Optimal control of a mobile manipulator for spraying and suckering tasks in viticulture

Vatavuk, Ivo

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ivo Vatavuk

OPTIMAL CONTROL OF A MOBILE MANIPULATOR FOR SPRAYING AND SUCKERING TASKS IN VITICULTURE

DOCTORAL THESIS

Zagreb, 2023

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ivo Vatavuk

OPTIMAL CONTROL OF A MOBILE MANIPULATOR FOR SPRAYING AND SUCKERING TASKS IN VITICULTURE

DOCTORAL THESIS

Supervisor: Professor Zdenko Kovačić, PhD

Zagreb, 2023

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ivo Vatavuk

OPTIMALNO UPRAVLJANJE MOBILNIM MANIPULATOROM ZA ZADATKE PRSKANJA I PLIJEVLJENJA U VINOGRADARSTVU

DOKTORSKI RAD

Mentor: prof. dr. sc. Zdenko Kovačić

Zagreb, 2023.

Doctoral thesis is written at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering.

Supervisor: Professor Zdenko Kovačić, PhD

Doctoral thesis has 95 pages

Dissertation No.: _____

About the Supervisor

Zdenko Kovačić (https://www.fer.unizg.hr/en/zdenko.kovacic) is a full professor at the University of Zagreb's Faculty of Electrical Engineering and Computing. He founded and has been leading the Laboratory for Robotics and Intelligent Control Systems (LARICS) since 1996. In 1990/91 he was a visiting researcher at IREX in the Bradley Department of Electrical Engineering at Virginia Polytechnic Institute and State University, Blacksburg, USA. From 2004 to 2008, he was the head of the Department of Automation and Computer Engineering. In 2013, he received the "Fran Bošnjaković" award from the University of Zagreb for his outstanding scientific contribution to the promotion and development of control systems, flexible automation and robotics. He was also awarded the "Josip Lončar" prize of the Faculty of Electrical Engineering and Computing in 2018. In 2022, the Croatian IEEE section honored him with the "Nikola Tesla" award for his exceptional professional and scientific contributions in the field of robotics and automation. He has been the principal investigator in over forty successfully completed international and Croatian R&D projects. He is currently leading three research and development projects in collaboration with industrial partners and is the coordinator and team leader for three ESIF funded scientific projects (EKOKOMVOZ, HEKTOR, ASAP). He also collaborates on several ongoing Horizon 2020 projects, including DATACROSS, AEROTWIN, ENCORE and AerialCore. He is involved in HorizonEurope projects such as AeroStream and Marble. He is also a participant and board member of COST Action CA19104, which promotes social inclusion through technology and empowerment (a-STEP). He has co-authored two scientific monographs published by international publishers Springer Verlag and Taylor Francis (CRC Press), the latter of which has also been published in Chinese. He has also authored one university textbook on robotics, manufacturing systems, and intelligent control. He has over 200 scientific publications to his credit, including book chapters, journal articles, and conference papers. He has actively participated in the organization of numerous international conferences, workshops, and academic seminars. He was a jury member for the Georges Giralt EuRobotics PhD Award (2016-2023) for the best European PhD thesis in robotics. He is an IEEE Senior Member. From 2010 to 2013 he was the president of the Croatian IEEE Robotics and Automation Section. From 2005 to 2010 he was the president of Croatian Robotic Association (which he also founded) and from 2011 to 2014 its vice president. From 2012 to 2015, he was elected president of the Croatian Robotic Alliance. He is also a founding member and part of the Presidium of KoREMA - Croatian Society for Communication, Computing, Electronics, Measurement and Control. He is a member of the editorial boards of several journals. Being passionate about popularization of science, he has participated in various events, such as science festivals at the Technical Museum in Zagreb, the Festival of Technical Culture, robotics schools organized by the Croatian Robotics Association, demonstration seminars for

high school teachers, and many others. From 2017 to 2023, he was the chairman of the committee for awarding the national prize for technical culture, Faust Vrančić. He is the holder of a patent. He received a gold medal for innovation at the 13th International Fair on Innovations in the Field of Agriculture, Food Industry and Agricultural Innovation, AGRO ARCA 2022.

