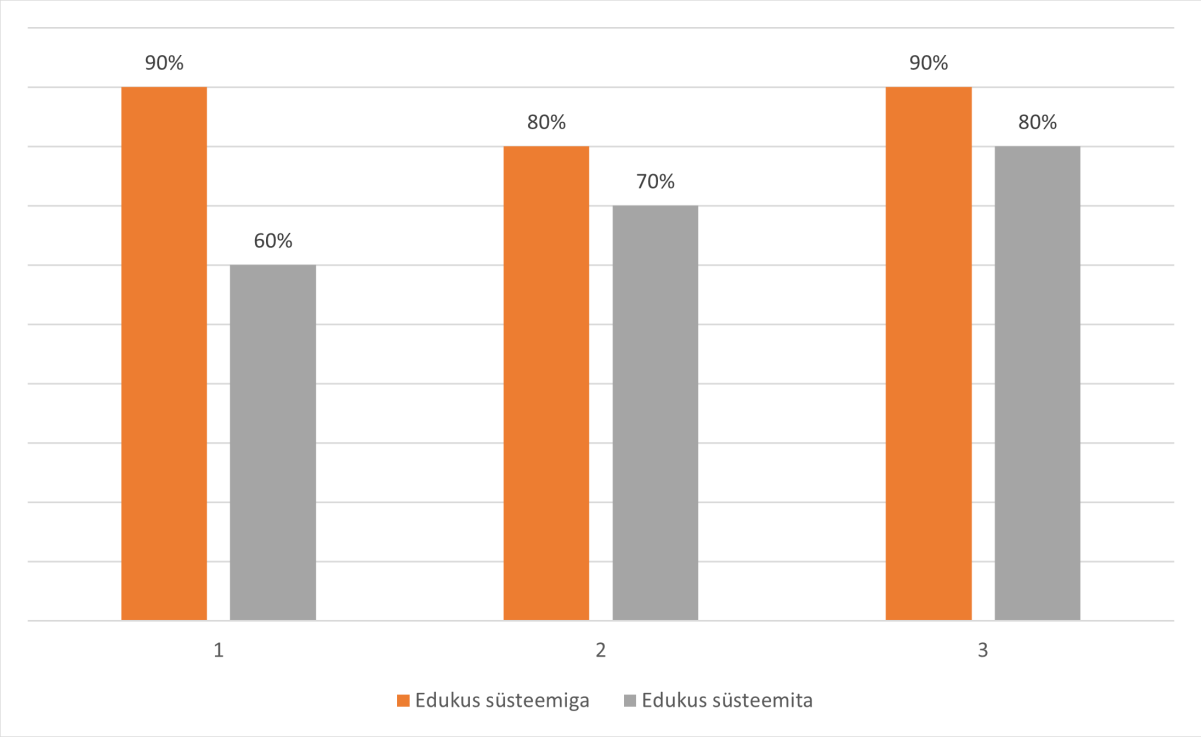
Joonis 18. Süsteemi testimiseks loodud takistuste asetused.

viimasest takistusest möödunud. Vastavalt võrreldi süsteemi edukusprotsenti ning asetuse läbimiseks kulunud aega süsteemi kasutamise korral ning täielikult manuaalselt robotit kaamera pildi abil läbi labürindi juhtides. Katsete tulemused on kujutatud tulpdiagrammides joonistel 19 ja 20.



Joonis 19. Süsteemi testimise ajalised tulemused kolme takistuste asetuse korral.

Katsete tulemuste põhjal oli takistuste asetuste keskmine läbimisaeg süsteemi kasutades ~ 29 sekundit. Süsteemita takistuste asetuste läbimiseks kulus keskmiselt ~ 51 sekundit.



Joonis 20. Asetuste läbimise edukusprotsendid.

Katse loeti mitte-edukaks juhul, kui robot tabas takistust või kui süsteem andis veateate raja planeerimise võimatusest. Süsteemi kasutades oli 30 katse vältel neli mitte-edukat katset, mis annab süsteemile edukusprotsendi $\sim 87\%$. Süsteemita takistusi läbides tabas operaator takistusi üheksal korral, mis annab manuaalse opereerimise edukusprotsentiks $\sim 70\%$.

