

Tartu Ülikool  
Loodus- ja täppisteaduste valdkond  
Tehnoloogiainstituut

Tarvi Tepandi

**Segareaalsusel põhinev kasutajaliides mobiilse roboti kaugjuhtimiseks  
Microsoft HoloLens 2 vahendusel**

Bakalaureusetöö (12 EAP)  
Arvutitehnika eriala

Juhendaja:  
kaasprofessor Karl Kruusamäe

Tartu 2021

# Resümee/Abstract

## Segareaalsusel põhinev kasutajaliides mobiilse roboti kaugjuhtimiseks Microsoft HoloLens 2 vahendusel

Segareaalsus on võimas tööriist, mille täielik potentsiaal on veel kasutamata erinevates valdkondades, sealhulgas robotikas. Kuigi liit- ja virtuaalreaalsus võimaldavad luua innovatiivseid kasutajaliideseid, puudub selline lahendus ROS platvormil põhineva Robotondi roboti kaugjuhtimiseks. Käesoleva töö eesmärk on luua Microsoft HoloLens 2 põhinev kasutajaliides sihtkoha seadmiseks ROSi-põhistele robotitele, mis kasutavad navigeerimiseks *move\_base* kimpu.

Valmis lahendus, kus visiiril on kujutatud roboti mudel, siiani roboti poolt kaardistatud 2D kaart ning marker, mida liigutades kaardil saab seada roboti kaardistamisalgoritmile sihtkoha. Tulemusele saab juurde arendada veel intuitiivsemalt mõistetavaid sihtkoha seadmise viise (näiteks päris roboti suhtes või näpuga näidates), lisada tugi 3D kaardistamis algoritmidele või parandada jõudlust, et kasutajaelamus oleks parem.

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

**Märksõnad:** segareaalsus, Microsoft HoloLens 2, ROS

## Mixed-reality user interface for teleoperating mobile robots with Microsoft HoloLens 2

Mixed reality is a powerful tool, the full potential of which is not used in various fields, among them robotics. Although augmented- and virtual reality can create innovative user interfaces, one does not exist for the ROS-based robot platform Robotont teleoperations. The purpose of this thesis is to create a Microsoft HoloLens 2-based user interface for setting the movement goal for any robot that uses the *move\_base* action for navigation.

The created solution displays the robot model, the map resulting of a mapping algorithm and a marker, which can be manipulated on the map to set the navigation goal. The solution can be improved upon by adding different more immersive ways of setting the goal (such as relative to the actual robot or by pointing a finger), adding support for 3D mapping algorithms or improving the performance so that that the user experience is better.

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

**Keywords:** mixed reality, Microsoft HoloLens 2, ROS

# Sisukord

<b>Resümee/Abstract</b>	<b>2</b>
<b>Sisukord</b>	<b>3</b>
<b>Jooniste loetelu</b>	<b>5</b>
<b>Lühendid, konstandid, mõisted</b>	<b>6</b>
<b>1 Sissejuhatus</b>	<b>7</b>
1.1 Probleemi tutvustus . . . . .	7
1.2 Töö eesmärk ja ülevaade . . . . .	7
<b>2 Kirjanduse ülevaade</b>	<b>8</b>
2.1 Segareaalsus . . . . .	8
2.2 Segareaalsus robotika rakendustes . . . . .	9
<b>3 Töövahendite ülevaade</b>	<b>12</b>
3.1 Hololens . . . . .	12
3.2 Unity . . . . .	12
3.3 ROS . . . . .	12
3.3.1 Robotont . . . . .	12
3.4 ROS Sharp . . . . .	14
3.4.1 Rosbridge . . . . .	14
<b>4 Lahenduse ülevaade</b>	<b>15</b>
4.1 Nõuded . . . . .	15
4.1.1 Töö eesmärk . . . . .	15
4.1.2 Funktsionaalsed nõuded . . . . .	15
4.1.3 Mitte-funktsionaalsed nõuded . . . . .	15
4.2 Disain . . . . .	15
4.2.1 Tööpõhimõte . . . . .	15
4.2.2 Arhitektuur . . . . .	16
4.2.3 ROSi poolne konfiguratsioon . . . . .	17
4.2.4 Unity poolne konfiguratsioon . . . . .	18
<b>5 Tulemuste analüüs ja ülevaade</b>	<b>20</b>
5.1 Valminud lahendus . . . . .	20
5.2 Võimalikud edasiarendused . . . . .	20
5.2.1 Teised viisid sihtkoha andmiseks . . . . .	20
5.2.2 3D kaardistamine . . . . .	20

5.2.3 Jõudluse parandamine . . . . .	20
<b>6 Kokkuvõte</b>	<b>22</b>
<b>Kokkuvõte</b>	<b>22</b>
<b>Kasutatud kirjandus</b>	<b>23</b>
<b>Lihtlitsents</b>	<b>27</b>



# Jooniste loetelu

2.1	Kaks inimest vaatavad virtuaalset kujutist [1]. . . . .	8
2.2	Kuvatõmmis videost, kus demonstreeriti segareaalsuse abil roboti trajektööri programmeerimist [2]. . . . .	9
2.3	Radioaktiivsete või keemiliste leketega piirkondades kaugjuhtimiseks loodud robot [3]. . . . .	10
2.4	Meditšiinis saab roboti poolt tehtavaid löikeid piirata virtuaalsete kinnitustega (kuvatud punase koonusena), mis määrab ala, mille seintest robot tööriistaga välja ei tohi sattuda [4]. . . . .	11
3.1	Microsoft Hololens 2 segareaalsusvisiir [5]. . . . .	13
3.2	Robotont robotiplatvorm [6]. . . . .	13
4.1	Lahenduse kontseptsioon. Rohelisena kujutatud sihtkoha seadmine ning roboti liikumistee. Kujutised pärit allikatest [6] [7]. . . . .	16
4.2	Lahenduses valminud kasutajaliides, kus kujutatud on valge silindrikujuline marker, roboti mudel ning 2D kaart. . . . .	17
4.3	Hololensi ja roboti vahelises suhtluses saadetakse informatsioon. Siiani kaardistatud kaart ja roboti positsioon on kuvatud visiiril. Visiiril sihtkoha seades saadetakse see edasi robotile. . . . .	17
4.4	Hololensis scriptidega töödeldatav baasfunktsionaalsuse tagav informatsioon. Punasega on tähistatud lõputöö jooksul valminud scriptid/objektid. Tärniga on tähistatud imporditud sõnumitüübid/objektid. . . . .	18
4.5	Sihtkoha seadmise protsess. Algselt on kujutatud robot kaardil (samm 1), korjates üles markeri (samm 2) on võimalik seda manipuleerida, selle maha asetades (samm 3) hakkab robot selle suunas liikuma. . . . .	19

# Lühendid, konstandid, mõisted

**API** - *Application Programming Interface*, eeldefineeritud eeskujul erinevate tarkvarakomponentide vahel suhtlust võimaldav rakendusliides

**JSON** - *JavaScript Object Notation*, andmevahetusel kasutatav parsimisformaad

**MRTK** - *Mixed Reality Toolkit*, tööriistade kogum segareaalsuse arendamiseks

**ROS** - *Robot Operating System*, tarkvararaamistik robotitarkvara kirjutamiseks, mis varustab endas tööriistu, teeke ja tavaid

**URDF** - *Unified Robot Description Format*, robotimudeli kujutamise vorming

**UWP** - *Universal Windows Platform*, Microsofti poolt loodud arendusplatvorm, et ühtsustada tarkvara, mis on loodud erinevale riistvarale ja operatsioonisüsteemile

# 1 Sissejuhatus

Segareaalsus on võimas tööriist tänapäeval mitmes erinevas valdkonnas, sealhulgas robotikas [4]. Kuigi turule on jõudnud erinevad visiirid, mis toetavad kas liit- või virtuaalreaalsust ning arendatud on ka võimalusi kasutada nutitelefoni liidesena, on tehnoloogia veel arenemas ning turule jõuavad tihti uued segareaalsuspõhised lahendused [8].

