

Tartu Ülikool

Loodus- ja täppisteaduste valdkond

Tehnoloogiainstituut

Georg Astok

Roboti juhtimine virtuaalreaalsuses kasutades ROS-raamistikku

Bakalaureusetöö (12 EAP)

Arvutitehnika eriala

Juhendajad:

robotika dotsent Karl Kruusamäe

nooremteadur Robert Valner

Tartu 2019

1. Resümee/Abstract

Roboti juhtimine virtuaalreaalsuses kasutades ROS-raamistikku.

Robotite kasutamine on muutunud vältimatuks tänapäeva maailmas. Nende intuitiivsemaks juhtimiseks on võimalus kasutada virtuaalreaalset (VR) keskkonda. ROS-raamistik on robotikas laialdaselt kasutusel. Raamistikus on olemas mitmed huvitavad kimbud, programmid ja tööriistad, mis aitavad virtuaalreaalsust luua. Praegusel hetkel puudub ROS-kimp, mis kasutaks ROSi võimalusi ja looks ainult ROSi põhineva kasutajaliidese.

Selle bakalaureusetöö eesmärk on uurida erinevaid ROS-raamistiku võimalusi virtuaalreaalsuse loomiseks, luua roboti juhtimiseks virtuaalreaalsuse kasutajaliidest sisaldav ROS-kimp ja seda testida. Kasutajaliidese loomisel kasutatakse maksimaalselt olemasolevaid ROSi vahendeid nagu RViz. Loodud keskkond kuvatakse omakorda kasutaja peas olevasse OSVR visiiri. Kasutaja interakteerub robotiga kasutades Leap Motion kontrolleri (LM-kontroller), mida kasutades on kasutaja käed vabad.

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia; T125 Automatiseerimine, robotika, control engineering;

Märksõnad: ROS, Leap Motion, virtuaalreaalsus, OSVR, virtuaalreaalne kasutajaliides

Creating virtual reality user interface using only ROS framework.

Using robots is unavoidable in modern world. We can use virtual reality (VR) to control them more intuitively. ROS framework is widely used in robotics. The framework has a lot of packages, programs and tool to build VR, but we are missing a ROS package that takes all these parts and puts them all together.

The purpose of this thesis is to research different ROS framework opportunities on creating virtual reality, to build a virtual reality user interface and to test it. User interface is created using visualization markers that are presented in ROS visualization program RViz. Using RViz plugin, OSVR headset is used as a head mounted display (HMD). User can interact with the package using Leap Motion controller, which will keep user's hands free.

CERCS: T120 Systems engineering, computer technology; T125 Automation, robotics, control engineering;

Keyword: ROS, Leap Motion, virtual reality, OSVR, virtual reality user interface

Sisukord

1. Resümee/Abstract	2
Sisukord	3
Lühendid, konstandid, mõisted	5
2. Sissejuhatus	6
3. Kirjanduse ülevaade	7
3.1. Virtuaalreaalsus ja selle saavutamine	7
3.2. Kaasaegsed lahendused	9
3.2.1. Visiirid ehk HMD	9
3.2.2. Kontrollerid ehk HID	10
3.3. ROS (Robot Operating System) ning VR	13
3.3.1. ROSi üldkirjeldus	13
3.3.2. ROS ja virtuaalreaalsus	14
3.4. Leap Motion kontroller	18
4. Eesmärk ja vajalikkus	20
5. VR kasutajaliides	21
5.1. Nõuded ja kontseptsioon	21
5.2. Disain	22
5.3. Seadmete vaheline suhtlus	23
5.4. Tarkvaralised lahendused	24
5.5. VR kasutajaliidese kirjeldus	26
5.5.1. Tähised	26
5.5.2. Roboti juhtimine	28
5.5.3. Roboti soovitud poosi määramine	30
5.5.4. Tundlikkusliugur	30
5.5.5. Alammenüü	30
5.5.6. Roboti mõjuri suuna muutmine ja suuna tähis	31
5.5.7. Roboti teekonna planeerimine	31
5.5.8. Roboti liigutamine	32
5.5.9. Kasutajavaate muutmine	32

5.6.	Koordinaadistik	33
5.7.	Lahenduse käivitamine	34
6.	Peamised tehnilised järeldused	36
6.1.	Probleemid Leap Motion kontrolleriiga	36
6.1.1.	LM-kontroller OSVRi küljes	36
6.1.2.	LM-kontroller kaelas	37
6.1.3.	Käte asend LM-kontrolleri suhtes	38
6.1.4.	Üldine käte tuvastamine	38
6.2.	Probleemid ROSi ja lahendusega	38
6.2.1.	Piiratud tööpiirkond	38
6.2.2.	Roboti mõjuri suuna määramine	39
6.2.3.	Tundlikkuse muutmine	40
6.2.4.	Mugavus	40
6.3.	Üldised probleemid	40
7.	Kokkuvõte	41
8.	Viited	42
	Lihtlitsents	48

Lühendid, konstandid, mõisted

HID (*human interface device*) – kasutaja kontrollitav seade, mis on süsteemile sisendiks

HMD (*head mounted display*) – pähe paigaldatav virtuaalreaalsuse komplekt

IMS Lab (*Intelligent Materials and Systems Lab*) – Tartu Ülikooli Tehnoloogiainstituudi arukate materjalide ja süsteemide labor

LM-kontroller – Leap Motion kontroller

OSVR (*Open Source Virtual Reality*) – avatud lähtekoodiga HMD

ROS (*Robot Operating System*) – roboti kontrollimise raamistik

RViz (*ROS Visualization*) - ROSi programm, mis võimaldab visualiseerida erinevaid sõnumeid, mudeleid ja tähiseid

UT (*University of Tartu*) – Tartu Ülikool

VR (*Virtual Reality*) – virtuaalreaalsus

2. Sissejuhatus

Tavainimene mõtleb üldjuhul virtuaalreaalsust (VR) eelkõige meelelahutuse kontekstis, näiteks erinevad VR mängud. Inimestel puudub suurem arusaam, mis VR on ja et seda on võimalik kasutada ka paljudes teistes eluvaldkondades.

Üks valdkondadest, kus VR-i kasutatakse on robotika, kus sellel on tohutult suur potentsiaal. Juba praegu kasutatakse virtuaalreaalsust robotikas võrdlemisi palju, kuid järgnevate aastatega kasvab see eeldatavasti veelgi. Näiteks kasutatakse seda erinevate robotite kaugjuhtimisel.

ROS-raamistik on robotite juhtimiseks mõeldud ja on väga laialdaselt kasutatud raamistik. Raamistikus on olemas mitmeid erinevaid kimppe, tööriistu ja programme, mis toetavad VR-i loomist. Virtuaalreaalsus ise aga tekitatakse kasutades raamistikust välja poole jäävaid programme või raamistikke. Hetkel puudub ainult ROS-raamistikul põhinev virtuaalreaalne kasutajaliides. Mõned projektid on küll arendamisel, nt ARViz [1] ja Rvis2AR [2]. Ainult ROS-raamistikul põhinev rakendus tagab projekti kompaktsuse ja parema suhtluse erinevate osade vahel.

Käesoleva bakalaureusetöö eesmärk on analüüsida erinevaid ROSi võimalusi VR keskkonna loomiseks ning luua ROS-kimp, mis võimaldaks kuvada virtuaalreaalset kasutajaliidest. Kasutajaliidesega on võimalik planeerida roboti liikumise teekonda ning liigutada robotit. Samuti on olemas lisavõimalused, mis toetavad roboti kontrollimist, nt tundlikkuse liugur, kasutajavaate muutmise ja mõjuri suuna määramise võimalused. Loodav kimp kasutab väljundseadmena OSVR visiiri ja sisendseadmena Leap Motion kontrolleri. Loodud kimp testitakse läbi ja tehakse avalikult kättesaadavaks UT IMS Robotics GitHubi koodihoidlas.

3. Kirjanduse ülevaade

3.1. Virtuaalreaalsus ja selle saavutamine

Virtuaalreaalsus (VR) on arvuti poolt loodud tehislik kolmedimensiooniline keskkond või reaalsus, mis tekitab kasutajale tunde, et ta viibib füüsiliselt selles keskkonnas [3].

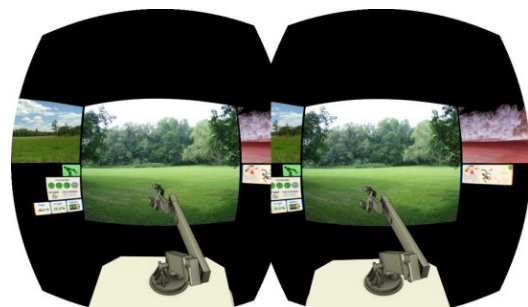
Virtuaalreaalsuse tekitamiseks on kasutatakse erinevaid sisend- ja väljundseadmeid. Väljundseadmeks on visiir ehk HMD (ingl *head mounted display*), millega visualiseeritakse kasutajale VR keskkond, ning sisendseadmeteks erinevad kasutaja poolt kontrollitavad seadmed ehk HID (ingl *human input device*), mille abiga kasutaja interakteerub VR keskkonnaga [4].

VR-i visuaalseks kogemiseks asetatakse kasutaja mõlema silma ette ekraanid, millel kuvatakse virtuaalset keskkonda. Sügavustaju tekitamiseks kuvab kumbki ekraan sama keskkonda eri vaatenurgast, mis simuleerib kahe silma lahususest tulenevat parallaxi [5]. Selleks, et kasutajale loodud kogemus oleks reaalsem, kasutatakse ka heli, liikumisplatvorme ning erinevaid haptilisi seadmeid [4].

Sõltuvalt VR-i rakendusest, on kasutaja jaoks väga oluline, mida väljundseadmetes kuvatakse ja kuidas kasutaja interakteerub virtuaalkeskkonnaga. Kui edastada kasutajale liiga vähe informatsiooni, siis ei pruugi saadav kogemus olla intuitiivne ega päriselus saadava kogemusega võrreldav. Teisalt, kui anda kasutajale liiga palju informatsiooni, võib see kasutajat liigselt koormata ja segadusse ajad.



(a)



(b)

Joonis 3.1. TAROS roboti juhtimiseks loodud kaks lahendust - (a) kaamera pildi otse silmade ette kuvamine ja (b) virtuaalse kontrollruumi kasutamine [4].

Samuti on tähtis, kuidas informatsiooni esitatakse. Kõige lihtsam viis on roboti kaamerapildi otse kasutaja silmade ette kuvamine [4] (joonis 3.1-a) . Kuna aga roboti kaamera asendit ei kontrollita HMD liigutamise abil vaid kaugjuhtimispuldiga, siis esineb kasutajatel peeringlus ja iiveldus. Selle vähendamiseks kasutatakse nn kontrollruumi lahendust, kus kaamerapildid kuvatakse virtuaalreaalsuse abil loodud kolmemõõtmelise virtuaalsesse juhtimiskeskusse asetatud paneelidele (joonis 3.1-b). Selline lahendus vähendab iiveldustunnet [6] ning tõstab märgatavalt keerukate ülesannete täitmise õnnestumist, näiteks Lego klotside kokkupanemist [7].

VR-i kasutamisel robotikas on mitmeid plusse ja see täidab mitmeid erinevaid ülesandeid. Mõned sellised positiivsed küljed on:

- robotite jaoks mõeldud juhtimisruumid on keerukad ning kallid, virtuaalsed kontrollkeskused on palju odavamad [6]
- robotit on võimalik opereerida eemalt, inimelu ohtu panemata [6] [8]
- VR-i on saab kasutada treenimiseks [9]
- VR on intuitiivne [9]
- VR-s on võimalik asju lihtsustada, näiteks eemaldades kasutajat segavad elemendid [9].

Erinevates teadusprojektides läbi katsetatud tegevused ja nende potentsiaalne kasu:

- kirurgilise roboti juhtimine - väikeste ja täpsust vajavate operatsioonide teostamine [10]
- Lego klotside kokkupanemine - detailide kokkupanek näiteks ehitusel [5]
- torude kokku keeramine - täpsust vajavate detailide kokku panemine näiteks ehitusel [11]
- miltitaarse mehitanud maismaasõiduki juhtimine - roboti juhtimine lahinguväljal [6]
- keevitamine - eemalt täpsete keevituste tegemine [12]
- kulbi liigutamine potis, harja ja kühvli kasutamine, särgi voltimine, klaviatuuril nupu vajutamine - inimtegevuse jäljendamine [13]
- ruumi kaardistamine - automaatne maastiku ja ruumide kaardistamine [14].

3.2. Kaasaegsed lahendused

3.2.1. Visiirid ehk HMD



(a)



(b)



(c)

Joonis 3.2. Mõned näited kaasaegsetest HMD-dest - (a) HTC Vive [15], (b) Oculus Rift [16] ja (c) OSVR HDK2 [17].

Kõige elementaarsem seade VR-i tekitamiseks on kuvarprillid ehk HMD (ingl *head-mounted display*), mis kujutavad endast 1 või 2 ekraaniga visiiri. Hetkel on saadaval suur valik erinevaid HMD-sid. HMD-l on üldjuhul sisseehitatud ekraan ning andurid, mis tagavad informatsiooni pea liikumise kohta [18]. HMD-d grupeeritakse statsionaarseteks, mis vajavad toimimiseks välist arvutit, ning juhtmevabadeks, millel on sisseehitatud arvuti [4]. Tuntumad turul leitavad VR HMD-d on HTC Vive [19] (joonis 3.2-a), Oculus Rift [20] (joonis 3.2-b) ja OSVR HDK2 [21] (joonis 3.2-c).



(a)



(b)

Joonis 3.3. Mõned nutitelefonide jaoks mõeldud VR-kestad - (a) Samsung Gear VR [22] ja (b) Zeiss VR One [23].