Süsteemi võimalikuks edasiarenduseks on süsteem muuta 360 kraadiselt juhitavaks, sest hetkel on robotit võimalik suunata vaid 180 kraadise vabadusega roboti esiosast. Süsteemi abil võiks olla võimalik robotit juhtida ka tahapoole ning vastavalt sinna radu planeerida.

7 Kokkuvõte

Robotisüsteemidel on lõpmata palju erinevaid rakendusvaldkoni, kuid need kõik jagavad ühist eesmärki – inimeste elu lihtsustamine. Kuigi enamik robotid on võimelised teatud määral autonoomselt tegutsema, pole need tänapäeval võimelised pikemate ajaperioodide vältel tegutsema. Robotisüsteemide juhtimiseks on seega kujunenud poolautonoomsed lähenemised, mis jagavad juhtimist inimese ning roboti vahel.

Jagatud juhtimine jagab täidetava ülesande alamülesanneteks, mille täitmine jaguneb operaatori ning roboti vahel. Jagatud juhtimine aitab muuta roboti juhtimist täpsemaks ning ohutumaks.

Käesoleva bakalaureuse töö raames uuriti olemasolevaid võimalusi mobiilse roboti kaugjuhtimiseks, kujundati kaugjuhtimissüsteem jagatud juhtimise põhimõttel ning hinnati selle kasutuskõlblikust ja jõudlust.

Töö tulemusena valminud süsteemi lähtekood, testimiseks loodud keskkonnad ning parameetrid võib leida töö raames loodud GitHub'i repositooriumist [49].

Kasutatud kirjandus

[1] *Roomba® Robot Vacuum Cleaners | iRobot®;* (i.a.) URL: https://www.irobot.com/roomba (vaadatud 28.04.2022).

[2] *Industrial Robots Portfolio | ABB Robotics*. (i.a.) URL: https://new.abb.com/products/robotics/industrial-robots (vaadatud 28.04.2022).

[3] *Auve Tech | Iseauto R&D*. (i.a.) URL: https://auve.tech/rd/ (vaadatud 28.04.2022).

[4] George A Bekey. *Autonomous robots: from biological inspiration to implementation and control*. MIT press, 2005.

[5] David A. Abbink *et al.* „A Topology of Shared Control Systems—Finding Common Ground in Diversity". *IEEE Transactions on Human-Machine Systems* 48.5 (2018), lk. 509–525. DOI: 10.1109/THMS.2018.2791570.

[6] George P. Moustris, Andreas I. Mantelos ja Costas S. Tzafestas. „Shared control for motion compensation in robotic beating heart surgery" (2013), lk. 5819–5824. DOI: 10.1109/ICRA.2013.6631414.

[7] Satya Veera Pavan Kumar Maddukuri *et al.* „A low cost sensor based autonomous and semi-autonomous fire-fighting squad robot" (2016), lk. 279–283. DOI: 10.1109/ISED.2016.7977097.

[8] Vitalii Pruks, Kwang-Hyun Lee ja Jee-Hwan Ryu. „Shared Teleoperation for Nuclear Plant Robotics Using Interactive Virtual Guidance Generation and Shared Autonomy Approaches" (2018), lk. 91–95. DOI: 10.1109/URAI.2018.8441814.

[9] Günter Niemeyer, Carsten Preusche ja Gerd Hirzinger. „Telerobotics". Teoses: *Springer Handbook of Robotics*. Toim. Bruno Siciliano ja Oussama Khatib. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, lk. 741–757. ISBN: 978-3-540-30301-5. DOI: 10.1007/978-3-540-30301-5_32. URL: https://doi.org/10.1007/978-3-540-30301-5_32.

[10] Barzin Doroodgar, Yugang Liu ja Goldie Nejat. „A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims". *IEEE Transactions on Cybernetics* 44.12 (2014), lk. 2719–2732.

[11] Zhengming Huang *et al.* „Teleoperate system of underwater cleaning robot based on HUD" (2017), lk. 2675–2679. DOI: 10.1109/ASCC.2017.8287599.

[12] Lawrence Stark *et al.* „Telerobotics: Problems and research needs". *IEEE transactions on aerospace and electronic systems* 24.5 (1988), lk. 542–551.