Virtuaalreaalsusplatvormid nagu Vive [9] ja Oculus Quest 2 [10], liitreaalsusplatvormid nagu Microsoft HoloLens 2 [11] ja Magic Leap 1 [12] kui ka erinevad nutitelefonidel põhinevad lahendused [13] on vaid mõned näited tänapäeva segareaalsusvõimekusest. Platvormide mitmekesisusest tulenevalt on aga ühe rakenduse kõikidele platvormidele välja andmine väga keeruline [8].

## 1.1 Probleemi tutvustus

Robotikas saab segareaalsust kasutada innovatiivselt ära nii tarbijatehnoloogias, militaarvaldkondades, meditsiinis kui ka programmeerimisel [4] [14]. Üks laialdasemalt kasutatud eesmärkidest on kaugjuhtimine, militaarvaldkondades näiteks selleks, et hoida inimest ohupiirkonnast eemal, tarbijatehnoloogias meelelahutuslikult drooni lennutamiseks [4] [15]. Kuigi teleoperatsioonide rakendusi luuakse pidevalt juurde, puudub siiani hea viis ROS platvormil põhineva Robotondi roboti juhtimiseks segareaalsuse abil [8] [16].

## 1.2 Töö eesmärk ja ülevaade

Käesoleva bakalaureusetöö eesmärgiks on luua segareaalsusplatvormil põhinev kasutajaliides mobiilse roboti kaardistamis- ja rajaplaneerimisalgoritmi juhtimiseks.

Lõputöös räägitakse segareaalsuse rakendustest tänapäeval (peatükk 2), kirjeldatakse valminud lahenduses kasutatud tööriistu (peatükk 3), püstitatakse täpsed nõuded ja disain valminud lahendusele (peatükk 4) ning analüüsitakse valminud lahendust (peatükk 5).

## 2 Kirjanduse ülevaade

### 2.1 Segareaalsus

Segareaalsus, mis koosneb liit- ja virtuaalreaalsusest, on tehnoloogia, mis võimaldab kujutada kasutajale virtuaalseid ja pärismaailma elemente segamini (joonis 2.1) [17]. Võimalus olla ümbritsetud virtuaalsete elementidega, samal ajal nähes tegelikku maailma, kasutada häälkäsklusi ning virtuaalsete elementide manipuleerimine neid kätega üles korjates on vaid mõni tehnoloogiline võimekus, mida kaasaegsed segareaalsuse platvormid võimaldavad [8].

Kõik segareaalsuse elemendid koostöös annavad arendajale hea aluse luua erinevaid kasutajali-

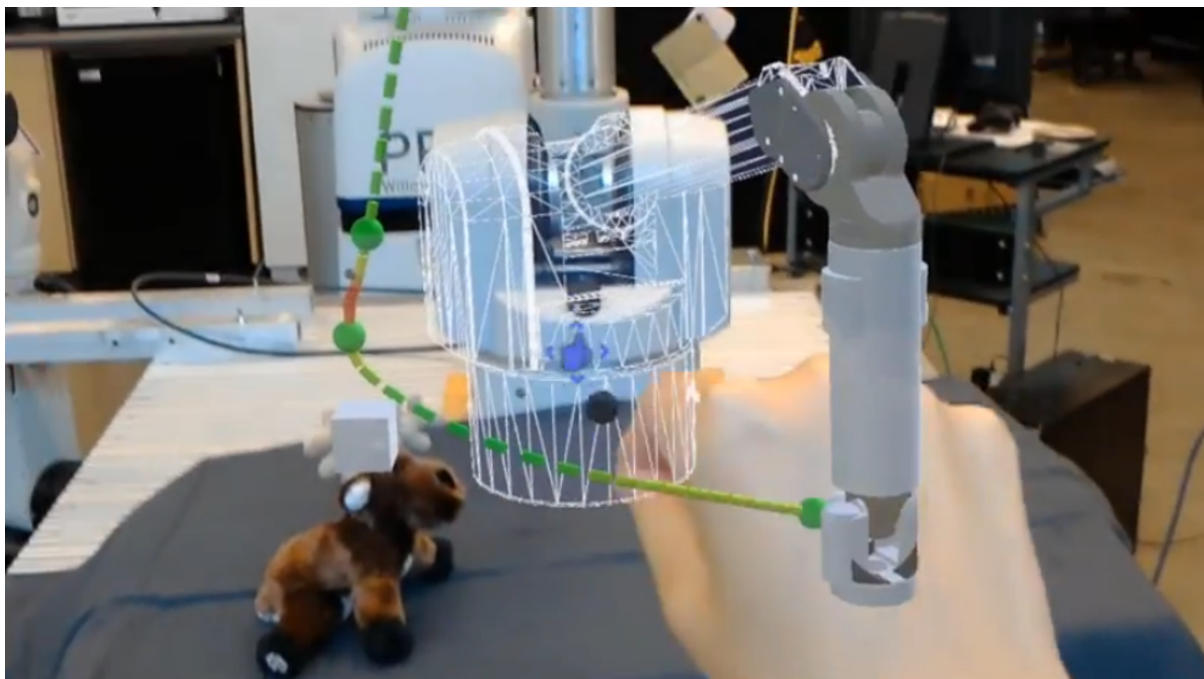


Joonis 2.1: Kaks inimest vaatavad virtuaalset kujutist [1].

ideseid erinevatele valdkondadele. Liit- ja virtuaalreaalsuse eelis, võrreldes seniste tehnoloogiatega, on see, et nendele omapäraseid võimekusi ära kasutades on võimalik luua intuiitselt aru saadavad lahendused. Erinevalt tüüpilise arvuti ja monitori kasutamisest, ei ole kasutaja piiratud kahedimensioonilise piltväljundi, klaviatuuri ja hiirega. [18]

Segareaalsusele on juba praegu palju erinevaid kasutusvaldkondi. Turismitööstuses virtuaalne giid [19] või õppetöös koolitamisvahend [20]. Interaktiivne õpe läbi mängude [21] ning meditatiinis patsiendiga virtuaalse konsultatsiooni vahend [22]. Liit- ja virtuaalreaalsuse abil loodud intuiivsete lahenduste puhul on võimalik kasutajale kujutada rohkem informatsiooni, saada täpsemaid sisendeid ning luua vähem koolitamist nõudvaid rakendusi [23].