Tootjad nagu Samsung (joonis 3.3-a) ja Zeiss (joonis 3.3-b) pakuvad võimalust kasutada nutitelefoni HMD-na. Sellisel juhul paigutatakse nutitelefoni selleks mõeldud kesta, mis omakorda asetatakse kasutaja pähe, silmade ette [4]. Selle tulemuseks on juhtmevaba HMD.

3.2.2. Kontrollerid ehk HID

Mõeldes VR-i peale, tuleb inimestel tihti meelde ainult väljundseade nagu näiteks HMD, kuid tihti jäetakse kõrvale sisend ehk viis, kuidas VR-ga suhelda: nt VR keskkonnas edasi liikuda või sealseid objekte mõjutada. Selliseid sisendseadmeid nimetatakse kontrolleriteks ehk HID (ingl *human input device*).

Kõige lihtsam kontroller on arvuti klaviatuur, kus süsteemi kontrollitakse näiteks nooleklahve vajutades. Sellise lahenduse kasuks räägib selle lihtsus ja soodsus, samas on see piiratud rakendustega ja kasutaja ei saa käsi vabalt liigutada [24].



(a)



(b)



(c)

Joonis 3.4. (a) *Sony PlayStation DualShock* [25], (b) *Microsoft Xbox Wireless* [26] mängupuldid ja (c) *Thrustmaster T-Flight Stick X* [27] juhtkang.

Mängupult on kontrolleri, mida hoitakse käes ja millega kontrollimine toimub nuppude ja kangidega [28]. Mängupulte on juhtmega ja juhtmevabasisid. Head näited sellistest pultidest on Sony PlayStation DualShock [29] (joonis 3.4-a), Microsoft Xbox Wireless [30] (joonis 3.4-b) ning Logitech F710 [31]. Mängupulte on lihtne kasutada, neid on lihtne ühendada VR-ga (näiteks Xbox Wirelessi ühendamine Samsung Gear VR-ga [32]) ning kasutaja saab ruumis vabalt ringi liikuda, samas ei tagasta pult informatsiooni enda täpse liikumise ja asukohta kohta. Juhtkang (ingl *joystick*) on kangiga mängupult, mis sobib väga hästi erinevate sõidukite ja masinate juhtimiseks, näiteks lennusimulaatorite juhtimiseks [33]. Üks sellistest juhtkangidest on Thrustmaster T-Flight Stick X [34] (joonis 3.4-c). Uuringud on näidanud, et mõningaid süsteeme, näiteks allveerobotit, on lihtsam kontrollida juhtkangiga [8].



(a)



(b)



(c)

Joonis 3.5. Mõned näited HID-dest - (a) Oculus Touch [27], (b) HTC Vive Controller [28] ja (c) PlayStation VR Aim [29].

Paljudel HMD-del eksisteerivad neile disainitud kontrollid. Sellised puldid on näiteks Oculus Touch [35] (joonis 3.5-a) ja HTC Vive Controller [36] (joonis 3.5-b). Lisaks tavalisele disainile on olemas ka kindlale rakendusele suunatud kontrollid, nt relvakujuline PlayStation VR Aim tulistamismängudele [37] (joonis 3.5-c). Selliste kontrollite positiivseteks külgedeks on hea sobitumine HMD-ga, täpsus ning asjaolu, et kasutaja käed saavad ruumis vabalt ringi liikuda. Negatiivseks küljeks on aga hind (umbes 115 eurot [38]) ning et kasutaja peopesad ei ole tühjad. Olukorras, kus kasutaja peopesad on vabad, tunnetab kasutaja oma käsi paremini. Seda sellepärast, et tal on võimalik näha oma käsi loomulikumana ja võimalus kasutada liigutusi, mis muidu oleks võimatud, nagu näiteks näpistusliigutus.

Kaamerapõhised kontrollid kasutavad ühte või rohkemat kaamerat ning pilditöötlust VR-i juhtimiseks. Kõige lihtsam selline lahendus on tavaline veebikaamera, kus pilditöötlustarkvara, nt OpenPose [39], tuvastab pildilt kasutaja keha, jäsemed ja pea.

Veebikaamera kasutamise positiivne külg on selle soodne hind ning kasutaja käte liikumisvabadus.



(a)



(b)

Joonis 3.6. Sügavuskaamerad (a) Microsoft Kinect [40] ja (b) Intel RealSense D435 [41].

Sügavuskaamera on seade, mis koosneb kahest kaamerast ja infrapuna projektorist ning mis tekitab kolmedimensioonilise sügavuskaardi [42]. Head näited sellistest kaameratest on Microsoft Kinect [40] (joonis 3.6-a) ja Intel RealSense D435 [41] (joonis 3.6-b). Sügavuskaamerate headeks külgedeks on kasutaja liikumisvabadus ning sügavuskaart. Leap Motion kontrolleri [43] on sügavuskaamera, mis jälgib ainult kasutaja käte asendit, liikumist ja žeste. Leap Motion kontrolleri positiivseks küljeks on selle lihtsus ning kasutaja käte liikumisvabadus, negatiivseks pooleks aga piiratud kasutusala (kuni 80 cm [44]) ning fakt, et seade jälgib ainult käsi, mitte tervet keha.



Joonis 3.7. Nutikinnas BeBop Sensor VR Forte nutikinnas [45].

Nutikinnas on kindakujuline seade, mis edastab informatsiooni kasutaja käe ja sõrmede asukoha ning žestide kohta [46], näiteks BeBop Sensor VR Forte nutikinnas [47] (joonis 3.7). Sellise seadme heaks küljeks on käte asendi täpne tuvastamine.

3.3. ROS (Robot Operating System) ning VR

3.3.1. ROSi üldkirjeldus

ROS (Robot Operating System) on avatud koodiga vabavaraline roboti kontrollimiseks mõeldud tarkvararaamistik, mis koosneb paljudest erinevatest teکیدest ja tööriistadest [48]. ROS võimaldab ligipääsu riistvarale, roboti kontrollimist ja juhtimist, protsesside vahelist sõnumivahetust, kasutada erinevaid roboti draivereid, visualiseerimistööriistu (nt RViz [49]) ja simulatsioonikeskkondi (nt Gazebo [50]) ning teegi- ja paketi haldust [51]. ROS-raamistiku üks eesmärk on pakkuda võimalikult suurt modulaarsust kasutades kogukonnas loodud teike ja kimpe robotite arendamisel [51]. ROS on robotikas laialdaselt kasutatud.

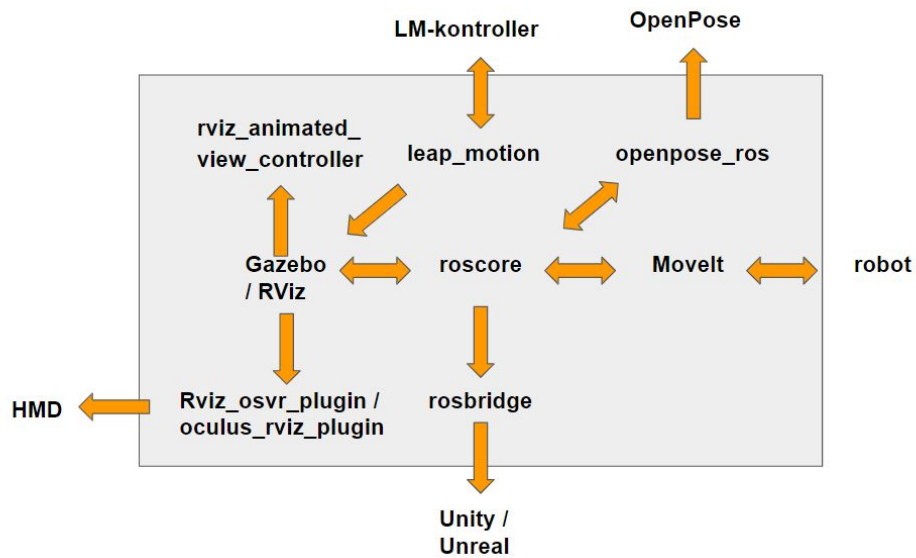
ROSi kasutamiseks peab arvutisse olema paigaldatud mõni operatsioonisüsteem. Parimat ROSi tuge pakuvad Linuxi tuumal põhinevad operatsioonisüsteemid, nt Ubuntu. ROSi saab kasutada ka Mac OS X operatsioonisüsteemil [51] ja Microsoft Windowsi operatsioonisüsteemidel [52].

ROSi kasutamise eeliseid on mitmeid. ROS-raamistik lubab kasutada erinevaid programmeerimiskeeli, näiteks Pythonit ja C++. Samuti on ROS riistvaraülene, mis tähendab, et loodud tarkvara töötab eri riistavaraga [53]. Raamistik on disainitud võimalikult õhukeselt, mis tähendab, et selle kasutamiseks ei ole vaja väga võimast arvutit. Seetõttu on seda võimalik kasutada ka näiteks Raspberry Pi peal [54]. ROSi moodulaarsus ja avatud lähtekood tähendab, et mitmed vajalikud teegid on kättesaadavad.

ROSi arendamine käib kogukonna põhiselt. Sellepärast on väga tähtis tarkvara dokumentatsioon. ROSi dokumentatsioon põhineb ROS Wikil. Igal ROSi paketil on oma vikileht. Samuti on kõikidel registreeritud kasutajatel võimalik koostada, uuendada ja parandada õpetusi ja juhendeid [55].

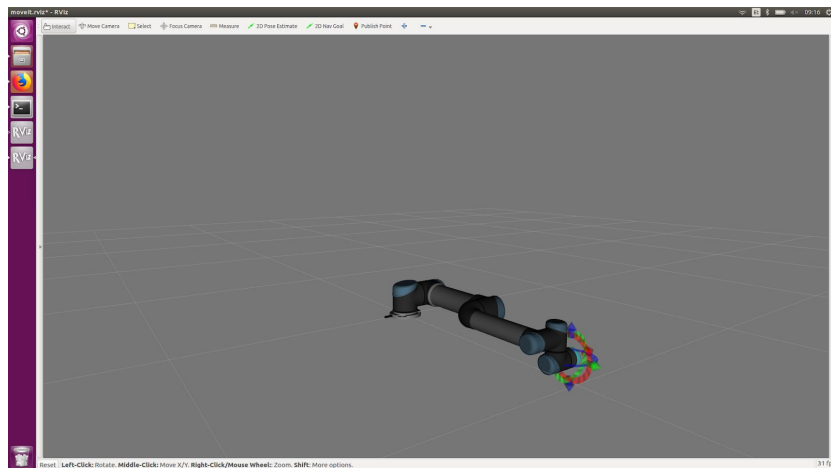
3.3.2. ROS ja virtuaalreaalsus

Nagu eelnevas peatükis sai märgitud, on ROSga võimalik kasutada mitmeid erinevaid kimpe, programme ja tööriistu, millest mitmeid toetavad virtuaalreaalsuse loomist ja kasutamist.



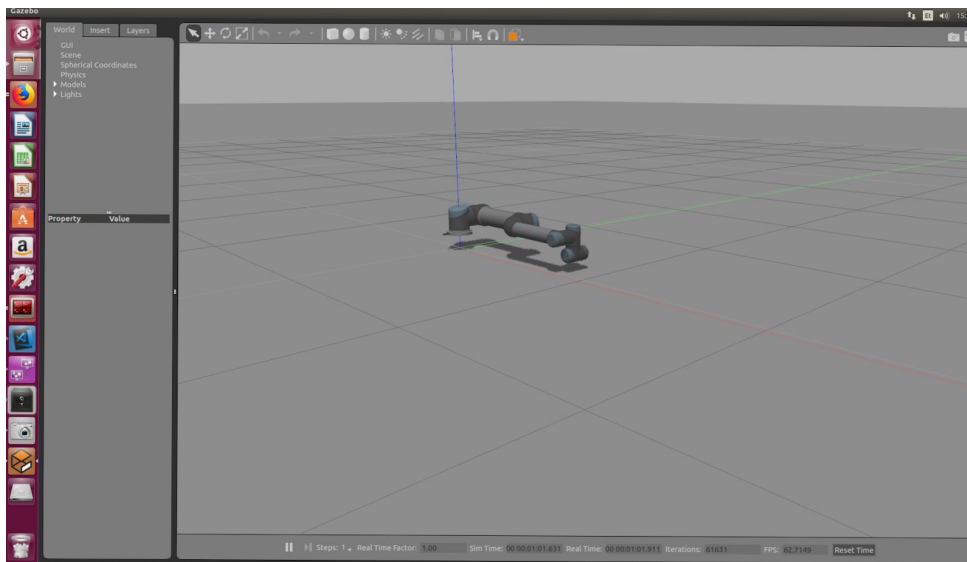
Joonis 3.7. Üldistav skeem ROSi kimpude, tööriistade, simulaatorite ja pistikprogrammide (halli kasti sees), roboti, HMD, mängumootorite, LM-kontrolleri ning OpenPose'i vahelistest suhetest.

Selleks, et juhtida ROSis robotit, kasutades VR-i, tuleb luua ühendus erinevate väliste seadmetega ning programmidega. Sellised seadmed ja programmid on näiteks HMD-d, robotid, välised mängumootorid, HID-d nagu LM-kontroller ja kaamerapõhised programmid nagu OpenPose (joonis 3.7).



Joonis 3.9. Ekraanitõmmis RVizist UR5 roboti mudeliga.

RViz (ROS Visualization) on tööriist, kus saab 3D visualiseerida ROSist tulevat informatsiooni [49] (joonis 3.9) . RViz on paindlik tööriist, sest sellele on võimalik lisada paljusid erinevaid erkaanielemente [56]. Lisaks on saab lisada paljusid erinevaid pistikprogramme (ingl *plugin*). Näiteks on olemas *rviz_animated_view_controller* pistikprogramm, mis võimaldab kasutajal kontrollida vaate liikumist RVizi aknas [57]. Samuti on olemas pistikprogrammid erinevatele VR visiiridele, näiteks Oculus Riftile [58] ja OSVRle [59].



Joonis 3.10. Ekraanitõmmis Gazebost UR5 roboti mudeliga.