O mentoru

Zdenko Kovačić (https://www.fer.unizg.hr/zdenko.kovacic) redoviti je profesor na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva. Osnivač je i voditelj Laboratorija za robotiku i inteligentne sustave upravljanja (LARICS) od 1996. godine. Proveo je 1990./91. kao gostujući istraživač IREX-a na Bradley Department of Electrical Engineering, the Virginia Polytechnic Institute and the State University, Blacksburg, SAD. Od 2004.-2008. bio je predstojnik Zavoda za automatiku i računalno inženjerstvo. Godine 2013. dobio je nagradu "Fran Bošnjaković" Sveučilišta u Zagrebu za posebno istaknuti znanstveni rad u promicanju i razvoju sustava upravljanja, fleksibilne automatizacije i robotike. Dobitnik je i nagrade Fakulteta elektrotehnike i računarstva "Josip Lončar" za 2018. U 2022. od Hrvatske sekcije IEEE dobio je nagradu "Nikola Tesla" za izniman stručni i znanstveni doprinos u tehničkom području robotike i automatike. Bio je glavni istraživač više od četrdeset uspješno završenih međunarodnih i hrvatskih R&D projekata. Trenutno je voditelj tri razvojno-istraživačka projekta s industrijskim partnerima, koordinator je i voditelj tima tri znanstvena projekta financirana od ESIF-a (EKOKOMVOZ, HEKTOR, ASAP) te suradnik na nekoliko tekućih projekata iz programa Obzor 2020, uključujući DATACROSS, AEROTWIN, ENCORE i AerialCore. Suradnik je na HorizonEurope projektima AeroStream i Marble. Također je sudionik i član upravnog odbora COST Action CA19104 Unapređenje socijalne uključenosti kroz tehnologiju i osnaživanje (a-STEP). Koautor je 2 znanstvene monografije međunarodnih izdavača Springer Verlag i Taylor Francis (CRC Press), potonja također objavljena na kineskom jeziku), 1 sveučilišnog udžbenika, sve u područjima robotike, proizvodnih sustava i inteligentnog upravljanja. Ima više od 200 znanstvenih publikacija (poglavlja u knjigama, članci u časopisima, radovi sa skupova). Aktivno je sudjelovao u organizaciji brojnih međunarodnih konferencija, radionica i akademskih seminara. Član je međunarodnog žirija za dodjelu nagrade Georges Giralt EuRobotics-a (2016.-2023.) za najbolji europski doktorat iz robotike. Član je IEEE-a (Senior Member). Od 2010. do 2013. bio je predsjednik hrvatske IEEE sekcije za robotiku i automatizaciju. Bio je predsjednik Hrvatskog robotičkog društva 2005.-2010. (ujedno i osnivač) i dopredsjednik (2011.-2014.). Od 2012. do 2015. bio je izabrani predsjednik Hrvatskog robotičkog saveza. Također je osnivač i član Predsjedništva KoREMA - Hrvatskog društva za komunikacije, računalstvo, elektroniku, mjerenje i upravljanje. Član je nekoliko uredništva časopisa. Aktivno je sudjelovao u popularizaciji znanosti sudjelujući u različitim manifestacijama kao što su Festivali znanosti (Tehnički muzej, Zagreb), Festival tehničke kulture, Škole robotike u organizaciji HDR-a, demonstracijski seminari za srednješkolske profesore i drugo. Od 2017. do 2023. predsjednik je Odbora za dodjelu državne nagrade tehničke kulture Faust Vrančić. Nositelj je jednog patenta. Dobitnik je zlatne medalje za inovaciju na 13. međunarodnom sajmu inovacija u poljoprivredi, prehrambenoj industriji i poljoprivrednoj mehanizaciji AGRO ARCA 2022.

Acknowledgements

First of all I would like to express my gratitude to my advisor professor Zdenko Kovačić, without whom I would not be where I am today, and would not be the person I am today. His support, guidance, encouragement and belief in myself is what made me even consider taking this step in my life, and without all of those things it could never have been successfully completed.

I would like to extend my gratitude to all of my colleagues at LARICS (Laboratory for Robotics and Intelligent Control Systems) for their selfless support, their friendship, and for confirming my life long suspicion that serious and challenging work is being done by people who know how to enjoy life, who refuse to take themselves too seriously, and who do their work with a dose of quality banter. It was a pleasure to have an opportunity to work with people who share my sense of humor, love for music, sailing and much more.

I would also like to express my deepest gratitude to my family and friends. Without their love, support, and encouragement throughout my entire life its hard to image where and who I would be today.

Abstract

This thesis deals with the application of optimization and optimal control for the execution of viticulture related tasks with a mobile manipulator. The focus is on viticultural procedures that require manipulation for their successful execution, namely viticultural spraying and suckering. The central challenge tackled in this thesis is the application of prioritized optimization and optimal control to resolve different kinds of redundancies that arise in the execution of mentioned tasks. Depending on the needs of the specific task, redundancy resolution is performed in various spaces: task space, joint velocity space and joint position space.

Viticultural spraying refers to the process of applying a protective spraying agent to grapevine plants. Two variants of the spraying task are considered: applying the spraying agent to the entire plant canopy and to a specific part of the plant. During the treatment of the entire canopy, referred to as continuous spraying, the position of the spraying agent is controlled by both the mobile vehicle and the robot arm. This introduces the first kind of redundancy addressed in this thesis, which is resolved predictively in the task space. A model predictive control algorithm is employed to anticipate changes in the reference trajectory of the spraying agent and optimize the coordinated control actions for both the mobile vehicle and the robot arm accordingly. The control of the robot arm itself for this task presents a second type of redundancy addressed within this thesis. The spraying nozzle used in this thesis is an axis-symmetric tool, and its orientation around its approach axis does not affect the application of the spraying agent. The task of controlling the spraying nozzle is a 3T2R task, where three translational and two rotational degrees of freedom impact task execution. Prioritized task space control is employed to address this redundancy, approaching the control of the spraying nozzle as a prioritized 3T2R task. Linear velocity of the spraying agent is prioritized over its angular velocity, with an additional priority fully constraining the optimization problem. The task of spraying a particular plant area, referred to as selective spraying, is tackled using a similar approach, but in the joint position rather than velocity space. A prioritized positional inverse kinematics solver is developed for this task, using iterative prioritized task space control to determine the joint positions that optimize a number of prioritized positional tasks.

The viticultural suckering task refers to the removal of shoots, buds, and suckers emerging at the lower part of the grapevine plant. Both the robotic tool developed for this task and the accompanying control method are presented. A brush-shaped robotic suckering tool employs the direct drive design philosophy, enabling precise and high-bandwidth torque measurements. This capability allows the developed tool to function as a tactile sensor. An experiment was conducted to investigate the correlation between the overlap of the brush bristles with an obstacle and the torque exerted by the tool. The established relationship between these values is employed to convert torque measurements into estimates of obstacle overlap, enabling the

intuitive design of an overlap controller. The developed brush-shaped tool also exhibits axis symmetry, reintroducing the same type of redundancy encountered in controlling the spraying nozzle. This redundancy is also addressed through the application of prioritized task space control. The suckering control method utilizes the torque-sensing capability of the developed tool for tactile exploration, maintaining physical contact with the plant trunk during the suckering process. This enables the robot arm to explore and map the trunk relying solely on tactile feedback, without the need to previously detect its shape.

