

Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Tehnoloogiainstituut

Hans Pärtel Pani
ROS draiver pehmerobootika haaratsile

Bakalaureusetöö (12 EAP)
Arvutitehnika eriala

Juhendajad:
PhD Karl Kruusamäe
PhD Indrek Must

Tartu 2022

Resümee

ROS draiver pehmerobootika haaratsile

Pehmerobootika haaratsite jaoks leidub vähe ROS põhiseid lahendus. ICL tehislihasest haaratsi jaoks polnud veel tehtud ROS draiverit. Töö eesmärgiks oli uurida kas pehmerobootika haaratsit saaks juhtida sarnaselt tavarobootika haaratsiga. Töö käigus valmis ROS draiver, millega on võimalik kontrollida ICL tehislihasest pehmerobootika haaratsit. Peale draiveri loodi ka lihtne testhaarats draiveri funktsionaalsuse testimiseks.

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia; T125 Automatiseerimine, robootika

Märksõnad: ICL, ROS, pehmerobootika, haaratsid

Abstract

ROS Driver for Soft Robotic Gripper

There are a small number of ROS drivers for soft robotics and none existed for ICL type artificial muscles. The aim of the thesis was to investigate whether a soft robotic gripper could be controlled in a similar way to a regular gripper. A ROS driver was developed for controlling artificial muscle grippers made of ICL material. Furthermore a simplified test gripper was constructed to verify the operation of the driver.

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

Keywords: ICL, ROS, Soft Robotics, grippers

Sisukord

1 Sissejuhatus	4
2 Ülevaade	5
2.1 ROS	5
2.2 ROS ja tavarobootika haaratsid	6
2.3 ROS ja pehmerobootika haaratsid	9
2.3.1 Hübriid pick-and-place robotkäsi	9
2.4 Tehislihased	10
2.4.1 ICL tehislihas	10
3 Töö eesmärk ja nõuded	11
4 ROS draiver	12
4.1 Draiveri programmikood	12
4.2 Draiveri ja täituri vaheline suhtlusliides	14
5 Tehislihasest täitur	15
5.1 Täituri kontrollid	15
5.1.1 Tehislihas laadimine	16
5.1.2 Tehislihas mõõtmine	17
5.1.3 Täituri püsivara	17
5.1.4 Kasutatud tehislihas	18
6 Tulemused ja arutlus	19
6.1 Võrdlus tavahaaratsi draiveriga	20
6.2 Testimine	21
7 Kokkuvõte	22
8 Viited	23
Lisad	25
Lihtlitsents	26

Lühendid

ROS - roboti operatsioonisüsteem (ingl *Robot Operating System*)

ICL - ioonne ja mahtuvuslik laminaat (ingl *ionic and capacitive laminate*)

PWM - pulsilaiusmodulatsioon (ingl *pulse width modulation*)

ADC - analoog-digitaalmuundur (ingl *analog digital converter*)

1 Sissejuhatus

Pehmerbootika hõlmab endas roboteid millel on ülim mугanemisomadus, millest tulenevalt on robotil võimatu või võimalikult väike võimalus kahjustada ümbritsevat keskkonda või mõnda muud objekti[1,2]. Pehmerrobot on tihti valmistatud elastsetest ja järelandlikest materjalidest[3]. Mугandumine teeb pehmeroboti kasutuse eelistatuks meditsiinis, katastroofi piirkondades ja üldisemalt inim- ning looduskeskkonnas[3–5]. Pehmerrobotid võivad olla bioinspireeritud - nende liikumisviis või mõni muu omadus on inspireeritud looduses leiduva organismi poolt[6,7].

Pehmetäitur on seade mis on tehtud elastsest materjalist. Materjali omadustest tulenevalt väheneb vajadus väliste sensorite ja täpse kontrolli jaoks, sest materjal kohaneb iseenesest mingi mõjuteguri pärast. Võib öelda, et selline materjal on intelligentne[3]. Nende omaduste materjali sisse viimine võimaldab teha järjest väiksemaid ja energiasäästlikumaid täitureid ning roboteid[8].

ICL (iooniline ja mahtuvuslik laminaat) on üks näide targast materjalist. Elektrivoolu abil on võimalik seda painduma panna ja samas on materjal ka piisavalt elastne, et võimaldab mõne objekti ümber kujuda, ilma seda vigastamata. See omadus teeb võimalikuks ICL materjali kasutada pehme haaratsina, mis suudab mingi objekti ümber kinni võtta ja ka kergemaid objekte liigutada[9,10].

Varasemalt on leitud tehislihastele mitmeid kasutusi, sealhulgas ka erinevad haaramislahendusi[10], kuid praeguseini pole veel tehtud tehislihasest haaratsile kontrollimislahendust, mis lubaks seda kontrollida sarnaselt mõnele olemasolevale haaratsile. Varasemaid lahendusi on kontrollitud limiteeritud tarkvaraga, mille integreerimine olemasolevate robotite peale oleks ajakulukas ja iga uue roboti jaoks spetsiaalselt tehtav lahendus. Siit tulebki selle lõputöö idee: teha tehislihase kontrollimiseks draiver, mis lihtustaks pehmetäituri integreerimist olemasoleva roboti külge. See hõlmab endast tehislihase tarkvaralise kontrollimise arendust, võttes aluseks olemasolevad tavarobootika haaratsid.

2 Ülevaade

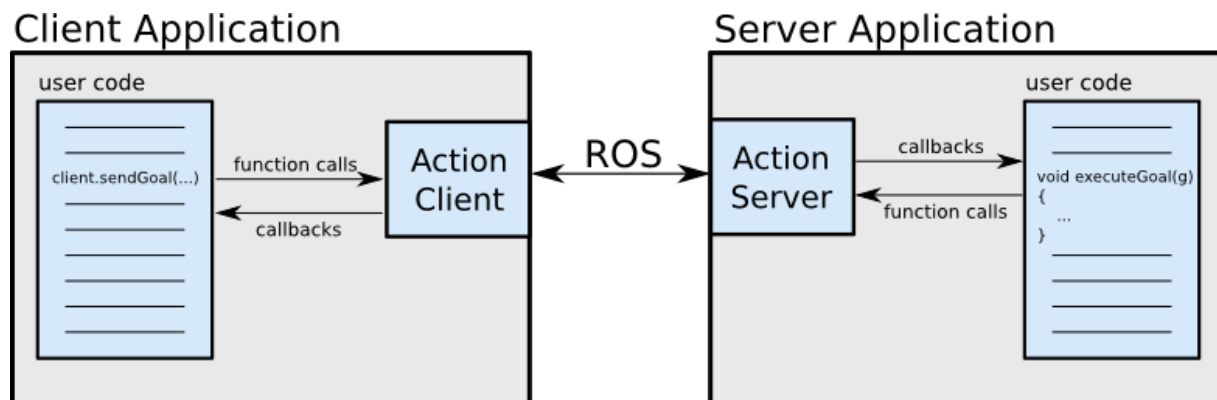
2.1 ROS

ROS (roboti operatsioonisüsteem) on modulaarne ja vabavaraline tarkvara kogum ning raamistik, mis aitab luua robotikasüsteeme. See on laialt kasutusel kommerts ja hariduslikel eesmärkidel[11]. ROSi peetakse oma valdkonnas samaväärseks standardiga[12].

ROSi organiseeritakse erinevad projektid kimpudeks (ingl *package*). Olenevalt otstarbest võivad kimbud sisaldada mitmeid erinevaid komponente, selle lõputöö jaoks nendest tähtsamad: sõlm (ingl *node*), rubriik (*topic*), sõnum (ingl *message*), teenus (ingl *service*) ja tegum (ingl *action*). Sõlm on protsess, mis teeb teatud tööd. ROSi parimate praktikate alla kuulub modulaarsus, mis näeb ette, et iga sõlm tegeleb võimalikult väikse ülesandega. Sõlmedevaheline suhtlus toimub läbi rubriikide, teenuste ja tegumite[13–15].

