Tartu Ülikool

Loodus- ja täppisteaduste valdkond

Tehnoloogiainstituut

Taaniel Küla

ROS2 platvormile keevitusroboti tarkvara portotüübi loomine kasutades UR5 robot-manipulaatorit

Bakalaureusetöö (12 EAP)

Arvutitehnika eriala

Juhendaja:

Robotitehnoloogia lektor Veiko Vunder

Tartu 2022

Resümee/Abstract

ROS2 platvormile keevitusroboti tarkvara prototüübi loomine kasutades UR5 robot-manipulaatorit

Robotite kasutus erinevates tööstussektorites kasvab iga aastaga, seda just nende hinna languse ja tehnoloogia kiire arengu pärast. Kuigi keevitusrobotite kasutuselevõtu trend on tõusuteel, pole ühel populaarseimal roboti tarkvara platvormil ROS2 keevitusroboti tarkvara valik lai. Selle bakalaureusetöö eesmärk on luua ROS2 platvormile kergesti kasutatav keevitusroboti tarkvara prototüüp, kasutades selleks erinevaid tööriistu ROS2 tarkvararaamistikult.

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

Märksõnad: ROS2, UR5, keevitusrobot, robootika

Welding robot software prototype for ROS2 using UR5 robot arm

The usage of robots in different sectors rises every year thanks to falling prices and technological advancements of these robots. Even though the usage of welding robots is on the rise as well, the software choice on ROS2 is lacking. The goal of this bachelor thesis is to create an easy-to-use welding software prototype for ROS2, using its many capabilities and tools for easy engineering.

CERCS: T120 System engineering, computer technology; T125 automation, robotics, control engineering

Keywords: ROS2, UR5, welding robot, robotics

Sisukord

Resümee/Abstract **2**

Jooniste loetelu **5**

Lühendid, konstandid, mõisted **6**

1 Sissejuhatus **7**

2 Eksisteerivad lahendused **9**

2.1 Tasulised keevitustarkvarad 9

2.2 Vabavaralised keevitustarkvarad 11

3 Bakalaureusetöö nõuded **15**

4 Tarkvaravahendite ülevaade **16**

4.1 ROS2 16

4.2 MoveIt2 17

4.3 RVIZ2 17

4.4 PCL 18

4.5 Universal Robots ROS2 Driver 19

4.6 ROS2 Wrapper for Intel® RealSense™ Devices 19

5 Lahendus **20**

5.1 Kasutajaliidese ülevaade 20

5.1.1 Paneeli pistikprogramm 21

5.1.2 Punktivalija pistikprogramm 22

5.2 Peaprogramm 22

5.2.1 Skaneerimine 22

5.2.2 Objekti maailma lisamine 23

5.2.3 Keevitustee otsimine 23
5.3 Robotmanipulatsioon 25
5.3.1 Roboti liigutamine 25
5.3.2 Roboti peatamine 25
6 Lahenduse testimine 26
6.1 UR5 robot-manipulaator 26
6.2 Intel RealSense D435i 26
6.3 Robotile tööriistade kinnitamine 26
6.3.1 Keevituskäpa kinnitus 27
6.3.2 Kaamera 28
6.4 Tarkvara testimine 29
Kokkuvõte 30
Summary 31
Kirjandus 32
Lisad 34
Lisa 1. Lõputöö raames valminud tarkvara prototüübi koodibaas 34
Lihtlitsents 35

Jooniste loetelu

Joonis 1. Verbotics Weld kasutajaliides 9
Joonis 2. Keevituste tuvastamine 10
Joonis 3. Keevitamise seaded 10
Joonis 4. ROSWELDi keevitusprotsessi jälgimise paneel [6] 11
Joonis 5. SmartWeldi WELD-3D ennustusmudel 12
Joonis 6. Tarkvara töövoog [8] 13
Joonis 7. Scilabis loodud kasutajaliides [9] 14
Joonis 8. RVIZ2 - tuvastatud objektide kuvamine [14] 18
Joonis 9. Pakutava lahenduse arhitektuur 20
Joonis 10. Operaatori sammud koos etapisiseste nuppudega 21
Joonis 11. Operaatori paneel 22
Joonis 12. Tee leidmise protsess 24
Joonis 13. Robot koos kinnituste ja tööriistadega 27
Joonis 14. Keevitusrobot integreeritud lahendusega [26] 28
Joonis 15. Adapteri disain 29

Lühendid, konstandid, mõisted

ROS – *Robot Operating System*, robotitele tarkvara kirjutamist hõlbustav raamistik

punktipilv – ruumiliselt laiali olev punktide kogum, mille igal punktil on X, Y ja Z koordinaat

PCL – *Point Cloud Library*, erinevaid 2D/3D piltide ja punktipilvede töötlusalgoritme sisaldav teek

IMU – *inertial measurement unit*, liikmist tuvastav seade

1 Sissejuhatus

Tänapäeva maailmas on robotite kasutustrend tõusuteel. Roboteid võetakse aina rohkem kasutusse täpsust vajavate ja korduvate ülesannete jaoks [1]. Seda on soodustanud fakt, et tööstusrobotite hinnad on langenud suuremates arenenud riikides rohkem kui 50% viimase kolmekümne aasta jooksul [2].

Roboteid kasutatakse kõige rohkem tööstussektorites. Suurimad kasutajad on transpordifirmad ja elektroonikatootjad, omades kahepeale kokku kuni 75% kõikidest maailma robotitest. Väiksem protsent robotitest on kummi-, plastik- ja metallitöö sektorites. Nendele kuulub umbes 12% maailma robotitest. [2]

Keevitusrobotite ülesseadmine ja kasutamine on töörohke ettevõtmine. Samuti tähendab töörohke ettevõtmine industriaalmaailmas ka suuri väljaminekuid. Maksimaalse kasumi jaoks on vaja robotit kasutada efektiivselt. See tähendab tarka masinat, mis vajab väikest ülesseadmise aega ja kasutab tarka algoritmi enda käitamiseks. Tänapäeval kasutatakse veel suuresti iga liigutuse manuaalset planeerimist. Üks manuaalsetest planeerimise viisidest on salvestada roboti asendid läbi robotile lisatud juhtpaneeli. Asendite vahel liikumise planeerib roboti juhtarvuti. Nii tekib juhtarvutisse programm, mida robot saab hiljem korrata. See planeerimise viis nõuab operaatorilt keerukamate tööde puhul kõrget pädevust ja erinevate robotite puhul vastavaid koolitusi. Samuti tuleb iga uue töövoo jaoks luua uus programm [3]. Nendel põhjustel on kogu see protsess väga aeganõudev. Manuaalne planeerimine ei ole ka enamasti optimaalne ei energiatarbe ega mootorite läbitud teepikkuse osas. Nimetatud põhjuste pärast proovitakse seda protsessi aina rohkem automeerida [1]. Kuigi on mitmeid tasulisi tarkvarasid, mis seda võimekust juba pakuvad, on vabavaraliste tarkvarade valik väike.

Selle bakalaureusetöö eesmärk on luua avatud tarkvararaamistikul põhinev automaatse keevitustarkvara prototüüp, kuna täieliku keevitustarkvara loomine on liiga mahukas protsess. Töö raames tuleb

1. kaardistada olemasolevad vabavaralised tarkvarakimbud
2. koostada demonstratsioon tuntud tööstusrobotiga
3. disainida ja valmistada keevituskäpa adapter.

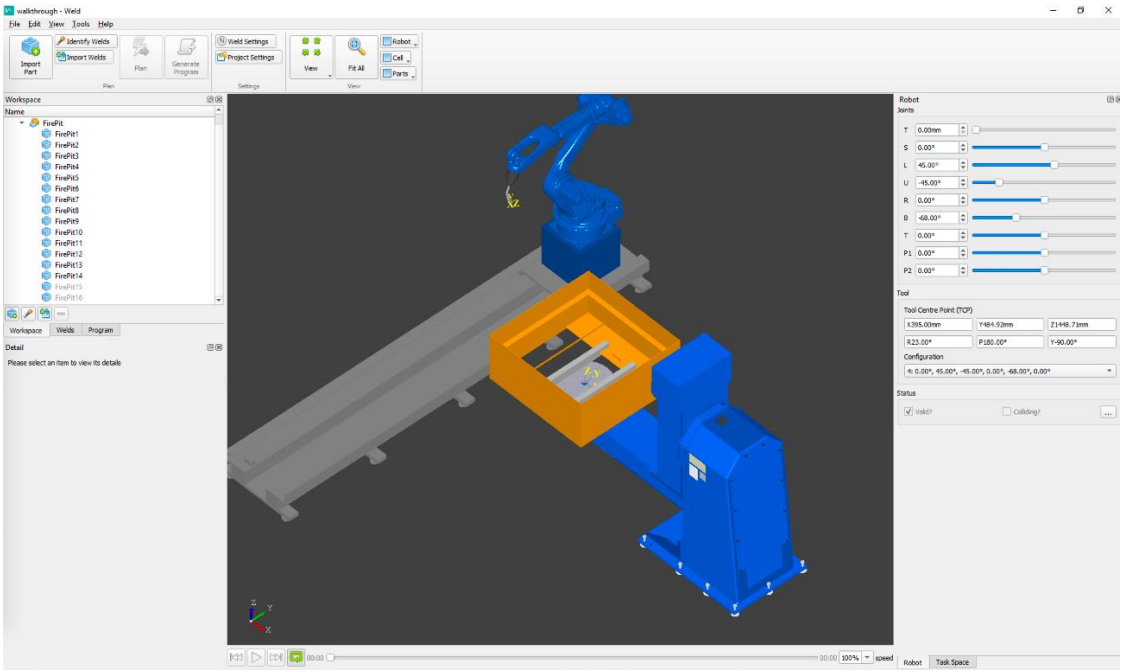
Lõputöös tutvustatakse olemas olevaid lahendusi, kirjeldatakse täpsemalt lõpptulemuse jaoks püstitatud nõudeid, antakse ülevaade kasutatud tööriistadest, tutvustatakse ja testitakse valmis tehtud lahendust.