Ometigi, et segareaalsuse kasutamine alternatiivide asemel arendaks kasutajaelamust paljudes valdkondades, on selle kasutamisele võtmisel takistavaid tegureid. Näiteks arendamise keerukus ja kulukus ning visiiride tootmise lisakulud. Kuigi kokkuvõttes lubavad liit- ja virtuaalreaalsus



Joonis 2.2: Kuvatõmmis videost, kus demonstreeriti segareaalsuse abil roboti trajektööri programmeerimist [2].

luua intuitiivsemaid lahendusi, on segareaalsuse igapäevasele kasutusele võtmisel õppeprotsess. [18]

## 2.2 Segareaalsus robotika rakendustes

Alates üheksakümnendatest on uuritud segareaalsuse võimalusi robotikas [8]. Aastate jooksul on selle tõttu tekkinud palju võimalusi kasutada ära segareaalsust robotite ja inimeste koostöös. Üks peamistest põhjustest arendada robotikas segareaalsusrakendust on, et segareaalsus võimaldab kasutajal loomulikumalt robotiga suhelda [24].

Liit- ja virtuaalreaalsuse üks kasutusala on robotite programmeerimine. Näiteks, robotkäe liikumistrajektoriks oleva punktide järjendi saab sisestada liitreaalsuses (joonis 2.2). On juba valminud lahendus, kus häälkäskluste ja žestide põhjal oli võimalik programmeerida robotit, kusjuures virtuaalses ruumis lisatud punktid ja robot olid läbi Hololensi näha [2]. Sarnaselt on demonstreeritud segareaalsuse põhist viisi teisaldada kätežestid robotkäsklusteks [25] ning roboti programmeerimist läbi virtuaalreaalsuse kasutades roboti digitaalset kaksikut [14].

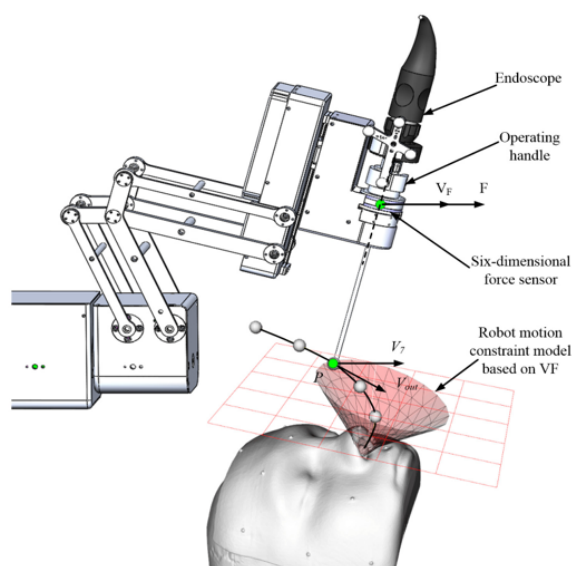
Võimalusi kasutada segareaalsust kaugjuhtimiseks ehk teleoperatsioonideks on palju. ROSi ja Hololensi koostööd kasutati uuringus, et koostada virtuaalruum robotitele [26]. Virtuaalruumis kasutajale oli kujutatud füüsiliste robotite liikumine olenemata, et robot polnud silmnähtav. Militaarvaldkondades robotite kaugjuhtimine (joonis 2.3) kasutab segareaalsust eesmärgiga vältida inimeste kokkupuudet ohtlike olukordadega [4]. Kaugel ohupiirkonnas asetsevaid roboteid saab juhtida kas žestidega või käes hoitava märgiga, mille tõttu pole vaja inimesel ohtlikku olukorda sattuda.



Joonis 2.3: Radioaktiivsete või keemiliste leketega piirkondades kaugjuhtimiseks loodud robot [3].

Kasutades segareaalsust saab vähendada ka ohtu meditsiinivaldkonnas, kui operatsiooni viib läbi robot inimese juhtimisel [4]. Kirurgil on võimalik teha mitu erinevat virtuaalset lõiget, kus tööriista liikumisteed salvestatakse roboti jaoks. Enne, kui robot lõike teeb, saab kirurg segareaalsust kasutades näha kõiki salvestatud virtuaalsete lõigete teekondi. Kuvatavad lõiked teeb roboti virtuaalne teisik, mis asetseb päris robotiga kohakuti. Valides nähtud virtuaalruumi lõigetest parima, teostab just sellise lõike päris robot. Kuna lõike teeb robot, on ohutuse suurendamiseks võimalik lisada juurde ka robotile piiranguid virtuaalsete kinnituste (ingl *virtual fixture*) kujul [4]. Virtuaalsed kinnitused ei luba roboti poolt juhitud tööriistal minna määratud ruumist välja ning seega välistatakse võimalus teha patsiendile kahju (joonis 2.4) [4].





Joonis 2.4: Meditsiinis saab roboti poolt tehtavaid löikeid piirata virtuaalsete kinnitustega (kuvatud punase koonusena), mis määrab ala, mille seintest robot tööriistaga välja ei tohi sattuda [4].

## 3 Töövahendite ülevaade

### 3.1 Hololens

Microsoft Hololens 2 (joonis 3.1) on segareaalsusvisiir, mis ilmus turule 7. novembril 2019. aastal [11]. Olles arendatud edasi eelmiselt Hololens seadmelt, parandati selle jõudlust, ekraani ulatust ning käte tuvastamist ning lihtsustati seadmele arendamise protsessi [27]. Lisatud sensorite hulgas on näiteks kiirendusmõõtur, güroskoop, magnetomeeter, sügavuskaamera, video-kaamera, valgussensor ja mikrofoni. Funktsionaalsuselt lisandub veel käteliigutuste- ja hääletuvastus võimekus. Kuna see kasutab ka varem tuntuid Wi-Fi ja Bluetoothi standardeid suhtluseks, on kommunikatsioon seadmega võrdlemisi lihtne [11].

### 3.2 Unity

Unity on platvorm, millega saab luua nii 3D, 2D, veebipõhiseid, virtuaalreaalsuse, liitreaalsuse kui ka nutiseadmerakendusi [28]. Kuigi algselt loodud mängude arendamiseks, leiab see kasutust ka teaduse, inseneeria, kunsti, filmi ja tööstuse valdkondades. Kuigi ise kirjutatud programmeerimiskeeles C++, saab selles tehtud programmidele luua *scripte* keeles C#. [29]

### 3.3 ROS

ROS ehk *Robot Operating System* on Unixi-põhine raamistik robotitarkvara kirjutamiseks [30]. See varustab endas tööriistu, teeke ja tavaid. Olles loodud põhimõtetega olla skaleeritav ning innustada koodi taaskasutamist, on need vaid mõned põhjused, miks ROSi populaarsus kasvab [31]. ROSile on loodud põhjalik dokumentatsioon ning nii ROS ise kui ka paljud kolmandate osapoolte lisad on avatud lähtekoodiga [32].