Rubriigis toimub suhtlus kuulutaja(*publisher*)-telliija(*subscriber*) mustriga. Igas rubriigis saab olla mitu kuulutajat ehk suhtlus toimub mitmelt-mitmele. Suhtlus käib asünkroonselt, ning puudub informatsioon selle kohta, kes ja kas saatis või võttis vastu mingi sõnumi. Rubriigi sõnumite sisu on defineeritud vastava andmestruktuuri failiga[16].

Teenus võimaldab sõlmede vahelist suhtlust klient-server mustriga. Suhtlus toimub üks-ühele ja on kliendi jaoks ette nähtud sünkroonsena, s.t. teenuse väljakutsumine ootab kuni vastuse saamiseni. Ühe vastuse ja blokeerimise tõttu ei ole teenused sobilikud pikaajalisemate protsesside jaoks[17,18]. Pikaajaliste protsessi jaoks on mõeldud tegum (joonis 1). Sarnaselt teenusega kasutab tegum klient-server arhitektuuri, kuid erinevalt teenusega toimub suhtlus asünkroonselt ehk mitteblokeerivalt. Asünkroonsusest tulenevalt võimaldab tegum tagasisidet kogu pikemaajalise operatsiooni vältel. Lisaks võimaldab tegum infot oma oleku kohta ning ka võimalust end tühistada[15].



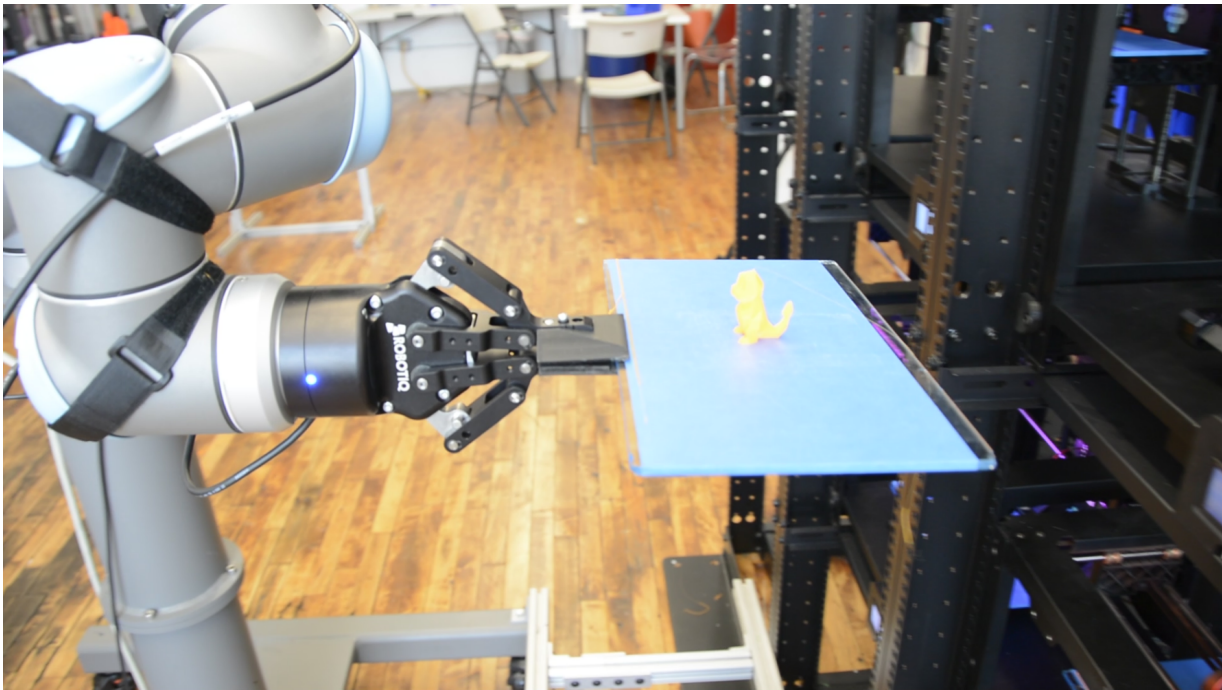
Joonis 1. ROS tegumi asünkroonse klient-server arhitektuuri illustatsioon, ROS haldab suhtlust kliendi ja serveri sõlmede vahel[15].

Draiver on programm mis liidestab tarkvara ja riistvara. ROSi puhul saab selle definitsiooni puhul draiveriks pidada sõlme, mis liidestab mõnda täiturit või riistvarakomponenti ROSiga. Liidestatud komponenti võimaldatakse kontrollida läbi ROSi.

ROSigas on kaasas `control_msgs` kimp, milles on defineeritud mitmed sõnumid, teenused ja tegumid, mis võivad roboti draiveri loomisel kasuks tulla. Haaratsite kontrollimiseks on selles kimpus defineeritud `GripperCommand` tegumi andmestruktuur. `GripperCommand` võimaldab kontrollida haaratsi positsiooni, maksimaalset pingutust. Tegumi tagasiside annab teada saavutatud positsioonist, tegelikust pingutusest ja kas on jõutud sihtmärgini või kas haarats on seiskunud[19].

2.2 ROS ja tavarobootika haaratsid

ROS on robootikas laialt kasutusel. Selles lõputöös on näitena võetud Robotiq 2F-85 ja 2F-140 haaratsid (joonis 2). Need on kommertsiaalsed haaratsid mis on võetud esinduslike näidetena, kuna nende jaoks leidub mitu erinevat ROS draiverit[20,21]. Robotiq 2F haaratsite ametlikult toetatud kontrollimisliides kasutab Modbus protokollit, üle mille on võimalik kontrollida haaratsi vahe suurust, haarde jõudu ja haaratsi sulgumise ning avanemise kiirust[22,23].

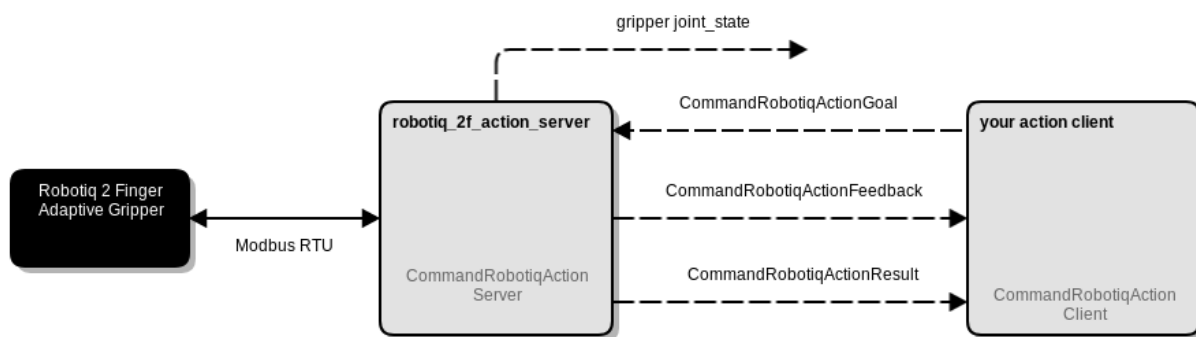


Joonis 2 Robotiq 2F haarats haaramas klaasist plaati[24].

Üks 2F haaratsite ROS draiveritest on tehtud tootja enda pool. Draiver on jagatud kaheks: haaratsi ja ROSi vahel suhtlust korraldavad sõlmed ning kasutaja käske haldavad draiversõlmed. Haaratsiga suhtluseks on võimaldatud kaks erinevat varianti: üks sõlm mis kasutab modbus rtu protokollit ja teine mis kasutab modbus tcp protokollit. Mõlemad töötavad draiversõlmede vaatepunktist samamoodi - kasutavad samu rubriike ja võtavad vastu ning edastavad samu andmeid. Kasutajal on võimalus haaratsit juhtida käsurealt või kasutades tegumeid. Käsurea draiveriga on võimalik haaratsit avada, sulgeda, liigutada haaratsi sõrmi soovitud positsiooni, muuta haaratsi sõrmede liikumiskiirust ja maksimaalset kasutatavat haardejõudu. Käsurea draiveris puudub haaratsi poolt pakutav hädaolukorra avamise käsk. Tegumipõhine draiver kasutab suhtlusliidesena control_msgs kimbus olevat GripperCommand tegumit. See valik limiteerib haaratsi kontrollimist GripperCommand tegumi seatud piiridesse. GripperCommand'is puuduvad haaratsi hädaolukorra vabastamise ja kiiruse parameetrid, need seatakse ette draiveri poolt ning kasutaja neid muuta ei saa[21].