2 Eksisteerivad lahendused

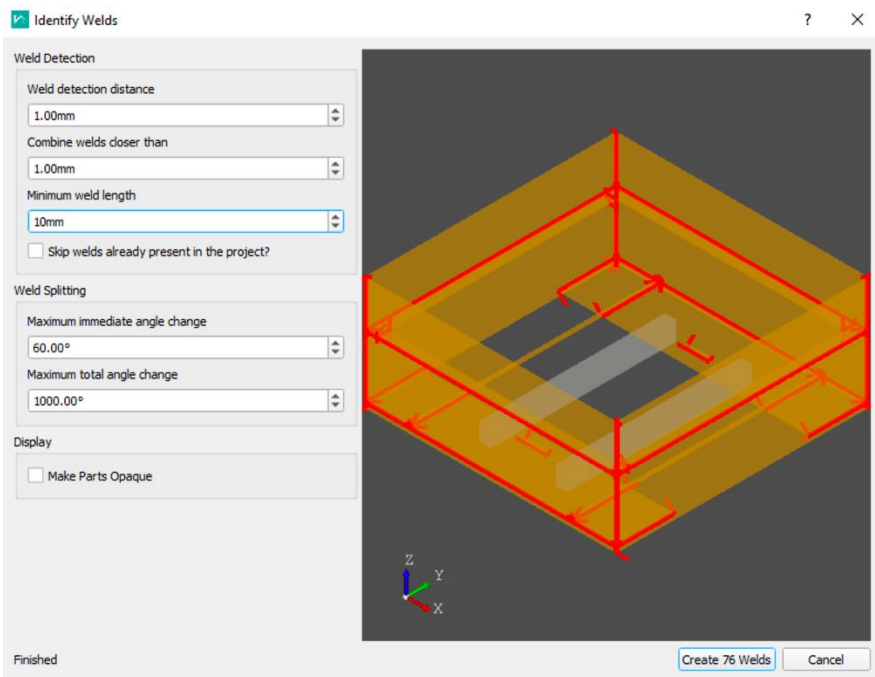
Praeguseks on maailmas juba mitmeid eksisteerivaid keevitustarkvarasid. Käesolev peatükk annab ülevaate erinevatest olemasolevatest keevitusroboti tasulistest ja vabavaralistest lahendustest.

2.1 Tasulised keevitustarkvarad

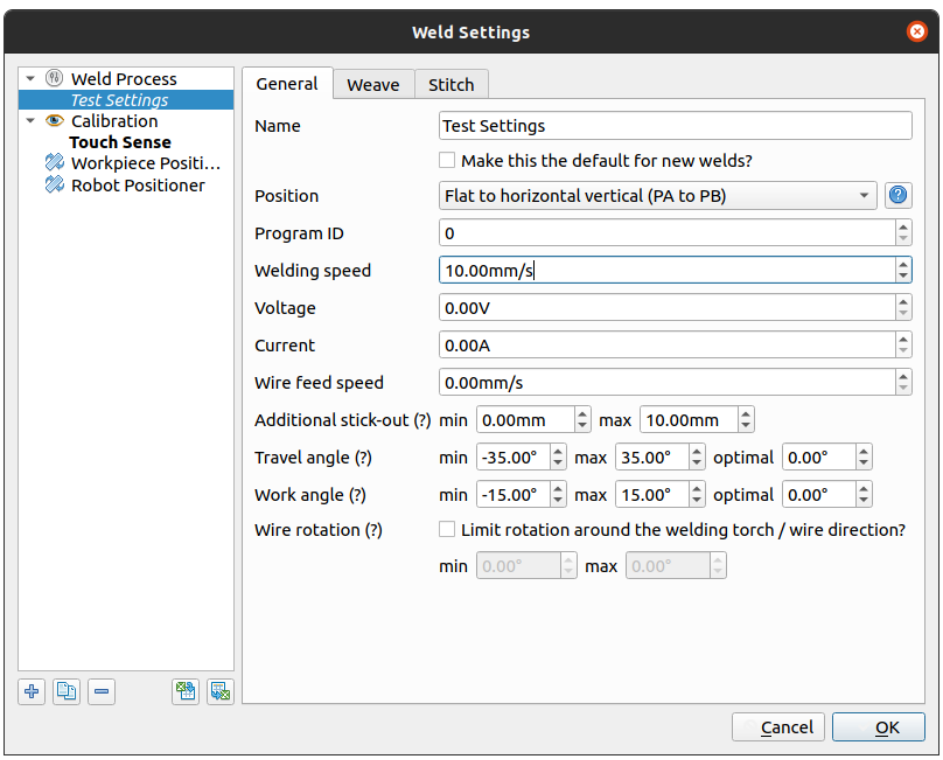
Verbotics Weld pakub graafilist tarkvara, mis on loodud keevituste planeerimise automatiseerimiseks (joonis 1) [4]. Töö alustamiseks tuleb kõigepealt programmi sisse tuua 3D mudel detailist, mida soovitakse keevitada. Seejärel tuvastab tarkvara ise automaatselt vajalikud keevituskohad. Keevituste tuvastamisel saab mitmeid muutujaid vajadusel kohandada. Näiteks saab kohandada keevituse miinimumpikkust ja keevituste kombineerimist sõltuvalt nende kaugusest (joonis 2) [4]. Kui keevituskohad on paigas, planeerib tarkvara parima teekonna robotile keevituste sooritamiseks. Verbotics Weld pakub ka roboti automaatse kalibreerimise võimalust. Töö alustamiseks genereerib tarkvara robotkoodi, mille saab edastada keevitusroboti kontrollerile kinnitamiseks ja täide viimiseks. Robotil saab täpsemalt ka seadistada keevitamise seadeid (joonis 3) [4]. Tarkvaral on ka keevituse simuleerimise võimekus. See toetab järgmisi robotite programmeerimiskeeli: ABB RAPID, FANUC TP, Yaskawa Motoman INFORM. [5]



Joonis 1. Verbotics Weld kasutajaliides



Joonis 2. Keevituste tuvastamine

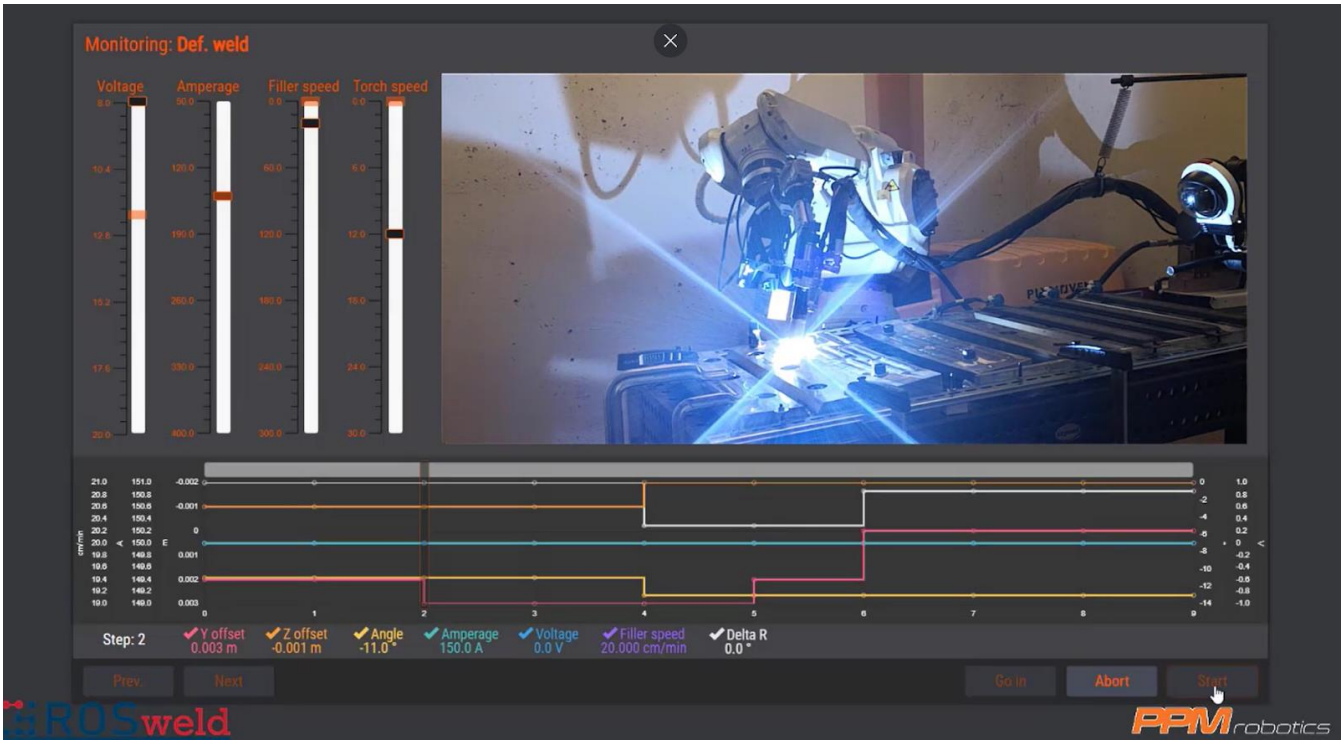


Joonis 3. Keevitamise seaded

Almacam Weld on automaatne keevitustarkvara, millel on samad funktsioonid nagu Verbotics Weldil. See tuvastab automaatselt potentsiaalsed keevitused, planeerib parima tee keevituste sooritamiseks ja võimaldab konfigureerida keevitusroboti seadeid. Keevitatavad detailid ja roboti tööriistad on peab sisse tooma kolmanda osapoole 3D projekteerimise tarkvarast. Toetatud robotite tootjad: ABB, Cloos'i, Fanuc'i, IGM'i, Kawasaki, Kuka, Panasonic'i, Reis'i, Staubl'i, Hyndai, OTC Daihen'i, Yaskawa/Motoman. [6]

