

Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Tehnoloogiainstituut

Kristo Pool

MoveIt 2 õppematerjalid

Bakalaureusetöö (12 EAP)

Arvutitehnika eriala

Juhendaja:

robotika kaasprofessor Karl Kruusamäe

Tartu 2023

Resümee/Abstract

MoveIt 2 õppematerjalid

Populaarsel ROS2 baasil töötaval robotmanipulatsiooni tarkvaral on loodud uus versioon MoveIt 2. Tarkvara kasutama õppimiseks on vaja lihtsasti mõistetavaid juhendmaterjale. Käesoleva bakalaureusetöö eesmärk on luua õppematerjalid MoveIt 2 jaoks.

Bakalaureusetöö käigus valmisid juhised ning koodinäited tarkvara kasutamiseks. Valminud materjale järgides saab õppida MoveIt 2 kasutama nii C++ kui ka Pythoni programmeerimiskeeles. Juhiste järgi saab õppida liikumise teostamist Franka Emika Panda ja xArm robot-manipulaatoritega.

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

Märksõnad: ROS2, MoveIt 2, Franka Emika Panda, xArm

Learning materials for MoveIt 2

Popular ROS2-based robotic manipulation platform has developed a new version MoveIt 2. To learn how to use this software, there is a need for easy-to-understand learning materials. The goal of this bachelor's thesis is to create such learning materials.

The created materials contain instructions and code examples. Following the instructions, beginners can learn to use MoveIt 2 using C++ and Python programming languages for robot motion planning on Franka Emika Panda and xArm robot manipulators.

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

Keywords: ROS2, MoveIt 2, Franka Emika Panda, xArm

Sisukord

Resümees/Abstract	2
Jooniste loetelu	4
Lühendid, konstandid, mõisted	5
1 Sissejuhatus	6
1.1 Töö eesmärk	6
2 Kirjanduse ülevaade	7
2.1 Robot-manipulaatorid	7
2.2 Juhtimistarkvarad	8
3 Kasutatud töövahendid	10
3.1 ROS2	10
3.2 MoveIt 2	10
3.3 Colcon	11
4 Olemasolevad õppematerjalid	12
4.1 MoveIt 1 õppematerjalid	12
4.2 MoveIt 2 õppematerjalid	14
4.2.1 Dokumentatsioon	14
5 Õppematerjalide loomine	16
5.1 Nõuded	16
5.2 Lahendus	16
5.2.1 Töökeskkonna ettevalmistus	17
5.2.2 C++ kimp	18
5.2.3 Pythoni kimp	21
5.2.4 Kasutamine teiste robotitega	24
6 Kokkuvõte	25
Kasutatud kirjandus	26
Lihtlitsents	28

Jooniste loetelu

Joonis 1. Franka Emika Panda robot-manipulaator [1]	7
Joonis 2. Kuvatõmmis MoveIt 1 juhendi sissejuhatusest	12
Joonis 3. Kuvatõmmis kordinäidete peatükist MoveIt 1 juhendis	13
Joonis 4. Kuvatõmmis uue roboti lisamise peatükist MoveIt 1 juhendis	13
Joonis 5. MoveIt 2 dokumentatsiooni avaleht koos sisukorraga [2]	14
Joonis 6. Kuvatõmmised raskesti mõistetavast dokumentatsiooni kordinäitest (üleval) ja sama funktsionaalsust täitvast töö raames valminud lihtsustatud kordinäitest (all)	15
Joonis 7. Töö raames loodud koodihoidla struktuur	17
Joonis 8. Kuvatõmmis avajuhendi sissajuhatavast osast	18
Joonis 9. Kuvatõmmis C++ juhendist	19
Joonis 10. C++-keelse kordinäite osad kasutades <i>MoveIt Move Group Interface</i> liidest	20
Joonis 11. Franka Emika Panda robot enne ja pärast <i>Pose goal</i> liikumist “ <i>ros2 run cpp_examples pose_goal</i> ” käsu abil	20
Joonis 12. C++-keelsete kordinäidete käitamise tööpõhimõte MoveIt 2 ja ROS2-ga	21
Joonis 13. Kuvatõmmis Pythoni juhendist	22
Joonis 14. Pythoni kordinäite osad	23
Joonis 15. Pythoni programmeerimiskeeles kordinäite käitamise tööpõhimõte MoveIt 2 ja ROS2-ga	23

Lühendid, konstandid, mõisted

ROS – *Robot Operating System*, robotite jaoks mõeldud avatud lähtekoodiga raamistik.

MoveIt – Avatud lähtekoodiga tarkvara, mille abil saab robot-manipulaatorite liikumise rakendusi luua.

GitHub – Veebimajutusteenus tarkvaraarenduseks, mille kasutajad saavad luua oma koodihoidla.

API – *Application Programming Interface*, protokollide, määratluste ja tööriistade komplekt rakendustarkvara ehitamiseks.

1 Sissejuhatus

Tänapäeval on robotite kasutamine tööstustes jätkuvalt tõusuteel [3, 4]. Tööstuses on laialdaselt kasutusel robot-manipulaatorid. Need on inimkätt meenutavad, tavaliselt mitmest lülist koosnevad multifunktsionaalsed masinad (nt joonis 1). Robot-manipulaatori kasutamiseks tuleb selle liikumist eelnevalt programmeerida [5]. Robot-manipulaatorite programmeerimise lihtsustamiseks on olemas erinevaid tarkvaravahendeid, millest üks on MoveIt.

MoveIt on ettevõtte PickNik Robotics poolt välja arendatud tarkvara, mille abil on võimalik programmeerida robot-manipulaatorite liikumist [6]. 2020. aastal avaldati selle järgmine versioon MoveIt 2 [7]. Bakalaureuse- ja magistriõppe tudengid ning ettevõtete töötajad, kes MoveIti tarkvara kasutavad, tuleb õppida kasutama ka uut versiooni. Selle jaoks on vajalikud uued ja ajakohased õppematerjalid.

1.1 Töö eesmärk

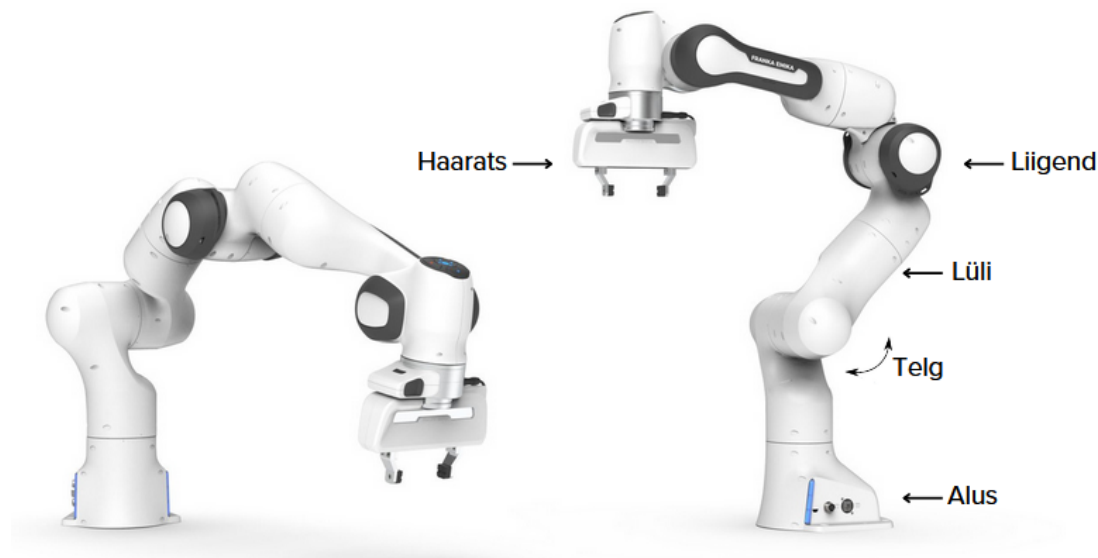
Bakalaureusetöö eesmärk on teha uued ja ajakohased õppematerjalid MoveIti versiooni 2 jaoks. Loodavad õppematerjalid on mõeldud kasutamiseks Tartu Ülikooli bakalaureuse- ja magistriõppe tudengitele robotite programmeerimise alaste oskuste edendamiseks. Lisaks sellele saavad õppematerjale kasutada ettevõtted Eestis ja Euroopas oma töötajate oskuste arendamiseks.