Alternatiivne ROS draiver nendele haaratsitele on Danfoa/robotiq_2finger_grippers. Selles on kasutusel ainult üks sõlm, mis võtab vastu kasutaja käske ja korraldab ka suhtlust haaratsiga üle modbus rtu protokollit (joonis 3). Kasutaja poolt kasutatav suhtlusliides on lahendatud ROS tegumiga. Haaratsi tootja pakutud draiveri tegumist eristavad seda edasi arendatud

kontrollimisvõimalused. Kasutajal on võimalus kontrollida haaratsi sõrmed vahelist positsiooni, liikumise kiirust ja haarde maksimaalset jõudu. Lisaks on võimaldatud haaratsi hädaolukorra avamine ja liikumise seiskamise käsk. Tegumi tagasisides kajastub haaratsi sõrmede vaheline kaugus, soovitud kaugus, veateade ja lipud mis näitavad kas haarats on valmis, algseadistatud, liikumas ja kas haarats on kinni haaranud mingist objektist. Veateatega antakse teada olukorrast kus haarats on seiskunud ja ei suuda kasutaja käsku täita. Peale haaratsi kontrollimise on ka võimaldatud haaratsit vahetult robotkäe külge ühendava lüli kontrollimine. Selleks on kasutusel eraldi tegum, mis kasutab control_msgs kimbust robotkäe lüli jaoks olevat lahendust[20].



Joonis 3 Danfoa/robotiq_2finger_grippers tegumi serveri andmete liikumisskeem[20].

2.3 ROS ja pehmerobootika haaratsid



Joonis 4. ROSi kasutavad pehmehaaratsid[25–27].

Võrreldes tavarobootikaga, leiab ROS vähe kasutust pehmerobootika valdkonnas. Mõndade näidetena saab tuua jõutagasisidega pehmehaaratsi (joonisel 4, a)[25], pneumaatiline pehmehaarats põllumajandussaaduste korjamiseks (joonisel 4, b)[26] ja hübriidne robotkäsi pneumaatilise pehmehaaratsiga(joonisel 4, c)[27].

2.3.1 Hübriid pick-and-place robotkäsi

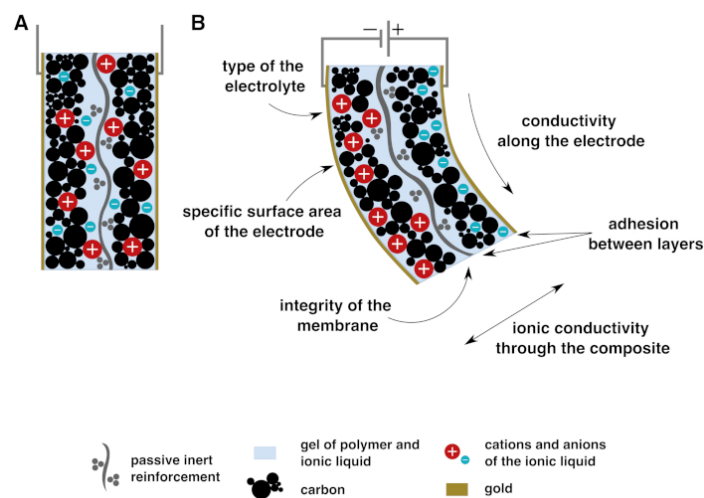
See *pick-and-place* robotkäsi koosneb tavarobootika robot käest, mille otsa on ühendatud pneumaatiline pehmerobootika haarats. Sellega on robotkäe täpne ja materjalitasandil muganeva pehmehaaratsi kontroll terviklikuks haaratsiks. Selle ROS draiveris on kasutusel eraldi sõlmed terve süsteemi kontrollimiseks, robotkäe liigutamiseks, haaratsi kontrollimiseks ja haaratsi rõhu lugemiseks. Sõlmedevaheline suhtlus on lahendatud rubriikidega. Süsteemi kontrollid haldab käe liigutamist ja haaratsi avamist ning sulgemist. Haaratsi sõlm tellib rõhu sensori rubriiki ja hoiab sellega haaret süsteemi kontrolleri soovitud tasemel. Erinevalt varasema näitena toodud Robotiq 2F haaratsitega, on hübriidse haaratsi kasutajapoolne kontroll lahendatud ROSi väliselt. Robotkäsi võtab sisendiks faili milles on g-koodis kirjeldatud soovitud liigutused. Uue liigutuse tegemiseks peab asendama draiveri sisse loetava faili sisu ja draiveri uuesti käivitama. ROSi liidestuse vaates teeb see selle robotkäe kasutuse ebamugavaks[27].

2.4 Tehislihased

Üks pehmerobootika valdkondadest on tehislihased, mis on päris lihaseid imiteerivad bioinspireeritud pehmetäiturid. Olenevalt ehitusest võib tehislihas painduda, tugevneda või teha mõnda muud kujumuutust või liigutust. Tehislihase liigutus kutsutakse esile elektrivoolu, gaasi rõhu või mõne muu mõjuteguri poolt[28]. Kuna selles lõputöös on kasutusel ICL (ioonine ja mahtuvuslik laminaat) tüüpi tehislihas, siis nendest ka pikemalt järgnevas alapeatükis.

2.4.1 ICL tehislihas

ICL tehislihas koosneb kahest kullast voolukogujast, mille vahel on kaks kõrge eripinnaga süsinikelektroodi ja elektrodide vahel poorne separaator, mis juhib ioone, aga mitte elektrone. Kogu poorne süsteem on täidetud elektrolüüdi lahusega; enamasti kasutatakse elektrolüüdina ioonseid vedelikke. Rakendades ICL täituri elektrodidele voolu, hakkab ICL laaduma. Laadumisel tekib elektrodide vahel elektriväli, mis omakorda paneb ioones vedelikus olevad ioonid liikuma. Kuna vastasmärgilised laengud tõmbuvad, siis anioonid ja katioonid liiguvad vastasmärgiliselt laetud elektrodide juurde. Peale eri suuruse või mobiilsusega ionide ümberpaigutumist on tehislihase ühel poolel kogunenud rohkem materjali kui teisel. See ümberpaigutumine tekitabki tehislihase paindumise (joonis 6). Kuna ICL tehislihas on ehituslikult sümmeetriline, on pinge polaarsuse muutmisega võimalik tehislihase paindumise suunda muuta[9,29–31].



Joonis 5 ICL tehislihase painutuse ja ionide liikuvuse sõltuvus[31].

3 Töö eesmärk ja nõuded

Bakalaureusetöö eesmärgiks oli uurida kas on mõttekas ja võimalik tehislihasest pehmetäiturit kontrollida sarnaselt mõnele tavarobootika haaratsile. Selle jaoks tehti ROS draiver tehislihasest haaratsi laadse pehmetäituri kontrollimiseks. Eeskujuks võeti olemasolevad haaratsite ROS draiverid. Valminud draiverit on võimalik kasutada sarnaselt tavarobootika haaratsite draiveritega.

Funktsionaalsed nõuded:

- Võimaldama juhtida pehmetäitur-haaratsit, kus haare objektiga toimub materjalitasandil mугanemise abil.
- Draiver on implementeeritud modulaarses ja laialt levinud programmeerimiskeskkonnas ROS.
- Standardkuju: draiver implementeeritud olemasolevate haaratsite draiverite eeskujul.
- Universaalsus: draiver on kasutatav ilma konkreetse tehislihase ehitusest või spetsiifikast midagi teadmata.