2.2 Vabavaralised keevitustarkvarad

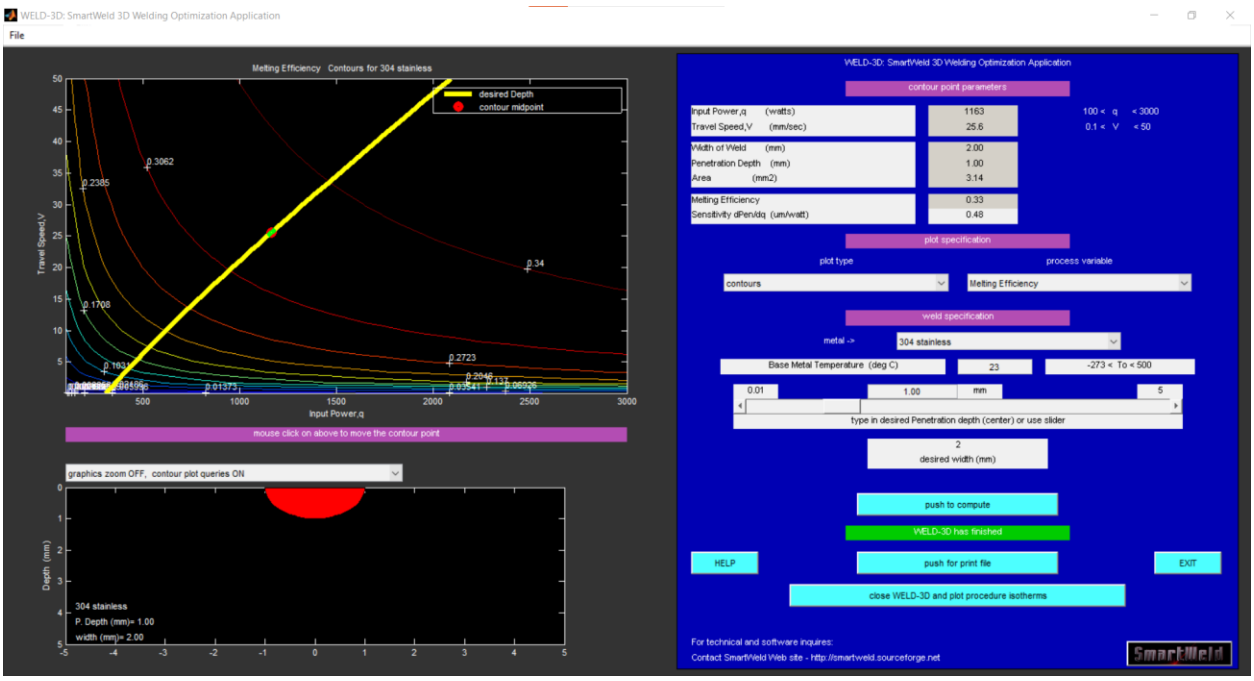
ROSWELD on automaatne keevitus tarkvara, mis põhineb maailma ühel populaarseimal robotiraamistikul ROS. See võimaldab joonistada 3D mudelitele keevitusteid ja planeerida automaatselt parima võimaliku viisi keevitusteede läbi viimiseks. Samuti on võimalik muuta keevitusseadeid nagu keevituskiirus, pinge ja amperaaž (joonis 4) [7]. ROSWELD toetab NACHI ja Hyundai keevitusroboteid. [7]



Joonis 4. ROSWELDi keevitusprotsessi jälgimise paneel

SmartWeld on vabavaraline tarkvara, mis aitab inseneril valida paremini keevitusseadeid ja materjale. Seda ei kasutata keevitustee tuvastamiseks ega planeerimiseks, vaid keevitusseadete

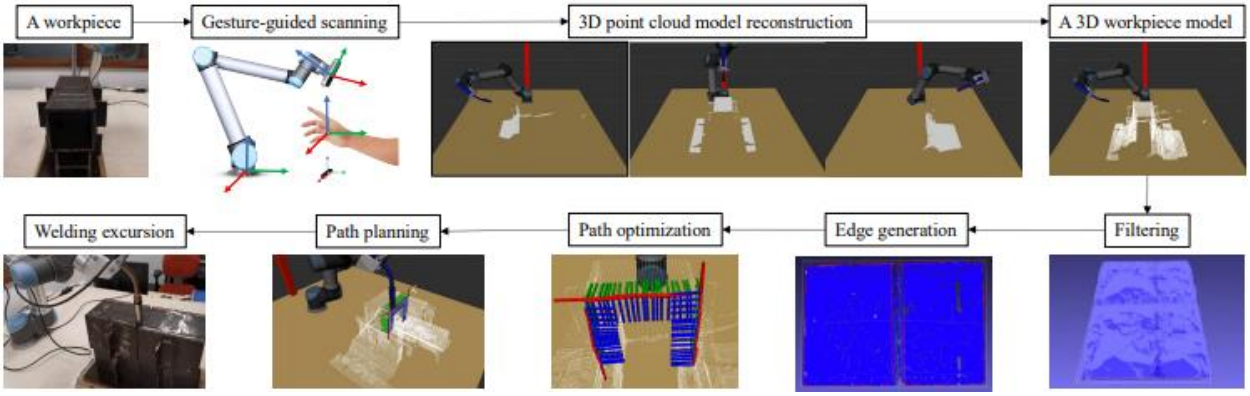
optimeerimiseks ja paremaks konfigureerimiseks. See võimaldab kasutajal luua ennustus- ja uurimismudeleid kasutades muutujaid nagu temperatuur, keevituse suurus, materjal ja keevitaja seaded (joonis 5) [8]. SmartWeldi arendamisega alustati aastal 1993 ja nüüdseks on see projekt kasvanud üle 15000 koodireani. [8]



Joonis 5. SmartWeldi WELD-3D ennustusmudel

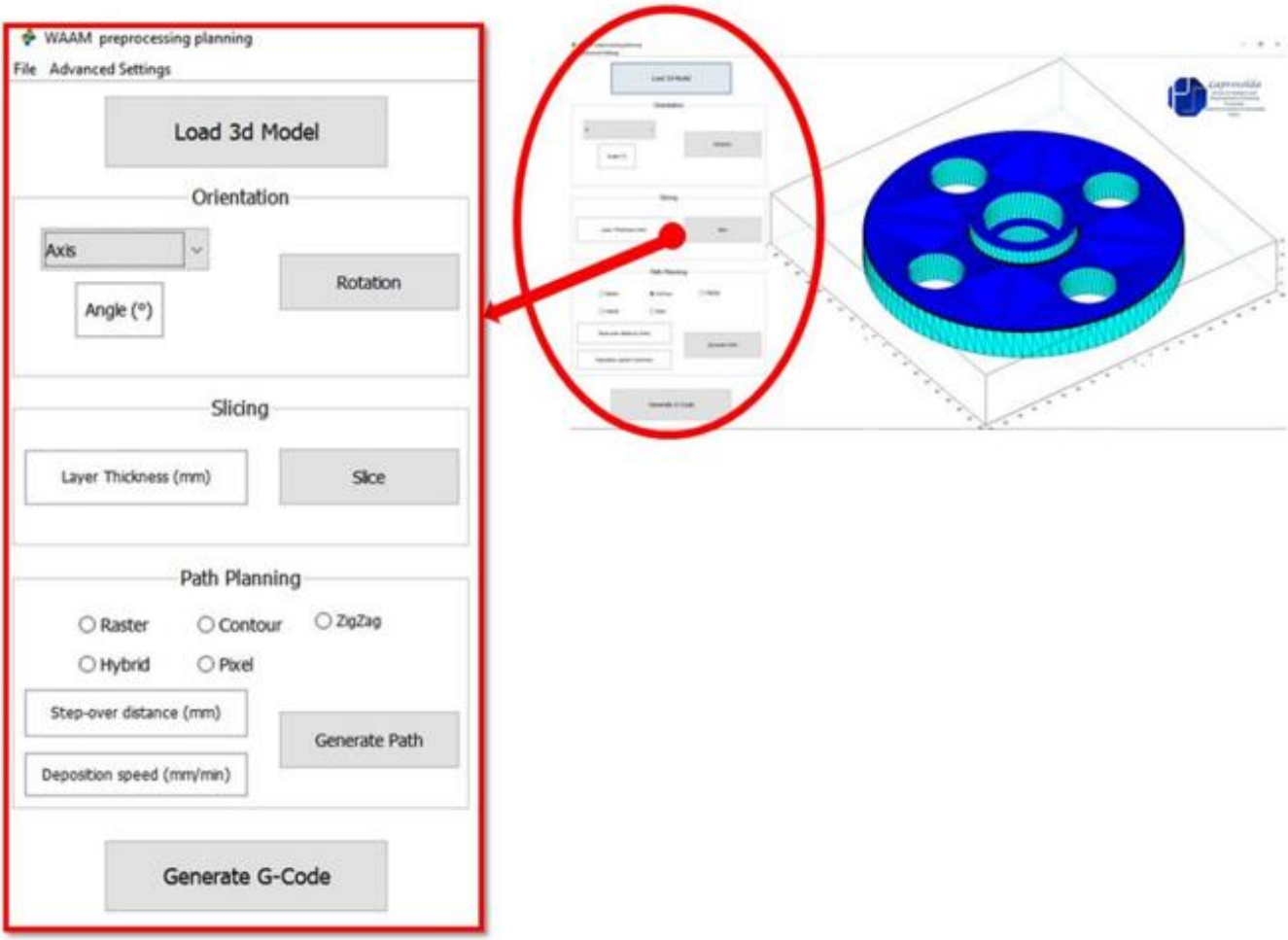
Tasuta keevitustarkvarade saadavus ei ole suur, kuid on tehtud palju uuringuid ja teste, mis pakuvad portotüüplahendusi automaatse keevitustarkara loomiseks. Üks prototüüptarkvara loodi Hong Kong Polytech University's, kus tehti automaatse liitekoha tuvastamisega tarkvara, mis kasutab käeviipeid detaili skaneerimise protsessi juhtimiseks [9]. Esimese sammuna skaneerib operaator detaili sisse kasutades roboti juhtimiseks käeviiped. Roboti liikumist kontrollitakse mõlema käega – parema käega kontrollitakse roboti mõjuri positsiooni ja pildistamist, vasaku käega kontrollitakse ülejäänud robotit. Kui kõik vajalikud pildid detailist on tehtud, kombineeritakse need kasutades iteratiivset lähima punkti (ingl *iterative closest point)* algoritmi, mis ühendab kõik punktipilved kokku üheks. Seejärel filtreeritakse punktipilvest müra. Peale müra filtreerimist tuvastatakse potentsiaalsed liitekohad kasutades järsu punna muutuse tuvastamise algoritmi. Alles jäänud punktidest genereeritakse robotile läbitav keevitustee. Tarkvara testimisel

leiti, et see meetod on efektiivne ja robustne viis robotitele automaatse keevitustee planeerimiseks. [9] Kogu töövoog on illustreeritud joonisel 6 [9].