There are three main contributions of this thesis:

- *Task space model predictive control based method for vineyard spraying with a mobile manipulator* (Chapter3)
- *A direct drive brush-shaped vine suckering robotic tool with torque sensing* (Chapter4)
- *Prioritized task space control based method for compliant vine suckering using a direct drive brush-shaped robotic tool (Chapter4)*

Keywords: Mobile Manipulation, Optimization and Optimal Control, Redundancy Resolution, Agricultural Robotics

Optimalno upravljanje mobilnim manipulatorom za zadatke prskanja i plijevljenja u vinogradarstvu

Ova doktorska disertacija bavi se primjenom optimizacije i optimalnog upravljanja za izvršavanje zadataka povezanih s vinogradarstvom mobilnim manipulatorom. Fokus je na vinogradarskim zadacima koji zahtijevaju manipulaciju za uspješnu izvedbu, prskanje i plijevljenje. Središnji izazov kojim se disertacija bavi je primjena prioritizirane optimizacije i optimalnog upravljanja za razrješavanje različitih redundancija koje su prisutne u izvođenju navedenih zadataka. Ovisno o potrebama pojedinog zadatka, razrješavanje redundancije se izvršava u različitim prostorima: prostoru zadatka, prostoru brzina zglobova te prostoru pozicija zglobova.

Vinogradarsko prskanje odnosi se na zadatak nanošenja zaštitnog sredstva za prskanje na biljke vinove loze. Dvije varijante zadatka prskanja su razmotrene: nanošenje sredstva na čitavu krošnju te na specifične dijelove biljke. Tijekom tretiranja čitave krošnje, koje se još naziva i kontinuirano prskanje, pozicija sredstva za prskanje upravljana je i mobilnim vozilom i robotskom rukom. Ovo rezultira prvom vrstom redundancije kojom se ova disertacija bavi, koja je razriješena prediktivno, u prostoru zadatka. Algoritam modelskog prediktivnog upravljanja korišten je za predviđanje promjena u referentnoj trajektoriji sredstva za prskanje te optimizaciju koordiniranih upravljačkih akcija za mobilno vozilo i robotsku ruku u skladu s predikcijom. Samo upravljanje robotskom rukom za navedeni zadatak predstavlja drugu vrstu redundancije kojom se ova disertacija bavi. Mlaznica korištena za prskanje je osno-simetričan alat te njena orijentacija oko vlastite osi prilaska ne utječe na nanošenje sredstva. Zadatak upravljanja mlaznice za prskanje je 3T2R zadatak, kod kojeg tri translacijska i dva rotacijska stupnja slobode utječu na izvršavanje zadatka. Prioritizirano upravljanje u prostoru zadatka korišteno je za razrješavanje navedene redundancije, promatrajući upravljanje mlaznice kao prioritizirani 3T2R zadatak. Translacijska brzina sredstva za prskanje prioritizirana je nad njegovom rotacijskom brzinom, uz dodatni prioritet koji u potpunosti ograničava optimizacijski problem. Zadatak prskanja određenog dijela biljke, koji se još naziva i selektivno prskanje, riješen je na sličan način no u prostoru pozicija umjesto brzina zglobova. Algoritam prioritizirane pozicijske inverzne kinematike razvijen je za ovu svrhu, koristeći iterativno prioritizirano upravljanje u prostoru zadatka za određivanje pozicija zglobova koje optimiraju više prioritiziranih pozicijskih zadataka.

Zadatak vinogradarskog plijevljenja odnosi se u ovom kontekstu na zadatak uklanjanja izdanaka i pupova koji se pojavljuju na donjem dijelu biljke vinove loze. Prezentirani su robotski alat razvijen za ovu svrhu te odgovarajuća upravljačka metoda. Robotski alat za plijevljenje u obliku četke razvijen je u skladu s dizajnom izravnog pogona, što omogućava precizno mjerenje momenta s visokom frekvencijom. Navedena sposobnost omogućava korištenje razvijenog alata kao senzora dodira. Proveden je eksperiment za evaluaciju odnosa između prekla-

panja čekinja četke i prepreke te momenta proizvedenog od strane alata. Uspostavljeni odnos između tih vrijednosti koristi se za pretvaranje mjerenja momenta u estimaciju preklapanja, što omogućava intuitivan dizajn regulatora preklapanja. Razvijeni alat u obliku četke također je osno-simetričan alat, što rezultira istim tipom redundancije kao kod upravljanja mlaznicom. Ova redundancija također je razriješena korištenjem prioritiziranog upravljanja u prostoru alata. Upravljačka metoda za plijevljenje koristi sposobnost mjerenja momenta razvijenog alata za istraživanje na temelju dodira, održavajući kontakt s deblom biljke vinove loze tijekom procesa plijevljenja. Ovo omogućava robotskoj ruci da istraži i mapira deblo oslanjajući se isključivo na taktilnu povratnu informaciju, bez potrebe za prethodnom detekcijom njegovog oblika.

Disertacija je podijeljena u pet poglavlja.

U prvom poglavlju dan je uvod u istraživanje opisano u ovoj disertaciji te u šire istraživačko područje poljoprivredne robotike. Predstavljeni su motivacija i ciljevi ovog istraživačkog područja, uz diskusiju o povijesti poljoprivrede te značaju tehnološke inovacije u poljoprivredi. Također, provedena je diskusija o benefitima koje nove tehnologije imaju potencijal uvesti u različite poljoprivredne procedure, gdje korištenje robotske tehnologije može ponuditi dosad neviđenu razinu autonomije i preciznosti. Objašnjeni su motivacija i ciljevi projekta HEKTOR (Heterogeni autonomni robotski sustav u vinogradarstvu i marikulturi) u sklopu kojeg je istraživanje predstavljeno u disertaciji provedeno. Predstavljen je mobilni manipulator korišten u disertaciji uz kratku argumentaciju izbora njegovih komponenti. Naglasak je na mogućnost ovakvog sustava da izvršava zadatke koji zahtijevaju manipulaciju njegove okoline, zauzimajući aktivnu ulogu u poljoprivrednim procedurama. Konkretne procedure razmatrane u disertaciji su vinogradarsko prskanje i plijevljenje. Predstavljeni su istraživački ciljevi u kontekstu mobilne manipulacije za izvršavanje navedenih procedura. Zadatak vinogradarskog prskanja podijeljen je na kontinuirano i selektivno prskanje, dok su za zadatak plijevljenja razmotrene teme dizajna robotskog alata te razvoja upravljačke metode.