Süsteemi nõuded:

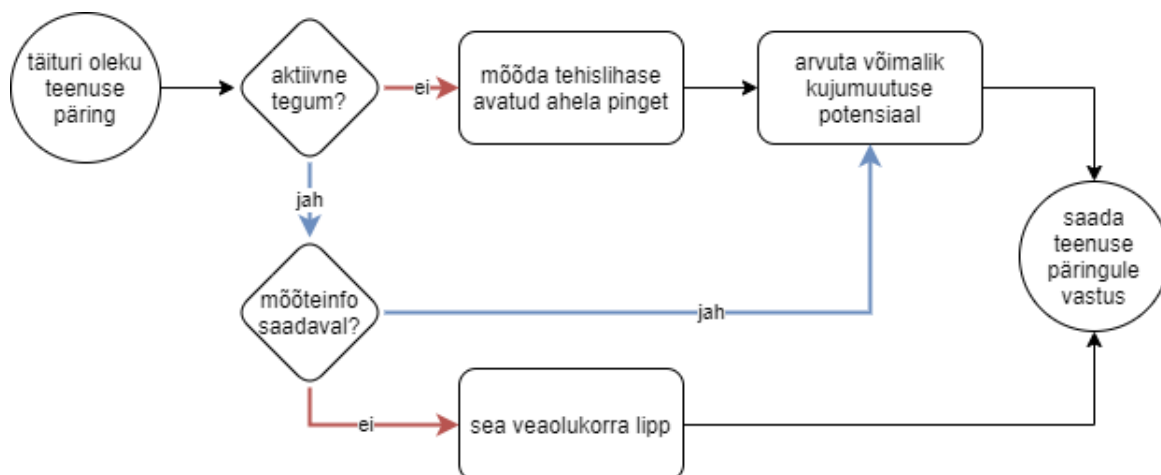
- Arendatud viimase ROS 1 redaktsiooniga.
- Draiveri kimbu struktuur järgib ROSi parimaid praktikaid.
- Kasutusliides võimaldab täiturit kontrollida ROSi tegumiga.
- Liidestus ICL tehislihasega:
 - Bipolaarselt lisada ja eemaldada tehislihaselt laengut. (kinni-lahku haaratsi kontekstis)
 - Arvestama kasutuses oleva tehislihase tehnoloogiliste piiridega (voolutihedus, potentsiaalide erinevus elektroodide vahel).

4 ROS draiver

Valminud draiver võimaldab ROS1 süsteemis kontrollida tehisliahast kui haaratsit (vt. lisa 2). Draiveris on eraldi kimpudes sõnumi-, teenuse- ja tegumifailid ning draiveri sõlme programmikood. Kasutusliides on tehtud võimalikult üldiseks ja detailid spetsiifilise tehisliahase kontrollimise kohta on peidetud abstraktsioonide taha. Nii saab seda draiverit kasutada kasutaja, kellel on minimaalsed teadmised pehmerobootikast. Kuna draiveri loomisel ei leidunud pehmerobootika valdkonnast piisavalt ROSi võimalusi kasutavaid eeskujusid, siis võeti eeskujuks ka olemasolevate tavarobootika haaratsite draiverid.

4.1 Draiveri programmikood

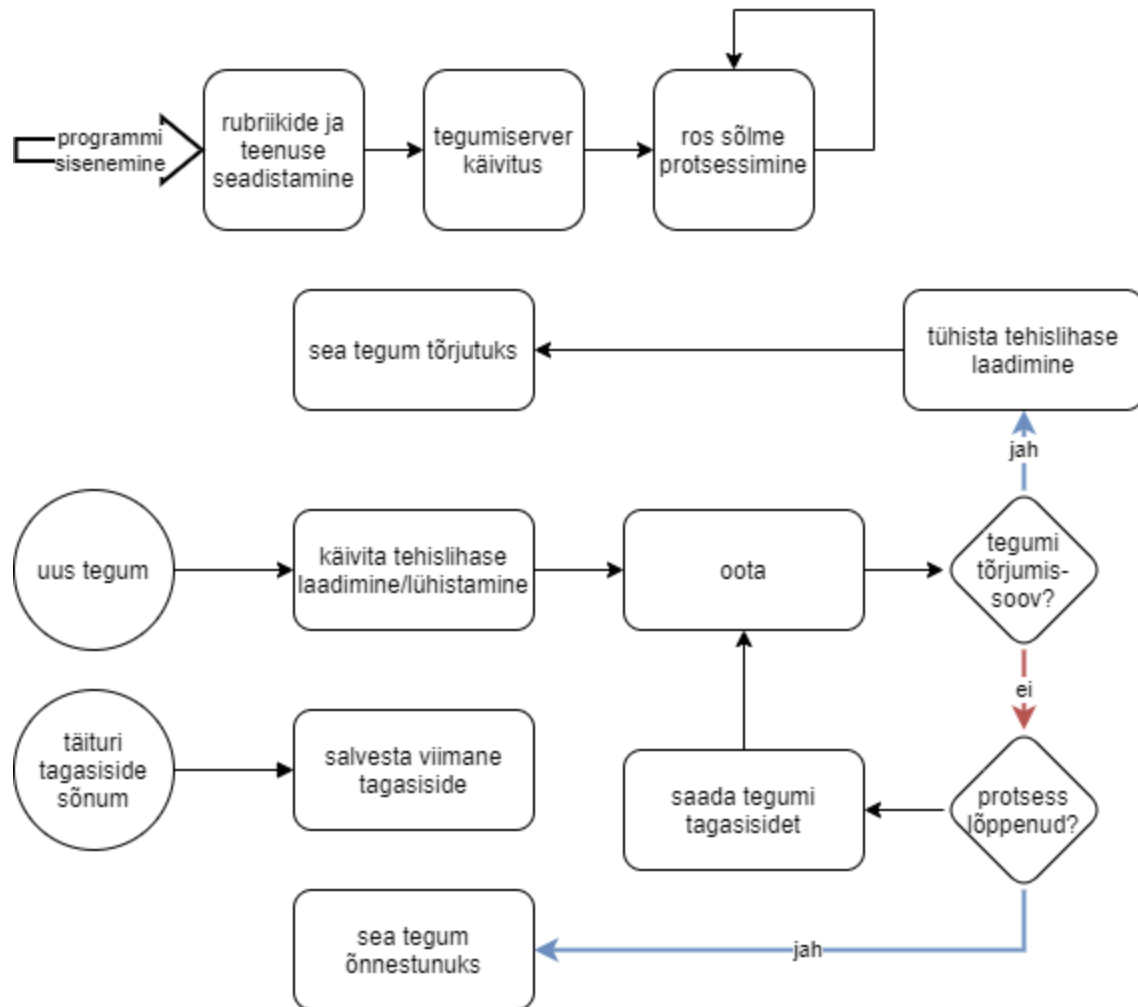
Kasutajal on draiveriga tehisliahase kontrollimiseks võimalik kasutada ROS tegumit, mille sisuks on soovitud lokaalse kujumuutuse suund: vasakule, paremale, neutraalne ja soovitud haarde kiiruse protsent haaratsi maksimumkiirusest. Tegumi väljundiks on väärtus, mis võrdleb haarde kujumuutuse potentsiaali enne liigutust olnud algseisundiga. ICL tehisliahase puhul sai kujumuutuse hinnanguks kogunenud laengu muutus ilma ühikuta. Lisaks on kasutajal võimalik draiveri ROS teenuselt saada hinnang eeldatavale potentsiaalse lokaalse kujumuutusele vastavalt vasakule ja paremale liigutusele. Hinnanguks on kasutusel 4 väärtust: 0=mõttetu kujumuutus, 1=väike kujumuutus, 2=keskmine kujumuutus, 3=suur kujumuutus. Potentsiaalse lokaalse kujumuutuse suuruse hinnang arvutatakse tehisliahase avatud ahela pingest. Kui täiturit lühistatakse või mõõteinfo pole saadaval, tagastakse hinnangu asemel vealipp, mis tähistab, et hinnangut ei olnud võimalik läbi viia (joonis 6).



Joonis 6 haaratsi liigutuse mõttekuse hinnangu ROS teenuse skeem.