Bakalaureusetöö käigus luuakse ingliskeelsed õpetused ja vajalikud koodinäited C++ ja Python programmeerimiskeeltes. Valmivad õppematerjalid peavad olema paraja keerukusega, et neid saaks järgida ilma taustateadmisteta ning minimaalse programmeerimisoskusega. Juhiseid järgides saab simulatsioonis teha nii roboti liikumisplaani koostamise kui ka teostamise. Eduka simulatsiooni läbiviimisel saab samasid materjale rakendada ka reaalse roboti peal.

2 Kirjanduse ülevaade

2.1 Robot-manipulaatorid

Robot-manipulaator on masin, mis koosneb mitmest lülist, mis on omavahel seotud liigenditega (joonis 1). Viimase liigendi otsas asub haarats või mõni muu seade, mille abil robot soovitud ülesannet läbi viib. Liigendites asuvate mootorite abil saab robot ennast keerata ümber oma telgede. Suurem telgede (vabadusastmete) arv muudab manipulaatori liikumise paindlikumaks [5]. Teisalt teeb suurem telgede arv roboti rakendamise üldjuhul keerulisemaks ja kallimaks ning muudab roboti töö aeglasemaks [8].



Joonis 1. Franka Emika Panda robot-manipulaatorid [1].

Robot-manipulaatori liikuma panemiseks tuleb juhtida roboti telgi liigutavaid mootoreid. See käib juhtimistarkvara abil [5]. Selle jaoks on juhtimistarkvaral vaja järgnevaid võimekusi:

- Päri- või pöördkinemaatika – roboti liigenditele sobiva väärtuse arvutamine, et saada haarats soovitud asukohta;
- Liikumistrajektoori planeerimine – haaratsi algpunktist lõpp-punkti jõudmiseks on vaja välja arvutada terve trajektoor, mida manipulaator läbib, seejuures vältides kokkupõrkeid keskkonnaga;
- Liikumisplaani läbiviimine – robotile sobivate väärtuste edastamine, et saada soovitud liikumine.

Sellist funktsionaalsust pakkuvaid tarkvarasid on mitmeid.

2.2 Juhtimistarkvarad

Robot-manipulaatorite juhtimiseks on olemas mitmeid tarkvaralahendusi. Erinevad juhtimisprogrammid sisaldavad endas erinevaid võimekusi. Järgnev tabel 1 toob välja erinevad tarkvarad, mida kasutatakse robot-manipulaatorite liikumistee planeerimiseks ja läbi viimiseks, ning neis sisalduvad tarkvaralised võimalused. Kuna erinevaid juhtimistarkvarasid on palju, on võrdluses vaid tasuta kasutatavad tarkvarad.

	MoveIt 2	OpenRAVE	Drake	CoppeliaSim	Robotics Library	DART 6
Päri- või pöördkinemaatika	✓	✓	✓	✓	✓	✓
Planeerimine	✓	✓	✓	✓	✓	✓
Juhtimine	✓	✓	✓	✓	✓	✓
Kokkupõrgete vältimine	✓	✓	✓	✓	✓	✓
ROS2 integratsioon	✓	✗	✓	✓	✗	✗
Simuleerimine	✓	✓	✓	✓	✓	✓
Uue roboti lisamise tugi	✓	✓	✓	✓	✗	✓
Päriselus robotite juhtimine	✓	✗	✓	✓	✗	✓
Programmeerimis-keelte tugi	C++, Python	C++, Python	C++, Python	C, C++, Python, Java	C++, Python	C++, Python
Toetatud platvormid	Windows, Linux	Windows, Linux	Linux, macOS	Windows, Linux, macOS	Windows, Linux, macOS	Windows, Linux, macOS
Vabavara	✓	✓	✓	✓*	✓	✓

Avatud lähtekood	✓	✓	✓	✓	✓	✓
Väljalaskeaasta	2020	2012	2019	2019	2009	2016

Tabel 1. Juhtimistarkvarade võrdlus. **Kommertseesmärgil kasutamine on tasuline*

Eelneva tabeli põhjal on näha, et ka paljud teised robotite juhtimistarkvarad sisaldavad endas piisavalt võimekusi, et robot-manipulaatoreid edukalt juhtida. Samas saab öelda, et robot-manipulaatorite jaoks on just MoveIt 2 väga hea tarkvara, kuna sisaldab kõike vajalikku. MoveIt 2 üks suurimaid eeliseid on ühilduvus ROS2-ga, kuna tuleviku väljavaadete poolest liiguvad robotika alased rakendused just ROS2 kasutamise suunas. Lisaks sellele on MoveIt 2 puhul kasulik see, et paljud funktsionaalsused, mida teiste tarkvarade jaoks tuleb eraldi juurde lisada (nt pluginate või kõrvaliste koodihoidlate kaudu), on MoveItisse juba sisse ehitatud ning eraldi ettevalmistust ei vaja.

3 Kasutatud töövahendid

3.1 ROS2

ROS (*Robot Operating System*) on avatud lähtekoodiga raamistik, mis annab kasutajale vajalikud tarkvaravahendid erinevate robotipõhiste rakenduste loomiseks. ROSi suureks väärtuseks on tarkvaraline modulaarsus, mis võimaldab kasutajatel lihtsasti siduda mitmeid erinevaid robotika alaseid rakendusi ROSiga. Lisaks sellele on ROS tasuta kasutatav [9, 10].

ROS ja selle põhjal loodud rakendused koosnevad kimpudest (ingl *package*), mis võivad sisaldada mitmeid sõlmi (ingl *node*). Sõlmedevaheline suhtlus käib läbi rubriikide (ingl *topic*) kasutades sõnumeid (ingl *message*) [9].

ROSi edasiarendus ROS2 avaldati 2017. aastal [11]. ROS2 arhitektuuris on eelkäijaga võrreldes mitmeid muutusi. Uue versiooni põhilised eelised on:

- Turvalisus – kasutusele võeti DDS võrguprotokoll, mis on eelnevast turvalisem ja töökindlam;
- ROS Master sõlme ära jätmine – varasemal versioonil oli sõlmedevahelise suhtluse alustamiseks vaja ROS Master sõlme. ROS2 puhul saavad sõlmed omavahel kohe otse suhelda;
- Erinevate platvormide tugi – ROS1 töötas ainult Linuxi põhjal, kuid ROS2 sobib ka Windowsi ja macOS süsteemidele;
- Ühtne C-keelne alustek – lihtsam arendada erinevate programmeerimiskeelte tuge ROS2-le [12].

Käesolevas töös on kasutuses ROS2 Humble Hawksbill, mis töötab Ubuntu 22.04 peal.