U drugom poglavlju predstavljeni su i proučeni temeljni teoretski principi korišteni za razvoj predloženih rješenja, potrebni za potpuno razumijevanje disertacije. Uvedena je tema redundancije u kontekstu upravljanja robotskim sustavima, kao i različiti tipovi redundancije koji se susreću u praksi. Intrinzična redundancija odnosi se na zadatak upravljanja čitavom pozom nekog članka u kinematičkom lancu, dok se funkcionalna redundancija odnosi na upravljanje jednog dijela te poze. Oba alata korištena u disertaciji su osno-simetrična, što uvodi funkcionalnu redundanciju u problem upravljanja robotskom rukom. Osno-simetričan alat zahtjeva upravljanje čitavom trodimenzionalnom pozicijom alata te njegovom dvodimenzionalnom orijentacijom, zanemarujući orijentaciju oko same osi simetrije. Redundancija te njeno razrješavanje usko su povezani uz teme kinematike i upravljanja u prostoru zadatka. Objašnjena je uloga Jacobian matrice u kinematici te upravljanju u prostoru zadatka. Provedena je diskusija

o uvjetima u kojima je robotski sustav redundantan te o povezanosti redundancije i svojstava Jacobian matrice. Predstavljen je problem pozicijske kinematike s objašnjenjem numeričkih iterativnih metoda za pozicijsku inverznu kinematiku. Zatim, dan je uvod u matematičko područje optimizacije. Definiran je generalni optimizacijski problem te su objašnjeni pojmovi kriterijske funkcije, optimizacijske varijable te ograničenja. Kratko je objašnjena klasifikacija optimizacijskih problema s obzirom na svojstva kriterijske funkcije i ograničenja. Pobliže je proučena klasa kvadratičnog programiranja, okarakterizirana optimizacijskim problemom s kriterijskom funkcijom kvadratičnog oblika uz afina ograničenja. Ovakav optimizacijski problem pogodan je za upravljanje robotskom rukom u prostoru zadatka. Također, predstavljena je i tema prioritizirane optimizacije kao alternative korištenju parametara težina za multikriterijsku optimizaciju. Objašnjen je algoritam za prioritizirano kvadratično programiranje bez i s ograničenjima te je provedena diskusija o korištenju istih za razrješavanje redundancije kod upravljanja u prostoru zadatka. Posljednja tema predstavljena u ovom poglavlju je modelsko prediktivno upravljanje. Modelsko prediktivno upravljanje metoda je optimalnog upravljanja koja koristi model sustava za predikciju njegova ponašanja kroz određeni vremenski horizont. U slučaju linearnog sustava, modelsko prediktivno upravljanje također se može riješiti optimizacijom kvadratičnog programa. U ovoj disertaciji, modelsko prediktivno upravljanje se koristi za praćenje reference koja se mijenja u vremenu. Predstavljeni su različiti oblici kriterijskih funkcija te ograničenja koja se u ovom slučaju mogu koristiti za modelsko prediktivno upravljanje.

U trećem poglavlju opisana su predložena rješenja za izvršavanje zadatka vinogradarskog prskanja mobilnim manipulatorom. Poglavlje započinje definicijom problema mobilne manipulacije za kontinuirano i selektivno prskanje, odnosno, za nanošenje tekućeg zaštitnog sredstva na čitavu krošnju te na specifične dijelove biljke vinove loze. Provedena je diskusija o različitim načinima kako se ovi zadatci obično izvršavaju u vinogradima. Dana je argumentacija za izvršavanje navedenih procedura mobilnim manipulatorom te za izbor alata. Kombinacija robotske tehnologije i alata za ručno prskanje nudi mogućnost autonomnog izvođenja zadatka uz visoku preciznost nanošenja zaštitnog sredstva na biljku. Zatim, predstavljen je pregled područja robotskog prskanja u poljoprivredi, planiranja trajektorije za mobilnu manipulaciju, razrješavanja funkcionalnih redundancija te povezanih tema. Predstavljena je upravljačka metoda za kontinuirano prskanje, kao kombinacija modelsko prediktivnog upravljanja u prostoru zadatka te prioritiziranog upravljanja u prostoru zadatka. Metoda za kontinuirano prskanje mobilnim manipulatorom koristi opis oblika krošnje biljke za generiranje referentne trajektorije zaštitnog sredstva, omogućavajući pokrivenost krošnje bez ekscesivnog trošenja sredstva. Upravljačke akcije u prostoru zadatka za mobilno vozilo te robotski manipulator dobivene su algoritmom modelsko prediktivnog upravljanja za praćenje reference promjenjive u vremenu. Opisane su kriterijska funkcija te funkcije ograničenja korištene za odgovarajući optimizacijski problem.