Gazebo on kolmedimensiooniline füüsika simulaator, mis võimaldab simuleerida roboti liikumist ja kontrollimist erinevates keskkondades [60] (joonis 3.10). Gazebol on mitmeid versioone, neist kõige uuem on Gazebo 10 [61]. Gazebole on olemas mitmed pistikprogrammid, millega toetatakse mitmeid VR-i visiire [62], näiteks Oculus Riftile [63] ning HTC Vive'ile [64]. Gazebo ei ole küll ROSi osa, aga töötab väga hästi koos sellega [65].

ARViz [1] ja Rvis2AR [2] on projektid, mille eesmärk on välja töötada lahendused, kus liitreaalsust kasutatakse roboti kontrollimiseks ja juhtimiseks. Mõlemad projektid on ROS-põhised.

Kinect ROS-kimp [66] on mõeldud Microsoft Kinecti [40] ühendamiseks ROSi raamistikuga. Seda kasutades on võimalik anda kasutajale tagasisidet roboti ja ruumi kohta, kuvades RVizis Kinecti loodud punktipilve.



Joonis 3.11. *Inimeste tuvastamine kasutades veebikaamerat ja OpenPose'i [39].*

OpenPose [39] on projekt, mis kasutades veebikaamerat tuvastab inimese kehaosad ning ühendab need terviklikuks mudelskelektiks [67] (joonis 3.11). OpenPose'i kasutamiseks on loodud pakend (ingl *wrapper*), mis võimaldab seda ROSi raamistikus kasutada [68].

Movel! on ROS-kimpude kogum, mis pakub võimalusi roboti liikumise ja tee planeerimiseks, kokkupõrke avastamist ning kolmedimensioonilist ruumi tunnetust [69].

Rosbridge ROS-kimp, mis võimaldab mitte ROSil põhinevatel platvormidel suhelda ROSil töötava riist- ja tarkvaraga [70]. Selleks loob Rosbridge serveri ja JSON pesa, kuhu on võimalik teisi seadmeid või internetilehitsejat ühendada [71]. Kasutades Rosbridge'i on võimalik kontrollida ROSil põhinevat robotisüsteemi kasutades näiteks Unity [72] või Unreal [73] mängumootorit.

3.4. Leap Motion kontrolleri



(a)



(b)

Joonis 3.9. (a) Leap Motion kontrolleri käsi tuvastamas [67] ning (b) Leap Motion kontrolleri kinnitatuna OSVR visiiri külge.

Leap Motion (LM) kontrolleri on USB abil arvutiga ühendatav seade (joonis 3.9-a), mis jälgib kasutaja käsi, sõrmi ja žeste [74] [75]. LM-kontroller on mõeldud liit- ja virtuaalreaalsuse kontrollimiseks, kuid seda võib väga edukalt kasutada ka hiire asemel [76]. LM-kontrollerit on võimalik paigutada visiiri külge (joonis 3.9-b) ning sellises lahenduses on võimalik tuvastada kasutaja käsi VR-s [77]. LM-kontrollerit saab kasutada näiteks Oculus Rifti, OSVRi ja HTC Vive'iga [74]. LM-kontrollerit on võimalik kasutada nii Windowsi, Linuxi kui ka Mac OS operatsioonisüsteemidega [74]. LM-kontrolleril on olemas ka ROS-kimp [78], mis valmis 2018. aastal Tartu Ülikooli üliõpilase Kristo Allaje bakalaureusetöö raames [79].

LM-kontrolleri positiivseteks külgedeks on:

- seadme võrdlemisi odav hind - 94 eurot [80]
- lihtne disain - mõõtmed on 7,6 cm x 3 cm x 1,3 cm [81]
- olemasolev tarkvara ja rakendused on lihtsasti kasutatavad [82]

- hea täpsus, kui käed asuvad seadmest mõõdukal (lähemal kui 80 cm [44]) kaugusel [83]
- võimalus tuvastada mitmeid erinevaid žeste [84] ja näpistusliigutust [85].
- LM-kontrollerit on edukalt kasutatud robotmanipulaatori abil täpsete liigutuste sooritamiseks [86].

LM-kontrolleri negatiivseteks külgedeks on:

- vahel esinev liigne tundlikkus [82]
- madal täpsus, kui käed asuvad seadmele liiga lähedal või kaugel [83]
- vananenud Linuxi tarkvara
- piiratud jälgimisala - kuni 80 cm [44]
- mitme LM-kontrolleri samaaegsel kasutamisel võib tekkida ka käte mitmekordne tuvastamine [87]
- kuna LM-kontroller kasutab infrapunakaameraid, siis päiksevalgus, ekraanidelt ja halogeenlampidest kiirgav infrapunakiirgus segavad seadme tööd [88].

LM-kontroller suudab tuvastada järgnevad liigutused [84] ja žesti :

- näpu ringikujuline liikumine (ingl *circle*)
- kõrvale lükkamine (ingl *swipe*)
- ekraani vajutus (ingl *screen tap*)
- nupu vajutus (ingl *key tap*)
- näpistusliigutus (ingl *pinch*).

Mõned näited projektidest, kus kasutatakse LM-kontrollerit roboti juhtimisel:

- käe trajektoori jälgimisel [11]
- kahe vindiga toru kokku keeramine [11]
- keevitamisel [12]
- allveerobotikas [8]
- meditsiiniliste robotite juhtimiseks [10]
- radioaktiivsete keskkondade inspekteerimisel [89].

4. Eesmärk ja vajalikkus

Tänapäeva kiiresti areneva tehnoloogiaga on ilmselge, et VR-i tähtsus suureneb ning selle kasutamine muutub laialdasemaks. Seda sellepärast, et VR-i kasutamine on odav, intuitiivne, seda saab kasutada treenimiseks ning tänapäeval sarnaneb see juba väga reaalmaailmas kogetavaga [9]. Kindlasti leiab see laialdasemat kasutust ka robotikas. Robotikas VR-i kasutamise tähtsaimaks küljeks on võime juhtida roboteid eemalt, panemata inimelu ohtu. Lisaks on VR võrdlemisi odav võrreldes füüsiliste kontrollruumidega [6].

ROS-raamistikku on laialdaselt kasutatud küll robotite osteseks kontrollimiseks, kuid VR keskkonna loomiseks kasutatakse siiski mõnda muud mängumootorit, näiteks Unityt. Samas on ROSil endal mitmeid programme, simulaatoreid, tööriistu ja pistikprogramme, mille toel on võimalik luua VR keskkond. Käesoleval hetkel puudub ROS-kimp, kus ei kasutataks eraldiseisvat mängumootorit, vaid ROSi visualiseerimistarkvara.

Antud lõputöö eesmärgiks on:

- analüüsida ROSi pakutavaid võimalusi
- välja töötada lahendus, kus kasutatakse nii roboti juhtimiseks kui ka VR-i tekitamiseks ainult ROSi, selle programme, tööriistu ja pistikprogramme
- testida välja töötatud lahenduse riistvaralisi ja tarkvaralisi erinevusi
- teha lahendus kõigile vabalt kättesaadavaks kasutades Githubi koodihoidlat.

5. VR kasutajaliides

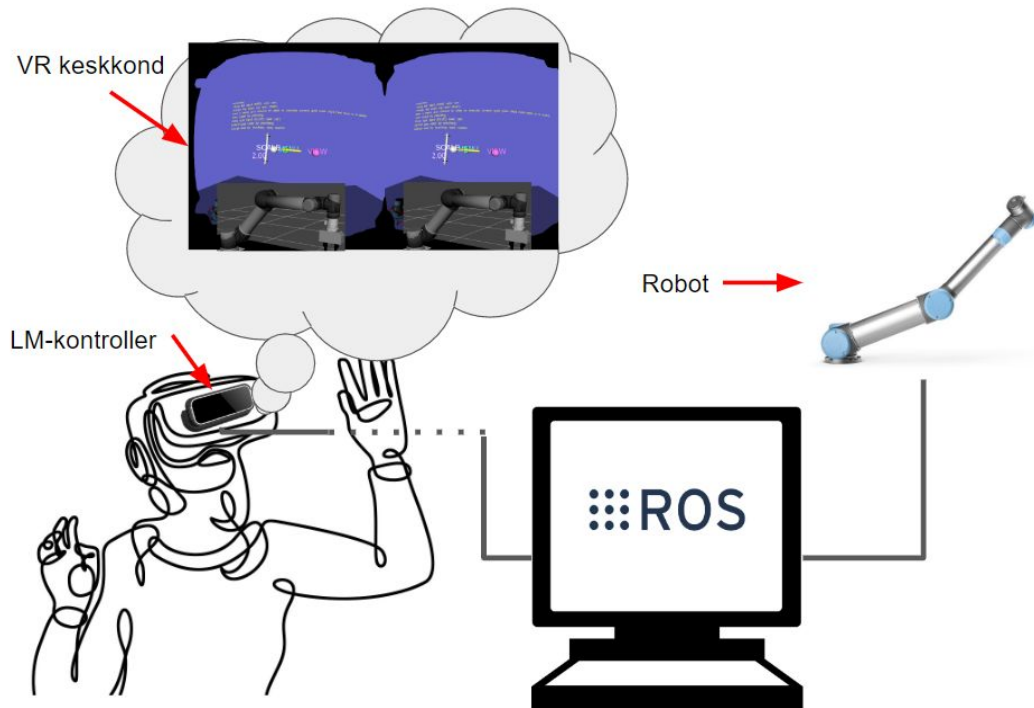
Antud peatükis kirjeldatakse eelnevalt püstitatud eesmärkidele vastavat lahenduse riist- ja tarkvaralist disaini.

Eelnevates peatükkides on välja toodud erinevad VR osad, mis on vajalikud virtuaalkeskkonna tekitamiseks ja juhtimiseks. ROS-raamistik on laialdaselt kasutatud üle maailma, kuid siiski pole olemas ainult ROS-raamistikul põhinevat VR-i. Selle põhjuseks on üks puuduv osa - ROS-kimp, mis kõik komponendid üheks seob. Loodava kimbu nimeks saab vrui_rviz.

5.1. Nõuded ja kontseptsioon

Loodud lahendus peab lubama kasutajal võimalikult lihtsalt juhtida robotit. Lahendus peab olema intuiitivne, nii et ka väikese VR kogemusega kasutajal on võimalik robotit kontrollida. Kasutajaliides peab olema lihtne ja arusaadav. Lahenduses peab olema võimalik planeerida ja muuta roboti ning mõjuri asendit.

Kasutajale kuvatakse virtuaalne keskkond kasutades visiiri. Kuna kasutaja käed peavad olema vabad, kasutatakse kasutaja käte jälgimiseks LM-kontrollerit. Kontroller paigutatakse visiir külge, suunaga kasutaja käte poole. Lahenduse üldise kontseptsiooni skeem on välja toodud joonisel 5.1.



Joonis 5.1. Lahenduse üldine kontseptsioon. Joonisel kasutatud kujutised pärinevad allikatest [90] [91] [92].

Lahendusega valmiv kimp peab ühendama erinevaid komponente, kasutades selleks ROSi osi. Välja töötatud programmis on vajalik lahendada järgnevad ülesanded:

- virtuaalse keskkonna loomine
- roboti juhtimine ja tagasiside saamine
- sisendseadmega ühendamine ja suhtlemine
- väljundseadmega ühendamine ja suhtlemine
- kasutajavaate muutmine.

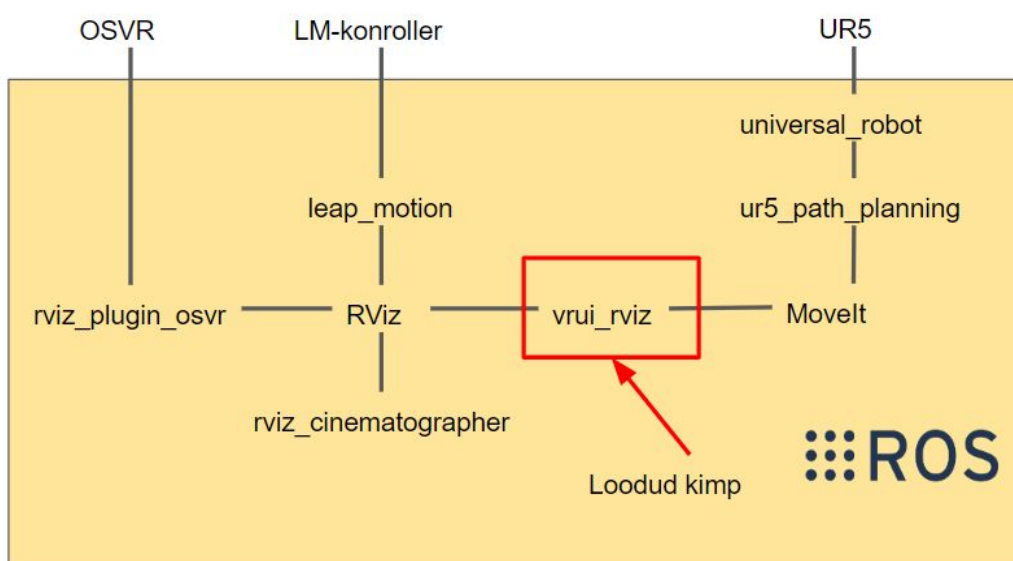
5.2. Disain

Konseptsioonis püstitatud ülesanded on lahendatud järgnevalt:

- Virtuaalse keskkonna loomiseks kasutatakse RViz programmi, sest sellel on olemas vajalikud pistikprogrammid ning kimbud suhtlemiseks sisend- ja väljundseadmetega.

- Roboti juhtimiseks ja sellega suhtlemiseks kasutatakse MoveIt [93] programmi. Lahenduses kasutatakse UR5 robotit.
- Sisendseadmena kasutatakse Leap Motion kontrolleri, sest seda kasutades on kasutaja käed vabad ning LM-kontrolleri jaoks on olemas ROS-kimp [79]. LM-kontrollerit ei paigutata lauale, vaid see paikneb kasutaja keha või pea küljes.
- Väljundseadmena kasutatakse OSVRi, sest see on lihtsasti programmeeritav, vabavaraaline ning sellele on olemas RViz pistikprogramm [94]
- kasutajavaate muutmiseks kasutatakse rviz_cinematographer pistikprogrammi [95].