3.2 MoveIt 2

MoveIt on laialdaselt kasutuses tänapäevaste ROSi peal töötavate robotite juhtimises. MoveIt vajab ROSi vahevara, kuna programmi töötamisel kasutatakse ROSi sõlmi, sõnumeid, rubriike jm. MoveIti abil saavad robotid oma ümbruskonda tajuda, luua liikumisplaan ja neid täide viia, samal ajal kokkupõrkeid vältides [13]. Veel pakub MoveIt roboti ja selle liikumise visualiseerimist kolmemõõtmelises ruumis RVizi abil ja automaatset haaratsi kontrolli haaramisel. Lisaks sellele saab MoveIti abil juhtida ükskõik millist robot-manipulaatorit, kui see on korrektselt ära kirjeldatud [14]. See tähendab, et on olemas

vajalikud failid, mis sisaldavad roboti lülide ja liigendite kirjeldusi, liikumise piiranguid jms. MoveIt sisaldab uue robot-manipulaatori kirjeldamise lihtsustamiseks *MoveIt Setup Assistance* graafilist kasutajaliidest [15].

ROS raamistik on pidevas arenduses ning valminud on selle edasiarendus ROS2. Kuna ROS2 arhitektuuris on uuel versioonil palju muudatusi, siis MoveIti esimene versioon sellega ei ühildu ja on vajalik ka MoveIt ajakohasena hoida. 2020. aasta alguses avaldas PickNik Robotics uue versiooni MoveIt 2, mis on sobilik ROS2 peal kasutamiseks, pidades seda isegi ROS2 võtmeosaks [7].

Käesoleva töö raames valminud juhendid on MoveIt 2 õppimiseks.

3.3 Colcon

Colcon on käsureapõhine tööriist, mis töötab ROSi põhjal. Colcon lihtsustab rakenduste loomist ja testimist. Colconi abil saab erinevaid kimpe siduda terviklikuks töökeskkonnaks, samal ajal automaatselt järgides ROSi tavasid [16].

Käesoleva töö raames valminud õppematerjalides kasutatakse töökeskkonna loomiseks Colconi.

4 Olemasolevad õppematerjalid

On olemas mitmeid õppematerjale MoveIt 1 jaoks [17, 18] ning ametlikud juhised ka MoveIt 2 jaoks [2]. Käesolev peatükk annab ülevaate olemasolevatest juhenditest.

4.1 MoveIt 1 õppematerjalid

Tartu Ülikoolis on olemas komplekt MoveIt 1 õppematerjale, mida on kasutatud tudengite õpetamiseks “Robotitehnoloogia” aines (joonis 2). Juhend sisaldab roboti-manipulaatorite tööpõhimõtte selgitusi ning õpetusi tarkvara paigaldamiseks ja kasutamiseks. Õpetusega käib kaasas koodihoidla [18], mis sisaldab C++-keelseid koodinäiteid roboti liikumisplaani tegemiseks ja täide viimiseks. Koodinäidete kasutamiseks tuleb õppuril juhiste järgi MoveIti tarkvara paigaldada ning koodinäidete hoidla oma tööruumi sisse tuua. Seejärel saab juhiseid järgides roboti liikumise koodinäiteid läbi teha.

LOTI.05.057 Robotics Technology
Karl Kruusamäe
Sandra Schumann

Spring 2022
University of Tartu

ROS: Motion Planning with MoveIt

The objective of this lab is to learn how to use ROS MoveIt for motion planning. Get familiar with the key concepts and go through the process of setting up MoveIt! configuration package for an industrial robot. In order to complete the lab, we are going to use the digital twins of three different robot manipulators depicted in Figure 1.

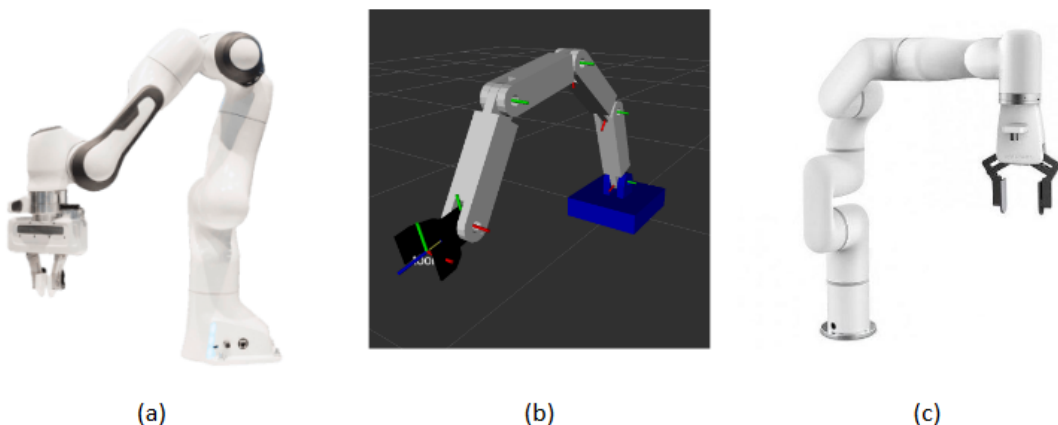


Figure 1. (a) Franka Emika Panda; (b) Aironbot; (c) UFactory xArm7.

Joonis 2. Kuvatõmmis MoveIt 1 juhendi sissejuhatusest.

Juhend keskendub kolmele robotile: Franka Emika Panda, Aironbot ja xArm7. Koodinäidetega saab MoveGroup C++ liidese abil panna xArm7 roboti simulatsioonis ja ka päriselus liikuma (joonis 3). Eraldi koodinäide on olemas järgnevate liikumisplaanide loomise viiside jaoks:

- *Pose goal* – roboti haaratsile lõpp-asukoha koordinaatide andmine;
- *Joint value goal* – roboti liigenditele kindla nurgaväärtuse andmine;
- *Named goal* – eelnevalt määratud nimelise asendi andmine;
- *Cartesian path* – mitme punkti määramine, mida robot oma liikumisel läbib;
- *Dual pose goal* – kahele eraldi robotile haaratsi lõppasendi andmine;
- *Gripper close* – haaratsi nimeline sulgemine;
- *Gripper joint goal* – haaratsile väärtuse andmine, kui palju see avanema peab.

MOVEGROUP C++ INTERFACE

In this section we are going to learn how to program a manipulator robot using the MoveGroup C++ interface. Here's the list of potentially relevant learning nuggets.

- 1) [MoveGroup C++ interface](#)
- 2) [MoveGroup C++ interface: planning to a pose goal](#)
- 3) [MoveGroup C++ interface: planning to a joint value goal](#)
- 4) [MoveGroup C++ interface: planning to a named goal](#)
- 5) [MoveGroup C++ interface: planning a Cartesian path](#)

All the example code solutions can be found in the repository for [movegroup interface demo](#).

Joonis 3. Kuvatõmmis koodinäidete peatükist MoveIt 1 juhendis.

Lisaks koodinäidetele on juhendis RVizi kasutamise õpetus ja teiste robotite lisamise õpetus *MoveIt Setup Assistant* abil (joonis 4).

MOVEIT SETUP ASSISTANT FOR GENERATING A MOVEIT CONFIGURATION PACKAGE

Next, let's take a look at how to start using MoveIt with any serial manipulator we may have or develop. Complete the following two learning nuggets to understand what a MoveIt configuration package is.