ROSi tarkvara koosneb kimpudest (ingl *package*), mis sisaldab endas sõlmesid (ingl *node*). Erinevad sõlmed kasutavad üksteise vaheliseks suhtluseks sõnumeid (ingl *message*) ning rubriike (ingl *topic*). [33]

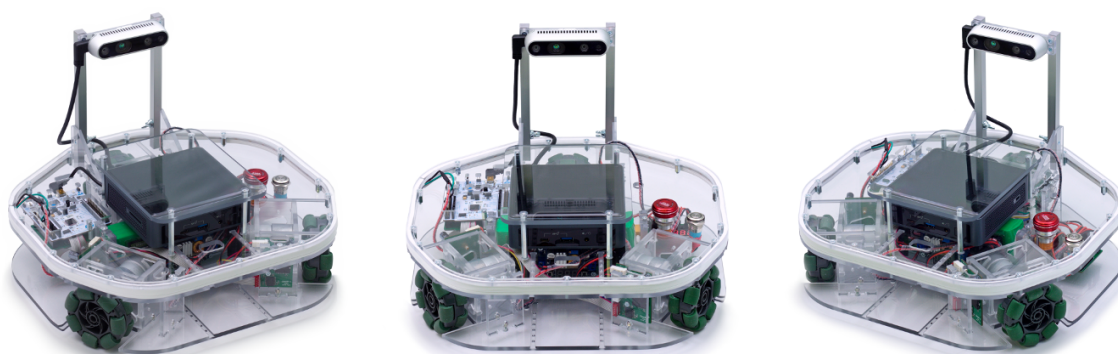
#### 3.3.1 Robotont

Robotont on mobiilne maas omniliikuv ROSi-põhine ja avatud lähtekoodiga robotiplatvorm (joonis 3.2) [34]. Selle sensorsüsteemi osaks on ka sügavuskaamera, mis võimaldab sellel kaardistada oma keskkonda [16]. Arendatuna Tartu Ülikooli tudengite poolt on sellele valminud ka põhjalik dokumentatsioon ning eestikeelsed õppematerjalid [35].





Joonis 3.1: Microsoft Hololens 2 segareaalsusvisiir [5].



Joonis 3.2: Robotont robotiplatvorm [6].

## 3.4 ROS Sharp

ROS# on avatud lähtekoodiga teekide ja tööriistade kogum ROSi ja .NET [36] raamistikul põhinevate rakenduste (eriti Unity) vaheliseks suhtluseks. [37] UWP ehk *Universal Windows Platform* [38] (ja ühtlasti Hololensi) toe lisamiseks on olemas ROS# baasversiooni edasiarendus [39].

### 3.4.1 Rosbridge

Rosbridge tagab JSON API [40] suhtluseks ROSi ja teiste programmide vahel. Koosnedes nii serverist kui kliendist, ROS# kasutab klienti, et suhelda ROS platvormi robotiga, millel käib Rosbridge server. [41]

## 4 Lahenduse ülevaade

### 4.1 Nõuded

#### 4.1.1 Töö eesmärk

Luuu rajaplaneerimisalgoritmiga käiva mobiilsele robotiplatvormile liitreaalsusvisiiri-põhine kasutajaliides liikumissihtmärgi andmiseks.

#### 4.1.2 Funktsionaalsed nõuded

1. Visiiril on kujutatud siiani kaardistatud kaart, mida uuendatakse pidevalt.
2. Kujutatud on ka roboti mudel, mille asukoht kaardi suhtes on korrektne.
3. Robotile sihtkoha andmine toimub kasutajale intuiitiivsel viisil läbi visiiril kujutatavate objektide manipuleerimise.
4. Sihtkoha seadmine toimib kõikidel ROSi-põhistel robotitel, mis kasutavad navigeerimiseks *move\_base* kimpu.

#### 4.1.3 Mitte-funktsionaalsed nõuded

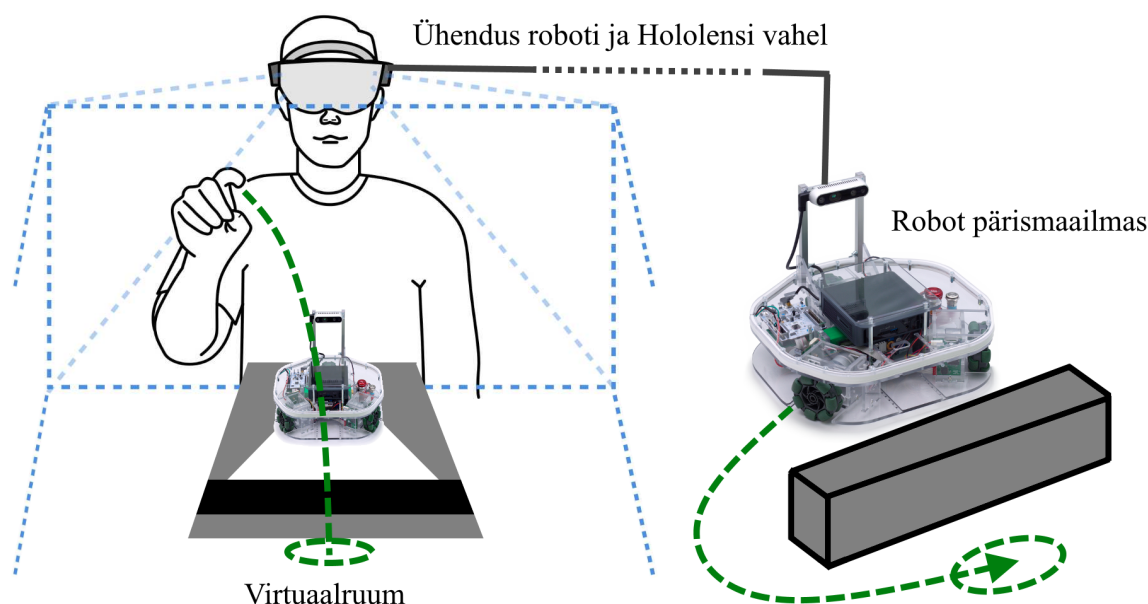
1. Segareaalsusvisiirina on kasutusel Microsoft Hololens 2.
2. Mobiilse robotiplatvormina on kasutusel Robotont.
3. Kasutatav mobiilne robotiplatvorm on juhtmevabas ühenduses visiiriga.
4. Robotondi pardaarvuti olemasolevat tarkvara ei muudeta (küll aga võib paigaldada ja käivitada täiendavaid mooduleid/ programme).

### 4.2 Disain

#### 4.2.1 Tööpõhimõte

Valminud lahenduses on võimalik visiiril kujutatava sihtkoha markeri manipuleerimisel saata sellega võrguühenduses olevale robotile sihtkoht, kuhu liikuda (joonis 4.1). Kasutatav mobiilne robot jooksub rajaplaneerimis- ja kaardistamisalgoritmi.

Kasutajale on näha segareaalsusvisiiri Microsoft Hololens 2 abil järgmised kujutised (joonis 4.2):



Joonis 4.1: Lahenduse kontseptsioon. Rohelisena kujutatud sihtkoha seadmine ning roboti liikumise. Kujutised pärit allikatest [6] [7].