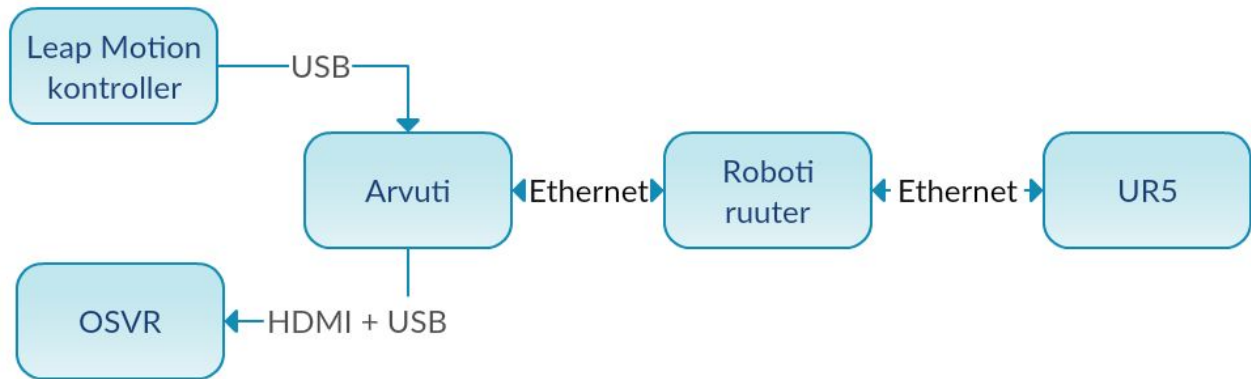
ROS-kimp on kirjutatud C++ programmeerimiskeeles. Lahenduse struktuur on näha joonisel 5.2.



Joonis 5.2. Välja töötatud lahenduse tarkvaraline ja riistavaraline skeem [92]. Punases kastis on välja toodud loodav kimp.

5.3. Seadmete vaheline suhtlus

Kiiruse ja turvalisuse tagamiseks on kõik riistvaralised komponendid ühendatud erinevate füüsiliste kaablitega. Joonisel 5.3 on välja toodud erinevate komponentide vaheline suhtlus. Robot UR5 ühendatud roboti ruuteriga kasutades Ethernet kaablit. Arvuti ja roboti ruuteri on samuti ühendatud kasutades Etherneti kaablit. Arvuti ja LM-kontroller on ühendatud USB kaabliga. Arvuti ja OSVR on ühendatud nii HDMI, kui ka USB kaabliga.



Joonis 5.3. Üldskeem lahenduse riistvaraliste komponentide ühendusviisidest.

5.4. Tarkvaralised lahendused

Komponendid on tarkvaraliselt omavahel ühendatud kasutatdes ROS-raamistikku. Lahendus on välja töötatud ning testitud Linux operatsioonisüsteemil Ubuntu 16.04 ja ROS Kinetic Kame [96] versioonil. Seda seetõttu, et Linuxil on kõige laialdasem ROS-raamistiku tugi. Süsteemi tarkvaralise poole saab jagada viieks:

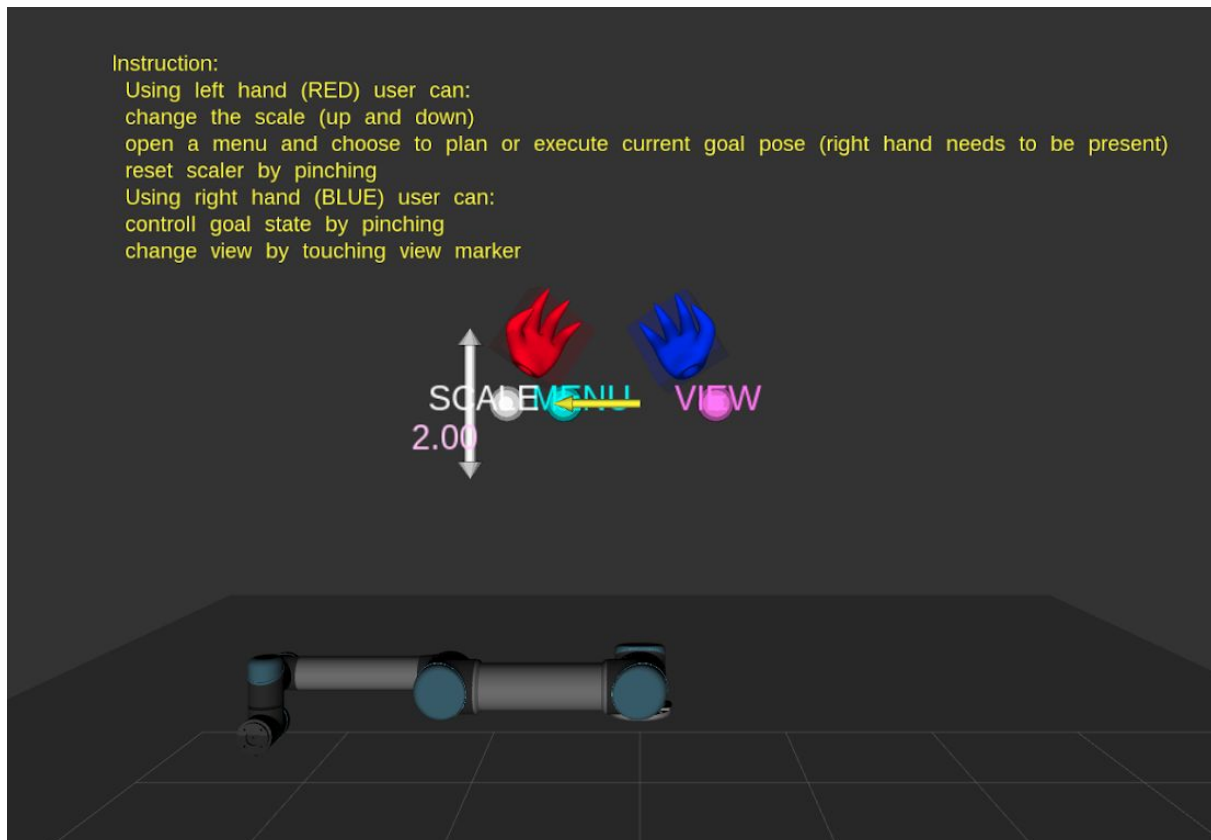
- suhtlus robotiga
- suhtlus LM-kontrolleriga
- suhtlus OSVRga
- VR keskkonna tekitamine
- kasutajaliides.

Lahenduse kasutamise eelduseks on ROS Kinetic Kame'i olemasolu arvutis. Lisaks sellele, tuleb eelnevalt paigaldada tabelis 5.1 välja toodud kimbud ja pistikprogrammid.

Tabel 5.1. Lahenduse käivitamiseks vajalikud kimbud ja pistikprogrammid. Välja on toodud kimbu või pistikprogrammi nimi, vajalikkus ja GitHub koodihoidla link.

Kimbu või pistikprogrammi nimi	Vajalikkus	GitHub koodihoidla link
vrui_rviz	Lahenduse põhikimp	https://github.com/ut-ims-robotics/vrui_rviz/
leap_motion	LM-kontrolleri kimp	https://github.com/ros-drivers/leap_motion
universal_robot	UR5 robotit kirjeldav kimp	https://github.com/ros-industrial/universal_robot
ur5_path_planning	UR5 roboti liigutamise kimp	https://github.com/ut-ims-robotics/ur5_path_planning
rviz_plugin_osvr	OSVRi pistikprogramm RViz-le	https://github.com/UTNuclearRoboticsPublic/rviz_plugin_osvr
rviz_cinematographer	RVizi kaamera liigutamiseks mõeldud pistikprogramm	https://github.com/AIS-Bonn/rviz_cinematographer

5.5. VR kasutajaliidese kirjeldus



Joonis 5.4. Kasutajale kuvatava VR keskkond RVizis.

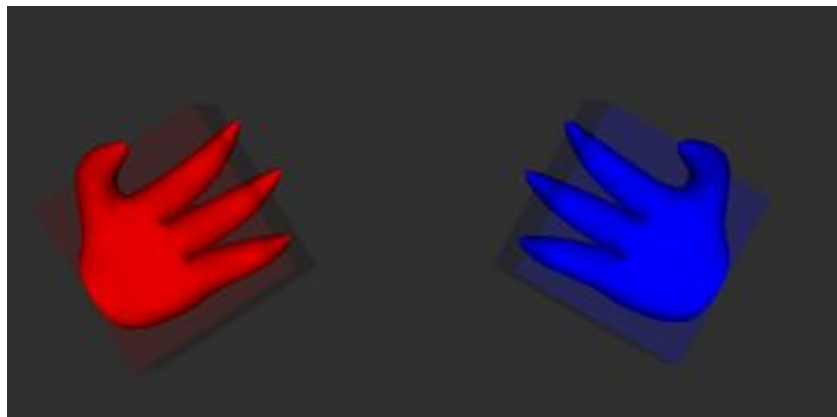
Loodud kasutajaliides võimaldab roboti liigutamist ja kasutajavaate muutmist. Kasutajale kuvatakse järgnevad elemendid:

- roboti liigutamiseks ja kasutajavaate muutmiseks loodud mitmed tähised, mida puudutades sooritatakse vastav tegevus.
- roboti mudel, mis näitab roboti hetkeasendit. Robot on paigutatud risttahuka külge, mis sümboliseerib lauda. Kui roboti teekonnaplaneerija tuvastab, et robot liigub läbi laua, muutub mudel punaseks ja liikumine peatatakse
- seletav kasutusjuhend taustal (joonis 5.4 taustal)
- kasutaja parem (sinine) ja vasak käsi (punane) (joonis 5.5).

5.5.1. Tähised

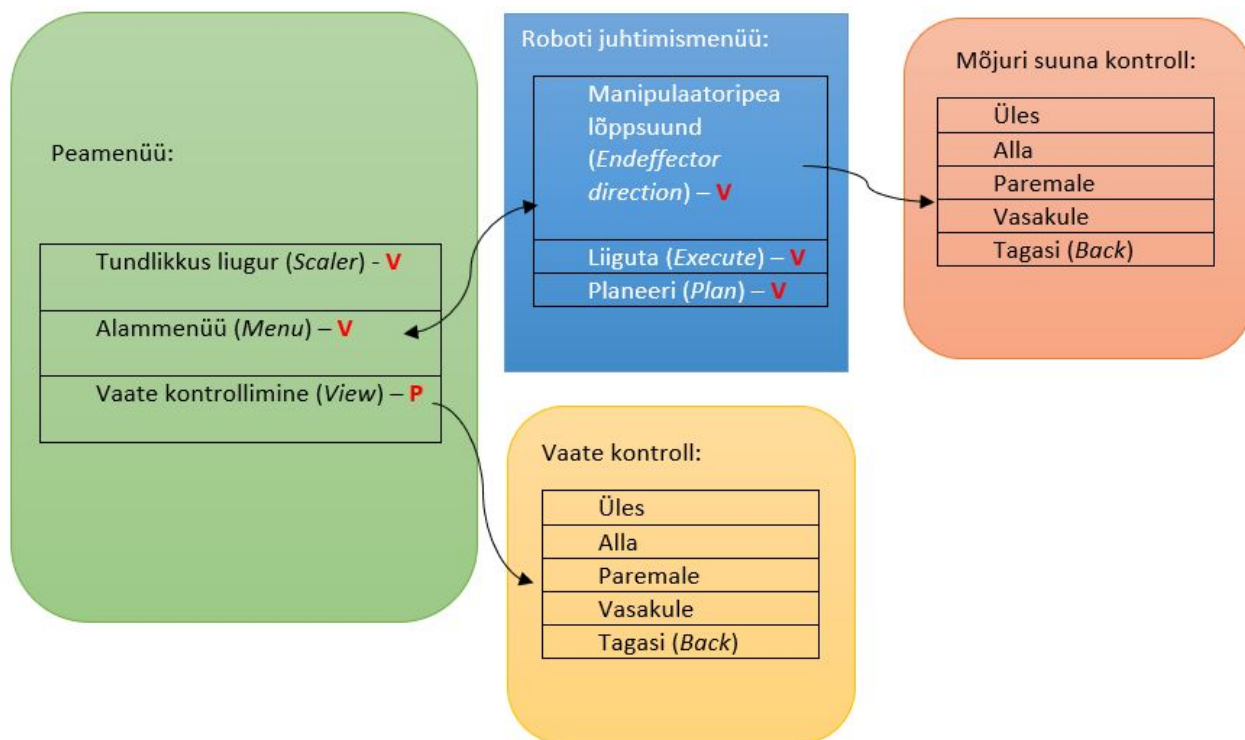
Roboti ja kasutajavaate juhtimiseks kasutatakse ROSi visualiseerimistähiseid (*visualization_msgs/Marker*) [97]. Tähis on ROS-raamistiku sõnumitüüp, mida kasutatakse objektide kuvamiseks RVizis [97]. Sõnumis saab seadistada mitmeid erinevaid parameetreid, näiteks tähise asukoht, suurus, värv ja nimi [98]. Tähised on mitme erineva kujuga. Käesolevas töös on kasutatud noole-, rööptahuka- ning kerakujulisi tähiseid. Samuti on võimalik kasutada erikujundusega tähiseid, mille kuju loetakse sisse STL-failist, näiteks käte visualiseerimiseks kasutatav tähis selles lõputöös (joonis 5.5).

Käesolevas lahenduses kasutatakse tähiseid, mis on kasutaja keha koordinaadistiku suhtes statsionaarselt paigutatud, näiteks planeerimise tähis, ning tähiseid, mille positsioon muutub keha koordinaadistiku suhtes pidevalt, näiteks käte tähised.