- 1) [Creating a kinematic description for a manipulator robot](#)
NB! Do not add `aironbot_description` as a submodule to your catkin workspace. Instead make a carbon copy by cloning `aironbot_description` and then deleting the `.git` folder inside the `aironbot_description` folder.
- 2) [Using MoveIt Setup Assistant to generate a MoveIt configuration package](#)

Joonis 4. Kuvatõmmis uue roboti lisamise peatükist MoveIt 1 juhendis.

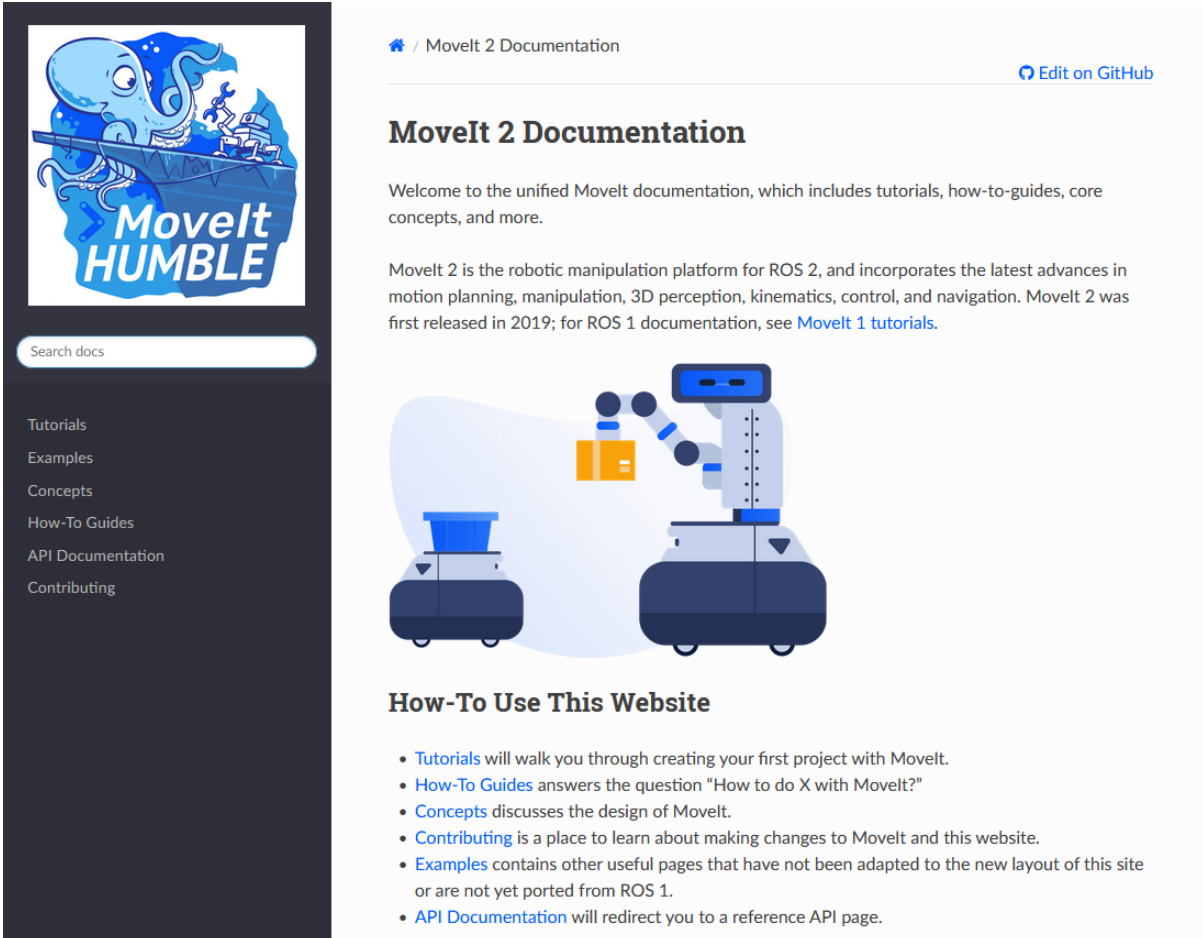
Käesoleva töö raames valminud õppematerjalid toetusid tugevalt sellele juhendile.

4.2 MoveIt 2 õppematerjalid

Juhiseid MoveIt 2 kasutamiseks leiab MoveIti kodulehel olevast ametlikust dokumentatsioonist [2]. Lisaks sellele on PickNik Robotics teinud ka videoseeria, milles on mõned algtasemel õpetused ja demod [19].

4.2.1 Dokumentatsioon

MoveIt 2 ametlik dokumentatsioon sisaldab õpetusi, näiteid ja erinevate tarkvaraga seotud kontseptsioonide selgitusi. Need osad on jaotatud eraldi peatükkidesse (joonis 5) [2].



Home / MoveIt 2 Documentation [Edit on GitHub](#)

MoveIt 2 Documentation

Welcome to the unified MoveIt documentation, which includes tutorials, how-to-guides, core concepts, and more.

MoveIt 2 is the robotic manipulation platform for ROS 2, and incorporates the latest advances in motion planning, manipulation, 3D perception, kinematics, control, and navigation. MoveIt 2 was first released in 2019; for ROS 1 documentation, see [MoveIt 1 tutorials](#).

How-To Use This Website

- [Tutorials](#) will walk you through creating your first project with MoveIt.
- [How-To Guides](#) answers the question "How to do X with MoveIt?"
- [Concepts](#) discusses the design of MoveIt.
- [Contributing](#) is a place to learn about making changes to MoveIt and this website.
- [Examples](#) contains other useful pages that have not been adapted to the new layout of this site or are not yet ported from ROS 1.
- [API Documentation](#) will redirect you to a reference API page.

Joonis 5. MoveIt 2 dokumentatsiooni avaleht koos sisukorraga [2].

Õpetuste peatükk katab MoveIt 2 kasutamise algtasemest. Ette on näidatud MoveIt 2 ja Colcon'i paigaldamine ning tööruumi valmis seadmine. Sellele alapeatükile on viidatud ka käesoleva töö raames valminud juhendis. Lisaks sellele on näidatud RViz'i kasutamine läbi graafilise kasutajaliidese, ROS2 kimbu loomine MoveIti tarbeks ja lihtsa liikumisplaanide tegemine.

Näidete peatükis tutvustatakse *Move Group C++ Interface* liidest ja MoveIti kasutamist otse läbi C++ API. Veel sisaldab näidete peatükk juhendit uue roboti kasutamiseks MoveIt 2-ga. Sellele alapeatükile on viidatud ka töö raames valminud uue roboti lisamise juhistes. Lisaks on olemas õpetused erinevate planeerijate jaoks.

Kontseptsioonide peatükis on selgitatud MoveIt 2 tarkvara arhitektuur, liikumise planeerimise teoreetiline külg, *move_group* sõlme töö ja roboti ning ümbruskonna jälgimise võimekus.

Kokkuvõttes sisaldab dokumentatsioon palju vajalikku informatsiooni, et tarkvara edukalt kasutada. Dokumentatsioonil on ka miinuseid. Ette näidatud koodinäited ei ole kuigi algajasõbralikud. Koodinäited on küll kommenteeritud, kuid read on kohati keeruliselt kirjutatud, kuigi saaks lihtsamini (vt joonis 6). Lisaks sellele on paljud olulised osad erinevate peatükkide alla ära jaotatud, mille leidmine ja seostamine võib tekitada segadust.

```
// Create a plan to that target pose
auto const [success, plan] = [&move_group_interface]{
    moveit::planning_interface::MoveGroupInterface::Plan msg;
    auto const ok = static_cast<bool>(move_group_interface.plan(msg));
    return std::make_pair(ok, msg);
}();

// Create a plan to that target pose and check if that plan is successful
moveit::planning_interface::MoveGroupInterface::Plan my_plan;
bool success = (move_group_interface.plan(my_plan) == moveit::core::MoveItErrorCode::SUCCESS);
```

Joonis 6. Kuvatõmmised raskesti mõistetavast dokumentatsiooni koodinäitest (üleval) ja sama funktsionaalsust täitvast töö raames valminud lihtsustatud koodinäitest (all).