Provedena je diskusija o utjecaju parametara kriterijske funkcije na ponašanje sustava te o izboru parametara za postizanje željenog ponašanja te željenog načina razrješenja redundancije. Upravljačke akcije robotskog manipulatora dobivene ovim algoritmom transformirane su iz prostora zadatka u prostor zglobova korištenjem prioritizirane optimizacije. Argumentirana je prioritizacija linearne nad kutnom brzinom upravljanog koordinatnog sustava te korištenje konačnog prioriteta pozicija pojedinih zglobova. Prezentirani su rezultati metode kontinuiranog prskanja na različitim primjerima, uz korištenje različitih ograničenja na prioritizirani optimizacijski problem. Različita ograničenja utječu na ponašanje sustava, odnosno na način razrješenja redundancije prisutne u upravljačkom problemu. Prezentiran je algoritam prioritizirane pozicijske inverzne kinematike korišten za zadatak selektivnog prskanja. Algoritam je temeljen na iterativnom prioritiziranom upravljanju u prostoru zadatka. Prezentirani su i različiti parametri korišteni u algoritmu, čiji izbor utječe na vrijeme izvođenja te preciznost rješenja. Metoda selektivnog prskanja koristi slične prioritete kao ona kontinuiranog prskanja, prioritizirajući poziciju nad orijentacijom upravljanog koordinatnog sustava. Provedena je diskusija o različitim ponašanjima kojima prezentirani algoritam rezultira ovisno o zadanim vrijednostima pojedinih prioriteta i mogućnosti ostvarivanja ovih vrijednosti.

U četvrtom poglavlju opisana su predložena rješenja za izvršavanje zadatka vinogradarskog plijevljenja mobilnim manipulatorom. Kao što je već spomenuto, plijevljenje se u ovom kontekstu odnosi na proceduru uklanjanja izdanaka i pupova koji se pojavljuju na deblu biljke vinove loze. Poglavlje započinje definicijom problema te raspravom o različitim načinima kako se ovaj zadatak obično izvršava, uz predstavljanje alata koji se obično koriste za ručno i mehanizirano izvršavanje navedenog zadatka. Argumentiran je izbor robotskog alata uz raspravu o potencijalnim benefitima korištenja robotske tehnologije za izvršavanje zadatka plijevljenja. Predstavljen je pregled područja robotskog plijevljenja, robotskih alata s izravnim pogonom, taktilnog istraživanja prostora, taktilnog mapiranja i povezanih tema. Prezentiran je dizajn robotskog alata u obliku četke korištenog za ovaj zadatak, temeljenog na izravnom pogonu. Izravni pogon u ovom kontekstu označava manjak mehaničkog prijenosa između rotora i vanjske osovine aktuatora. Ovakav dizajn aktuatora omogućava precizno mjerenje proizvedenog momenta uz visoku frekvenciju mjerenja, korištenjem mjerenja struje. Predloženi robotski alat koristi navedeno svojstvo za ostvarivanje taktilne povratne veze, s ciljem izvršavanja zadatka praćenja oblika debla biljke vinove loze bez korištenja vizualne povratne veze. Objašnjen je eksperiment evaluacije odnosa između preklapanja čekinja četke i prepreke te momenta proizvedenog od strane alata. Upravljačka metoda koja omogućava robotskoj ruci praćenje oblika debla temeljena je na upravljanju navedenog preklapanja korištenjem mjerenja momenta kao povratne veze. Objašnjen je automat stanja korišten za ostvarivanje željenog ponašanja obilaska debla biljke vinove loze. Prioritizirano upravljanje u prostoru zadatka korišteno je za upravljanje robotskom rukom temeljeno na taktilnoj povratnoj vezi. Definiran je korišteni prostor zadatka, kao i različiti ko-

ordinatni sustavi korišteni za upravljanje. Predstavljeni su prioriteti korišteni za razrješavanje redundancije, uz njihovu argumentaciju. Objašnjena je metoda korištena za taktilno mapiranje debla biljke, uz predloženi vjerojatnosni model senzora. Taktilno mapiranje temelji se na izgradnji više dvodimenzionalnih karata prostora, koje su kombinirane za dobivanje trodimenzionalnog modela debla biljke. Poglavlje je zaključeno diskusijom o provedenim eksperimentima te postignutim rezultatima.

U petom poglavlju dan je zaključak ove disertacije.

Ova disertacija sadrži tri glavna znanstvena doprinosa:

- *Metoda za prskanje vinograda mobilnim manipulatorom temeljena na modelskom prediktivnom upravljanju u prostoru zadatka* (Poglavlje3)
- *Robotski alat za plijevljenje vinove loze u obliku četke s izravnim pogonom i mjerenjem momenta* (Poglavlje4)
- *Metoda temeljena na prioritiziranom upravljanju u prostoru zadatka za podatno plijevljenje vinove loze korištenjem robotskog alata u obliku četke s izravnim pogonom (Poglavlje4)*

Ključne riječi: mobilna manipulacija, optimizacija i optimalno upravljanje, razrješenje redundantnosti, poljoprivredna robotika

Contents

CHAPTER 1

Introduction

1.1Motivation

The research presented within this thesis is a part of an exciting and rapidly growing research field of **agricultural robotics**. In recent years, the research community has devoted continuous efforts to the development and utilization of robotic systems in agriculture, aimed at both automating and enhancing various agricultural tasks. In order to gain a better understanding of the underlying motivation driving this research, it is worthwhile to take a glimpse into the history of agriculture.

Agriculture played a fundamental role in the birth of civilizations. The transition from a nomadic, hunter-gatherer lifestyle to settled agricultural communities was a historical shift in human history whose significance cannot be overstated. The advent of agriculture laid the foundation for the development of societies, cultures and, consequently, the development of technology. Driven, to a large extent, by technology, agriculture has evolved through several distinct stages, known as "agricultural revolutions":

Stage 1:First Agricultural Revolution (circa 10,000 BCE - 4,000 BCE): The first agricultural revolution marked the transition from a nomadic hunter-gatherer lifestyle to settled agricultural communities. Early humans began cultivating crops and domesticating animals, leading to the establishment of agricultural practices as a foundation for civilization. During this stage, basic agricultural practices were developed, along with the primitive hand-held tools used to assist in their execution [1].