Joonis 6. Tarkvara töövoog

Keevitustarkvara prototüüpe ei tehta ainult klassikalise keevitamise jaoks. 3D printimise tõusuga on esile kerkinud ka uus viis keevitusrobotite kasutuseks – metalli 3D printimine (ingl *wire arc additive manufacturing).* Selle meetodi jaoks kasutatakse keevitusroboteid, mis loovad keevitusi kihiliselt üksteise peale. Üks pakutud lahendustest implementeeriti Scilabis, mis on vabavaraline andmetöötlus programm (joonis 7) [10]. Esimese sammuna laeb operaator sisse detaili 3D mudeli, mis orienteeritakse ümber parimaks prindi tulemuseks. Järgmisena genereeritakse detailist kõik vajalikud keevituskihid, mis ükshaaval üksteise peale tehakse. Genereeritud kihtide kõrgust on võimalik operaatoril reguleerida. Peale seda luuakse robotile kõikidest kihtidest robotile läbimiseks tee. Tee genereerimiseks on võimalik kasutada mitmeid erinevaid keevitusmeetodeid simuleerivaid algoritme. Enne tee genereerimist tuleb valida kahe kõrvuti oleva keevituspunkti keskpaikade omavahelised kaugused. Sellega on võimalik reguleerida keevituste kattuvust. Peale keevitustee genereerimist, on programmist võimalik väljastada keevitusrobotile sobiv G-kood. [10]



Joonis 7. Scilabis loodud kasutajaliides

3 Bakalaureusetöö nõuded

Bakalaureusetöö eesmärgiks on valmis teha tarkvaralahenduse prototüüp, mida kasutades saab operaator kergesti sooritada keevitust. Selle töö üks oluline aspekt on universaalsus – tarkvara peab olema kasutatav erinevate riistvaraliste osadega. Lahendusele esitati järgmised nõuded:

1. Tarkvara peab olema võimeline looma 3D mudeli keevitatavatest objektidest.
2. Kasutajaliides peab võimaldama kasutajal intuitiivselt määrata soovitud keevitustee.
3. Tarkvara peab põhinema vabavaralisel ROS2 tarkvararaamistikul, mis tagaks, et loodud lahendus oleks riistvaraülene, tulevikku orienteeritud ja võimaldaks tarkvara vähese vaevaga integreerida olemasolevate robotsüsteemidega.
4. Tarkvaral peab olema kasutajaliides, mis on lihtsasti kasutatav tehnoloogia valdkonnas kogenule operaatorile, kes pole varem keevitustarkvaradega kokku puutunud.
5. Tarkvara funktsionaalsus peab olema demonstreeritud füüsilisel robotil, mis on toetatud ROS2 platvormil.
6. Tarkvara peab vältima detailiga kokkupõrkeid ja ohu korral võimaldama operaatoril robot-manipulaatorit iga hetk peatada.

4 Tarkvaravahendite ülevaade

Selles peatükis kirjeldatakse tarkvarabaasi ja tähtsamaid tarkvarakimpe, mida lõpplahenduse arendamisel kasutati.

4.1 ROS2

ROS2 (*Robot Operating System 2*) on vabavaraline operatsiooni-süsteemiga sarnast ülesannet täitev raamistik, mis on loodud robotitele tarkvara kirjutamiseks. ROS2 on mitmeplatvormiline raamistik, mis tähendab, et sellel kirjutatud tarkvara töötab igal ROS2 poolt toetatud platvormil. Kõige suuremaks ROS2 kasutamise eeliseks, on selle kommuuni poolt loodud, vabalt saadaval olevate tarkvarakimpude hulk. Suur kimbubaas teeb oma lahenduste valmis arendamise kiireks, kuna on võimalik võtta sisse eksisteerivad tarkvarakimpe ja kombineerida need kokku vastavalt enda vajadustele valmislahenduseks. [11]

ROS2 on edasiarendus ROS1st, mis arendati välja mõttega teha sellest täielikum tööriist. ROS1 on ka tarkvararaamistik robotite tarkvara kirjutamiseks. ROS2 arendati üles nullist koos uue arhitektuuri ülesehitusega. Suurte fundamentaalsete muutuste tõttu ei ole originaalsel ROSil töötava tarkvara üle toomine ROS2le optimaalne, kuna selle jaoks on vaja kasutada muutusi siluvaid vahelülisid, ja tihti modifitseerida olemasoleva peaprogrammi struktuuri.

Arhitektuurse ümberehitusega muudeti ROSi funktsionaalne sõltuvus programmeerimise keele kimbust (rospy või roscpp) ümber kõigile ühisele funktsionaalsuse kihile. See tähendab, et iga funktsiooni tööloogikat ei pea implementeerima mõlema keele jaoks eraldi. Funktsioon implementeeritakse ühises tööloogika kihis ja sellele tuleb mõlemas keeles ainult luua siduv viide. See avab võimaluse kergelt teha ROS2 jaoks kasutatavaks ka teisi programmeerimise keeli peale *Pythoni* ja C++'i. Läbi selle meetodi on ka kõigil ühine API, mis toetab selle kiiremat arengut ja stabiilsemat tarkvarabaasi kvaliteeti. Samuti muutuvad uued funktsionaalsused kõigile korraga kasutatavaks. Tänu arhitektuuri ümberehitusele, on võimalik ROS2 käitada erinevate platvormide peal. Kui ROS1 oli põhiliselt toetatud Linux-is, siis ROS2-s on toetatud kõik enamlevinud operatsioonisüsteemid: Ubuntu Linux, Windows, MacOS. [12]

Teiseks suureks muutuseks on tsentraalse infovahetusüsteemi puudumine. Originaalselt sõltus ROSis sõlmede (ROSi sees jooksev programm) vaheline suhtlus kesksest infoserverist, millega

iga sõlm oli otseses ühenduses. See tähendab, et enne sõlme tegemist, pidi ROS1-s looma infoserveri. ROS2-s on iga sõlm iseseisev ja ühenduses ainult nende sõlmedega, millega sel ühendust vaja on. See meetod võimaldab ilma eelse ülesseadmiseta luua sõlmi ja samuti korraldada nende vahel otsest infovahetust. [12]

Selle töö raames on kasutuses ROS2 Rolling Ridley, mida käitatakse Ubuntu 20.04 peal.

4.2 MoveIt2

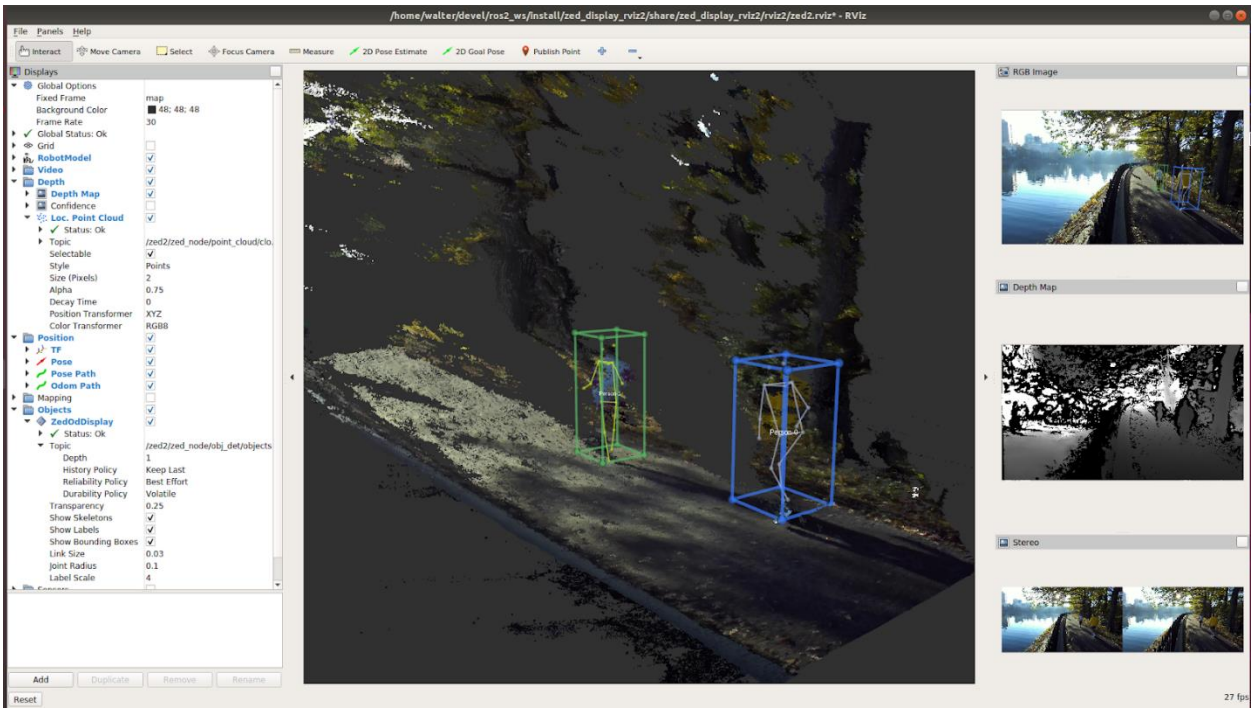
MoveIt on vabavaraline robotmanipulatsiooni platvorm ROSile. See hõlmab endas uusimaid teeplaneerimis, navigeerimis ja kinemaatika algoritme. MoveIt2 on edasiarendus MoveItist, mis on loodud ROS2 jaoks [13]. Sarnaselt teiste ROS programmidele, kasutatakse ka MoveIt-i juhtimiseks mitmeid erinevaid ROS sõnumeid, teenusekutseid ja parameetreid. Selleks, et MoveIt-i kasutamine oleks võimalikult lihtne, on nii MoveIt kui ka MoveIt2-le loodud eraldi programmeerimisliides *MoveGroup Interface*, mis on saadaval C++ ja Python programmeerimiskeeltes. Näiteks saab kirjeldatud liidese kaudu robotit liigutada kõikide telgede soovitud asetuse, mõjur positsiooni või mitmeid mõjuri positsioone sisaldava nimekirja põhjal. Lisaks saab lisada robotiga samasse virtuaalmaailma objekte. See on kasulik kui päris maailmas on roboti liikumisalal potentsiaalseid kokkupõrkekohti. Nii oskab planeerimise algoritm valida tee, millega neid välditakse. MoveIt2-ga on võimalik vältida objekte ka reaalajas kasutades hübriidplaneerijat. See tähendab, et planeerija võtab sisse muutuva maailma kohta infot ja kohandab selle järgi juba läbitavat trajektoori. Hübriidplaneerija on kombinatsioon lokaalsest ja globaalsest planeerijast [14]. See funktsionaalsus puudus originaalses MoveIt-is.