5 Õppematerjalide loomine

Töö eesmärgiks on teha valmis õppematerjalid, mida järgides saab õppida kasutama MoveIt 2 tarkvara. Loodud materjalid sisaldavad juhiseid ning koodinäiteid, mis demonstreerivad tarkvara kasutamist robot-manipulaatorite liikuma panemiseks.

5.1 Nõuded

1. Juhised peavad olema inglise keeles.
2. Õppematerjalid peavad sisaldama koodinäiteid nii C++ kui ka Pythoni programmeerimiskeeltes.
3. Juhendi järgi saab MoveIt 2 abil juhtida Franka Emika Panda ja xArm roboteid.
4. Koodinäited peavad olema lihtsad ja kommenteeritud.
5. Juhendit peab olema võimalik järgida ka eelnevate robotika alaste teadmisteta.

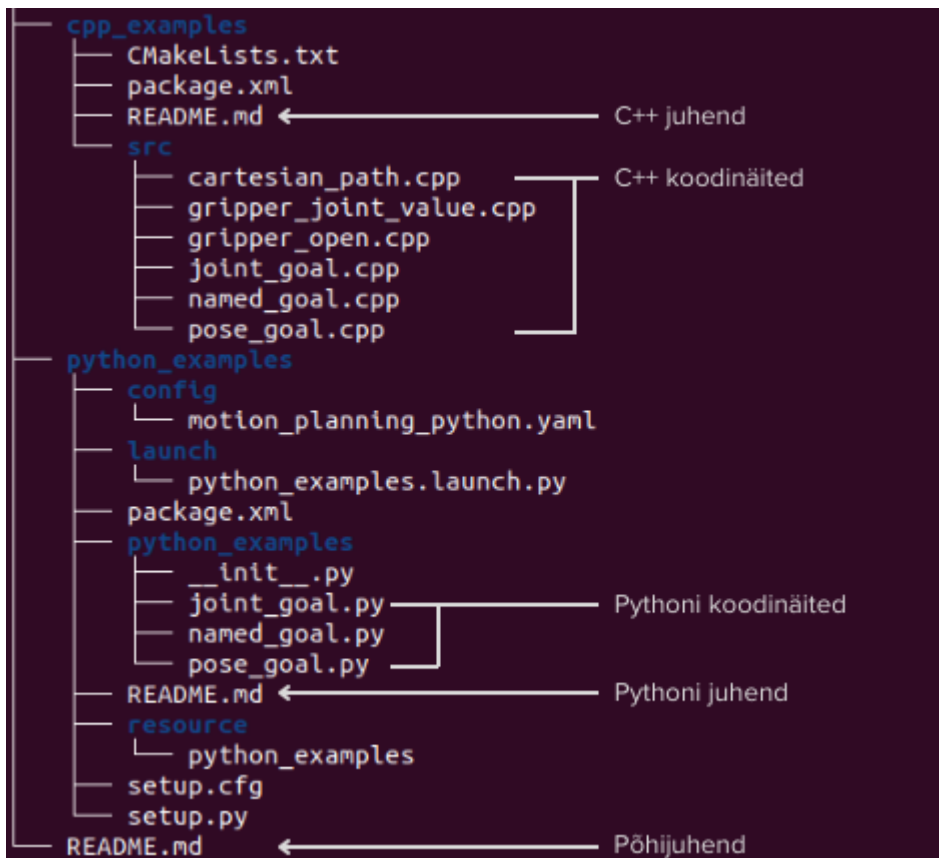
5.2 Lahendus

Valminud materjalid: <https://github.com/ut-ims-robotics/pool-thesis-2023-moveit2-examples/>

Juhend asub GitHub keskkonnas ning selle peamine eesmärk on tutvustada kasutajale MoveIt 2 kasutamist koodinäidete abil. Juhend põhineb koodihoidlal, mis hoiab endas kahte ROS2 kimpu. Koodihoidla struktuur on näidatud joonisel 7. Kimbud sisaldavad koodinäiteid kasutamaks Franka Emika Panda robotit erinevates programmeerimiskeeltes.

Koodihoidla avalehel on juhised, kuidas vajalikke programme ja töökeskkonda valmis seada. Avajuhendi lõpus on lisaks juhend, kuidas kasutada MoveIt 2 ka xArm roboti peal ning viide ametlikele juhistele, kuidas seada valmis ükskõik millise roboti kasutamist MoveIt 2 abil.

Esimene kimp *cpp_examples* on mõeldud C++ programmeerimiskeeles MoveIt 2 kasutamise õppimiseks. Teise kimbu *python_examples* eesmärk on tutvustada kasutajale MoveIt 2 kasutamist Pythoni programmeerimiskeeles. Mõlemad kimbud sisaldavad endas eraldi juhiseid, kuidas vastavat kimpu kasutada.



Joonis 7. Töö raames loodud koodihoidla struktuur.

5.2.1 Töökeskkonna ettevalmistus

Koodihoidla avaleht sisaldab juhiseid, kuidas oma töökeskkonda õigesti üles seada (joonis 8). Alustuseks on kirjeldatud vajalikud eeltingimused, mida on MoveIt 2 kasutamiseks vaja. Selle jaoks viitab juhend ROS2, MoveIt 2 ning Colconi ametlikele installatsiooniahjuhistele. Järgnevalt on näidatud, kuidas töö raames valminud kimpusid oma töökeskkonda lisada ning oma töökeskkond lõplikult valmis panna.

Movelt 2 examples



This repository is made for learning Movelt 2.

It contains two packages with code examples that demonstrate the usage of Movelt 2 through C++ and Python API for the Franka Emika Panda robot.

When using the commands provided in this guide, replace `ws_moveit2` with the name of your workspace if it is different.

Prerequisites

- ROS2 (Humble Hawksbill)

[ROS2 installation guide](#)

- Movelt 2 source code and Colcon

[Movelt2 and Colcon installation guide](#)

It is important that we use Movelt 2 Tutorials as our source, because it also contains the description for the Panda robot and visualising tools.

Joonis 8. Kuvatõmmis avajuhendi sissajuhatavast osast.

5.2.2 C++ kimp

Koodinäited demonstreerimaks MoveIt 2 kasutamist C++ programmeerimiskeeles asuvad *cpp_examples* kimbus. Kimp sisaldab 6 erinevat roboti liikumise koodinäidet. Kimbuga on kaasas eraldi juhend, milles on kirjeldatud kimbus sisalduvad näited ning juhised näidete kasutamiseks (joonis 9). Kuna C++ on põhiline toetatud programmeerimiskeel MoveIt 2 kasutamiseks, siis piisab koodinäite kasutamiseks lihtsalt näidisfaili käitamisest *ros run* käsuga.

Package for learning MoveIt 2 Move Group C++ Interface

This package contains code examples for visualizing and controlling a Franka Emika Panda robot with MoveIt 2 Move Group Interface.