Joonisel 7 on kujutatud draiveri programmi üldskeem. ROS tegumite halduseks on kasutusel SimpleActionServer serveri implementatsioon, mis haldab tegumite käivitust. Samaaegselt on lubatud ainult ühel tegumil tööd teha[32]. Selle draiveri on see vajalik, kuna täiturit saab samal hetkel laadida ainult üks kasutaja. Tegumi saamise peale hakkab draiver vastavalt soovitud kujumuutusele vasakule või paremale tehislühast laadima, või neutraalsesse olekusse naasmise olukorras lühistama. Laadimist haldab täitur ise, lülitades selle välja kui tehislüh on laetud 1.3V avatud ahela pingeni. Lühistamise puhul laseb draiver täituril kõigepealt avatud ahela pinget mõõta ja siis draiver lühistab täiturit, kuni tehislühase suletud ahela pinge keskmine on vähem kui 0.01V. Lühistamise pinge piir 0.01V tuleb ADC (analoog-digitaalmuundur) mõõtetudes esildubast mürast. 12bitisest ADC'st mis on kasutusel differentiaalmoodis on ADC näit 0-4095 vahel, mis peab differentiaalmoodis jagunema mõlemat pidi mõõtmise vahel s.t koguskaala on -3.3V kuni 3.3V või kokku 6.6V ja mõõtmise täpsuseks on $6.6/4095=0.0016V$ [33].

Kui mõne tegumi täitmise ajal on mõni tõrjuv tegum tegumi serveri poolt vastu võetud, siis käsilolev tegum peatab esimesel võimalusel laadimise, see seatakse tegumi serveris tõrjutud olekusse ja lõpetatakse väljutakse tegumi tööfunktsioonist. Uus tegum käivitatakse SimpleActionServer'i poolt kohe peale seda kui eelneva tegumi tööfunktsioonist on väljutud. Kogu tegumi vältel ja ka selle lõppedes on kasutajale võimaldatud tagasiside, mis annab teada tehislühase liigutuse muutuse relatiivsest suuruselt. Selleks on kasutusel tehislühasele salvestunud laengu arv alates laadimise algushetkest. Kasutaja mõttes on see ühikuta suurus, mida saab võrrelda varasemate liigutuste muutustega.



Joonis 7 Draiveri loogika üldskeem.

4.2 Draiveri ja täituri vaheline suhtlusliides

Täituri ja draiveri vaheliseks suhtluskanaliks on ROS rubriigid. Iga käsu jaoks on eraldi rubriik, kuna see võimaldab sõnumiga saata ainult selle jaoks vajaliku info. See sai tehtud, et vähendada igas sõnumis saadetud andmete hulka. Olemas on rubriigid tehislühase laadimiseks, lühistamiseks, avatud ahelaks tegemiseks, avatud ahela pinge mõõtmiseks ja tehislühasele kogunenud laengu nullimise käskudeks ning tehislühasest täituri tagasisideks. Igal käsusõnumil on saatmise hetke parameeter. Rosserial kimbu poolt tehtud ühendus arduino ja arvuti vahel võib desünkroniseerida ja enne taasühendamist saadetud sõnumid ei pruugi kohale jõuda. Laadimiskäsu teisteks parameetriteks on soovitud pinge ja maksimaalne lubatud laadimisvool

amprites. Täituri tagasiside sõnumi sisuks on selle hetkeseisund, tehislihase avatud ja suletud ahela pinged, tehislihasele kogunenud laengu väärtus ning täituri poolt viimase vastuvõetud käsust selle saatmise hetk. Kuna kasutusel on rubriigid millel puudub tagasiside, siis tagasiside sõnumis kajastatud täituri hetkeseisund ja viimase vastuvõetud käsu aeg annavad draiverile võimaluse kontrollida, kas täitur on käsud kätte saanud ja kas ta on oodatud olekus. Tehislihase avatud ahela pinge on kasutusel kujumuutuse potentsiaali arvutamiseks ja suletud ahela pinge tehislihase lühistamise progressi järgimiseks.

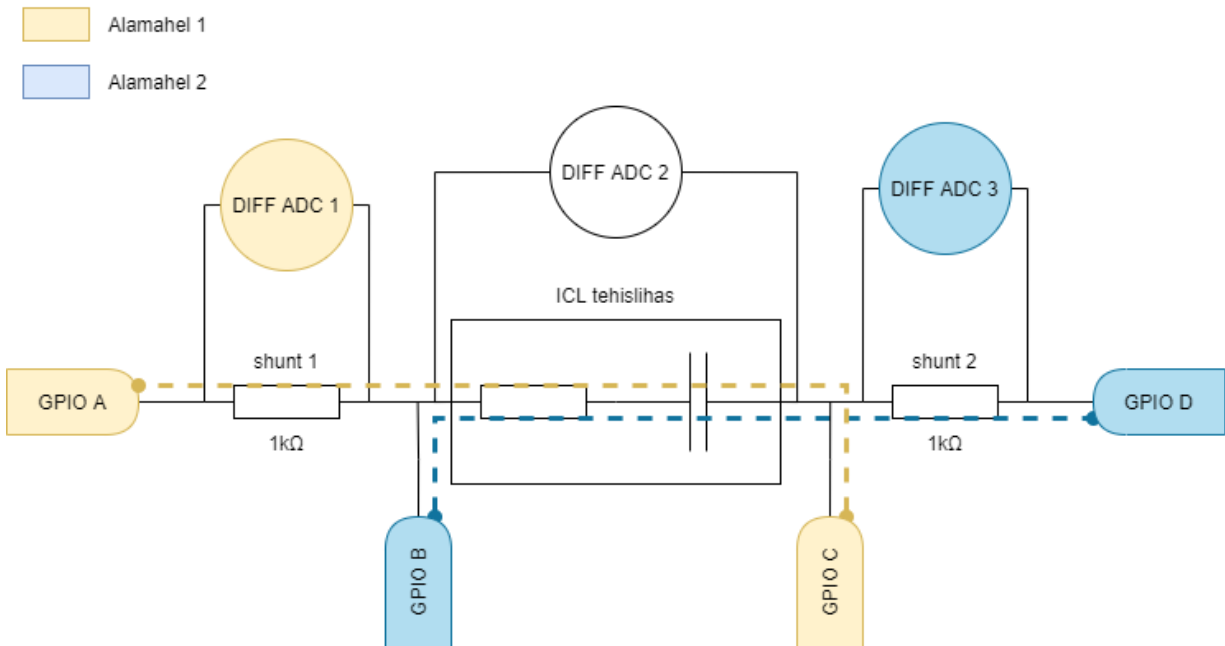
5 Tehislhasest täitur

Töö käigus valmis lihtne tehislhaset täitur, mille peal on võimalik draiverit testida (vt. lisa 1 ja lisa 3).

5.1 Täituri kontroll

Täituri kontrollimiseks on kasutusel Arduino DUE arendusplaat. Muud kasutatud vahendid on 1000 oomise shunt takistid, maketeerimislaud ja HC-06 bluetooth moodul. Arduino DUE sai valitud kuna see on kergesti saadaval, omab suurt taktsagedust, kasutab 3.3V loogikat ja võimaldab ADC kasutust differentsiaalmoodis, mis teeb tehislihase avatud ahela pinge mõõtmise kergemaks[33]. HC-06 lihtsasti integreeritav bluetooth moodul[34]. Kuna HC-06 serial liides kasutab 3.3V loogikat, siis Arduino DUE võimaldab seda ühendada ilma lisakomponentideta. Bluetooth moodul on üheks võimalikuks ühenduseks arvutiga, milles töötab ROS süsteem.

Selleks, et vältida mikrokontrolleri poolt mittemõõdetava negatiivse pinge olukorda, on tehislihase laadimis- ja mõõtmiskeem jagatud kaheks alamahelaks: 1 ja 2 (joonis 8). Tehislihase pärisuunda liigutamiseks kasutatakse 2. alamahelat, kus GPIO B viiakse PWMi (pulsilaiusmodulatsioon) poolt kõrgeks ja GPIO D viiakse madalaks. Pärisuunaliseks lühistamiseks kasutatakse 1. alamahelat, kus mõlemad GPIO A ja C viiud viiakse madalaks. Vastassuunas laadides analoogselt kasutusel alamahel 1. laadimiseks alamahel 2. Lühistamiseks.