Selle töö raames on kasutuses MoveIt2, mis on kohandatud ROS2 Rolling Ridley jaoks.

4.3 RVIZ2

RVIZ2 on graafiline liides ROS2 platvormile, millega saab visualiseerida roboteid ja muud vajaliku informatsiooni (näiteks roboti sügavuskaamerast tulev punktipilv) [15]. RVIZ2 kasutamise üks suuri eeliseid on selle universaalsus, mis luuakse pistikprogrammidega. Kui kasutaja vajab võimekust, mida RVIZ2i arendajad pole sellele lisanud, on see võimalik ise luua. Pistikprogramme on võimalik luua kolme eri tüüpi – tööriist, paneel ja kuva. Tööriista tüüpi pistikprogrammidega on võimalik suhelda maailmaga, mis on kuvatud andmete visualiseerimise aknas. Paneeli pistikprogrammidega on võimalik luua kasutajaliideseid taustal töötavate

protsesside kontrollimiseks ja kasutajaliideses info kuvamiseks. Kuva pistikprogrammidega on võimalik visualiseerimise aknas luua kohandatud disaine informatsiooni edasi andmiseks [16]. RVIZ2 graafilise liidese näidet on näha joonisel 8 [15].



Joonis 8. RVIZ2 - tuvastatud objektide kuvamine

4.4 PCL

PCL ehk Point Cloud Library on vabavaraline 2D/3D piltide ja punktipilvede töötlemise teek. See sisaldab filtreid, erinevaid andmepuusid, visualiseerimisvahendeid, registratsiooni ja muid andmete töötlemise võimekusi [17].

Enamus PCLi funktsionaalsusest on kasutatav ROSis ilma vahelülideta, kuid mõnel juhul on vaja informatsioon viia teistsugusele formaadile. Näiteks on vaja sõlmede vahel info vahetamise jaoks viia PCL muutujatüüp ümber ROS sõnumi kujule. Selle jaoks on ROS2 platvormile loodud PCLi mugavaks kasutamiseks teek, mis viib PCLi andmetüübid üle ROSi jaoks kasutatavale andmetüüpidele [18].

4.5 Universal Robots ROS2 Driver

UR5 robot-manipulaatoril tarkvara testimiseks kasutatakse Universal Robots ROS2 Driverit, mis on ROS2 ja Universal Robotsi robotite vahelise suhtluse korraldamiseks loodud tarkvarakimp. See on vabavaraline draiver, mille arendamisele saab kaasa aidata ka ROS kommuun kuid haldajaks on Universal Robots ise. See sisaldab endas mitmeid erinevaid kontrollereid, mudeleid ja konfiguratsioone, mis on valmis kasutamiseks. Samuti on draiveris konfiguratsiooni- ja näitefailid UR manipulaatorite kasutamiseks MoveIt2-ga. [19]

Kuna see draiver töötab läbi ROSi on võimalik seda vajadusel kiirelt teise manipulaatori draiveri vastu välja vahetada ilma ülejäänud tarkvara ümber konfigureerimata.

Selle töö raames on kasutuses versioon, mis on kohandatud ROS2 Rolling Ridley jaoks [19].

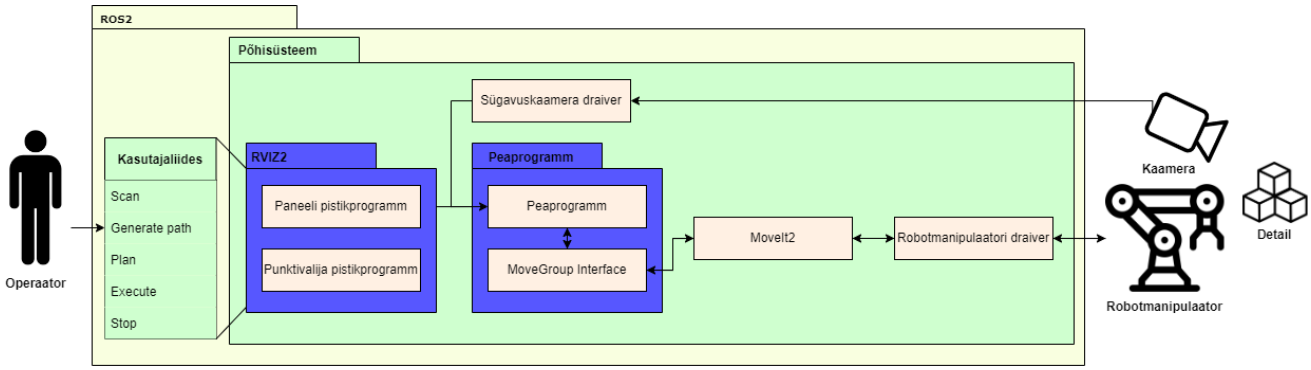
4.6 ROS2 Wrapper for Intel® RealSense™ Devices

Töö raames kasutatava sügavuskaamera ROSi sisese kasutamise jaoks, on vaja tootja poolt tehtud draiver koos vastava tarkvaraarenduskomplektiga. See viib kõik kaameratest sisse loetud info ROSi standarditele vastavale kujule ja võimaldab kaamerat ühendada teiste ROSi tarkvarakimpudega. Samuti toetab kaamera draiver seadete ja parameetrite muutmist läbi *Dynamic Reconfigure* graafilise liidese. See võimaldab reaalajas muuta näiteks säriaega või lülitada sügavuspildi leidmiseks kasutatavat laserit sisse/välja. [20]

5 Lahendus

Sellest peatükis räägitakse täpsemalt bakalaureusetöö raames valminud keevituslahenduse loomisest ning selle kasutamisest. Lahendus koosneb eksisteerivate tarkvarateekide ja ROS2 sõlmede omavahelises integreerimises ja puuduoleva funktsionaalsuse lisamine uute ROS tarkvarakimpude näol. Pakutava keevituslahenduse üldistatud arhitektuuri on kujutatud joonisel 9.

Lahenduse kõige pealmine osa on kasutajaliides, millesse lõin operaatoripaneeli ja keevitus alg- ja lõpp-punkti valimise võimekuse. Lahenduse madalamal tasemel on peaprogramm, mis koosneb kõikidest implementeeritud andmetöötlusfunktsioonidest. Peaprogramm saab käsklusi operaatorilt läbi operaatoripaneeli. Samuti on peaprogrammis kasutusel eksisteerivad vahelülid, mis korraldavad suhtlust sügavuskaamera ja robotmanipulaatoriga.

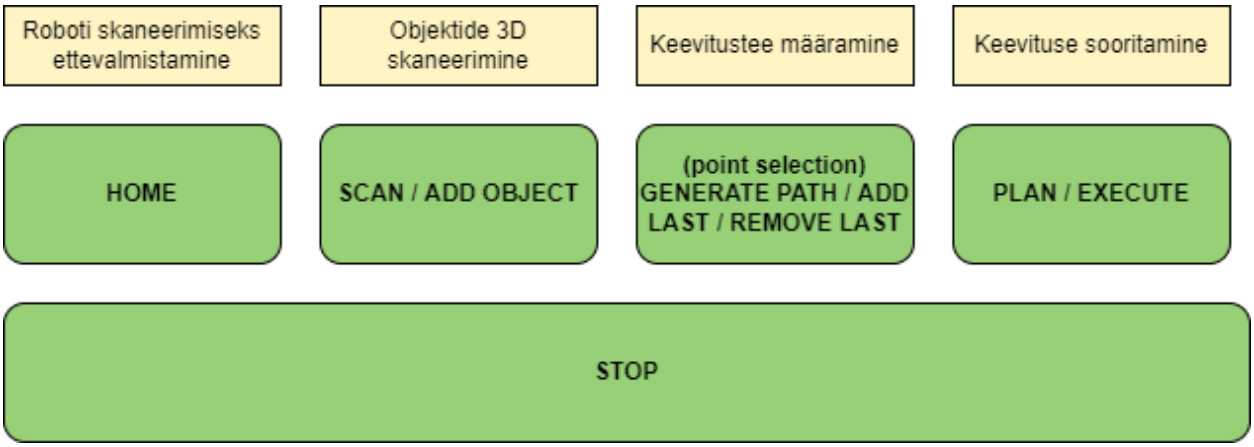


Joonis 9. Pakutava lahenduse arhitektuur

5.1 Kasutajaliidese ülevaade

Keevitustarkvara kasutajaliides põhineb RVIZ2 visualiseerimisvahenditel. Programmi käivitamisel tuleb esile RVIZ2 aken, kus on näha robotit ja selle kontrollimiseks vajalik graafiline kasutajaliides.

Lähtuvalt töö nõuetest jaotati kasutajaliidese kavandamisel operaatori tegevused joonisel 10 toodud etappideks.