Running with Panda

- Make sure you have this package in your workspace and it is built with Colcon.
- Make sure you have sourced ROS and your workspace every time you open a terminal window.
- Visualize the Panda Arm in RViz

```
ros2 launch moveit2_tutorials demo.launch.py
```

You can use the interactive markers to set goal poses for the robot and then press "Plan & execute" to move the robot.

- Demonstration of different goals

In another terminal, use the following commands for demonstration of different goals. Feel free to read the commented code for each example in src directory to understand how it works.

Pose goal - setting a target pose for the end-effector:

```
ros2 run cpp_examples pose_goal
```

Named goal - setting a target pose that is previously defined in panda.srdf configuration file:

```
ros2 run cpp_examples named_goal
```

Joint-space goal - setting a value to each joint of the robot:

```
ros2 run cpp_examples joint_goal
```

Cartesian path - giving a set of waypoints that the robot follows:

```
ros2 run cpp_examples cartesian_path
```

Gripper open - opening the gripper of the robot:

```
ros2 run cpp_examples gripper_open
```

Gripper joint value - giving a custom value to set how much the gripper is opened:

```
ros2 run cpp_examples gripper_joint_value
```

Joonis 9. Kuvatõmmis C++ juhendist.

Juhendi alguses on meeldetuletus, et selle kimbu kasutamiseks peab see olema töökeskkonda sisse toodud ning Colcon'i abil kogu töökeskkond valmis seatud. See on kirjeldatud avajuhendis ning selles kimbus eraldi õpetusi selle jaoks ei ole. Järgnevalt on näidatud, kuidas Franka Emika Panda robot-manipulaatorit RViz programmis kuvada. Seejärel on eraldi välja toodud iga koodinäite põhiline idee, mille järgi roboti liikumist läbi viiakse, ja käsk, kuidas vastava liikumise demonstratsiooni käivitada. Juhend soovib kasutajatel näidete koodi ise lugeda, kuna need on kommenteeritud ning see aitab mõista, mis on iga koodirea eesmärk roboti liikuma panemisel.

Koodinäited põhinevad *Move Group C++ Interface*-il, mis on lihtsaim liides, mille abil roboti liikumist C++ keeles programmeerida [20]. Iga C++-keelse koodinäite ülesehitus on järgnev (vt ka joonis 10):

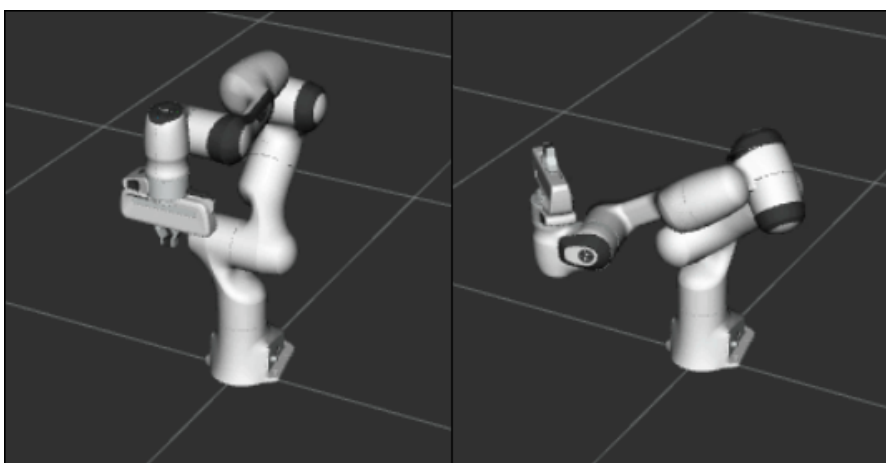
1. ROS2 sõlme ja logeri loomine – see on iga ROS-põhise rakenduse aluseks;
2. *Move Group C++ Interface* liidese loomine – pakub lihtsasti kasutatavat funktsionaalsust MoveIt 2 kasutamiseks C++ programmeerimiskeeles;
3. Roboti-manipulaatori lõppasendi seadmine – määratakse, milline peab olema roboti asend pärast koodi täitmist;
4. Liikumise teostamine – *Move Group C++ Interface* abil luuakse liikumisplaan soovitud asendisse ning sobiva trajektoori leidmisel viiakse see täide.



Joonis 10. C++-keelse koodinäite osad kasutades *MoveIt Move Group Interface* liidest.

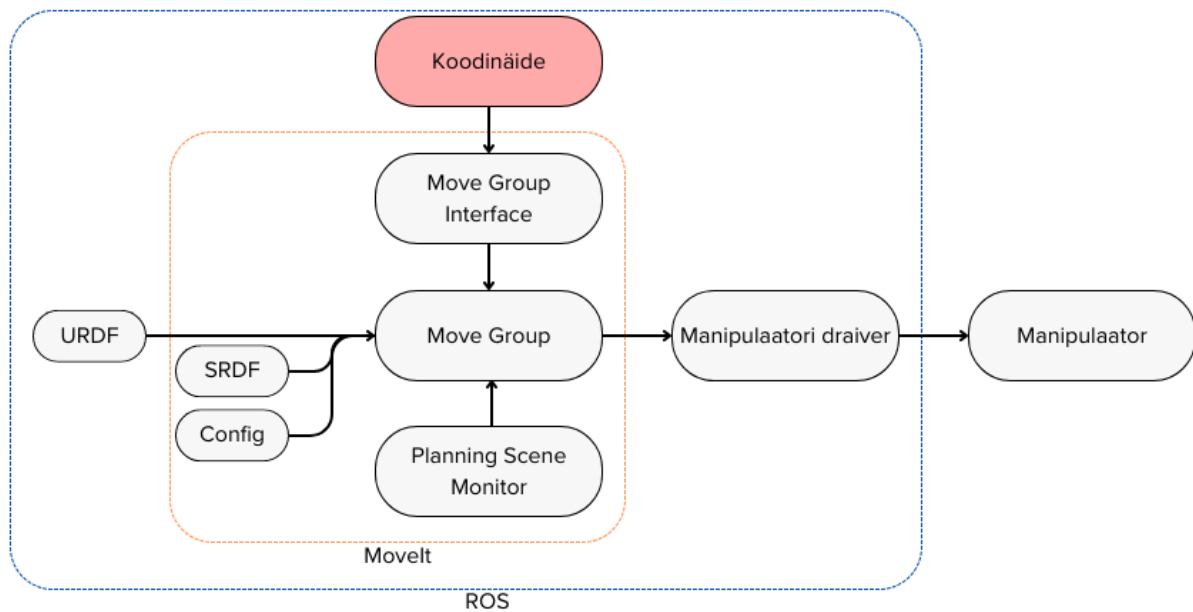
Iga koodinäite puhul on lõppasendi seadmine erinevalt määratud. Nendeks on:

- *Pose goal* – roboti haaratsile lõpp-asukoha andmine (joonis 11);
- *Joint goal* – roboti liigenditele nurgaväärtuse andmine;
- *Named goal* – robotile eelnevalt määratud nimelise asendi andmine, mis on kirjeldatud roboti SRDF failis, Franka Emika Panda puhul panda.srdf failis;
- *Cartesian path* – robotile mitme liikumispunkti andmine, mida sirgjooneliselt läbitakse;
- *Gripper open* – roboti haaratsi avamine;
- *Gripper joint value* – roboti haaratsile väärtuse andmine, kui palju haarats avanema peab.



Joonis 11. Franka Emika Panda robot enne ja pärast *Pose goal* liikumist “*ros2 run cpp_examples pose_goal*” käsu abil.

C++-keelsete koodinäidete käitamise tööpõhimõte ROS2 ja MoveIt 2-ga on illustreeritud joonisel 12.