Joonis 8 tehislhase laadimis- ja mõõtmiskeem.

5.1.1 Tehislhase laadimine

Täiturit laetakse 1kHz PWM signaaliga, mille madala faasi asemel on väljundviigud muudetud sisendviikudeks ehk kõrge impedantsiga olekusse.. Kuna tehislhas sisaldab elektrolüüti, on vajalik jälgida elektrootide vahel olevat pinget. See ei tohiks tõusta üle 1.3V, kuna sellest pingest kõrgemaga hakkab tehislhas toimuma õhust imendunud vee elektrolüüs, mis toodab plahvatusohtlikku gaasi. Kasutatud tehislhasel on suur sisetakistus, mille tõttu pole laadimise ajal mõõdetav pinge samaväärne avatud ahela pingega. Kasutatud PWM laadimine võimaldab tehislhase elektrootide vahelist pinget mõõta madala faasi hetkel.

Tehislhase voolutihedusega arvestamiseks on kasutusel PWM täituvus muutmine. PWM täituvus kirjeldab suhet kõrge ja madala PWM perioodi vahel. Täituvuse muutmise on võimalik kontrollida tehislhast läbivat keskmist voolu. Laadimise ajal mõõdetakse shunti

läbivat voolu, millest arvutatakse valemiga $T_{\text{täituvus}} = I_{\text{kesk}} * \frac{R_{\text{shunt}}}{V_{\text{shunt}}}$ maksimaalne lubatud PWM täituvustegur, ning uuendatakse PWM täituvust.

5.1.2 Tehislihase mõõtmine

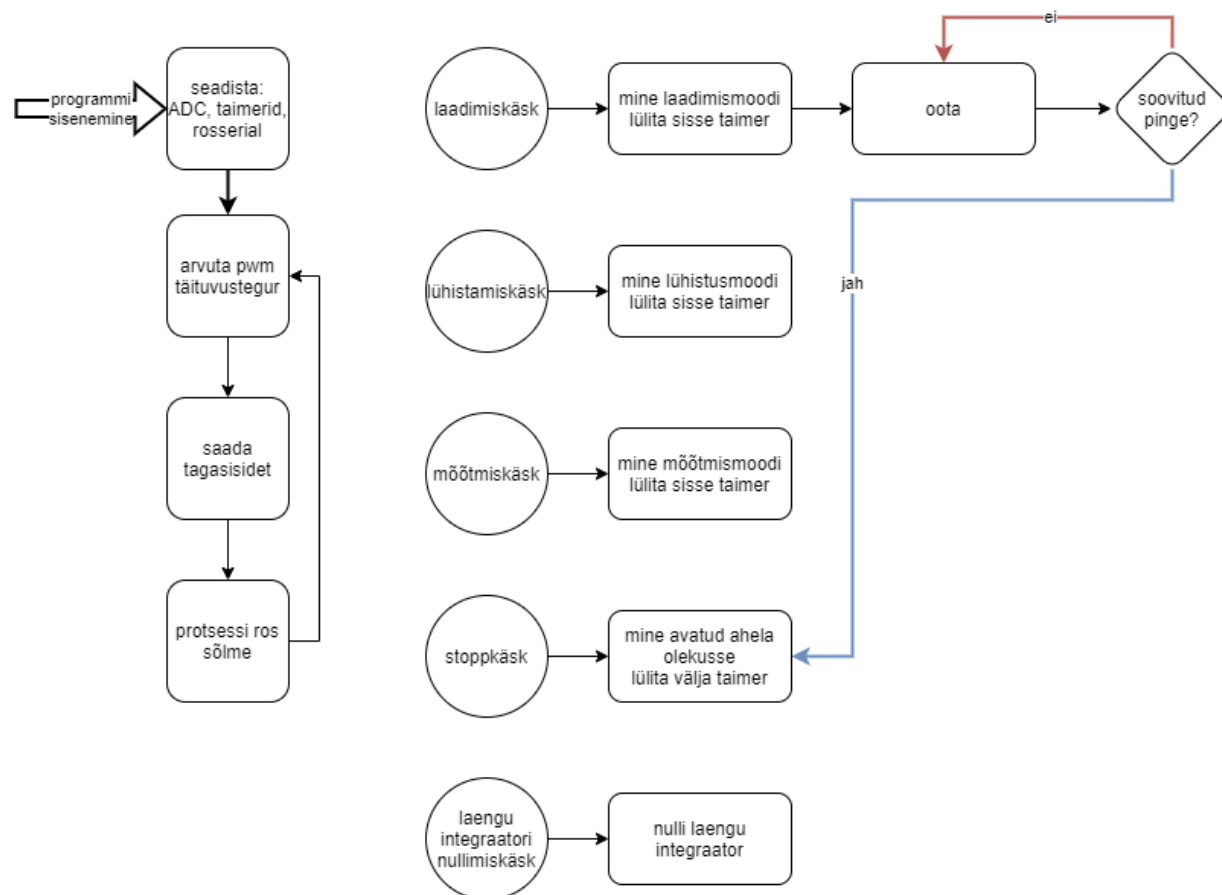
Tehislihase pinge mõõtmiseks on kasutusel kolm differentiaalmoodis ADCd. Ühepoolse ADC kasutamisel peaks tehislihase laadimise ajal maaga ühendatud pool jääma ühendatuks, muidu pole ühist maaga ühendust ja ühepoolse adc mõõdetud pinge ei oma tähendust. Differentisaal ADC teeb lahenduse kindlamaks, ei saa tekkida olukorda kus avatud ahela pinget ei saa mõõta.

Kuna ICL materjali mahtuvus sõltub selle sisend- ja väljundpinge sagedusest ning kasutusel on PWM laadimine, siis avatud ahela pinge ei sobi hästi suvalisel hetkel tehislihase laetuse taseme mõõtmiseks. Selle puudujäägi mõningaseks kompenseerimiseks integreeritakse shundilt mõõdetud laadimisvool, et leida tehislihasele laadimise käigus salvestunud laeng.

5.1.3 Täituri püsivara

Täituri kood on kirjutatud C++ keeles kasutades PlatformIO arenduskeskkonda. Joonisel 9 on kujutatud täituri püsivara üldine skeem. Programmikood kontrollib tehislihast võimaldades seda laadida, lühistada, mõõta avatud ahela pinget, teha avatud ahelaks ja nullida tehislihasele salvestunud laengu loendur. Programmis on kasutusel rosserial teek, mis võimaldab arduino koodis teha ROS sõlme. See lahendab andmete vahetuse mikrokontrolleri ja ROSi vahel, kuna rosserial haldab ise ühenduse loomist, taasloomist ja veatuvastust. Täituri ja draiveri vaheliseks suhtluseks on kasutusel rubriigid.

Joonisel 9 mainitud taimeri ülesanne on kontrollida GPIO A, B, C ja D viike vastavalt täituri olekule ja käivitada ADC lugemist. Käsud töötavad rubriikide sõnumite saamise tagasikutse meetoditega, neid kutsub välja rosserial teek. ROS sõlme protsessimine ajal teeb rosserial saadetakse välja tagasisidet ja võetakse vastu uut. Kuna täituril pole vaja draiveripoolset kiiret kontrolli, saadetakse tagasisidet maksimaalselt iga 100 uuenduse järel.



Joonis 9 tehislihas arduino programmikoodi loogikaskeem

5.1.4 Kasutatud tehislihas

Lõputöös kasutati 1.78mm x 7.06mm suurusega ICL tehislihasest tükki. Selle hoidmiseks sai tehtud spetsiaalne hoidja, mis koosnes kahest kullatud tükist, millest mõlema külge sai joodetud juhe ja hiljem liimitud neodüüm magnet.