- siiani kaardistatud kahedimensiooniline maailm, mida uuendatakse pidevalt;
- roboti mudel, mille asukoht on kaardist sõltuv;
- kaardil asetsev silindrikujuline marker, mida liigutades saab määrata robotile rajaplaneerimisalgoritmi liikumise sihtkohta.

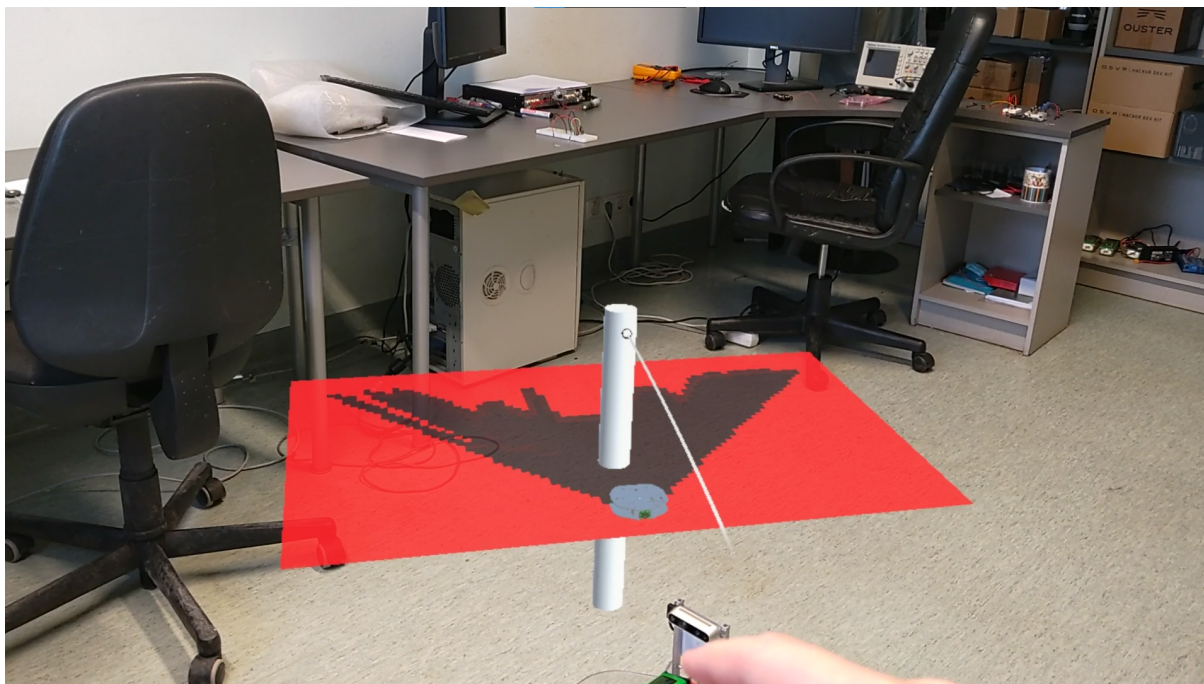
#### 4.2.2 Arhitektuur

Nõute ja funktsionaalsuse tagamiseks on esmatähtis ühendus visiiri ja roboti vahel. Ühenduse abil on võimalik küsida robotilt informatsiooni kaardistatud maailma ning oma asendi kohta, et seda kujutada visiiril. Lisaks, ühendus on vajalik, et saata robotil jooksva rajaplaneerimisalgoritmile sihtkoht (joonis 4.3).

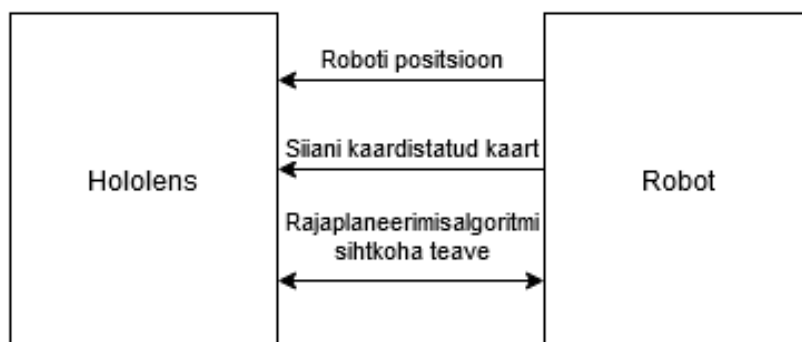
Süsteem on loodud kasutamaks visiirina Microsoft Hololens 2 Liitreaalsusprille ning ROS Navigation toega mobiilset robotiplatvormi. Antud visiir sai valitud oma võimekuse poolest olla kasutusel üldise segareaalsusplatvormina.

Microsoft Hololens 2 jaoks arendamisel antud töös kasutati Unity keskkonda. MRTK ning ROSsharp abil on võimalik ühendus roboti ja visiiri vahel.

Töös valminud kasutajaliides valmis viisil, et lahendus oleks võimalikult universaalne ning sobiks mistahes navigeerimiseks *move\_base* ROSi kimpu kasutavale robotile. Kommunikatsioonitoe lisamiseks on ROSi poolele vaja lisada *rosbridge\_server* kimp.



Joonis 4.2: Lahenduses valminud kasutajaliides, kus kujutatud on valge silindrikujuline marker, roboti mudel ning 2D kaart.



Joonis 4.3: Hololensi ja roboti vahelises suhtluses saadetakse informatsioon. Siiani kaardistatud kaart ja roboti positsioon on kuvatud visiiril. Visiiril sihtkoha seades saadetakse see edasi robotile.

### 4.2.3 ROSi poolne konfiguratsioon

Siiani kaardistatud kaardi kuvamiseks kasutatakse teavet `/map` rubriigilt. Sealne informatsioon on `nav_msgs/OccupancyGrid` [42] sõnumitüübis.

Et Hololensil oleks võimalik kuvada robotit korrektes asukohas, tuleb visiiril vastu võtta andmeid rubriigist `/odom`. See rubriik sisaldab endas roboti positsiooni, pööret ja kiirust.