Joonis 12. C++-keelsete koodinäidete käitamise tööpõhimõte MoveIt 2 ja ROS2-ga.

5.2.3 Pythoni kimp

Koodinäited, mis demonstreerivad MoveIt 2 kasutamist Pythoni programmeerimiskeeles asuvad *python_examples* kimbuses. Kimbuga on kaasas juhend, milles on kirjeldatud kimbuses sisalduvad näiteid ja õpetused vastavate näidete kasutamiseks (joonis 13). Kimp sisaldab 3 erinevat roboti liikumise koodinäidet. Lisaks sellele sisaldab kimp eraldi Pythoni jaoks vajalikku konfiguratsioonifaili, kuid kuna selle toimimine ei ole algajasõbralik ja on MoveIt 2 arendajate poolt eelnevalt valmis seatud, siis nende selgitusi juhendis ei ole. Kuna juhendi loomise hetkel on Pythoni tugi veel arendamisel, ei ole koodinäited niivõrd põhjalikud kui C++ programmeerimiskeeles, sest palju funktsionaalsust on veel puudu. Lisaks sellele tuleb Pythoni puhul MoveIt 2 kasutamiseks kasutada tarkvara lähtekoodi *main* haru, mis on mõeldud hetkel arenduses oleva funktsionaalsuse jaoks ning ei pruugi olla stabiilne.

This package is made for learning MoveIt 2 MoveItPy.

This package contains code examples for visualizing and controlling a Franka Emika Panda robot with MoveIt 2 using MoveItPy API.

Keep in mind that Python support is still in development for MoveIt 2. You should use the main branch of MoveIt 2.

Running with Panda

- Make sure you have this package in your workspace and it is built with Colcon.
- Make sure you have sourced ROS and your workspace every time you open a terminal window.
- Visualize the Panda Arm in RViz

```
ros2 launch moveit2_tutorials demo.launch.py
```

You can use the interactive markers to set goal poses for the robot and then press "Plan & execute" to move the robot.

- Demonstration of different goals

In another terminal, use the following commands for demonstration of different goals. Feel free to read the commented code for each example in `python_examples` directory to understand how it works.

Pose goal - setting a target pose for the end-effector:

```
ros2 launch python_examples python_examples.launch.py example_file:=pose_goal
```

Named goal - setting a target pose that is previously defined in `panda.srdf` configuration file:

```
ros2 launch python_examples python_examples.launch.py example_file:=named_goal
```

Joint-space goal - setting a value to each joint of the robot:

```
ros2 launch python_examples python_examples.launch.py example_file:=joint_goal
```

Note that we are not just running a script, but instead launching a launch file, which also initializes configurations that are required for using MoveIt 2 with Python.

Additionally, for now, using MoveIt 2 with Python lacks some functionalities that the C++ interface has. Because of this, there are less examples for Python.

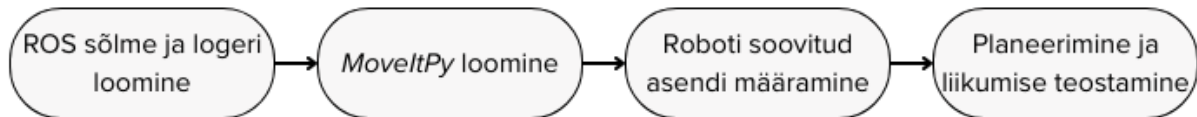
Joonis 13. Kuvatõmmis Pythoni juhendist.

Juhendi alguses on mainitud, et selle kimbu jaoks on vaja kasutada lähtekoodi vastavat haru ning kasutajal palutakse veenduda, et töökeskkond on õigesti üles seatud. Kuna töökeskkonna üles seadmise juhised on avajuhendis olemas, siis selle kimbu juhendis neid ei ole. Seejärel on juhised, kuidas Franka Emika Panda robotit RViz programmis kuvada. Järgnevalt on välja toodud iga näidiskoodi põhiline idee, mille järgi robot oma liikumist korraldab, ja käsk, millega vastavat näidist käivitada. Juhend soovib kasutajatel roboti liikumise koodinäideid ise läbi lugeda, kuna need on kommenteeritud ja aitavad roboti liikuma panemiseks vajalikke samme mõista.

Koodinäited põhinevad *MoveItPy* teegil, mis seob Pythoni programmeerimiskeeles olevad funktsioonid vastavate C++ funktsioonidega. Koodinäidete kasutamine käib *launch* faili abil, mis paneb töövalmis Pythoni kasutamise MoveIt 2 jaoks ning käivitab soovitud koodinäite. Iga koodinäide koosneb järgnevatest osadest (vt ka joonis 14):

1. ROS2 sõlme ja logeri loomine – see on iga ROS-põhise rakenduse aluseks;

2. *MoveItPy* konkretiseerimine – *MoveItPy* ette valmistamine Franka Emika Panda roboti jaoks;
3. Robot-manipulaatori lõppasendi seadmine – määratakse roboti soovitud lõppasend;
4. Liikumise teostamine – *MoveItPy* abil planeeritakse roboti liikumine algasendist soovitud lõppasendisse ning eduka trajektoori leidmisel viiakse see täide.

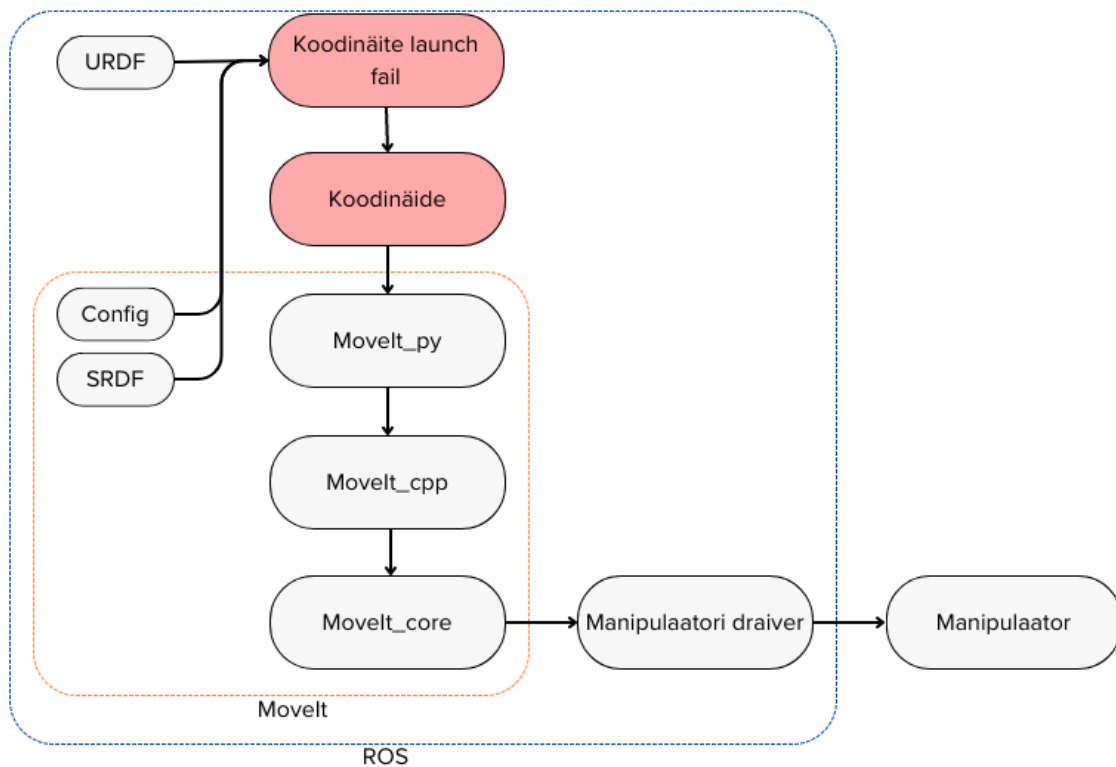


Joonis 14. Pythoni koodinäite osad.