Stage 2:Second Agricultural Revolution (17th - 19th centuries): Also known as the *British Agricultural Revolution*. This time period witnessed significant advancements in farming technology and techniques. Innovations such as crop rotation, improved

tools, and the enclosure movement in Europe led to increased agricultural productivity, which fueled population growth. Already during this period, interest arose for developing machines to help "automate" certain agricultural procedures. In 1701 (270 years before recording "Aqualung"!) Jethro Tull, an Oxford scholar and a rich landowner, developed the first seed drill that became a widely adopted innovation, used to uniformly and economically sow seeds on prepared land [1].

Stage 3:Third Agricultural Revolution (mid-20th century): Also known as the *Green Revolution*. This period was characterized by the introduction of high-yield crop varieties and modern agricultural practices. Synthetic fertilizers, pesticides, and advanced breeding techniques were employed to achieve higher crop yields. Except for maximizing crop yield, new breeding techniques were used to produce specific varieties of particular crops that could lend themselves to a larger degree of mechanization [1]. These techniques proved to be a profoundly effective in addressing the problem of global hunger. However, concerns arose regarding environmental sustainability and chemical dependence.

Stage 4:Digital Agricultural Revolution (ongoing): Also known as *Precision Agriculture*. This ongoing movement in agriculture involves the integration of technologies such as data analytics, artificial intelligence, robotics and large-scale automation into farming practices. Advanced machinery and sensor networks enable optimized planting, irrigation, and pest management, aiming for increased productivity and reduced environmental impact.

The common thread uniting all the highlighted advancements in agricultural production is their shared dependence on technological progress. From the primitive tools employed during the inception of agriculture, to Jethro Tull's pioneering crop planting machine, the evolution of fertilizers and pesticides, and the continuous refinement of modern machinery, technology has been the driving force behind these transformative shifts.

Innovations based on robotics are arguably the next big step in the field of agriculture, offering a potential solution to a variety of contemporary challenges, such as increasing global demand for food production and labor shortages [2]. The level of precision, autonomy, and modularity robotic systems offer to introduce is unprecedented in agricultural machinery. Robotic agricultural systems can perform acquisition of large amounts of data on the state of the crops, along with on-board processing of that data. Various tasks that require physical manipulation can be carried out precisely, efficiently, and more frequently with robotic systems than possible with manual labour. Perhaps the potentially most intriguing is the ability of a single robotic system to perform all of these tasks, and more. These agricultural robotic systems have the potential to offer a high level of modularity in terms of the tools they could use and the tasks they could perform. A single robot with sufficient capabilities for manipulation, sensing and

locomotion, could have a potential to be used for the large portion agricultural procedures that are performed manually to this day.

1.1.1HEKTOR Project

The research presented in this thesis was carried out as a part of the HEKTOR project [3,4], which aims to introduce heterogeneous autonomous robotic systems in agricultural fields of mariculture and viticulture. A heterogeneous robotic system refers to a collection of robots that are diverse in terms of their physical characteristics, functionalities, or capabilities. The heterogeneity of the robot agents offers the ability to leverage the strengths of different robot types to tackle complex tasks more efficiently and effectively.

This thesis focuses on the viticultural part of the HEKTOR project, where a heterogeneous robotic system comprising of an unmanned aerial vehicle and a ground mobile manipulator is employed for various viticultural tasks (Fig.1.1).

Figure 1.1: A visualization of the heterogeneous robotic system employed in a vineyard as a part of the HEKTOR project, consisting of an unmanned aerial vehicle and an autonomous mobile manipulator.

The aerial vehicle has a fairly unconstrained movement space, and it can traverse the entire vineyard with ease in a short amount of time. On the other hand, navigation of the ground vehicle presents a challenge due to the semi-structured nature of the vineyard environment. The obstacle-free area between the rows of grapevines is typically narrow, significantly restricting the movement of the vehicle. The strength of the ground mobile manipulator lies in its ability to manipulate its environment, allowing it to participate more actively in agricultural procedures, which is the focus of this thesis.

Figure 1.2: A mobile manipulator used in the HEKTOR project and throughout this thesis. It features a custom flipper-tracked vehicle as its mobile base, mounted with a Kinova Gen3 robot arm for the execution of manipulation tasks. Copyright [8]CC BY 4.0.

Some of the research objectives of the HEKTOR project are directed towards the navigation and localization capabilities of a mobile manipulator within the vineyard environment. The research conducted in this area addresses challenges such as autonomous navigation [5], localization and positioning of the mobile manipulator next to the grapevine trunks [6] and estimation of energy consumption during the execution of these tasks [7]. On the other hand, the central emphasis of this thesis is on control algorithms that employ the ability of such a robot to manipulate its environment for specific viticultural procedures.

The HEKTOR project focuses on vineyards situated on karst terrain, a characteristic feature of the Croatian islands and the broader Mediterranean region. The unique topography of such terrains, characterized by steep inclines and reduced accessibility, poses a significant challenge for conventional agricultural machinery. In these kinds of vineyards, a substantial portion of work is still carried out manually, a practice that is both time-consuming and physically demanding.

The mobile manipulator used in the HEKTOR project was designed considering two objectives: being able to traverse such difficult, high-slope terrain, and its manipulability (Fig.1.2). The Kinova Gen3 robot arm has 7 actuated degrees of freedom, making it *intrinsically* redundant, discussed in detail in section2.1. A recurring theme in this thesis is the exploitation of redundancy exhibited by the mobile manipulator to optimize execution of different tasks.