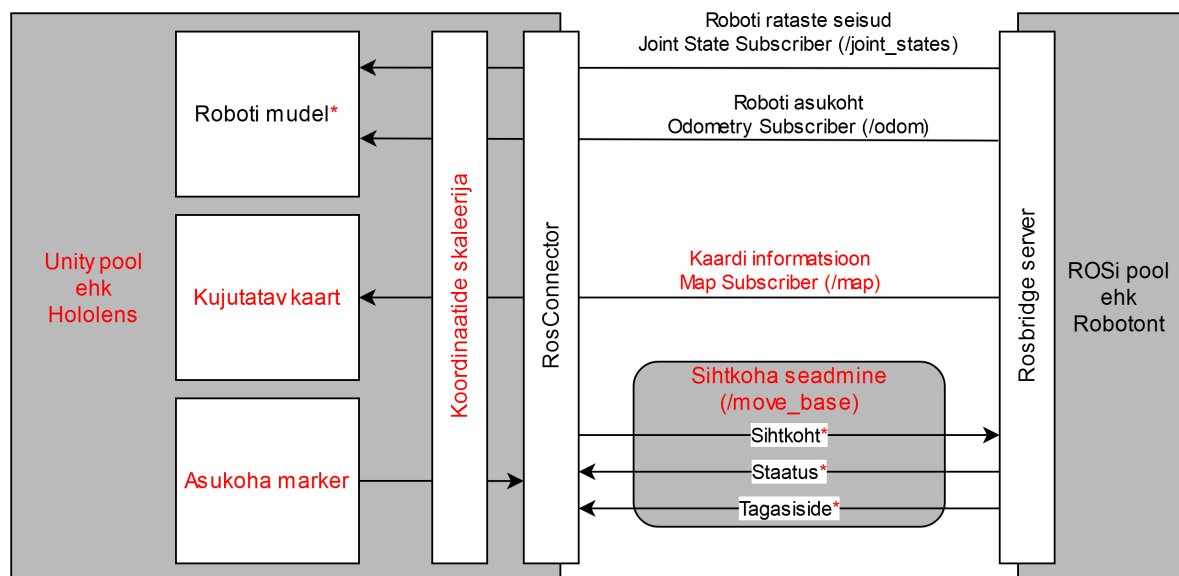
Sihtkoha seadmiseks kasutatakse `/move_base actioni`. Sihtkoha seadmiseks on `/move_base/goal` ning selle tühistamiseks `/move_base/cancel` rubriigid. Tagasiside, staatus ning tulemus kirjutatakse vastavalt rubriikidele `/move_base/feedback`, `/move_base/status` ja `/move_base/result`.

Kõik kasutatud sõnumitüübid olid kas varasemalt ROS# tööriistade hulgas juba implementeeritud või imporditavad ROSist Unity'sse kasutades ROS#'iga kaasas olevaid importimistööriistu. Sama kehtib roboti mudeli kohta, mis tuli importida URDF [43] vormingus Unity'sse.

#### 4.2.4 Unity poolne konfiguratsioon

Unity keskkond on virtuaalruum, mille abil kujutatakse visiidil kõik lahenduses kasutusel olevad objektid. Tagamaks sobivust Unity keskkonnas arendamisel Hololensi ja ROSi vahel, kasutati ROSsharp'i ning MRTK [44] tööriistu. Täpsemalt kasutati UWP (*Universal Windows Platform*) [38] toega ROSsharp baasversiooni edasiarendust [39]. Funktsionaalsuse tagamiseks oli vaja luua ja kasutada mitut erinevat komponenti (joonis 4.4).

Unity-poolse ühenduse robotiga loob *ros connector script*, mis on osa ROS#'ist. *Ros connector*



Joonis 4.4: Hololensis scriptidega töeldatav baasfunktsionaalsuse tagav informatsioon. Punasega on tähistatud lõputöö jooksul valminud scriptid/objektid. Tärniga on tähistatud imporditud sõnumitüübid/objektid.

võtab sisendiks *rosbridge\_serverit* jooksvat roboti IP aadressi, ühendusprotokolli (*Web Socket UWP*), serialiseerimismeetodi (*Newtonsoft\_JSON*) kui ka ühenduse *timeout* aja sekundites.

ROS#'iga kaasnenud *scriptidest* on töös kasutatud järgmisi, mis tagavad selle, et roboti asukoht on korrektne ka Unity poolel:

- *Joint State Patcher*;
- *Joint State Subscriber*;
- *Odometry Subscriber*.

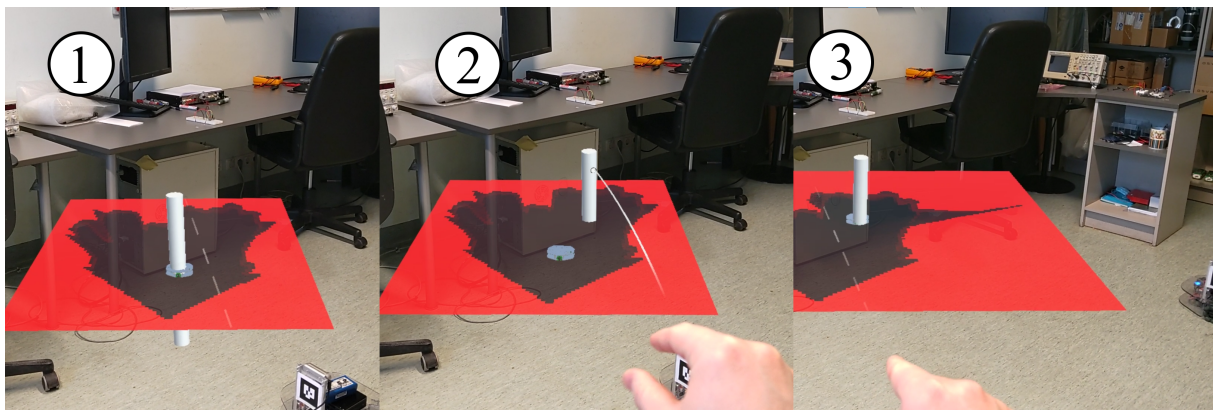
*Odometry Subscriber* saadab saadud info vahepealsele *scriptile*, et skaleerida roboti koordinaate, kuid mitte roboti suurust. See on vajalik selleks, et olenemata roboti ja kaardi suurusel,

on võimalik see kasutaja visiirile ära mahutada. Ühtlasi võimaldab vahepealse skaleerija olemasolu liigutada tervet kaarti korraga, et see ei oleks kinni seal, kus programm käivitati.

Töö jooksul loodud *script Map Subscriber* võtab sisendiks Unity tasapinna, millel kaart kujutata, ning robotilt kaardi rubriigilt */map*. Saades kaardi kohta informatsiooni kujutatakse see tasapinnal ja tasapind skaleeritakse viisil, et see oleks roboti mudeliga samas skaalas ning roboti algne asukoht (nullkoordinaat) jääks alati samaks.

*Unity Move Base Action Client script* on loodud suhtlemaks ROSipoolse *move\_base* kimbuga. Läbi selle *scripti* saadetakse rajaplaneerimisalgoritmile sihtkoht. *Unity Move Base Action Client script* põhineb abstraktsel klassil *ActionClient* ja kirjutab üle sealsed tööks vajalikud meetodid. Ühenduse andmetüübid on automaatselt genereeritud ROSis kasutatavate *move\_base\_msgs* sõnumitüüpide põhjal kasutades ROS#'iga kaasas olevat *RosBridgeClient* tööriista [45] [46].

Lahenduses kasutatakse visiiril asukoha määramisel silindrikujulist objekti (markerit, joonis 4.2), mida saab segareaalsuses üles korjata ja maha asetada. Markerit maha asetsemisel seab *Unity Move Base Action Client script* markeri koordinaadid roboti sihtkohaks (joonis 4.5).



Joonis 4.5: Sihtkoha seadmise protsess. Algselt on kujutatud robot kaardil (samm 1), korjates üles markeri (samm 2) on võimalik seda manipuleerida, selle maha asetades (samm 3) hakkab robot selle suunas liikuma.