Joonis 5.5. *Kasutaja käte tähised.*

Käte kuvamiseks kasutatakse erikujulist tähist (*Mesh*) (joonis 5.5), mis on võetud avalikust 3D mudelihoidlast [99]. Eristamise eesmärgil on parema käe tähis sinine ja vasaku oma punane. Kui käed ei ole kaamerale näha või tuvastatavad, siis ei kuvata ka käte tähiseid.



Joonis 5.6. Graafilise kasutajaliidese skeem. Kandiliste nurkadega ja kahesuunalise noolega plokk tähendab automaatselt sulguvat menüü osa. Ühesuunalise noolega ja ümarate nurkadega plokk tähendab eraldiseisvaid menüüsid, millest väljumiseks on vaja kasutada "Tagasi"-objekti. Punane täht näitab kumma käega on võimalik vastavat tähist puutuda - V tähendab vasakut kätt, P tähendab paremat kätt.

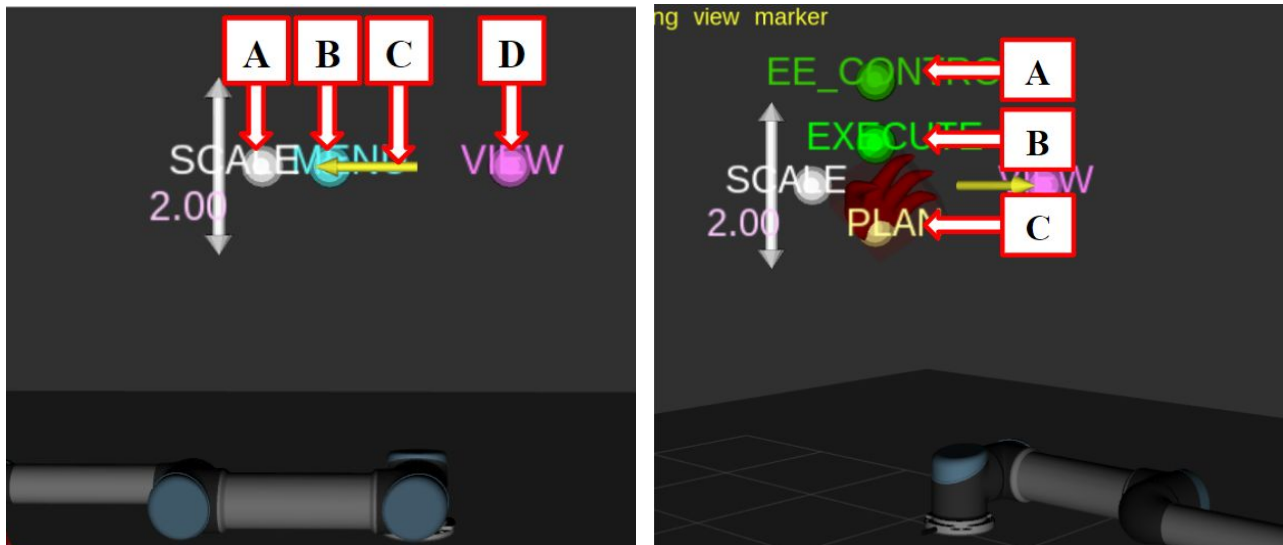
Erinevate käskude täitmiseks peab kasutaja viima käe tähise mõne teise tähise vastu. Kuna ROSis puudub tähistevaheline kokkupõrke tuvastamine, siis käesolevas lõputöös arvutatakse tähistevaheline kaugus ja, võttes arvesse nende mõõtmeid, tuvastatakse kontakt. Taoliselt avanevate menüüvalikute skeem on näha joonisel 5.6.

5.5.2. Roboti juhtimine

Roboti juhtimine toimib kasutades tähiseid ja kasutaja käte liigutusi. Kasutaja käte liigutusi kasutatakse roboti soovitud poosi mõjuri asukoha määramiseks. Roboti kontroll toimub kasutades järgnevaid tähiseid:

- tundlikkus liuguri tähis (*scaler*) (joonis 5.7-a-A)

- alammenüü tähis (*menu*) (joonis 5.7-a-B)
- mõjuri suuna tähis (kollane nool) (joonis 5.7-a-C)
- mõjuri suuna muutmise tähis (*ee_control*) (joonis 5.7-b-A)
- liigutamise tähis (*move*) (joonis 5.7-b-B)
- planeerimise tähis (*plan*) (joonis 5.7-b-C)
- mõjuri suuna määramise tähised.



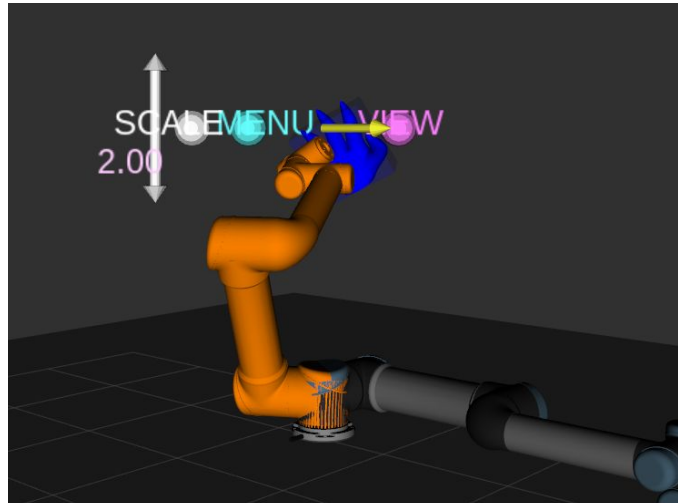
(a)

(b)

Joonis 5.7. (a) Kasutajale kuvatav kasutajaliides. A - Valge tähis koos noole ja numbriga on tundlikkusliugur, B - sinine tähis on alammenüü tähis, C - kollane noolekujuline tähis on mõjuri suuna tähis ja D - roosa tähis on kasutajavaate muutmise tähis.

(b) Kasutaja ette kuvatav vaade, kui alammenüü on avatud. A - Tumeroheline tähis on roboti mõjuri suuna muutmise tähis, B - heleroheline tähis on roboti liigutamise tähis ja C - helekollane tähis on roboti planeerimise tähis.

5.5.2.1. Roboti soovitud poosi määramine

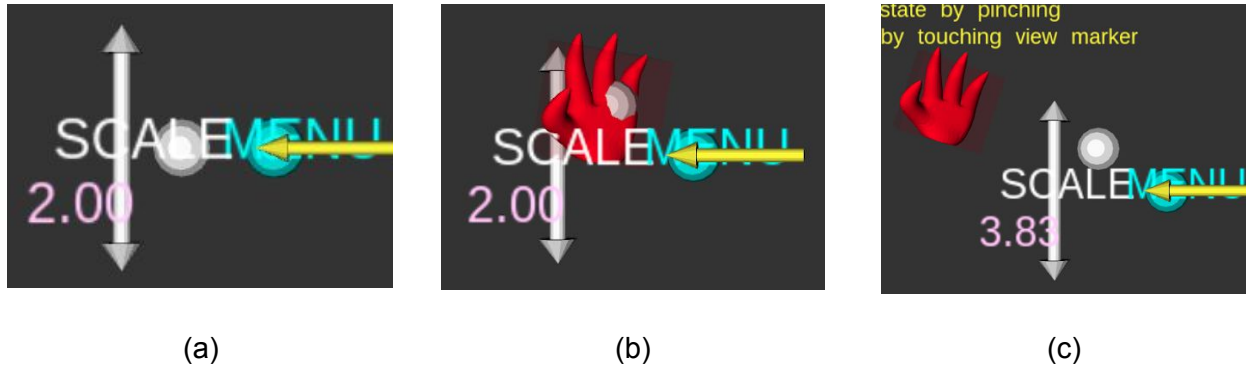


Joonis 5.8. Oranž roboti mudel kujutab roboti planeeritud poosi.

Roboti liigutamiseks tuleb kasutajal määrata soovitud asukoht. Selleks tuleb kasutajal viia oma parem käsi soovitud asukohta ruumis ning sooritada näpistusliigutus. Selle peale liigub oranž roboti päringukuju (ingl *query state*) (joonis 5.8) soovitud punkti. Oranž roboti mudel sümboliseerib soovitud roboti asendit. Tuleb meeles pidada, et selline roboti asend ei väljenda roboti reaalmaailma poosi, samuti ei väljenda roboti mudeli liikumine roboti mõjuri liikumist reaalmaailmas. Vaikeseadena on soovitud roboti poos roboti algpoos.

5.5.2.2. Tundlikkusliugur

Tundlikkusliugur (*scaler*) (joonis 5.7-a-A) on tähis, mida saab kasutaja liigutada puudutades tähist ning liigutades kätt üles ja alla. Liuguri kõrval on kuvatud arv, mis näitab käe koordinaatide kordajat, kirjeldav tekst ja nooled, mis viitavad suunale, kuhu saab tähist liigutada. Kordaja algväärtust (kordaja algväärtus on 2) saab taastada tehes vasaku käega näpistusliigutuse. Tundlikkusliuguri algasend on näha joonisele 5.9-a, kasutamine joonisel 5.9-b ning olukord, kus kordajat on muudetud, joonisel 5.9-b.



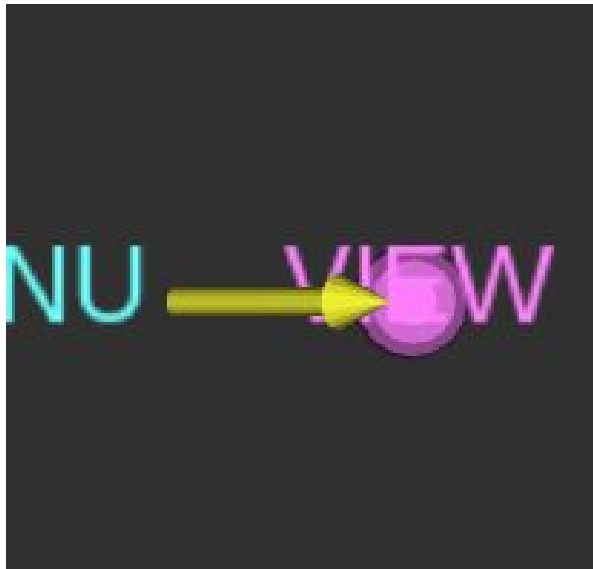
Joonis 5.9. (a) Tundlikkusliugur algasend, (b) tundlikkusliuguri kasutamine ja (c) tundlikkusliugur olukorras, kus kordajat on muudetud.

5.5.2.3. Alammenüü

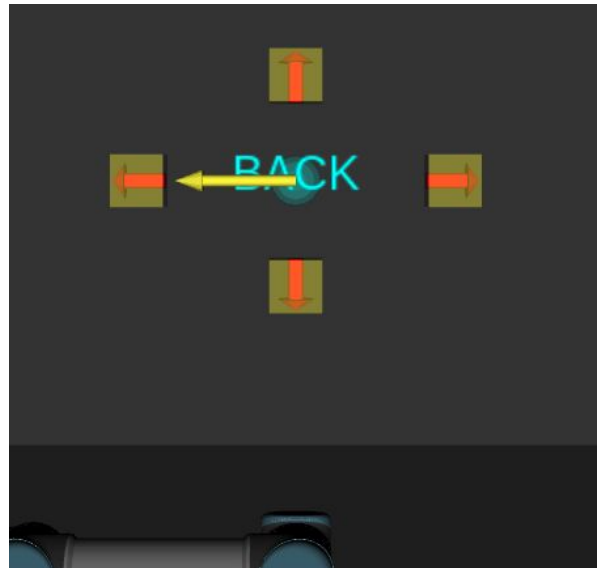
Planeerimise (*plan*) (joonis 5.7-b-C), liigutamise (*move*) (joonis 5.7-b-B) ning mõjuri suuna muutmise tähis (*ee_control*) (joonis 5.7-b-A) on algul varjatud. Nende nähtavale toomiseks peab kasutaja vasak käsi puutuma alammenüü tähist (*menu*) (joonis 5.7-a-B). Selle peale lõpetatakse alammenüü tähise kuvamine ning kasutajale kuvatakse eelnevalt mainitud tähised (joonis 5.7-b). Kui vasak käsi liigub alammenüü tähisest eemale või LM-kontroller ei tuvasta enam seda, siis suletakse alammenüü ning kuvatakse jälle algseis.

5.5.2.4. Roboti mõjuri suuna muutmise ja suuna tähis

Kasutaja vaatevälja keskel kuvatakse kollast noolt, mis on mõjuri suuna tähis (joonis 5.10-a). See tähis näitab kasutajale, millise suunaga on roboti mõjur planeerimise ja liigutamise järgselt.



(a)



(b)

Joonis 5.10. (a) Kollane noolekujuline tähis, mis näitab roboti mõjuri suunda. (b) Roboti mõjuri suuna muutmiseks kasutatavad tähised ning mõjuri suuna tähis.

Mõjuri suuna muutmise tähise (joonis 5.7-b-A) puutumisel kuvatakse kasutajale neli suuna määramise tähist ja üks tagasi minemise tähis (joonis 5.10-b). Suuna määramise tähise puutumisel keeratakse suuna tähist ümber vastava telje. Tähist saab pöörata ümber kahe telje.

5.5.2.5. Roboti teekonna planeerimine

Planeerimise tähise (*plan*) (joonis 5.7-b-C) eesmärk on anda ROSle käsk arvutada välja tee, kuidas robotit liigutada hetkeasendist soovitud asendisse, ning demonstreerida seda liigutades roboti mudelit vastavalt plaanile, muutmata roboti hetkeasendit [100]. Planeerimise demonstreerimise tähtsus seisneb selles, et kasutajale tekiks ettekujutus, milline on roboti eeldatav liikumisteed ja lõppasend.