[13] *Telerobotics - Scientific Figure on ResearchGate: R. C. Goertz demonstrating his teleoperation interface*. (i.a.) URL: https://www.researchgate.net/figure/Raymond-C-Goertz-used-electrical-and-mechan%5C_fig1%5C_227154177 (vaadatud 23.03.2022).

[14] Raymond C Goertz ja William M Thompson. „Electronically controlled manipulator". *Nucleonics (US) Ceased publication* 12 (1954).

[15] Indra Adji Sulistijono *et al.* „Teleoperated Food and Medicine Courier Mobile Robot for Infected Diseases Patient" (2021), lk. 620–625. DOI: 10.1109/TENCON54134.2021.9707377.

[16] Ramviyas Parasuraman *et al.* „A new ugv teleoperation interface for improved awareness of network connectivity and physical surroundings". *arXiv preprint arXiv:1710.06785* (2017).

[17] *The Britannica Dictionary: cruise control*. (i.a.) URL: https://www.britannica.com/dictionary/cruise-control (vaadatud 24.03.2022).

[18] *Telerobotics - Scientific Figure on ResearchGate: Different concepts for telerobotic control architectures*. (i.a.) URL: https://www.researchgate.net/figure/Different-concepts-for-telerobotic-control-architectures%5C_fig5%5C_227154177 (vaadatud 24.03.2022).

[19] Mauricio Marcano *et al.* „A Review of Shared Control for Automated Vehicles: Theory and Applications“. *IEEE Transactions on Human-Machine Systems* 50.6 (2020), lk. 475–491. DOI: 10.1109/THMS.2020.3017748.

[20] Bingqing Zhang, Catherine Holloway ja Tom Carlson. „A hierarchical design for shared-control wheelchair navigation in dynamic environments“ (2020), lk. 4439–4446. DOI: 10.1109/SMC42975.2020.9282838.

[21] Ko Matsuhiro *et al.* „Study on safe operation architecture in semi-autonomous remote control system for mobile robots using mobile communications“ (2022), lk. 950–956. DOI: 10.1109/SII52469.2022.9708784.

[22] Olimpiya Saha ja Prithviraj Dasgupta. „A Comprehensive Survey of Recent Trends in Cloud Robotics Architectures and Applications“. *Robotics* 7.3 (2018). ISSN: 2218-6581. DOI: 10.3390/robotics7030047. URL: https://www.mdpi.com/2218-6581/7/3/47.

[23] James Kuffner. „Cloud-enabled humanoid robots“. *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, Nashville TN, United States, Dec.* (2010). URL: https://ci.nii.ac.jp/naid/10031099795/en/.

[24] Hong Wang *et al.* „Ethical Decision Making in Autonomous Vehicles: Challenges and Research Progress“. *IEEE Intelligent Transportation Systems Magazine* 14.1 (2022), lk. 6–17. DOI: 10.1109/MITS.2019.2953556.

[25] Azim Shariff, Jean-François Bonnefon ja Iyad Rahwan. „Psychological roadblocks to the adoption of self-driving vehicles“. *Nat Hum Behav* 1 (oktoober 2017), lk. 694–696. DOI: https://doi.org/10.1038/s41562-017-0202-6. URL: https://www.nature.com/articles/s41562-017-0202-6.

[26] *ROS Documentation: Introduction*. (i.a.) URL: http://wiki.ros.org/ (vaadatud 08.03.2022).

[27] Morgan Quigley, Brian Gerkey ja William D Smart. *Programming Robots with ROS: a practical introduction to the Robot Operating System*. O'Reilly Media, Inc., 2015.

[28] *ROS Wiki: navigation*. (i.a.) URL: http://wiki.ros.org/navigation (vaadatud 15.05.2022).

[29] *ROS Wiki: Navigation stack setup*. (i.a.) URL: http://wiki.ros.org/navigation/Tutorials/RobotSetup?action=AttachFile%5C&do=view%5C&target=overview%5C_tf.png (vaadatud 23.03.2022).

[30] Morgan Quigley, Brian Gerkey ja William D Smart. *Programming Robots with ROS: a practical introduction to the Robot Operating System*. O'Reilly Media, Inc., 2015, lk. 157.

[31] *ROS Wiki: gmapping*. (i.a.) URL: http://wiki.ros.org/gmapping (vaadatud 13.03.2022).