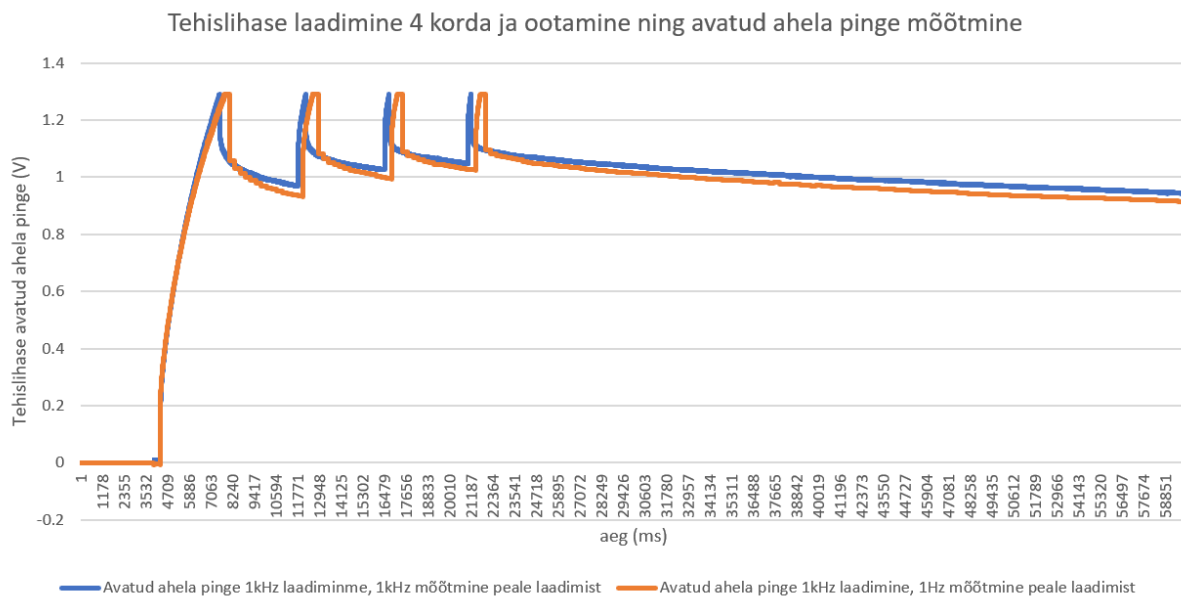
6 Tulemused ja arutus

Valmis ROS draiver materjali tasandil kohaneva pehmerobootika haaratsi jaoks. Sellega on võimalik kontrollida ICL tehislihasest haaratsit kasutades vastavat ROS tegumit. Materjali tasandil kohanemine, ilma lisasensoriteta, teeb võimalikuks teada haaratsi positsioon või jõudu. Sel põhjusel ei saanud sellise haaratsi draiverit teha samasuguseks haaratsi kohta absoluutset infot omavate draiveritega.

Tehtud draiveriga on võimalik muuta haaratsi lokaalse kujumuutuse potentsiaali vasakule, paremale ja neutraalsesse olekusse. Kasutajal on võimaldatud haaratsi draiveri tegumiga haaratsit kontrollida. See lahendus on sarnane haaratsi draiverile, mis täpse positsiooni asemel lubab haaratsit avada ja sulgeda. Neutraalsesse olekusse naasmine on tehilihase elektriliselt neutraalse positsiooni viimine, mida saab pidada sarnanesks robotiq haaratsi hädaolukorra avanemise võimekusega.

Üheks suuremaks erisuseks, võrreldes olemasolevate haaratsitega, sai kasutajale draiveri poolt võimaldatud haarde parandamise proovimise võimalus. Testhaaratsil pole teada mahtuvuse muutuse funktsioon sisend- ja väljundpinge suhtes, ega tehilihase pikaajalise laengu tühjenemise funktsiooni. Kuna tehilihase tühjenemist ei saa mõõta ja seda võidakse suvalises olekus arduino külge ühendada, siis pole integreeritud laengu summa usaldusväärne väljaspool ühte liigitust. Nendel põhjustel ei võimalda täitur tehilihast laadida täpsele laengu tasemele. Tehtud testhaaratsi puhul on kasutusel ka PWM laadimine. PWM laadimise lõppedes suureneb tehilihase mahtuvus ja avatud ahela pinge väheneb (joonis 10). Nii võiks täitur jääda väga pikaks laadima, kuna laadimise lõppedes avatud ahela pinge langeb ja võiks uuesti laadida. Kui tegum teatataks kasutajale, et soovitud liigutus on tehtud, aga samas täitur toimetab taustal edasi, siis on tekkinud vastuoluline olukord. Samas kui draiver jääks pikaks ajaks tegumit täitma, oleks see täiesti teistsugune teistest tegumipõhistest haaratsi draiveritest, mille puhul tegumil on kindel lõpp-punkt haarde saavutamiseks. Kui laadimise lõpul jääda tehilihase pinget hoidma 1.3V tasemel, on tehilihase mahtuvus pidevalt vähendatud ja haaret poleks võimalik kiirelt tugevdada, kuigi see võib võimalik olla. Mõni komplekssem haarats võib omada paremat kontrolli tehilihase üle, sel juhul haarde tugevdamist ei pruugigi vaja olla. Draiver haarde tugevdamist ei kohusta, ning hinnang on haaratsilt tulevate andmete põhjal tehtud. Lisaks on haaratsil võimalus laadimiskäsu peale ise otsustada mida tegema hakata.

Draiver kasutab haaratsiga suhtluseks ROS rubriike ja draiveris puuduvad kasutatud ICL haaratsiga seotud konstandid (v.a 1.3V piirang mis tuleb materjalist) s.t sellega on võimalik juhtida mõnda teist haaratsit, mis kasutab sama suhtlusliidest. ROS sõnumite kimp on eraldi draiveri kimbust ja seda on võimalik kasutada mõne teise pehmehaaratsi draiveri loomisel.



Joonis 10 tehislihase korduvalt laadimine ja mõõtmine.

6.1 Võrdlus tavahaaratsi draiveriga

Mõlema robotiq haaratsi draiveri puhul peab kasutaja saatma soovitud haaratsi positsiooni ja maksimaalse jõu. Danfoa draiveri puhul ka maksimaalse kiiruse. Lõputöös tehtud draiveri puhul on kasutusel liikumise suund ja maksimaalne kiirus. See sarnaneb robotiq haaratsi draiveritega, kui nende puhul saata minimaalne või maksimaalne positsioon täpse asemel, ning lasta jõusensoril haaratsi liikumine peatada. Pehmehaaratsi puhul maksimaalse jõu andmine, mida jõusensoriga kasutatakse, kohaneb materjal haaratava objekti ümber. Tagasiside on erinev - pehmehaaratsi puhul ei saa mõõta materjali kohanemist ilma mõne lisasensorita. Robotiq haaratsi draiverite puhul jõuab haarats kohe maksimaalse haardeni, pehmedraiveri puhul peab kasutaja haaret parandama kui ta teab, kuskilt draiveri väliselt saadud teadmisega, et seda on vaja teha.

Kokkuvõtteks - pehmeaaratsit on võimalik kontrollida sarnaselt aga mitte täpselt samamoodi tavahaaratsiga.

6.2 Testimine

Testimiseks anti draiverile 10 korda käsud järjestuses: vasakule, neutraalsesse ja paremale. Iga käsu vahel oodati 10 sekundit. Draiver täitis iga käsu kõigil kümnel korral. Liigutamise kujumuutuse hinnangu töötas ootuspäraselt s.t mida lähemal tehislühase maksimumpingeni, seda väiksem oli antud lokaalse kujumuutuse potentsiaal.