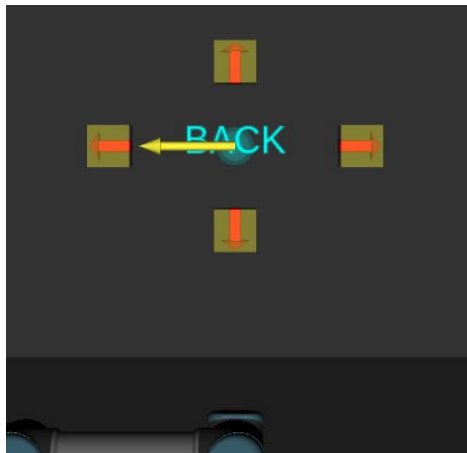
5.5.2.6. Roboti liigutamine

Liigutamise tähise (*move*) (joonis 5.7-b-B) puutumisel antakse süsteemile käsk planeerida roboti liikumisteedkond ning seejärel seda vastavalt plaanile liigutada [101].

Planeerimist ja liigutamist saab teostada ainult vasaku käega. Ohutuse mõttes peab kasutaja parem käsi samuti olema kaamerale tuvastatav.

5.5.2.7. Kasutajavaate muutmine

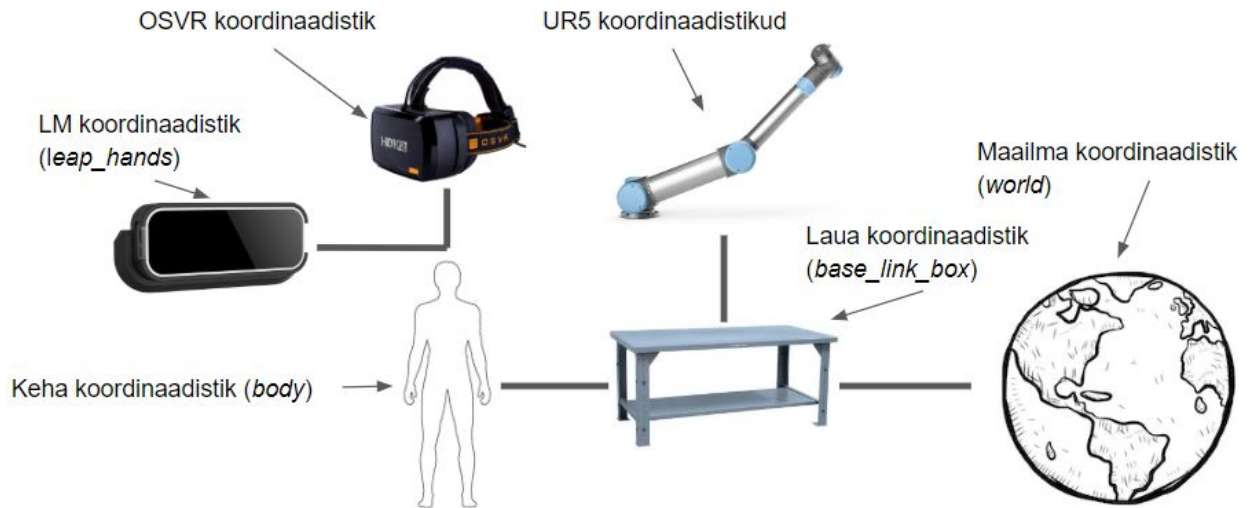
Välja töötatud lahendus on disainitud nii, et OSVR ja kasutaja käed on kinnitatud keha koordinaadistiku külge. See tähendab, et kasutajal on võimalik vaadata igasse suunda, liigutades oma pead, samal ajal kui kasutajaliides on püsivalt samas VR keskkonna osas. Selleks, et muuta, millisesse virtuaalse ruumi osasse käed ulatuksid, tuleb uuendada keha koordinaadistiku asendit.



Joonis 5.11. *Kasutajavaate muutmise tähised.*

Keha koordinaadistiku muutmiseks tuleb kasutajal parema käega puudutada vaate muutmise tähist (*view*) (joonis 5.7-b-D). Selle peale kuvatakse kasutajale neli vaatemuutmise tähist - üles, alla, paremale, vasakule - ning tagasi minemise tähis (joonis 5.11). Puudutades vaate muutmise tähist, pööratakse keha koordinaadistiku roboti koordinaadistiku suhtes mööda vastavat telge. Keha koordinaadistiku on võimalik keerata ümber kahe telje. Tagasi peamenüüsse minekuks tuleb puutuda tagasi minemise tähist (*back*).

5.6. Koordinaadistik



Joonis 5.12. Lahenduses kasutatavad koordinaadistikud ja nende suhted.

Välja töötatud lahendus kasutab mitmeid erinevaid koordinaadistike, mille suhted on välja toodud joonisel 5.12.

Maailma koordinaadistik (*world*) on üldine koordinaadistik, mis on RVizi aluskoordinaadistikuks. Laua koordinaadistiku (*base_link_box*) külge kinnitub roboti all olev laud.

Roboti igal liigesel on eraldi koordinaadistik. Keha koordinaadistik (*body*) sümboliseerib kasutaja keha ja selle asendit. Keha koordinaadistiku külge on kinnitatud OSVR. See tähendab, et OSVRi asukoht on määratud just selle koordinaadistikuga, kuid kasutajal on võimalik vaadata igasse suunda. Kasutaja käed ja erinevad juhtimiselemendid on samuti kinnitatud keha koordinaadistiku külge. Selleks on loodud LM koordinaadistik (*leap_hands*), mis on kinnitatud jäigalt keha koordinaadistiku külge, kasutaja poolt vaadates väikese nihkega, otse keha koordinaadistiku ette. Keha koordinaadistiku külge kinnitub ka RVizi kaameravaade.

5.7. Lahenduse käivitamine

Hetkel on välja töötatud kimp `vrui_rviz` avalikult kättesaadav UT IMS Robotics GitHubi koodihoidlas aadressil https://github.com/ut-ims-robotics/vrui_rviz. Kimbuga kaasas käivas seletavas README failis on juhised, kuidas paigutada ja kasutada loodud kimpu.

Lahendus töötab RViz-i põhjal.

Lahenduse käivitamiseks tuleb eraldi terminaliakendes anda järgmised käsud:

1. `sudo leapd`
2. `roslaunch vrui_rviz <käivituskripti nimi>.launch`, kus <käivituskripti nimi> tuleb asendada sobiva käivituskriptiga, mis on leitavad tabelis 5.2
3. `roslaunch ur_modern_driver ur5_bringup.launch robot_ip:=ROBOT_IP`, kus ROBOT_IP tuleb asendada kasutatava roboti IP-aadressiga
4. `roslaunch ur5_custom_config
ur5_on_table_moveit_planning_execution.launch`

Tabel 5.2. *Kimbus vrui_rviz olevad käivituskriptid ja nended kirjeldused.*

Käivituskript	Kirjeldus
<code>cam_without_osvr.launch</code>	Roboti mudeli kontrollimine RViz-s kasutades LM-kontrollerit ja veebikaamerat. Mõeldud testimiseks.
<code>leap_cam_osvr.launch</code>	Füüsilise roboti kontrollimine virtuaalreaalsuses kasutades UR5 robotit, LM-kontrollerit, veebikaamerat ja OSVRi. Lõplik veebikaameraga lahendus.
<code>leap_osvr.launch</code>	Füüsilise roboti kontrollimine virtuaalreaalsuses kasutades UR5 robotit, LM-kontrollerit ja OSVRi. Lõplik veebikaamerata lahendus.
<code>robot_without_osvr.launch</code>	Füüsilise roboti kontrollimine RViz-is kasutades UR5 robotit ja LM-kontrollerit.
<code>without_osvr.launch</code>	Roboti mudeli kontrollimine RViz-is kasutades LM-kontrollerit. Mõeldud testimiseks.

6. Peamised tehnilised järelused

6.1. Probleemid Leap Motion kontrolleringa

Lahenduse ilmnunud probleemide tõttu prooviti läbi olukorrad, kus LM-kontroller on visiiri küljes (joonis 6.1-a) ja kus ripub paelaga kasutajal kaelas (joonis 6.1-b).



(a)



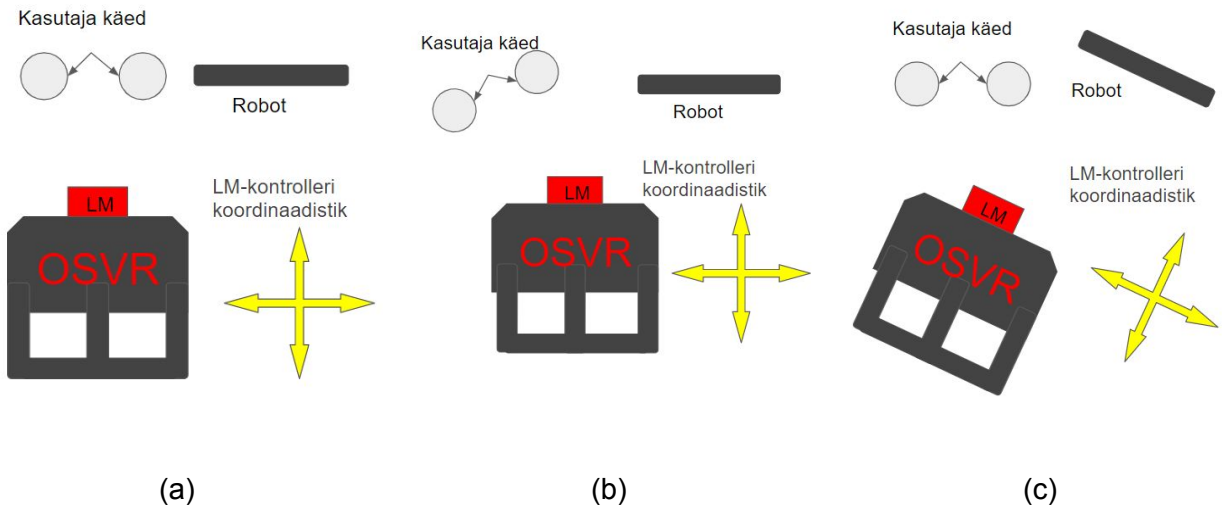
(b)

Joonis 6.1. (a) LM-kontroller kinnitatuna OSVR visiiri külge ja (b) LM-kontroller rippumas kaelas.

6.1.1. LM-kontroller OSVRi küljes

Nagu kirjeldatud, prooviti läbi kaks erinevat võimalust LM-kontrolleri kinnitamiseks kasutaja külge - kontroller visiiri küljes ja paelaga kaelas.

Kontrolleri kinnitamisel visiiri külge liigub LM-kontroller reaalmaailmas vabalt ringi, samas kui tarkvaraliselt on LM-kontrolleri koordinaadistik (*leap_hands*) jäigalt seotud virtuaalmaailma koordinaadistikuga (*world*). Probleem on näitlikustatud joonisel 6.1. Joonisel 6.1-a on näidatud algseis, kus OSVR ja LM-kontroller on suunatud kasutaja käte poole. Joonisel 6.1-c on kasutaja keeranud pead 25°, samas käed asuvad samas kohas. See tähendab, et käte asukoht muutub LM-kontrolleri koordinaadistiku suhtes. Joonisel 6.1-b on kirjeldatud olukorda, kus OSVR ja LM-kontroller on suunatud käte eelmisesse asukohta, aga käed on liigutatud 25° nurga alla. Selline olukord on süsteemi jaoks täpselt sama, mis on välja toodud joonisel 6.1-c.



Joonis 6.1. (a) OSVR ning LM-kontroller suunatuna otse kasutaja käte poole, (b) OSVR ning LM-kontroller suunatuna kasutaja käte esialgse asukoha poole, samal ajal kui kasutaja käed 25° nurga all, ja (c) OSVR ning LM-kontroller 25° nurga all, samal ajal kui kasutaja käed jäävad samase kohta.

Selline lahendus võib muutuda ohtlikuks, kui käte tähiste liikumisel puutuvad nad kokku näiteks liigutamise tähisega ja seetõttu antakse robotile soovimatu käsk liikuda. Seda probleemi oleks võimalik lahendada lugedes OSVRi inertsiaalanduritest (ingl *IMU* - *Inertial Measurement Unit*) andmeid visiiri liikumise kohta ja seejärel keerata keha koordinaadistikku vastassuunas niikaua kuni käed väljuvad kaamera pildist.

6.1.2. LM-kontroller kaelas

Leap Motioni kontrolleri kaela riputatades ilmnesid samuti mõned probleemid:

- käsi on mugavam kontrolleri kaugemal hoida. Mida kaugemal hoida käsi LM-kontrollerist, seda halvemini kontrolleri neid tuvastab.
- kuna LM-kontroller ripub kaelas suhteliselt vabalt paela küljes, siis liigub kontrolleri horisontaaltasandi suhtes kiiresti paigast ära, seda keha liikumise ja ühendusjuhtme tõttu. Seda probleemi saab lahendada kinnitades kontrolleri jäigemalt kasutaja keha külge.



Joonis 6.2. *Vaated Leap Motion kontrolleri kaamerast kui seade on paigaldatud kasutaja rinnale (vasakul) ja kui seade on kinnitatud OSVRi külge.*

6.1.3. Käte asend LM-kontrolleri suhtes

LM-kontroller on võimeline tuvastama käsi, peopesa kaamera poole pööratuna, kui ka peopesa kaamerast eemale pööratuna. Tulenevalt inimeste käte ülesehitusest on lihtsam ja täpsem tuvastada kätt siis, kui peopesa on kaamera poole pööratud, sest siis on väiksem võimalus, et mõni käe osa on varjatud. Eriti tuleb selline probleem esile kasutades näiteks näpistusliigutust, sest, olenevalt käeasendist, võib see liigutus olla kaamerale täielikult või osaliselt varjatud. Sellest tulenevalt on käesolevas lahendus parem hoida käsi peopesaga kaamera poole.