Joonis 10. Operaatori sammud koos etapisiseste nuppudega

Roboti kontrollimiseks on eraldi paneel järgnevate nuppudega:

- HOME – asetab roboti skaneerimise algusasendisse
- SCAN – alustab keevitatava objekti skaneerimist
- ADD OBJECT – skaneeritud objekti roboti maailmaruumi
- GENERATE PATH – genereerib keevitusteekonna valitud punktide vahel
- ADD LAST – lisab viimase genereeritud keevituse keevitusnimekirja
- REMOVE LAST – eemaldab viimase genereeritud keevituse keevitusnimekirjast
- PLAN – simuleerib kasutajale nähtavalt planeeritud tee läbi
- EXECUTE – alustab planeeritud tee läbimist
- STOP – peatab programmi ja roboti

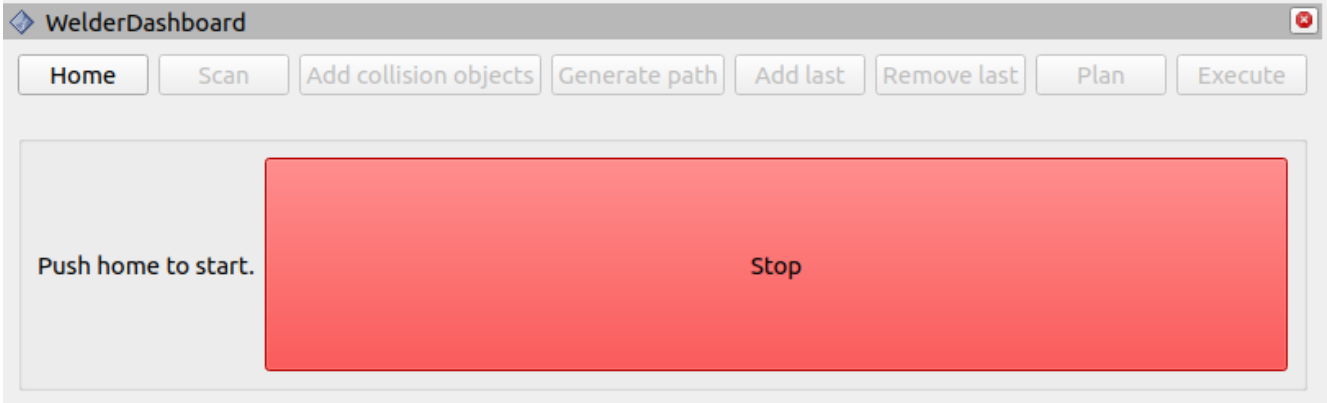
Kasutajaliideses on ka tööriist nimega “point selection“, mida kasutatakse kuvatud punktipilvest keevituse algus- ja lõpp-punkti valimiseks.

5.1.1 Paneeli pistikprogramm

Roboti kontrollimiseks on vaja kasutajale kergesti mõistvat ja kasutatavat viisi. Lihtsaim lahendus on luua graafiline liides, läbi mille saab kasutaja programmiga suhelda. Paneelide loomiseks on saab kasutada RVIZ2 pistikprogrammide *panel* tüüpi. Läbi selle on võimalik näiteks lisada nuppe, ligureid ja tekstikaste.

Selle töö raames loodud lahendus kasutab lihtsat disaini. Kogu paneel koosneb üheksast nupust. Iga nupp täidab programmist ühe etapi. Etapiline lähenemine annab kasutajale parema kontrolli

kogu protsessi üle. Samuti on nii kergem leida süsteemis vigu ja tänu sellele on programmi kasutamine ka turvalisem. Nupud on järjestatud töövoo järgi ehk nupud on etappide mõttes järjest. Loodud lahendust saab näha joonisel 11.



Joonis 11. Operaatori paneel

5.1.2 Punktivalija pistikprogramm

Keevituse algus- ja lõpp-punkti mugavaks valimiseks loodi eraldi pistikprogramm, mis võimaldab visualiseeritud punktipilve pealt valida punkte ja saata neid POINTCLOUD_SELECTION sõlmele edasi töötlemiseks. Punktivalija arendamisel võeti aluseks RVIZ2-s eksisteeriv pistikprogramm, mille otstarve on visualiseerimisaknas näidatud objekti peal asuva hiire koordinaatide väljastamine ROS sõnumina. Koordinaate saab väljastada ainult 3D objektide pealt. Valitud punkti täpse koordinaadi leidmiseks võetakse hiirevajutuse koordinaadid ja otsitakse punktipilvest punkt, millel on sellega kõige väiksem Eukleidiline kaugus. Valitud punktid saadetakse programmile PointStamped sõnumina läbi ROSi rubriigi nimega "/point_clicked". Programm võtab saadetud punktide seast alati esimese ja viimase punkti. Sellega välditakse probleeme, mis võivad tekkida kui kasutaja saadab kogemata rohkem kui 2 punkti korraga.

5.2 Peaprogramm

5.2.1 Skaneerimine

Keevitamise automeerimiseks on vaja keevitatav objekt sisse skaneerida. Seda on vaja teha, et kasutajal oleks mugav objekti pealt keevituskohti valida. Samuti saab nii robot ka info selle kohta, kus objekt asub keevitamiseks ja kuidas vältida tahtmatuid kokkupõrkeid.

Skaneerimine toimub poolringi kujulisel trajektooril roboti ees. Trajektoori läbides salvestab sügavuskaamera punktipilvesid, mis liidetakse juurde objekti täielikule punktipilvele. Enne liitmist konverteeritakse punktipilv ühtsesse fikseeritud koordinaadistikku. See on vajalik kuna pilt tehakse kaamera suhtes, mitte robotiga ühise maailma suhtes. Et hiljem pilve visualiseerides asetseks see õige koha peal, peab kõik punktid konverteerima robotiga ühise punkti suhtes. Peale punktipilve objekti pilvele juurde liitmist, visualiseeritakse täielik punktipilv RVIZ2 visualiseerimise aknas. Skaneerimise protsessi lõpus genereeritakse objektist tehtud pilvele ka pinna normaalid. Pinna normaalid on vektorid, mis on risti pinnaga antud kohas [21]. Normaalid genereeritakse PCL teegiga.

5.2.2 Objekti maailma lisamine

Kui skaneerimise protsess on lõpule jõudnud tehakse punktipilvest 3D võrk (ingl *mesh)*. Selle jaoks kasutatakse PCL kimbu kiire triangulatsiooni algoritmi. Triangulatsioon on viis pideva pinna ennustamiseks punktipilve baasil, ühendades kõik punktid oma naabritega [22]. Kui 3D võrk on valmis, lisatakse see läbi *MoveGroup Interface*'i planeerimise ruumi. Seal arvestatakse seda roboti liikumise planeerimisel, proovides sellega kokkupõrget vältida.

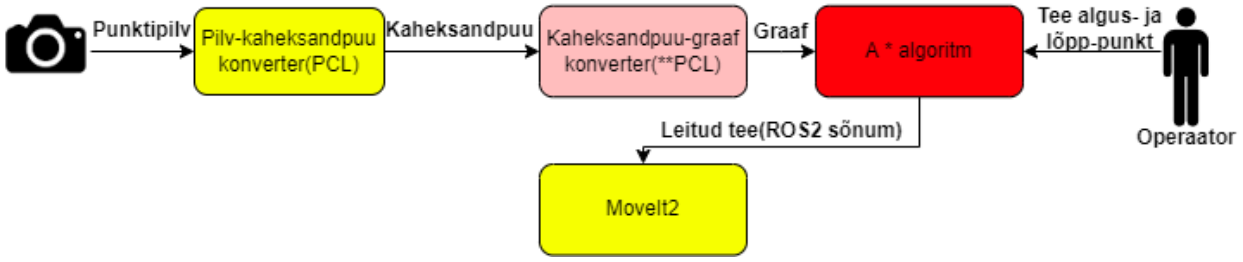
5.2.3 Keevitustee otsimine

Keevitustee genereerimise tähtsamad omadused selle projekti raames on kiirus ja täpsus. Seda just parimaks kasutajakogemuseks.

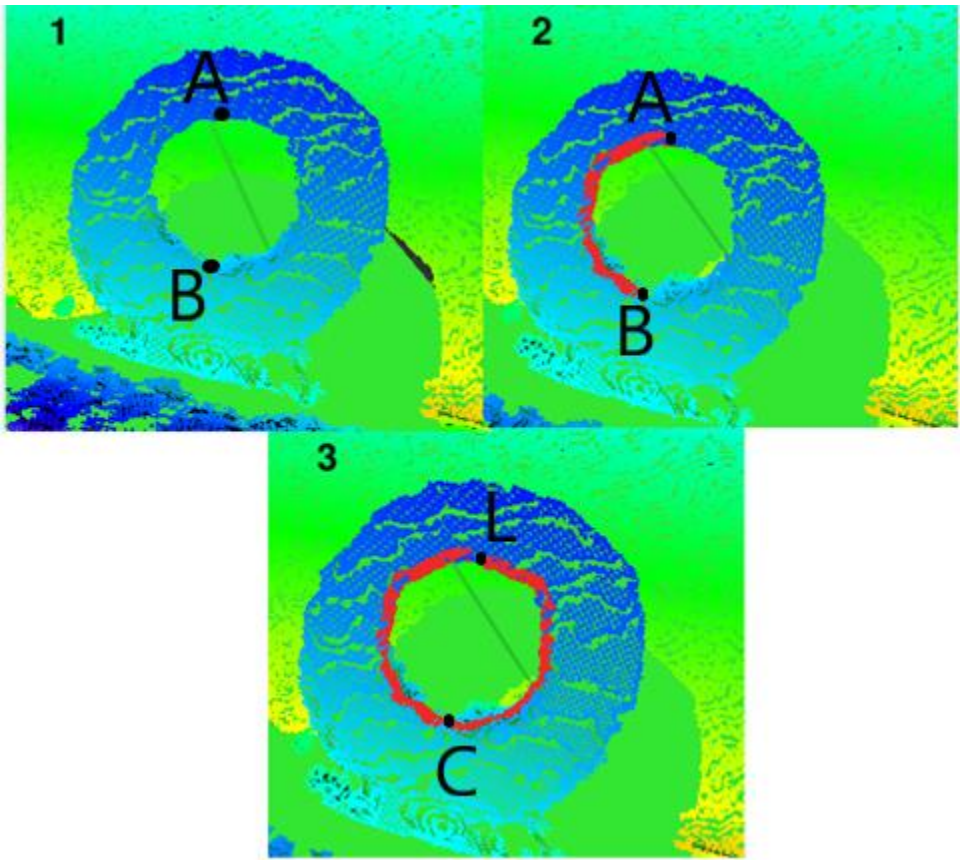
Keevitustee leidmiseks kasutatakse A* algoritmi. A * on tee otsimise algoritm, mis otsib kõikidest võimalikest teedest kõige väiksema kuluga tee. See on edasiarendus Djikstra tee otsimise algoritmist [23]. Seda algoritmi teostatakse graafil, mis on genereeritud sisse skaneeritud punktipilve kaheksandpuust. Kaheksandpuu on puu tüüpi andmestruktuur, mille igal tipul on täpselt 8 last. Kaheksandpuu kasutatakse enim 3D ruumi mitte kattuvate jaotuste genereerimiseks (ingl *spatial partitioning*). Kaheksandpuu struktuur võimaldab läbi viia kiireid naabrite otsinguid [24]. Tänu kiirele naabrite otsingule saab kaheksandpuust luua väikese ajaga graafi, mida vajab A *. Selles töös luuakse punktipilvest kaheksandpuu kasutades PCL teeki ja graaf minu poolt modifitseeritud PCLi funktsiooniga.