## 5 Tulemuste analüüs ja ülevaade

### 5.1 Valminud lahendus

Töö jooksul valminud lahendus vastab varem püstitatud nõuetele, mis määravad kuvatava informatsiooni. Puudusena, ei toimu sihtkoha seadmine täielikult intuitiivsel viisil, vaid peab üles korjama markeri sihtkoha seadmiseks (näiteks näpuga näidates sihtkoha seadmine pole võimalik). Lisaks, kuvatav kaart on kahedimensiooniline ning ühevärviline, millel on erinevate elementide tuvastamine keeruline. Sellest olenemata puuduvad kriitilised probleemid lahenduses.

### 5.2 Võimalikud edasiarendused

#### 5.2.1 Teised viisid sihtkoha andmiseks

Kuigi lahenduses sihtkoha saatmine läbi markeri liigutamise toimib, saaks seda arendada implementeerides teisi viise sihtkoha määramiseks. Olgu selleks markeri kaotamine ja näpuga näitamine soovitud sihtkohale või silindrilise markeri asendamine millegiks, et näha oleks ka selle nurk.

Alternatiivselt võib tekitada võimaluse saata sihtkoht päris robotist sõltuvalt, kaotades vajaduse kujutada kaarti ja roboti mudelit. Tuvastatades roboti asukoha Hololensiga, on võimalik lisada juurde virtuaalne marker päris roboti kõrvale, millega edaspidi sihtkohta saata.

#### 5.2.2 3D kaardistamine

Tulemusena valminud lahenduses kujutatakse kaarti pildina, millel asetsevad roboti mudel ja sihtkoha marker. Kuigi see lahendus toimib väheste puudustega liikudes tasasel pinnal, siis liikudes muutliku kõrgusega maastikul ei ole esitatud kaart enam korrektne. Selle probleemi lahendamiseks peab muutma Unity-poolset kaardi esitamise scripti viisil, mis suudab kaarti esitada kolmedimensioonilisena. Vaja on muuta ka kasutatav kaardistamisalgoritm roboti poolel.

#### 5.2.3 Jõudluse parandamine

Tulenevalt riistvaralistest piirangutest on lahenduse jõudlus murekohaks, sest kuvatavate kaadrite arv sekundis piirdus kahekümnega. Selle mure lahendamiseks tuleb kas vähendada kaardi uuendamise sagedust, lihtsustada kaardi lugemise scripti keerukust või kasutada võimekamat riistvara Hololensi poolel.

Lisaks oli latents, kus peale robotile sihtkoha seadmist ning roboti liikumist kulus kasutajaliidesel sekundeid aega, et uuendada kuvatavat kaarti ning roboti mudeli asukohta. Kuna latentsi



ei esinenud simuleeritud robotit juhtides, võib oletada, et probleemi allikaks on võrguühendus roboti ja Hololensi vahel.

## 6 Kokkuvõte

Käesoleva bakalaureusetöö raames valmis liitreaalsusvisiiri Microsoft Hololens 2 põhine kasutajaliides ROSi-põhise omniliikuva roboti kaugjuhtimiseks. Robotile, millel käib Rosbridge server ning sobilikud kaardistamis- ja rajaplaneerimisalgoritm, on võimalik ühendada juurde visiir kasutajaliidese kuvamiseks.

Valminud lahenduses on kasutajale kuvatud siiani roboti poolt kaardistatud 2D kaart, selle suhtes asetsev roboti mudel ning marker, millega anda rajaplaneerimisalgoritmile sihtkoht. Markeri asetamisel saadetakse robotile käsk liikuda kohta, kus marker asetseb. Liikumise jooksul roboti kaardistamisalgoritm täiendab pidevalt siiani poolt kaardistatud kaarti, mis kuvatakse kasutajale.

Lõputöö jooksul valminud sihtkoha seadmise *script* on kasutatav kõikide roboti rajaplaneerimisalgoritmidega, mis põhinevad ROS *move\_base* kimbul. Roboti poolt kaardistatud kaart kuvatakse ROSi kimbult */map*.

Valminud lahendus on Unity-põhine ning selle raames valminud *scriptide* põhjal saab tööd edasi arendada mitmes suunas. Lisades juurde kas erinevaid võimalusi sihtkoha andmiseks või kolmedimensioonilise kaardistamisalgoritmi toe laiendaksid lahenduse kasutusvaldkondi. Jõudluse parandamine teeks kasutajakogemuse sujuvamaks.

Töö jooksul valminud Unity *scriptid* on saadaval lingil

<https://github.com/ut-ims-robotics/unity-ros-navigation-demo>.

# Kasutatud kirjandus

- [1] D. Pavlovic, “Kujutis kirjeldamaks mis on segareaalsus,” august 2018 (vaadatud 17.05.2021). <https://store.hp.com/app/assets/images/uploads/prod/what-is-mixed-reality-hero1537387902041.jpg?impolicy=prdimg&imdensity=1&imwidth=1000>.
- [2] C. P. Quintero, S. Li, M. K. Pan, W. P. Chan, H. F. Machiel Van der Loos, and E. Croft, “Robot programming through augmented trajectories in augmented reality,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1838–1844, 2018 (vaadatud 19.05.2021).
- [3] K. Qian, A. Song, J. Bao, and H. Zhang, “Small teleoperated robot for nuclear radiation and chemical leak detection,” *International Journal of Advanced Robotic Systems*, vol. 9, no. 3, p. 70, 2012 (vaadatud 19.05.2021).
- [4] B. Robert, “The role of augmented reality in robotics,” *Industrial Robot: the international journal of robotics research and application*, vol. 47, no. 6, pp. 789–794, 2020 (vaadatud 19.05.2021).
- [5] Microsoft, “Mixed reality news and notes,” 2019 (vaadatud 28.04.2021). <https://docs.microsoft.com/en-us/windows/mixed-reality/out-of-scope/news>.
- [6] “Pilt robotondi platvormist,” (vaadatud 28.04.2021). [http://robotont.ut.ee/sites/default/files/robotont/files/robotont\\_gen\\_2\\_3\\_multi\\_view.png](http://robotont.ut.ee/sites/default/files/robotont/files/robotont_gen_2_3_multi_view.png).
- [7] Microsoft, “Getting around hololens 2,” 2019 (vaadatud 28.04.2021). <https://docs.microsoft.com/en-us/hololens/hololens2-basic-usage>.
- [8] S. Rokhsaritalemi, A. Sadeghi-Niaraki, and S.-M. Choi, “A review on mixed reality: current trends, challenges and prospects,” *Applied Sciences*, vol. 10, no. 2, p. 636, 2020 (vaadatud 19.05.2021).
- [9] “Vive virtuaalreaalsusprillid,” mai 2021 (vaadatud 17.05.2021). <https://www.vive.com/us/product/>.
- [10] “Oculus quest 2,” mai 2021 (vaadatud 17.05.2021). <https://www.oculus.com/quest-2/>.
- [11] “Hololens 2 hardware,” oktoober 2020 (vaadatud 19.04.2021). <https://docs.microsoft.com/en-us/hololens/hololens2-hardware>.