6.1.4. Üldine käte tuvastamine

LM-kontrolleri probleemiks on ühtlane käte tuvastamisega, kuna kasutaja liigutab käsi ruumis võrdlemisi kiiresti ning kaugus kontrollerist muutub palju. Lisaks üldisele tuvastamisele, on suur probleem vasaku ja parema käe eristamine. Siinses lahendus on väga tähtis, et tuvastatakse õige käsi, sest vasakul ja paremal on erinevad funktsioonid.

6.2. Probleemid ROSi ja lahendusega

6.2.1. Piiratud tööpiirkond

Kasutaja kätel on piiratud tööpiirkond, mis on väiksem roboti tööpiirkonnast. See tähendab, et ilma keha koordinaadistikku liigutamata ei ole võimalik robotit vabalt ruumis liigutada. Lahendusena on loodud keha koordinaadistiku keeramise võimalus, kasutades keha asendi muutmise tähist ning

selle alammenüüd. Selline lahendus ei ole aga väga intuitiivne. Teise lahendusena oleks käte tundlikkuse suurendamine, kuid selline lahendus muudab roboti enda kontrollimise keerulisemaks.

6.2.2. Roboti mõjuri suuna määramine

Roboti mõjuri suuna määramiseks testiti kahte erinevat viisi.

Esiteks prooviti, kas on võimalik mõjuri suuna määramisel kasutada LM-kontrolleri poolt edastatavat käe suunda. Selleks jäeti meelde käe orientatsioon hetkel, kui parem käsi tegi näpistusliigutuse (roboti soovitud positsiooni määramine). Hiljem, roboti liigutamisel, kasutati seda suunda. Järele proovides selgus, et selline suuna määramise viis on väga ebastabiilne, sest käe kohta edastatav informatsioon ei ole ühtlane. Lisaks on väga keeruline hiljem muuta mõjurisuunda nii, et roboti soovitud poos ei muutuks. Probleemiks on ka selliste suundade määramine, mille puhul ei suuda kaamera enam kätt ja selle liigutusi tuvastada, näiteks suunaga kasutajast eemal nii, et näpistusliigutus on kaamera eest varjatud.

Selleks, et mõjuri suunda saaks määrata ja see oleks stabiilne, kasutati sarnast lahendust, mida kasutati kasutajavaate muutmiseks. See tähendab, et mõjuri suunda saab muuta üle kahe telje, kasutades aga õigeid suuna tähiseid, on võimalik määrata ükskõik milline suund ruumis.

6.2.3. Tundlikkuse muutmine

Kasutaja käte tööpiirkonna suurendamiseks korrutatakse LM-kontrolleri poolt edastatavad koordinaadid läbi kordajaga. Kordaja algväärtuseks on 2. Kordaja muutmiseks kasutatakse liugurit. Selle kasutamiseks peab kasutaja vasaku käe viima liuguri peale ja liigutama seda vastavalt üles või alla. Nii kaua kui käsi puutub liugurit, muudetakse küll kordajat, kuid seda ei rakendata. Seda seetõttu, et muidu muutuks käe asukoht koheselt ja kordaja muutmine oleks keeruline. LM-kontrollerist saadavad andmed ei ole aga alati stabiilsed. Sellepärast esineb vahel olukord, kus liuguri kasutamise ajal rakendatakse käele uus kordaja ja käsi liigub eemale. Juhul kui kordaja on määratud väga väikseks, ei ole võimalik enam kordajat muuta, sest kätt pole võimalik enam liugurini viia. Sarnaselt sellele, juhul kui kordaja on väga suur, on raske kordajat vähendada. See probleem lahendati võimaldades kasutajal kordaja algväärtustada kasutades vasaku käe näpistusliigutust.

6.2.4. Mugavus

Käesoleva lõputöö eesmärk on luua võimalikult mugav töötav lahendus. Selleks, et lahendus oleks kasutatav, ei pruugi see alati kasutajale kõige mugavam olla.

Mõned mugavusega seotud probleemid:

- kohati on tähised liiga tihedalt koos
- käte tundlikkuse liuguri probleemid
- LM-kontroller ei tuvasta alati käsi korrektselt
- LM-kontroller ei tuvasta alati näpistusliigutust
- LM-kontrolleri programm lõpetab pidevalt töö.

6.3. Üldised probleemid

Üldistest probleemidest on suurimaks vananenud ning puudulik informatsioon erinevate kimpude ning programmide juhistes:

- `rviz_plugin_osvr`-i juhend on kohati vananenud ja puudu informatsioon, kuidas ühendada OSVRi uuemate videokaartidega Linux operatsioonisüsteemis
- `leap_motion`i kimbu juhendis puudub informatsioon, kuidas kasutada žeste

7. Kokkuvõte

Käesoleva bakalaureusetöoga loodi ainult ROSil põhinev virtuaalreaalsuse kasutajaliidese ROS-kimp. Kimp võimaldab planeerida roboti liikumist ning liigutada robotit. Lisaks on loodud mitmed lisavõimalused, mis toetavad roboti kontrollimist, nt tundlikkusliugur, mõjuri suuna muutmise ja kasutajavaate muutmise võimalus. Virtuaalsesse keskkonda kuvatavad objektid on tähised. Tähised kuvatakse ROSi visualiseerimisprogrammis RViz, mis omakorda, kasutades OSVRi pistikprogrammi, kuvab VR keskkonna visiiris. Kimp kasutab väljundseadmena OSVR visiiri ning sisendseadmena Leap Motion kontrolleri. LM-kontroller tagab, et kasutaja peopesad on tühjad. Loodud lahenduse juurdes on ka mitmed käivitusskriptid, millega saab käivitada kimpu erinevates seadistustes.

Loodud kimp `vruj_rviz` on avalikult kättesaadav UT IMS Robotics organisatoorses GitHubi koodihoidlas. Kimbu aadress on https://github.com/ut-ims-robotics/vruj_rviz. Koodihoidla README failis on kirjeldatud loodud kimbu võimalused, instruktsioonid kimbu paigutamiseks ja nimekiri vajalikest kimpudest.

8. Viited

- [1] „ARViz: Augmented Reality Visualizer for ROS2“, *ROSIN*. .
- [2] „Rvis2AR - Visualization platform for AR / VR devices“, *ROSIN*. .
- [3] „What is Virtual Reality (VR)? - Definition from Techopedia“, *Techopedia.com*. [Võrgumaterjal]. Kättesaadav: <https://www.techopedia.com/definition/4784/virtual-reality-vr>. [Vaadatud: 07-märts-2019].
- [4] C. Anthes, R. J. Garcia-Hernandez, M. Wiedemann, ja D. Kranzlmuller, „State of the art of virtual reality technology“, *2016 IEEE Aerosp. Conf.*, lk 1, jaan 2016.
- [5] „Explained: How does VR actually work?“, *Wearable*, 26-dets-2017. [Võrgumaterjal]. Kättesaadav: <https://www.wearable.com/vr/how-does-vr-work-explained>. [Vaadatud: 07-märts-2019].
- [6] T. Kot ja P. Novák, „Application of virtual reality in teleoperation of the military mobile robotic system TAROS“, *Int. J. Adv. Robot. Syst.*, kd 15, nr 1, lk 172988141775154, jaan 2018.
- [7] „Baxter’s Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing“, *IEEE Robot. Autom. Lett. Robot. Autom. Lett. IEEE IEEE Robot Autom Lett*, nr 1, lk 179, 2018.
- [8] J. C. García et al., „A Natural Interface for Remote Operation of Underwater Robots“, *IEEE Comput. Graph. Appl.*, kd 37, nr 1, lk 34–43, jaan 2017.
- [9] „Advantages of virtual reality training“, *Virtual Reality Society*, 05-mai-2017. .
- [10] T. Zhou, M. E. Cabrera, T. Low, C. Sundaram, ja J. Wachs, „A Comparative Study for Telerobotic Surgery Using Free Hand Gestures“, *J. Hum.-Robot Interact.*, kd 5, nr 2, lk 1, sept 2016.
- [11] G. Du ja P. Zhang, „A Markerless Human–Robot Interface Using Particle Filter and Kalman Filter for Dual Robots“, *IEEE Trans. Ind. Electron.*, kd 62, nr 4, lk 2257–2264, apr 2015.
- [12] Y. Liu ja Y. Zhang, „Toward Welding Robot With Human Knowledge: A Remotely-Controlled Approach“, *IEEE Trans. Autom. Sci. Eng.*, kd 12, nr 2, lk 769–774, apr 2015.
- [13] D. Whitney, E. Rosen, D. Ullman, E. Phillips, ja S. Tellex, „ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots“, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018, lk 1–9.
- [14] S. Kim, Y. Kim, J. Ha, ja S. Jo, „Mapping System with Virtual Reality for Mobile Robot Teleoperation“, lk 3.
- [15] „HTC Vive - Casque Réalité Virtuelle HTC sur LDLC.com“. [Võrgumaterjal]. Kättesaadav: <https://www.ldlc.com/fr-be/fiche/PB00210019.html>. [Vaadatud: 25-apr-2019].
- [16] „Oculus Rift - Wikipedia“. [Võrgumaterjal]. Kättesaadav: https://en.wikipedia.org/wiki/Oculus_Rift. [Vaadatud: 25-apr-2019].
- [17] „Razer’s HDK2 VR headset is finally consumer ready at \$399.99, with an emphasis on tinkerers - Polygon“. [Võrgumaterjal]. Kättesaadav: <https://www.polygon.com/2017/4/24/15406112/razer-hdk2-open-source-hack>. [Vaadatud: 25-apr-2019].
- [18] „IMU - Virtual Reality and Augmented Reality Wiki - VR AR & XR Wiki“. [Võrgumaterjal]. Kättesaadav: <https://xinreality.com/wiki/IMU>. [Vaadatud: 06-mai-2019].
- [19] „VIVE™ | Discover Virtual Reality Beyond Imagination“. [Võrgumaterjal]. Kättesaadav: <https://www.vive.com/eu/>. [Vaadatud: 24-apr-2019].
- [20] „Oculus Rift: VR Headset for VR Ready PCs | Oculus“. [Võrgumaterjal]. Kättesaadav: <https://www.oculus.com/rift/>. [Vaadatud: 07-märts-2019].

- [21] „OSVR - Open-Source Virtual Reality for Gaming“. [Võrgumaterjal]. Kättesaadav: <http://www.osvr.org/>. [Vaadatud: 07-märts-2019].
- [22] „Samsung Gear VR with Controller“, *The Official Samsung Galaxy Site*. [Võrgumaterjal]. Kättesaadav: <http://www.samsung.com/global/galaxy/gear-vr/>. [Vaadatud: 07-märts-2019].
- [23] „ZEISS VR ONE Plus - Virtual Reality Glasses with Multi: Amazon.co.uk: Electronics“. [Võrgumaterjal]. Kättesaadav: <https://www.amazon.co.uk/ZEISS-VR-ONE-Plus-2174-931/dp/B01GHN88HU>. [Vaadatud: 06-mai-2019].
- [24] „These tips and tricks make it easier to use a keyboard while in VR“, *Windows Central*, 16-veebr-2018. [Võrgumaterjal]. Kättesaadav: <https://www.windowscentral.com/how-use-your-keyboard-while-wearing-vr-headset>. [Vaadatud: 18-mai-2019].
- [25] „Accessories“, *PlayStation*. [Võrgumaterjal]. Kättesaadav: <https://www.playstation.com/en-us/explore/accessories/>. [Vaadatud: 24-apr-2019].
- [26] „Xbox Wireless Controller – Black - Microsoft“, *Microsoft Store*. [Võrgumaterjal]. Kättesaadav: <https://www.microsoft.com/en-us/p/xbox-wireless-controller/8vcw8gln9vrf>. [Vaadatud: 24-apr-2019].
- [27] (C) 2019 Conrad Electronic, „Thrustmaster T-Flight Stick X Flight sim joystick USB PC, PlayStation 3 Black | Conrad.com“. [Võrgumaterjal]. Kättesaadav: <https://www.conrad.com/p/thrustmaster-t-flight-stick-x-flight-sim-joystick-usb-pc-playstation-3-black-906477>. [Vaadatud: 30-apr-2019].
- [28] „Mängupult“, *Vikipeedia, vaba entsüklopeedia*. 19-juuli-2017.
- [29] „DualShock 4“, *PlayStation*. [Võrgumaterjal]. Kättesaadav: <https://www.playstation.com/en-us/explore/accessories/gaming-controllers/dualshock-4/>. [Vaadatud: 10-apr-2019].
- [30] „Xbox Wireless Controller - Microsoft“, *Microsoft Store*. [Võrgumaterjal]. Kättesaadav: <https://www.microsoft.com/en-us/p/xbox-wireless-controller/8t2d538wc7mn>. [Vaadatud: 10-apr-2019].
- [31] „F710 Wireless Gamepad - Logitech“. [Võrgumaterjal]. Kättesaadav: <https://www.logitechg.com/en-roeu/products/gamepads/f710-wireless-gamepad.html>. [Vaadatud: 10-apr-2019].
- [32] „Use Your Xbox Wireless Controller on Samsung Gear VR“, *Xbox Support*. [Võrgumaterjal]. Kättesaadav: <https://support.xbox.com/en-US/xbox-one/accessories/use-samsung-gear-vr-with-xbox-controller>. [Vaadatud: 10-apr-2019].
- [33] „What is a Joystick?“, [Võrgumaterjal]. Kättesaadav: <https://www.computerhope.com/jargon/j/joystick.htm>. [Vaadatud: 24-apr-2019].
- [34] „PS3 PC Joystick - T.Flight Stick X (Flight Simulator Compatible) | Thrustmaster“. [Võrgumaterjal]. Kättesaadav: <http://www.thrustmaster.com/products/tflight-stick-x>. [Vaadatud: 06-mai-2019].
- [35] „Oculus“. [Võrgumaterjal]. Kättesaadav: <https://www.oculus.com/>. [Vaadatud: 07-märts-2019].
- [36] „VIVE™ | VIVE Virtual Reality System“. [Võrgumaterjal]. Kättesaadav: <https://www.vive.com/us/product/vive-virtual-reality-system/>. [Vaadatud: 30-märts-2019].
- [37] „PlayStation VR Aim Controller – PlayStation VR Accessories“. [Võrgumaterjal]. Kättesaadav: <https://www.playstation.com/en-us/explore/accessories/playstation-vr-aim-controller/>. [Vaadatud: 07-märts-2019].
- [38] „VIVE | Controller“. [Võrgumaterjal]. Kättesaadav: <https://www.vive.com/us/accessory/controller/>. [Vaadatud: 10-apr-2019].