1.2Research Objectives

The central emphasis of this thesis is on the developed control algorithms and methodologies that allow the mobile manipulator to perform viticultural tasks, utilizing optimization and optimal control. As the title suggests, this thesis focuses on two of the viticultural procedures that require manipulation from the robot: spraying and suckering. Different optimization based methods are utilized for different subtasks that arise in the execution of these procedures, with a shared objective of redundancy resolution. This chapter functions as a broad introduction to the research objectives, and a detailed description of method-specific objectives is provided in chapters dedicated to the individual tasks.

1.2.1Viticultural Spraying

Viticultural spraying involves the application of a protective spraying agent onto the leaves and fruits of grapevine plants. There are two widely used contemporary approaches to viticultural spraying:

1. *Manual spraying:* This approach involves human workers using handheld sprayers to apply the spraying agent directly to the leaves and fruits of the grapevine plants. While this allows for targeted application of spraying agent, it can be labor-intensive and time-consuming, especially in larger vineyards.
2. *Mechanical spraying:* Widely available mechanization for viticultural spraying comes in the form of specialized equipment such as atomizers or sprayers mounted on tractors or other human-operated vehicles. These machines disperse large amounts of the spraying agent through nozzles, covering a larger area much faster compared to manual spraying. Although this method is more time-efficient it does not allow for any targeted application of the spraying agent and poses an environmental sustainability concern.

One of the objectives of the presented research is the development of a methodology for robotic spraying, combining the best of the two worlds. The robot arm is equipped with a spraying pole for manual spraying, maintaining the advantages of manual spraying while capitalizing on the improved efficiency and precision provided by robotic technology. Two separate spraying applications are considered:

1. **Continuous spraying:** Referring to the problem of spraying the entire canopy of the grapevine plant. The term *continuous* refers to the continuous movement of the spray nozzle needed to cover the entire canopy area.
2. **Selective spraying:** Robotic technology introduces the potential of precise selective application of the spraying agent to specific parts of the plant, for example a single leaf or a single fruit cluster.

During the continuous spraying, both the robot arm and the mobile base control the position

of the spraying nozzle, introducing a first kind of redundancy discussed in this thesis. Another form of redundancy is present in both of the spraying task variants, posed by the fact that the spraying nozzle is axis-symmetric. The orientation of an axis-symmetric spraying nozzle around its approach axis has no influence on the application of the spraying agent. A common control objective is resolving the mentioned redundancies in an optimal way, even though different algorithms are used for their resolution, discussed in detail in later chapters.

1.2.2Viticultural Suckering

Suckering refers to the procedure of removing unwanted growth from the base of grapevines. This growth (known as shoots, buds or suckers) emerges at the trunk of the plant, and competes with the main, fruit-bearing vines for nutrients and water, potentially affecting the quality of the plant's yield. The objective driving the development of suckering methodologies presented in this thesis is to combine the elements of both manual and currently available mechanized approaches to the task with precision and efficiency of robotic technology, explained in more detail in later chapters.

The research conducted for the suckering task is broken down into two distinct parts:

1. **Suckering tool:** A novel brush-shaped robotic suckering tool is developed, suitable for its use as the end-effector of the robot arm. The design of the suckering tool draws inspiration from both contemporary mechanization and hand-held tools used for the task. One of the main design objectives for the tool is its ability to measure the torque exerted onto the plant, allowing it to be used as a tactile sensor.
2. **Suckering control method:** The suckering tool's torque sensing ability is used to generate the feedback for the tactile exploration control method, allowing the manipulator to perform the suckering task without the need to detect the shape of the trunk. Another objective for the control method is to estimate the shape of the trunk using tactile mapping. The trunk shape estimate has the potential to be reused for future executions of the task, improving its execution time.

The developed suckering tool is shaped as an axis-symmetric circular brush. Even though this tool and the nozzle used for the spraying task have completely different functionalities and capabilities, their shared property of axis-symmetry introduces the same kind of redundancy to the control problem. One of the control objectives for the suckering control method is to resolve this redundancy in an optimal way.

1.3Thesis Outline

This thesis is organized in five chapters, as follows:

Chapter2: This chapter provides a comprehensive introduction to the fundamental prerequisites essential for a clear understanding of the rest of the thesis. It explores the topics of redundancy, task space control, inverse kinematics, mathematical optimization, prioritized optimization, and model predictive control.

Chapter3: This chapter focuses on the algorithms and methodologies developed for viticultural spraying. The control of a mobile manipulator for both continuous and selective spraying task variants is explored. A model predictive control algorithm utilized for resolving the redundancy between the robot arm and the mobile vehicle during continuous spraying is presented and discussed in detail. A control method for the robot arm during continuous spraying is presented, handling the task as a prioritized task space control problem. A task of selective spraying is approached as a prioritized positional inverse kinematics problem. An algorithm for prioritized positional inverse kinematics and an implementation of a corresponding solver are presented. The implications of the algorithm and the performance of the solver are discussed.

Chapter4: Within this chapter, the algorithms and methodologies developed for the task of viticultural suckering with a mobile manipulator are presented. The design of the presented suckering tool is discussed, along with its torque sensing, signal filtering and obstacle estimation capabilities. The control algorithm developed for the task of suckering is presented, based on prioritized task space control. The tactile sensing capability of the presented suckering tool and its utilization to traverse a trunk of the grapevine plant without detecting its shape is discussed. The algorithm for tactile mapping of the trunk shape is presented, used to estimate the shape of the trunk traversed by the suckering tool.

Chapter5: Finally, the thesis is finished with a conclusion of the presented work.

CHAPTER 2

Fundamentals

2.1 Redundancy

One of the focus points of this thesis is the application of optimization and optimal control techniques to address various forms of redundancies that arise during the execution of viticulture-related tasks. Redundancy occurs when a robot system has more actuated degrees of freedom than those strictly required to execute a given task [9]. Generally, this means a commanded task can be achieved in multiple ways by the robot system. It is important to note that redundancy itself is not an inherent characteristic of the robot system, but rather a consequence of the specific task being performed.