Keevitustee leidmiseks annab kasutaja punktivalija pistikprogrammiga A * algoritmile algus- ja lõpp-punkti. Leitud tee viiakse PointCloud2 kujule ja väljastatakse ROS2 rubriigile, kust

programm saab sellega edasi tegeleda. Kogu keevitustee leidmise protsess on kirjeldatud joonisel 12. Joonisel 13 on näidatud keevitustee otsingutulemust punasega.



Joonis 12. Tee leidmise protsess



Joonis 13. 1. Skaneeritud detail 2. detail koos poolikult genereeritud teega (A-B) 3. detail koos täielikult genereeritud teega (A-B ja C-L)

5.3 Robotmanipulatsioon

5.3.1 Roboti liigutamine

Robotit on vaja liigutada skaneerimise ja keevitustprotsessi ajal. Roboti liigutamiseks kasutatakse *MoveGroup Interface'*i asyncM*ove* ja asyncE*xecute* käske. *asyncMove* liigutab roboti ette antud asendisse, aga *asyncExecute* liigutab robotit mööda ette antud trajektoori.

5.3.2 Roboti peatamine

Võimalikult kõrge ohutustaseme tagamiseks tuli lisada keevitustarkvarale võimekus robotit peatada igal hetkel. Selleks, et robotit oleks võimalik igal hetkel peatada, tuleb kõik roboti liigutamise protsessid teha asünkroonseks. Tavalise *move* käsuga ootab programm kuni robot on liigutatud enne kui täidab järgmise koodirea. See tähendab, et *stop* käsk ei jõua robotini enne kui liigutamine on lõpuni viidud. Asünkroonsuse jaoks on MoveIt2 sisse ehitatud *asyncMove*. Seda kasutades luuakse roboti liigutamiseks peaprogrammiga paralleelne protsess. Nii on võimalik vajadusel liigutamise paralleelprotsess tappa ja robot peatada. Robotit on operaatoril võimalik peatada läbi juhtpaneelil oleva STOP nupu.

6 Lahenduse testimine

Käesolevas peatükis testitakse loodud lahendust ja antakse ülevaade riistvarast, mida testimiseks kasutati.

6.1 UR5 robot-manipulaator

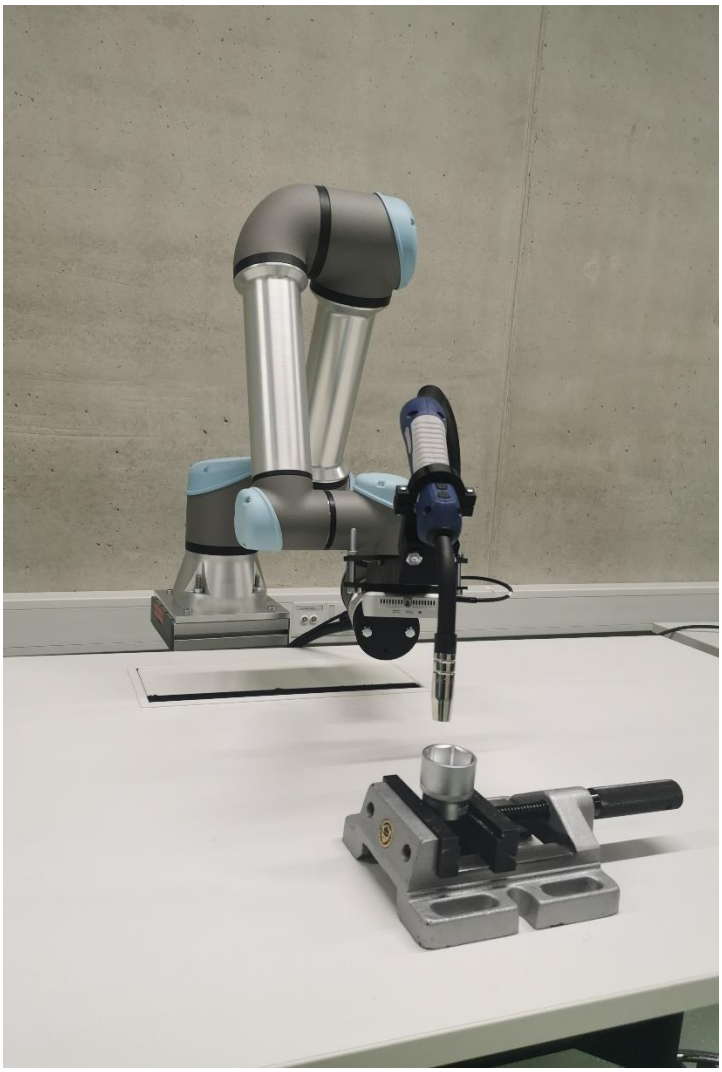
UR5 on robot-manipulaator, mida toodab *Universal Robots*. See on kuue teljeline robot, mille igal telje pöörlemisulatus +/- 360°, kiirus on kuni 180°/s. UR5 suudab tõsta kuni 5 kg, kaaludes ise 18.4kg. See robot on disainitud universaalseks robotiks, mis tähendab, et kasutaja saab selle vastavalt oma vajadusele üles seada. Universaalsust toetab ka lihtne programmeeritavus – erinevate telgede liigutusi saab programmeerida kasutades operaatoripaneeli õpetamise režiimis [24].

6.2 Intel RealSense D435i

D435i on sügavuskaamera koos inertsiaalanduriga (IMU ehk *inertial measurement unit)*. Sisemiselt kasutab IMU mitme erineva anduri kombinatsiooni, et tuvastada liikumist kolmel teljel ja pöördeid nende ümber. [26]

6.3 Robotile tööriistade kinnitamine

Lõpplahendust on näha joonisel 14.



Joonis 14. Robot koos adapteri ja tööriistadega

6.3.1 Keevituskäpa kinnitus

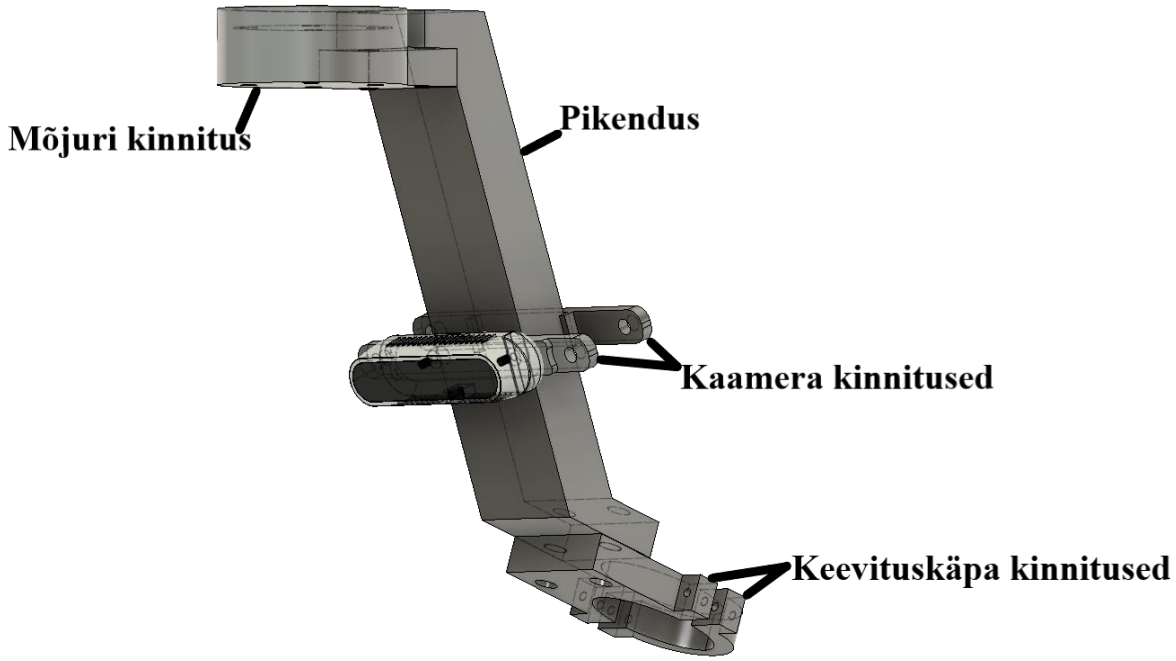
Lõppeesmärgini jõudmiseks, tuli robotile kinnitada keevituskäpp. Parima võimaliku disainini jõudmiseks uurisin juba eksisteerivaid lahendusi. Keevitusrobotite disaine on kaks – keevitusrobot, mis ongi loodud keevitusrobotiks ja robot-manipulaator, millele on keevitus hiljem juurde pandud. Esimesel lahendusel on keevituskäpp integreeritud roboti sisse, kuna see on parim võimalik lahendus (joonis 15) [27]. Robotid, kuhu keevitus on hiljem juurde pandud, teevad seda enamasti keevituskäpa kaugemalt nurga alt sisse toomisega. Selle disainiga saab minimeerida roboti kahjustamist, kuna kuumus ja sädemed hoitakse kaugemal robotist, mis ei pruugi olla tehtud sobivast materjalist.