- [12] “Magic leap 1,” mai 2021 (vaadatud 17.05.2021). <https://www.magicleap.com/en-us/magic-leap-1>.
- [13] “Tesseract holoboard,” mai 2021 (vaadatud 17.05.2021). <https://myholo.io/>.
- [14] A. Burghardt, D. Szybicki, P. Gierlak, K. Kurc, P. Pietruś, and R. Cygan, “Programming of industrial robots using virtual reality and digital twins,” *Applied Sciences*, vol. 10, no. 2, p. 486, 2020 (vaadatud 19.05.2021).
- [15] B. Huang, D. Bayazit, D. Ullman, N. Gopalan, and S. Tellex, “Flight, camera, action! using natural language and mixed reality to control a drone,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6949–6956, IEEE, 2019.
- [16] “Robotont overview,” 2021 (vaadatud 21.04.2021). <https://robotont.github.io/html/files/overview.html>.
- [17] Microsoft, “What is mixed reality?,” august 2020 (vaadatud 19.05.2021). <https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality>.
- [18] T. Watson, “Mixed reality benefits and uses in manufacturing,” november 2020 (vaadatud 17.05.2021). <https://skywell.software/blog/mixed-reality-benefits-uses-in-manufacturing/>.
- [19] CityGuideTour, “Rakenduse cityguidetour koduleht,” 2021 (vaadatud 17.05.2021). <https://www.cityguidetour.com/>.
- [20] S. Workman, “Mixed reality: A revolutionary breakthrough in teaching and learning,” juuli 2018 (vaadatud 17.05.2021). <https://er.educause.edu/articles/2018/7/mixed-reality-a-revolutionary-breakthrough-in-teaching-and-learning>.
- [21] S. Brewer, “Vr for kids and educational games,” oktoober 2020 (vaadatud 17.05.2021). <https://www.steampoweredfamily.com/education/vr-for-kids-educational-games/>.
- [22] J. Zhang, F. Gao, and Z. Ye, “Remote consultation based on mixed reality technology,” *Global Health Journal*, vol. 4, no. 1, pp. 31–32, 2020 (vaadatud 19.05.2021).
- [23] K. van Lopik, M. Sinclair, R. Sharpe, P. Conway, and A. West, “Developing augmented reality capabilities for industry 4.0 small enterprises: Lessons learnt from a content authoring case study,” *Computers in Industry*, vol. 117, p. 103208, 2020 (vaadatud 19.05.2021).
- [24] W. Hoenig, C. Milanes, L. Scaria, T. Phan, M. Bolas, and N. Ayanian, “Mixed reality for robotics,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5382–5387, IEEE, 2015 (vaadatud 19.05.2021).
- [25] S. Blankemeyer, R. Wiemann, L. Posniak, C. Pregizer, and A. Raatz, “Intuitive robot programming using augmented reality,” *Procedia CIRP*, vol. 76, no. 7th CIRP Conference on Assembly Technologies and Systems (CATS 2018), pp. 155–160, 2018 (vaadatud 19.05.2021).

- [26] D. Puljiz, E. Stöhr, K. S. Riesterer, B. Hein, and T. Kröger, “Sensorless hand guidance using microsoft hololens,” in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 632–633, 2019 (vaadatud 19.05.2021).
- [27] “Microsoft hololens vs hololens 2 comparison,” veebruar 2019 (vaadatud 19.04.2021). <https://program-ace.com/blog/microsoft-hololens-vs-hololens-2-comparison/>.
- [28] “Unity koduleht,” 2021 (vaadatud 21.04.2021). <https://unity.com/>.
- [29] J. Brodtkin, “How unity3d became a game-development beast,” juuni 2013 (vaadatud 21.04.2021). <https://insights.dice.com/2013/06/03/how-unity-3d-become-a-game-development-beast/>.
- [30] “Ros/introduction - ros wiki,” 2018 (vaadatud 21.04.2021). <http://wiki.ros.org/ROS/Introduction>.
- [31] “Ros community metrics - ros wiki,” 2020 (vaadatud 21.04.2021). <http://wiki.ros.org/Metrics>.
- [32] “Ros wiki,” 2020 (vaadatud 21.04.2021). <http://wiki.ros.org/>.
- [33] “Understanding ros nodes,” 2019 (vaadatud 20.05.2021). <http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes>.
- [34] R. Raudmäe, “Avatud robotplatvorm robotont,” 2019 (vaadatud 19.05.2021). [http://dspace.ut.ee/bitstream/handle/10062/64341/Raudmae\\_MSc2019.pdf?sequence=1&isAllowed=y](http://dspace.ut.ee/bitstream/handle/10062/64341/Raudmae_MSc2019.pdf?sequence=1&isAllowed=y).
- [35] “Robotont õppematerjalid,” (vaadatud 19.05.2021). <https://sisu.ut.ee/robotont/%C3%B5ppematerjalid>.
- [36] Microsoft, “.net raamistiku koduleht,” 2021 (vaadatud 19.05.2021). <https://dotnet.microsoft.com/>.
- [37] Siemens, “Ros# lähtekood githubis,” aprill 2021 (vaadatud 28.04.2021). <https://github.com/siemens/ros-sharp>.
- [38] Microsoft, “Universal windows platform documentation,” 2021 (vaadatud 19.05.2021). <https://docs.microsoft.com/en-us/windows/uwp/>.
- [39] EricVoll, “Fork of the rossharp github with uwp support,” veebruar 2021 (vaadatud 21.04.2021). <https://github.com/EricVoll/ros-sharp>.
- [40] “Json,” (vaadatud 19.05.2021). <https://www.json.org/json-en.html>.
- [41] J. Mace, “Rosbridge ülevaade ros wikis,” oktoober 2017 (vaadatud 28.04.2021). [http://wiki.ros.org/rosbridge\\_suite](http://wiki.ros.org/rosbridge_suite).
- [42] “Occupancygrid message documentation,” jaanuar 2021 (vaadatud 21.04.2021). [http://docs.ros.org/en/api/nav\\_msgs/html/msg/OccupancyGrid.html](http://docs.ros.org/en/api/nav_msgs/html/msg/OccupancyGrid.html).
- [43] “Unified robot description format kimp ros wikis,” 2019 (vaadatud 19.05.2021). <http://wiki.ros.org/urdf>.

- [44] polar-kev et al., “What is the mixed reality toolkit,” märts 2021 (vaadatud 21.04.2021). <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/>.
- [45] “move\_base\_msgs message documentation,” detsember 2013 (vaadatud 21.04.2021). [http://docs.ros.org/en/fuerte/api/move\\_base\\_msgs/html/index-msg.html](http://docs.ros.org/en/fuerte/api/move_base_msgs/html/index-msg.html).
- [46] “Rosbridgeclient tööriist rossharp githubis,” veebruar 2021 (vaadatud 21.04.2021). <https://github.com/siemens/ros-sharp/tree/master/Libraries/RosBridgeClient>.

# **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Tarvi Tepandi

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

**“Segareaalsusel põhinev kasutajaliides mobiilse roboti kaugjuhtimiseks Microsoft HoloLens 2 vahendusel”**

mille juhendaja on Karl Kruusamäe

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'i 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.

*Tarvi Tepandi*  
**20.05.2021**