7 Kokkuvõte

Bakalaureusetöö eesmärgiks oli uurida kas on mõttekas ja võimalik kontrollida materjalitasandil muganevat täituri haaratsina ROSis. Selleks loodi loodi ROS draiver tehislihasest täituri kontrollimiseks ja lihtne ICL tehislihasest täitur draiveri testimiseks. Draiver võimaldab pehmerobotika haaratsit kontrollida mingilmääral sarnaselt tavarobotika haaratsile. Tagasisides puudub info haaratsi positsiooni kohta, mis tavarobotika haaratsite draiveritel on. Kasutajale võimaldatakse hinnang haarde muutmise mõttekuse kohta. Draiver kasutab ROSi parimaid praktikaid ning on modulaarne. Modulaarsuse tõttu on sellega ka võimalik kontrollida mõnda teist ICL pehmehaaratsit mis kasutab sama suhtlusliidest.

8 Viited

1. ALICI G. Softer is Harder: What Differentiates Soft Robotics from Hard Robotics? *MRS Adv.* 2018;3(28):1557–68.
2. Polygerinos P, Correll N, Morin SA, Mosadegh B, Onal CD, Petersen K, et al. Soft Robotics: Review of Fluid-Driven Intrinsically Soft Devices; Manufacturing, Sensing, Control, and Applications in Human-Robot Interaction. *Adv Eng Mater.* 2017;19(12):1700016.
3. Laschi C, Mazzolai B, Cianchetti M. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Sci Robot.* 6. detsember 2016;1(1):eaah3690.
4. Yang GZ, Bellingham J, Dupont PE, Fischer P, Floridi L, Full R, et al. The grand challenges of *Science Robotics*. *Sci Robot.* 2018;3(14):eaar7650.
5. Drotman D, Jadhav S, Karimi M, de Zonia P, Tolley MT. 3D printed soft actuators for a legged robot capable of navigating unstructured terrain. 2017 IEEE International Conference on Robotics and Automation (ICRA). 2017. lk 5532–8.
6. Mazzolai B, Tramacere F, Fiorello I, Margheri L. The Bio-Engineering Approach for Plant Investigations and Growing Robots. A Mini-Review. *Front Robot AI.* 2020;7:573014.
7. Giordano G, Carlotti M, Mazzolai B. A Perspective on Cephalopods Mimicry and Bioinspired Technologies toward Proprioceptive Autonomous Soft Robots. *Adv Mater Technol.* 2021;6(12):2100437.
8. Hines L, Petersen K, Lum GZ, Sitti M. Soft Actuators for Small-Scale Robotics. *Adv Mater.* 2017;29(13):1603483.
9. Kaasik F, Must I, Baranova I, Põldsalu I, Lust E, Johanson U, et al. Scalable fabrication of ionic and capacitive laminate actuators for soft robotics. *Sens Actuators B Chem.* 2017;246:154–63.
10. Must I, Rinne P, Krull F, Kaasik F, Johanson U, Aabloo A. Ionic Actuators as Manipulators for Microscopy. *Front Robot AI* [Internet]. 2019;6. Available at: <https://www.frontiersin.org/article/10.3389/frobt.2019.00140>
11. ROS: Why ROS? [Internet]. [tsiteeritud 26. märts 2022]. Available at: <https://www.ros.org/blog/why-ros/>
12. Tiemerding T, Fatikow S. Software for Small-scale Robotics: A Review. *Int J Autom Comput.* 1. oktoober 2018;15(5):515–24.
13. Nodes - ROS Wiki [Internet]. [tsiteeritud 16. mai 2022]. Available at: <http://wiki.ros.org/Nodes>
14. ROS/Concepts - ROS Wiki [Internet]. [tsiteeritud 9. mai 2022]. Available at: <http://wiki.ros.org/ROS/Concepts>
15. actionlib - ROS Wiki [Internet]. [tsiteeritud 9. mai 2022]. Available at: <http://wiki.ros.org/actionlib>
16. Topics - ROS Wiki [Internet]. [tsiteeritud 9. mai 2022]. Available at: <http://wiki.ros.org/Topics>
17. Services - ROS Wiki [Internet]. [tsiteeritud 9. mai 2022]. Available at: <http://wiki.ros.org/Services>
18. rospy/Overview/Services - ROS Wiki [Internet]. [tsiteeritud 17. mai 2022]. Available at: <http://wiki.ros.org/rospy/Overview/Services>
19. control_msgs [Internet]. *ros-controls*; 2022 [tsiteeritud 9. mai 2022]. Available at: https://github.com/ros-controls/control_msgs

20. Ordonez D. Danfoa/robotiq_2finger_grippers [Internet]. 2022 [tsiteeritud 9. mai 2022]. Available at: https://github.com/Danfoa/robotiq_2finger_grippers
21. Robotiq ROS drivers github [Internet]. ROS-Industrial; 2022 [tsiteeritud 9. mai 2022]. Available at: <https://github.com/ros-industrial/robotiq>
22. 2F-85 and 2F-140 Grippers [Internet]. Robotiq. [tsiteeritud 9. mai 2022]. Available at: <https://robotiq.com/products/2f85-140-adaptive-robot-gripper>
23. Robotiq. Product-sheet-Adaptive-Grippers-EN.
24. Triple 3D Printing Production With Collaborative Robots [Internet]. Robotiq. [tsiteeritud 17. mai 2022]. Available at: <https://robotiq.com/resource-center/case-studies/triple-3d-printing-production-with-collaborative-robots>
25. Qiu Y, Zhang X, Li H, Wang R. ROS-Based Control Implementation of an Soft Gripper with Force Feedback. Liu XJ, Nie Z, Yu J, Xie F, Song R, toimetajad. Intelligent Robotics and Applications. Cham: Springer International Publishing; 2021. lk 528–38.
26. E. Navas, R. Fernández, D. Sepúlveda, M. Armada, P. Gonzalez-de-Santos. Soft Gripper for Robotic Harvesting in Precision Agriculture Applications. 2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). 2021. lk 167–72.
27. McKenzie RM, Barraclough TW, Stokes AA. Integrating Soft Robotics with the Robot Operating System: A Hybrid Pick and Place Arm. Front Robot AI [Internet]. 2017;4. Available at: <https://www.frontiersin.org/article/10.3389/frobt.2017.00039>
28. Mirvakili SM, Hunter IW. Artificial Muscles: Mechanisms, Applications, and Challenges. Adv Mater. 2018;30(6):1704407.
29. Kruusamäe K, Punning A, Aabloo A. Electrical Model of a Carbon-Polymer Composite (CPC) Collision Detector. Sensors. 2012;12(2):1950–66.
30. Must I, Kaasik F, Põldsalu I, Mikhels L, Johanson U, Punning A, et al. Ionic and Capacitive Artificial Muscle for Biomimetic Soft Robotics. Adv Eng Mater. 2015;17(1):84–94.
31. AU - Rinne P, AU - Põldsalu I, AU - Ratask HK, AU - Kruusamäe K, AU - Johanson U, AU - Tamm T, et al. Fabrication of Carbon-Based Ionic Electromechanically Active Soft Actuators. J Vis Exp. 25. aprill 2020;(158):e61216.
32. actionlib: actionlib.simple_action_server.SimpleActionServer Class Reference [Internet]. [tsiteeritud 20. mai 2022]. Available at: https://docs.ros.org/en/api/actionlib/html/classactionlib_1_1simple__action__server_1_1SimpleActionServer.html
33. Atmel. SAM3X / SAM3A Series Atmel | SMART ARM-based MCU DATASHEET [Internet]. 2015. Available at: https://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a_datasheet.pdf
34. HC-06 Bluetooth Module [Internet]. Components101. [tsiteeritud 19. mai 2022]. Available at: <https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet>

Lisad

Lisa 1. PlatformIO arduino projekt tehislhasest haaratsi jaoks: icl_test_gripper-main.zip

→ Saadaval ka https://github.com/panihans/icl_test_gripper

Lisa 2. ROS draiveri kimbud: artificial_muscle_actuator_ros-main.zip

→ Saadaval ka https://github.com/panihans/artificial_muscle_actuator_ros

Lisa 3. Testhaaratsi elektriskeem: Elektriskeem.pdf

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Hans Pärtel Pani,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
“ROS draiver pehmerobootika haaratsile”,

mille juhendajad on Karl Kruusamäe ja Indrek Must,

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.

Hans Pärtel Pani

22.05.2022