[32] *OpenSLAM: Give your algorithm to the community*. (i.a.) URL: https://openslam-org.github.io/gmapping.html (vaadatud 13.03.2022).

[33] Sebastian Pütz, Jorge Santos Simón ja Joachim Hertzberg. „Move Base Flex: A Highly Flexible Navigation Framework for Mobile Robots“. Teoses: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Software available at https://github.com/magazino/move_base_flex. Oktoober 2018. URL: https://github.com/magazino/move_base_flex.

[34] *ROS Wiki: move_base*. (i.a.) URL: http://wiki.ros.org/move%5C_base (vaadatud 15.05.2022).

[35] *ROS Navigation Basics - ROS Tutorials 0.5.2 Documentation*. (i.a.) URL: https://www.clearpathrobotics.com/assets/guides/melodic/ros/ROS%5C%20Navigation%5C%20Basics.html (vaadatud 13.03.2022).

[36] *ROS Wiki: costmap_2d*. (i.a.) URL: http://wiki.ros.org/costmap%5C_2d (vaadatud 13.03.2022).

[37] *Robotont: Avatud robotiplatvorm hariduses, teaduses ja tööstuses*. (i.a.) URL: http://robotont.ut.ee/ (vaadatud 14.03.2022).

[38] Renno Raudmäe. „Avatud robotplatvorm Robotont“. Magistritöö. Tartu: Tartu Ülikool, juuni 2019.

[39] *Intel® Realsense™ Depth Camera D435*. (i.a.) URL: https://www.intelrealsense.com/depth-camera-d435/ (vaadatud 06.04.2022).

[40] *Overview - Robotont 0.0.1 documentation*. (i.a.) URL: https://robotont.github.io/html/files/overview.html (vaadatud 15.05.2022).

[41] *Github: ROS Wrapper for Intel® RealSense™ Devices*. 2022. URL: https://github.com/IntelRealSense/realsense-ros (vaadatud 06.04.2022).

[42] *ROS Wiki: depth_image_to_laserscan*. (i.a.) URL: http://wiki.ros.org/depthimage%5C_to%5C_laserscan (vaadatud 06.04.2022).

[43] *ROS Docs: sensor_msgs/Image*. (i.a.) URL: http://docs.ros.org/en/noetic/api/sensor%5C_msgs/html/msg/Image.html (vaadatud 15.05.2022).

[44] *ROS Docs: sensor_msgs/LaserScan*. (i.a.) URL: http://docs.ros.org/en/api/sensor%5C_msgs/html/msg/LaserScan.html (vaadatud 06.04.2022).

[45] *ROS Wiki: joint_state_publisher*. (i.a.) URL: http://wiki.ros.org/joint%5C_state%5C_publisher (vaadatud 06.04.2022).

[46] *ROS Wiki: robot_state_publisher*. (i.a.) URL: http://wiki.ros.org/robot%5C_state%5C_publisher (vaadatud 15.05.2022).

[47] *ROS Wiki: joy*. (i.a.) URL: http://wiki.ros.org/joy (vaadatud 06.04.2022).

[48] *ROS Wiki: teb_local_planner*. (i.a.) URL: http://wiki.ros.org/teb%5C_local%5C_planner (vaadatud 13.03.2022).

[49] Markus Erik Sügis. *GitHub: joystick_planner_navigation*. 2022. URL: https://github.com/sygism/joystick_planner_navigation (vaadatud 22.04.2022).

[50] *OpenCV veebileht*. (i.a.) URL: https://opencv.org/ (vaadatud 15.04.2022).

[51] *ROS Wiki: cv_bridge*. (i.a.) URL: http://wiki.ros.org/cv_bridge (vaadatud 15.04.2022).

[52] *cvui - (very) simple UI lib built on top of OpenCV drawing primitives*. (i.a.) URL: https://dovyski.github.io/cvui/ (vaadatud 15.04.2022).

[53] *ROS Wiki: rviz*. (i.a.) URL: http://wiki.ros.org/rviz (vaadatud 16.04.2022).

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Markus Erik Sügis,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose,

   "Pideva kaugjuhtimissüsteemi kujundamine jagatud juhtimise põhimõttel",

   mille juhendajad on Karl Kruusamäe ja Arun Kumar Singh, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons litsensiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsensi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Markus Erik Sügis
22.04.2022