Iga koodinäite puhul on lõppasendi seadmine erinevalt määratud. Selles kimbus on nendeks:

- *Pose goal* – roboti haaratsile lõpp-asukoha andmine;
- *Joint goal* – roboti liigenditele nurgaväärtuse andmine;
- *Named goal* – robotile eelnevalt määratud nimelise asendi andmine, mis on kirjeldatud roboti SRDF failis, Franka Emika Panda puhul panda.srdf failis.

Pythoni-keelsete koodinäidete käitamise tööpõhimõte on illustreeritud joonisel 15.



Joonis 15. Pythoni programmeerimiskeeles koodinäite käitamise tööpõhimõte MoveIt 2 ja ROS2-ga.

5.2.4 Kasutamine teiste robotitega

Kuna õppematerjalides sisalduvad koodinäited on mõeldud kasutamaks Franka Emika Panda robotiga, on avajuhendi lõpus põgus sissejuhatus kasutamaks MoveIt 2 ka teiste robot-manipulaatoritega.

Selles juhendi osas on viidatud MoveIti ametlikus dokumentatsioonis olevatele juhistele, mille abil saab ise ära kirjeldada ükskõik millise roboti-manipulaatori. Lisaks sellele viitab dokumentatsioon koodihoidlatele, mis sisaldavad juba valmis olevaid robotite kirjeldusi MoveIti jaoks. Kuna MoveIt 2 on käesoleva töö kirjutamise hetkel veel küllaltki uus, on enamused olemasolevaid kirjeldusi mõeldud eelneva versiooni jaoks.

Järgnevalt on välja toodud juhised, kuidas kasutada MoveIt 2 xArm robot-manipulaatoriga. Juhiseid järgides saab õppematerjale kasutada nii xArm5, xArm6 kui ka xArm7 robotitega. Selle jaoks on juhendis õpetused xArm robotite lähtekoodi oma töökeskkonda sisse toomiseks, värskendamiseks ja paigaldamiseks. Seejärel on juhised xArm robot-manipulaatori planeerimissõlme käivitamiseks ning RViz programmis kuvamiseks. Peale seda on toodud välja xArmi arendajate poolt loodud liikumisnäidete kasutamine, mille abil saab robotit panna liikuma haaratsile lõppasendi ja roboti liigenditele väärtuse andmise järgi.

6 Kokkuvõte

Bakalaureusetöö raames valmisid õppematerjalid MoveIt 2 kasutamise õppimiseks. Materjalid sisaldavad endas juhiseid ja koodinäiteid.

Töö käigus valmis avajuhend, milles on juhised tarkvara paigaldamiseks ja töökeskkonna ülesseadmiseks. Lisaks sellele sisaldab avajuhend ka täpsemaid juhiseid xArm robot-manipulaatori liikumise demonstatsiooniks.

Valmis kaks ROS2 kimpu – nii C++ kui ka Pythoni programmeerimiskeeles õppimiseks – mis sisaldavad koodinäiteid robot-manipulaatori juhtimiseks MoveIt 2 abil, kasutades erinevaid robotile soovitud asendi määramise viise. Mõlema koodinäidete kimbuga on kaasas eraldi juhend, mille järgi saab näidisliikumisi läbi viia. Koodinäited on mõeldud kasutamaks Franka Emika Panda robot-manipulaatoriga ja on kommenteeritud, et oleks lihtsasti mõistetavad.

Saab öelda, et töö eesmärk ning õppematerjalidele esitatud nõuded said täidetud.

Töö raames valminud õppematerjalid on saadaval lingil

<https://github.com/ut-ims-robotics/pool-thesis-2023-moveit2-examples/>

Kasutatud kirjandus

1. Franka_Panda_204-scaled.jpg (JPEG Image, 2560 × 1440 pixels) — Scaled (64%). http://www.bolee.com.hk/wp-content/uploads/2021/09/Franka_Panda_204-scaled.jpg (vaadatud 4. mai 2023).
2. MoveIt 2 Documentation — MoveIt Documentation: Humble documentation. <https://moveit.picknik.ai/humble/index.html> (vaadatud 15. mai 2023).
3. Industrial Automation And Control Systems Market Report, 2030. <https://www.grandviewresearch.com/industry-analysis/industrial-automation-market> (vaadatud 25. aprill 2023).
4. Industrial Automation Global Market Report 2022. In: automation.com. <https://www.automation.com/en-us/articles/july-2022/industrial-automation-global-market-report-2022> (vaadatud 25. aprill 2023).
5. Tallinna Tööstushariduskeskus TÄITURMEHHANISMID. https://www.tthk.ee/MEH/Taiturid_13.html (vaadatud 25. aprill 2023).
6. Coleman D, Sukan I, Chitta S, Correll N (2014) Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study.
7. PR N (2020) MoveIt, the Popular Open Source Platform For Robotic Arms, Releases ROS 2 Version. PR Newswire US
8. Advantages and Disadvantages of Low-DOF Robots. In: Robots Done Right. <https://robotsdoneright.com/Articles/advantages-and-disadvantages-of-low-dof-robots.html> (vaadatud 4. mai 2023).
9. ROS/Introduction - ROS Wiki. <http://wiki.ros.org/ROS/Introduction> (vaadatud 6. mai 2023).
10. ROS: Why ROS? <https://www.ros.org/blog/why-ros/> (vaadatud 6. mai 2023).
11. Distributions — ROS 2 Documentation: Rolling documentation. <https://docs.ros.org/en/rolling/Releases.html> (vaadatud 6. mai 2023).
12. Elmofty O (2022) ROS2 — how is it better than ROS1. Medium
13. Chitta S, Sukan I, Cousins S (2012) MoveIt! [ROS Topics]. IEEE Robot Autom Mag Robot Autom Mag IEEE IEEE Robot Autom Mag 19:18–19
14. MoveIt Motion Planning Framework. <https://moveit.ros.org/> (vaadatud 1. mai 2023).
15. Examples — MoveIt Documentation: Humble documentation. <https://moveit.picknik.ai/humble/doc/examples/examples.html#integration-with-a-new-ro>

- [bot](#) (vaadatud 17. mai 2023).
16. colcon - collective construction — colcon documentation.
<https://colcon.readthedocs.io/en/released/> (vaadatud 15. mai 2023).
 17. MoveIt Tutorials — moveit_tutorials Noetic documentation.
https://ros-planning.github.io/moveit_tutorials/ (vaadatud 17. mai 2023).
 18. (2023) movegroup_interface_demo
https://github.com/ut-ims-robotics/movegroup_interface_demo (vaadatud 4. mai 2023).
 19. PickNik Robotics - YouTube. <https://www.youtube.com/@picknikrobotics/videos>
(vaadatud 15. mai 2023).
 20. Move Group C++ Interface — MoveIt Documentation: Humble documentation.
https://moveit.picknik.ai/humble/doc/examples/move_group_interface/move_group_interface_tutorial.html (vaadatud 15. mai 2023).

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Kristo Pool,

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

„MoveIt 2 õppematerjalid“

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 kaudu Creative Commons'i 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.

Kristo Pool

18.05.2022