- [39] *OpenPose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation: CMU-Perceptual-Computing-Lab/openpose*. CMU-Perceptual-Computing-Lab, 2019.
- [40] „Kinect“, *Wikipedia*. 22-märts-2019.
- [41] „Tracking camera T265 – Intel RealSense Depth and Tracking Cameras“, *Intel® RealSense™ Depth and Tracking Cameras*. [Vörgumaterjal]. Kättesaadav: <https://www.intelrealsense.com/tracking-camera-t265/>. [Vaadatud: 10-apr-2019].
- [42] A. McWilliams, „How a Depth Sensor Works - in 5 Minutes | Andrew McWilliams“. [Vörgumaterjal]. Kättesaadav: <https://jahya.net/blog/how-depth-sensor-works-in-5-minutes/>. [Vaadatud: 08-apr-2019].
- [43] „Leap Motion“. [Vörgumaterjal]. Kättesaadav: <https://www.leapmotion.com/>. [Vaadatud: 30-märts-2019].
- [44] „How Does the Leap Motion Controller Work?“, *Leap Motion Blog*, 09-aug-2014. [Vörgumaterjal]. Kättesaadav: <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>. [Vaadatud: 05-apr-2019].
- [45] „Bebop Sensors introduces wireless data glove at CES 2019 - Cerebral-Overload“. [Vörgumaterjal]. Kättesaadav: <https://cerebral-overload.com/2018/12/bebop-sensors-introduces-wireless-data-glove-at-ces-2019/>. [Vaadatud: 07-mai-2019].
- [46] „What is data glove? - Definition from WhatIs.com“, *WhatIs.com*. [Vörgumaterjal]. Kättesaadav: <https://whatis.techtarget.com/definition/data-glove>. [Vaadatud: 30-märts-2019].
- [47] „BeBop Sensors | Make Things Knowable“. [Vörgumaterjal]. Kättesaadav: <https://bebopsensors.com/>. [Vaadatud: 07-mai-2019].
- [48] „ROS.org | About ROS“.
- [49] „rviz - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: <http://wiki.ros.org/rviz>. [Vaadatud: 07-märts-2019].
- [50] „gazebo - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: <http://wiki.ros.org/gazebo>. [Vaadatud: 07-märts-2019].
- [51] „ROS/Introduction - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: <https://wiki.ros.org/ROS/Introduction>. [Vaadatud: 07-märts-2019].
- [52] „Bringing the power of Windows 10 to the Robot Operating System“, *Windows Experience Blog*, 28-sept-2018. [Vörgumaterjal]. Kättesaadav: <https://blogs.windows.com/windowsexperience/2018/09/28/bringing-the-power-of-windows-10-to-the-robot-operating-system/>. [Vaadatud: 05-apr-2019].
- [53] „cn/Industrial - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: <http://wiki.ros.org/cn/Industrial>. [Vaadatud: 08-märts-2019].
- [54] „ROSBerryPi - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: <http://wiki.ros.org/ROSBerryPi>. [Vaadatud: 08-märts-2019].
- [55] „ROS/Concepts - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: <https://wiki.ros.org/ROS/Concepts>. [Vaadatud: 07-märts-2019].
- [56] „rviz/DisplayTypes - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: <http://wiki.ros.org/rviz/DisplayTypes>. [Vaadatud: 08-märts-2019].
- [57] „rviz_animated_view_controller - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: http://wiki.ros.org/rviz_animated_view_controller. [Vaadatud: 08-märts-2019].
- [58] „oculus_rviz_plugins - ROS Wiki“. [Vörgumaterjal]. Kättesaadav: http://wiki.ros.org/oculus_rviz_plugins. [Vaadatud: 08-märts-2019].
- [59] *Use the power of OSVR library in RViz. Contribute to*

- UTNuclearRoboticsPublic/rviz_plugin_osvr development by creating an account on GitHub.* UTNuclearRoboticsPublic, 2018.
- [60] „Gazebo“. [Võrgumaterjal]. Kättesaadav: <http://gazebosim.org/>. [Vaadatud: 05-apr-2019].
- [61] „Gazebo : Blog“. [Võrgumaterjal]. Kättesaadav: <http://gazebosim.org/blog.html>. [Vaadatud: 08-apr-2019].
- [62] *General information that's easier to post on github than Google Docs, mainly set up for the wiki page: AS4SR/general_info.* Autonomous Systems for S.P.A.C.E. R.O.B.O.T.I.C.S. Laboratory, 2019.
- [63] „Gazebo : Tutorial : Oculus Rift“. [Võrgumaterjal]. Kättesaadav: <http://gazebosim.org/tutorials?cat=&tut=oculus&ver=3.0-5.0>. [Vaadatud: 08-apr-2019].
- [64] *Issue tracker for the Linux port of SteamVR. Contribute to ValveSoftware/SteamVR-for-Linux development by creating an account on GitHub.* Valve Software, 2019.
- [65] V. Mazzari, „Robotic simulation scenarios with Gazebo and ROS“, *Génération Robots - Blog*, 26-veebr-2015. .
- [66] „kinect - ROS Wiki“. [Võrgumaterjal]. Kättesaadav: <http://wiki.ros.org/kinect>. [Vaadatud: 22-apr-2019].
- [67] S. Qiao, Y. Wang, ja J. Li, „Real-time human gesture grading based on OpenPose“, *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, lk 1–6.
- [68] S. J. Jorgensen, *A ros wrapper for the CMU openpose library. Contribute to stevenjj/openpose_ros development by creating an account on GitHub.* 2019.
- [69] „FAQs | MoveIt!“ [Võrgumaterjal]. Kättesaadav: <https://moveit.ros.org/documentation/faqs/>. [Vaadatud: 08-märts-2019].
- [70] „Rosbridge - sdk-wiki“. [Võrgumaterjal]. Kättesaadav: <http://sdk.rethinkrobotics.com/wiki/Rosbridge>. [Vaadatud: 02-apr-2019].
- [71] „rosbridge_suite - ROS Wiki“. [Võrgumaterjal]. Kättesaadav: http://wiki.ros.org/rosbridge_suite. [Vaadatud: 05-apr-2019].
- [72] M. C. Thorstensen, *A Unity ROS tool. Contribute to MathiasCiarlo/ROSBridgeLib development by creating an account on GitHub.* 2019.
- [73] *Unreal Engine Plugin to enable ROS Support. Contribute to code-iai/ROSIntegration development by creating an account on GitHub.* Institute for Artificial Intelligence - University of Bremen, 2019.
- [74] „Documentation“, *Leap Motion Developer.* [Võrgumaterjal]. Kättesaadav: <http://developer.leapmotion.com/documentation>. [Vaadatud: 02-apr-2019].
- [75] „Leap Motion“, *Wikipedia*. 03-dets-2018.
- [76] „Touchless For Windows – Leap Motion Gallery“. [Võrgumaterjal]. Kättesaadav: <https://gallery.leapmotion.com/touchless-for-windows/>. [Vaadatud: 02-apr-2019].
- [77] „HTC Vive FAQ: What You Need to Know“, *Leap Motion Blog*, 05-apr-2016. [Võrgumaterjal]. Kättesaadav: <http://blog.leapmotion.com/leap-motion-htc-vive-faq/>. [Vaadatud: 05-apr-2019].
- [78] „leap_motion - ROS Wiki“. [Võrgumaterjal]. Kättesaadav: http://wiki.ros.org/leap_motion. [Vaadatud: 10-märts-2019].
- [79] K. Allaje, „Draiveripakettikäte jälgimise seadme Leap Motion™ kontrolleri kasutamiseks robotika arendusplatvormil ROS“, Tartu ülikool, 2018.
- [80] „Where to buy“, *Leap Motion.* [Võrgumaterjal]. Kättesaadav: <https://www.leapmotion.com/where-to-buy/>. [Vaadatud: 05-apr-2019].
- [81] L. E. | July 22 ja 2013, „Leap Motion Controller“. [Võrgumaterjal]. Kättesaadav: <https://www.laptopmag.com/reviews/accessories/leap-motion-controller>. [Vaadatud: 05-apr-2019].

- [82] „Review of Leap Motion Technology | Leap Motion Review“, *Farotech*, 11-okt-2013. [Võrgumaterjal]. Kättesaadav: <https://farotech.com/blog/review-leap-motion-new-touchless-technology/>. [Vaadatud: 02-apr-2019].
- [83] F. Weichert, D. Bachmann, B. Rudak, ja D. Fisseler, „Analysis of the Accuracy and Robustness of the Leap Motion Controller“, *Sensors*, kd 13, nr 5, lk 6380–6393, mai 2013.
- [84] „Gestures — Leap Motion C# SDK v2.3 documentation“. [Võrgumaterjal]. Kättesaadav: https://developer-archive.leapmotion.com/documentation/v2/csharp/devguide/Leap_Gestures.html. [Vaadatud: 22-apr-2019].
- [85] „Pinch Move – Leap Motion Gallery“. [Võrgumaterjal]. Kättesaadav: <https://gallery.leapmotion.com/pinch-move/>. [Vaadatud: 22-apr-2019].
- [86] K. Kruusamäe ja M. Pryor, „High-precision telerobot with human-centered variable perspective and scalable gestural interface“, *2016 9th International Conference on Human System Interactions (HSI)*, 2016, lk 190–196.
- [87] P. Zhang, X. Liu, G. Du, B. Liang, ja X. Wang, „A markerless human-manipulators interface using multi-sensors“, *Ind. Robot Int. J.*, kd 42, nr 6, lk 544–553, okt 2015.
- [88] „Running the Leap Motion Diagnostics“, *Leap Motion Support*. [Võrgumaterjal]. Kättesaadav: <http://support.leapmotion.com/hc/en-us/articles/223783868-Running-the-Leap-Motion-Diagnostics>. [Vaadatud: 02-apr-2019].
- [89] R. Valner, K. Kruusamäe, ja M. Pryor, „TeMoto: Intuitive Multi-Range Telerobotic System with Natural Gestural and Verbal Instruction Interface“, *Robotics*, kd 7, nr 1, lk 9, veebr 2018.
- [90] „UsingVRforHR_8b8f5e3b2e2ad9f1d8672930bc6d5d80.jpg (1200×700)“. [Võrgumaterjal]. Kättesaadav: https://rfsdelivers.com/media/zoo/images/UsingVRforHR_8b8f5e3b2e2ad9f1d8672930bc6d5d80.jpg. [Vaadatud: 16-mai-2019].
- [91] „UR5 Universal Robot - SCOTT Automation & Robotics » Scott“. [Võrgumaterjal]. Kättesaadav: <https://www.scottautomation.com/products/ur5-universal-robot/>. [Vaadatud: 16-mai-2019].
- [92] „ROS Logo - Works on Arm“. [Võrgumaterjal]. Kättesaadav: <https://www.worksonarm.com/explore/ros/ros-logo/>. [Vaadatud: 16-mai-2019].
- [93] „MoveIt Motion Planning Framework“. [Võrgumaterjal]. Kättesaadav: <https://moveit.ros.org/>. [Vaadatud: 19-mai-2019].
- [94] V. Vunder, R. Valner, C. McMahon, K. Kruusamäe, ja M. Pryor, „Improved Situational Awareness in ROS Using Panospheric Vision and Virtual Reality“, *2018 11th International Conference on Human System Interaction (HSI)*, 2018, lk 471–477.
- [95] *Easy to use tools to create and edit trajectories for the rviz camera. : AIS-Bonn/rviz_cinematographer*. AIS Bonn, 2019.
- [96] „kinetic - ROS Wiki“. [Võrgumaterjal]. Kättesaadav: <http://wiki.ros.org/kinetic>. [Vaadatud: 01-mai-2019].
- [97] „rviz/DisplayTypes/Marker - ROS Wiki“. [Võrgumaterjal]. Kättesaadav: <http://wiki.ros.org/rviz/DisplayTypes/Marker>. [Vaadatud: 22-apr-2019].
- [98] „visualization_msgs/Marker Documentation“. [Võrgumaterjal]. Kättesaadav: http://docs.ros.org/melodic/api/visualization_msgs/html/msg/Marker.html. [Vaadatud: 22-apr-2019].
- [99] „3D human hand“, *Yobi3D*. [Võrgumaterjal]. Kättesaadav: <https://www.yobi3d.com/q/human%20arm?page=3>. [Vaadatud: 12-dets-2018].
- [100] „planning_interface: moveit::planning_interface::MoveGroupInterface Class Reference“. [Võrgumaterjal]. Kättesaadav: http://docs.ros.org/melodic/api/moveit_ros_planning_interface/html/classmoveit_1_1planning_

[_interface_1_1MoveGroupInterface.html#ad72e3d9367374e388546c8e81f26035a](#). [Vaadatud: 22-apr-2019].

[101] „planning_interface: moveit::planning_interface::MoveGroupInterface Class Reference“. [Võrgumaterjal]. Kättesaadav: http://docs.ros.org/melodic/api/moveit_ros_planning_interface/html/classmoveit_1_1planning_interface_1_1MoveGroupInterface.html#a3513c41b0c73400fc6713b25bc6b1637. [Vaadatud: 22-apr-2019].

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina,

Georg Astok,

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

“Roboti juhtimine virtuaalreaalsuses kasutades ROS-raamistikku”,

mille juhendajad on Karl Kruusamäe ja Robert Valner,

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'i litsentsiga CC BY NC ND 3.0, 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 lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Georg Astok

02.06.2019