Selle projekti raames testitakse lahendust õppemanipulaatori peal, mis ei ole spetsiaalselt loodud keevitamiseks. UR5 on osaliselt kaetud plastikuga, mis võib saada kergesti kahjustada, kui ei kasutata vajalikke abinõusid. Just nendel põhjustel disainiti keevituskäpa kinnitus nii, et kogu keevitusprotsess toimuks robotist eemal. Keevituskäpa hoidja disain on näha joonisel 16.

Joonis 15. Keevitusrobot integreeritud lahendusega

6.3.2 Kaamera

Kaamera asukoha määramisel peab arvestama asjaoluga, et kõik sügavuskaamerad ei ole loodud keevituskeskkonna kindlaks. See tähendab, et kaamerat ei tohi asetada sellisele kohale, kus keevitamise protsess saaks seda kahjustada. Samuti ei saa asetada kaamerat keevituskäpast liiga kaugele, kuna see võib teha tee planeerimise tarkvara jaoks liiga keeruliseks või halvimal juhul võimatuks. Parimaks kaamera asukohaks osutus keevituskäpa hoidiku külge kinnituva pikenduse (see toob keevituskäpa kinnituse robotist kaugemale) keskkoht (joonis 16). See on piisavalt kaugel keevituskäpast ning ei tee roboti teeplaneerimist keerulisemaks.



Joonis 16. Adapteri disain

6.4 Tarkvara testimine

Tarkvara testiti esmaselt simulatsioonikeskkonnas Gazebo, kus toimus testimine terve arendusperioodi vältel. Seal mängiti läbi standard situatsioone nagu detaili skaneerimine, keevitusteede genereerimine ja keevitustee läbimine. Võimalikult realistliku tulemuse jaoks kasutati simulatsioonides UR5 robot-manipulaatori mudelit koos loodud adapteri ja keevituskäpaga. Simulatsioonisiseselt on tarkvara võimeline täitma kõiki implementeeritud funktsionaalsusi.

Tarkvara testiti ka reaalses maailmas koos varasemalt nimetatud riistvaraga, kus see on võimeline täitma kõik implementeeritud funktsionaalsusi.

Kokkuvõte

Käesoleva töö eesmärgiks oli luua ROS2 raamistikul põhinev avatud lähtekoodiga keevitustarkvara prototüüp, mis võimaldaks operaatoril keevitatava detaili sisse skaneerimist, selle pealt keevitusteede valimist ja keevitustee sooritamist.

Eesmärgi saavutamiseks pakuti välja neljast sammust koosnev terviklahendus: roboti skaneerimiseks ettevalmistamine, detaili skaneerimine, keevitusteede genereerimine, keevitusteede läbimine. Selle jaoks implementeeriti andmetöötlusalgoritmid, teeotsimise algoritm ja kasutajaliides, mis kombineeriti juba olemasolevate ROS2 funktsionaalsustega. Tulemuseks on graafilise liidesega keevitustarkvara prototüüp, mis täitis kõik ette määratud nõuded.

Tarkvara testiti kasutades UR5 robot-manipulaatorit ja RealSense D435i sügavuskaamerat. Testimiseks loodi keevituskäppa ja kaamerat hoidev adapter. Loodud lahendus on võimeline detaili sisse skaneerima, genereerima vastavalt sisendile keevitustee ja genereeritud keevitusteed läbima. Ohutuse nimel väldib robot-manipulaator kokkupõrkeid detailiga ja selle tööd on võimalik tarkvarasiseselt iga hetk peatada.

Summary

The goal for this thesis was to make an open-source prototype welding software in ROS2. It had to be able to scan an object, make it possible for the operator to select welding points on the scanned object and execute the found welding path.

To achieve the goal, a solution consisting of four steps was proposed: preparation of the robot for scanning, part scanning, generation of welding path, execution of welding path. For this purpose, data processing algorithms, a path search algorithm, and an user interface were implemented and combined with existing ROS2 functionalities. This resulted in a prototype welding software with a graphical interface that met all predefined requirements.

The software was tested using an UR5 robot-manipulator and a RealSense D435i depth camera. An adapter to hold the camera and welder handle was created for testing. The created solution can scan the object, generate a welding path according to the input and execute the generated welding path. For safety reasons, the robot avoids collision with the detail while executing the given path and the robots work can be stopped at any time within the software.

Kirjandus

[1] X. Wang, X. Zhou, Z. Xia, ja X. Gu, „A survey of welding robot intelligent path optimization“, *J. Manuf. Process.*, kd 63, lk 14–23, märts 2021, doi: 10.1016/j.jmapro.2020.04.085.

[2] G. Cette, A. Devillard, ja V. Spiezia, „The contribution of robots to productivity growth in 30 OECD countries over 1975-2019“, *Econ. Lett.*, kd 200, märts 2021, doi: 10.1016/j.econlet.2021.109762.

[3] „The Pros and Cons of 5 Robot Programming Methods“, *Automate*. https://www.automate.org/news/the-pros-and-cons-of-5-robot-programming-methods (vaadatud 30. aprill 2022).

[4] „Planning Walkthrough — Verbotics Manual“. https://docs.verbotics.com/en/0.11/tutorials/walkthrough/ (vaadatud 20. mai 2022).

[5] „Verbotics Weld · Verbotics“, 23. oktoober 2021. https://verbotics.com/weld/ (vaadatud 23. oktoober 2021).

[6] „Almacam Weld, CAD/CAM software for robotized welding - Almacam“, 23. oktoober 2021. https://www.almacam.com/products/almacam-weld/ (vaadatud 23. oktoober 2021).

[7] „ROSWELD – ROS based framework for planning, monitoring and control of multi-pass robot welding“, lk 35.

[8] P. W. Fuerschbach ja G. R. Eisler, „SmartWeld: Open Source Applications for Weld Analysis and Optimization“, lk 39.

[9] P. Zhou, R. Peng, M. Xu, V. Wu, ja D. Navarro-Alarcon, „Path Planning with Automatic Seam Extraction over Point Cloud Models for Robotic Arc Welding“. arXiv, 2020. doi: 10.48550/ARXIV.2011.11951.

[10] R. P. Ferreira, L. O. Vilarinho, ja A. Scotti, „Development and implementation of a software for wire arc additive manufacturing preprocessing planning: trajectory planning and machine code generation“, *Weld. World*, kd 66, nr 3, lk 455–470, märts 2022, doi: 10.1007/s40194-021-01233-w.

[11] „ROS/Introduction - ROS Wiki“. http://wiki.ros.org/ROS/Introduction (vaadatud 24. oktoober 2021).

[12] „ROS1 vs ROS2, Practical Overview For ROS Developers“, *The Robotics Back-End*, 25. mai 2020. https://roboticsbackend.com/ros1-vs-ros2-practical-overview/ (vaadatud 14. mai 2022).

[13] „MoveIt 2 Documentation — MoveIt Documentation: Rolling documentation“. https://moveit.picknik.ai/main/index.html (vaadatud 26. märts 2022).

[14] ROS-I Consortium, *ROS for robotic welding & Hybrid planning for MoveIt2*, (2021). Vaadatud: 18. mai 2022. [Online Video]. Available at: https://www.youtube.com/watch?v=A_g0fzIleLs

[15] „ROS 2 - Data display with Rviz2 | Stereolabs“. https://www.stereolabs.com/docs/ros2/rviz2/ (vaadatud 26. märts 2022).

[16] „rviz/plugin_development.md at ros2 · ros2/rviz“, *GitHub*. https://github.com/ros2/rviz (vaadatud 3. aprill 2022).

[17] „Point Cloud Library“, *Point Cloud Library*. https://pointcloudlibrary.github.io/ (vaadatud 26. märts 2022).
[18] *perception_pcl*. ROS Perception, 2022. Vaadatud: 26. märts 2022. [Online]. Available at: https://github.com/ros-perception/perception_pcl
[19] *Universal Robots ROS2 Driver*. Universal Robots A/S, 2022. Vaadatud: 28. märts 2022. [Online]. Available at: https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver
[20] „ROS & ROS2“, *Intel® RealSense™ Developer Documentation*. https://dev.intelrealsense.com/docs/ros-wrapper (vaadatud 26. märts 2022).
[21] E. W. Weisstein, „Normal Vector“. https://mathworld.wolfram.com/ (vaadatud 1. mai 2022).
[22] „Fast triangulation of unordered point clouds — Point Cloud Library 1.12.1-dev documentation“. https://pointclouds.org/documentation/tutorials/greedy_projection.html (vaadatud 1. mai 2022).
[23] C.-M. Hung, „PATHFINDING IN 3D SPACE - A*, THETA*, LAZY THETA* IN OCTREE STRUCTURE“, 2016.
[24] „Octree — Open3D 0.15.1 documentation“. http://www.open3d.org/docs/release/tutorial/geometry/octree.html (vaadatud 1. mai 2022).
[25] „UR5 collaborative robot arm | Flexible and lightweight cobot“. https://www.universal-robots.com/products/ur5-robot/ (vaadatud 28. november 2021).
[26] „Depth Camera D435i“, *Intel® RealSense™ Depth and Tracking Cameras*. https://www.intelrealsense.com/depth-camera-d435i/ (vaadatud 26. märts 2022).
[27] R. O. M. Team | 11/28/2017, „7 Common Types of Robotic Welding Processes and When They’re Used“, *Automate*. https://www.automate.org/blogs/7-common-types-of-robotic-welding-processes-and-when-they-re-used (vaadatud 14. mai 2022).

Lisad

Lisa 1. Lõputöö raames valminud tarkvara prototüübi koodibaas

https://github.com/ut-ims-robotics/ros2_welding_demo

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Taaniel Küla,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
   „**ROS2 platvormile keevitusroboti tarkvara portotüübi loomine kasutades UR5 robot-manipulaatorit**“

   mille juhendaja on Veiko Vunder
   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 Commonsi litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Taaniel Küla

20.05.2022