Controlling the pose of the end-effector is well-established as an inherent robotic task. The *pose* of a rigid body is a collective term for its position and orientation, fully describing its configuration in three-dimensional space. Robot systems that possess more than six degrees of freedom can generally achieve a single end-effector pose in multiple ways, and are commonly referred to as *intrinsically* redundant [10].

Functional redundancy represents a type of redundancy encountered when the control objective does not require controlled motion of the entire pose of the end-effector [11]. A commonly encountered example of functional redundancy is the control of axis-symmetric tools. Various robotic applications are executed with tools symmetric around their approach axis: drilling, milling, welding, paint spraying etc. As a consequence of the axis-symmetry, orientation of these tools around their approach axis has no impact on task execution. For example, in robotic drilling, both the position and the orientation of the drill bit are important for successful task execution, but the orientation of the drill bit around its approach axis is irrelevant. Thus, the task of robotic drilling requires controlled motion of all three components of the drilling tool's

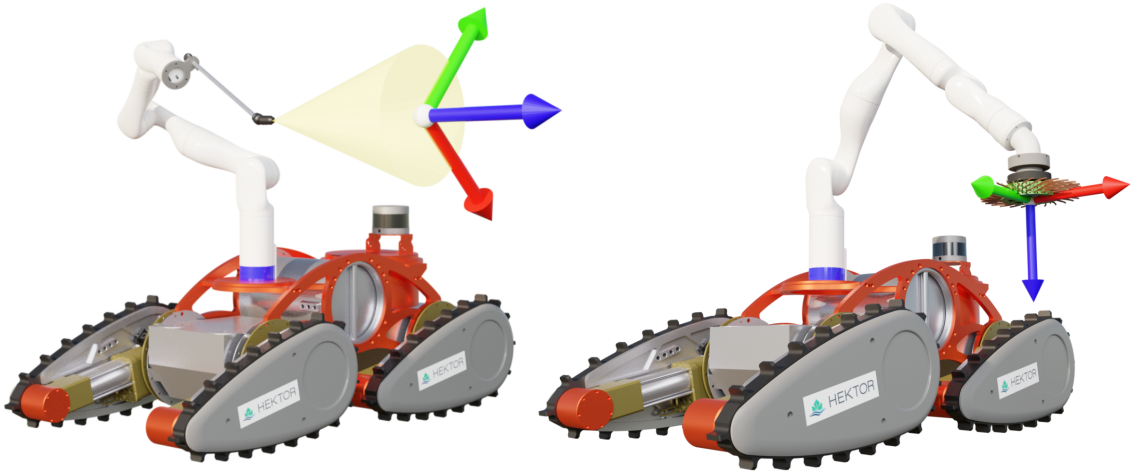

Figure 2.1: Robotic tools used for robotic viticultural spraying and suckering in this thesis are both examples of axis-symmetric tools. As a consequence, both of the tasks exhibit *functional* redundancy.

translation, and only two components of its orientation. These type of tasks are referred to as 3T2R tasks or pointing tasks in literature [10].

Viticultural tasks explored in this thesis, robotic spraying and suckering, are both performed with axis-symmetric tools: an axis-symmetric spray nozzle, and an axis-symmetric brush-shaped suckering tool (Fig.2.1). As already mentioned, the robot arm Kinova Gen3 used in this thesis has 7 actuated degrees of freedom, making the control of the specified tasks a redundant control problem.

The importance of redundancy lies in the flexibility and adaptability provided by a redundant robot system. Among all the ways a redundant system can achieve its primary task, some might be preferred over others. However, this enhanced flexibility comes at the cost of additional complexity, presenting a control challenge. In addition to satisfying the primary task for which the system exhibits redundancy, additional criteria have to be considered to select a single control strategy.

Redundancy resolution techniques play a crucial role in harnessing the full potential of redundant robot systems. These methods aim to find a suitable solution that satisfies the primary task while considering additional criteria and respecting various constraints. Additional criteria can be used to incorporate various secondary tasks, such as singularity avoidance, obstacle avoidance, etc. Various approaches to redundancy resolution exist, ranging from analytical methods to numerical techniques. These techniques consider factors like kinematic modeling, inverse kinematics, optimization algorithms, and artificial intelligence to determine the most suitable joint configuration for a given task. Redundancy resolution methodologies explored in this thesis are based on prioritized optimization and optimal control.

2.2Instantaneous Kinematics and Task Space Control

A robot system is typically mathematically described as a set of rigid bodies actuated by joints. The state and motion of a robot system with N_J actuated joints can be fully described in the *joint space*, employing the joint position vector $\boldsymbol{q} \in \mathbb{R}^{N_J}$ and its derivatives. Although the joint space fully describes the motion of the system, the desired system motion is usually more conveniently described in some other space, for example the end-effector space.

2.2.1Instantaneous Kinematics

Kinematics are a fundamental problem in robotics, studying motion of bodies in a robot system, while disregarding the forces/torques that cause that motion [12]. The primary objective of kinematics is to study the geometrical relationship between joint motion and the movement of individual links within the robot system. Kinematics, in a broader sense of the term, address general motion of the robot system, incorporating its velocities, accelerations, and all higher-order derivatives [12]. The term *instantaneous kinematics* refers to the kinematics specifically studying the velocities of the robot system, while the kinematics dealing with its positions will be referred to as *positional kinematics*. The two fundamental kinematics problems are *forward* and *inverse* kinematics.

The forward instantaneous kinematics problem is to determine the rate of change in the pose of the end-effector, given the positions and the velocities of all the joints. This problem has a unique solution:

$$\boldsymbol{v}_{ee} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2.1}$$

where $\boldsymbol{v}_{ee} \in \mathbb{R}^6$ is the *spatial* velocity of the end-effector, and $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{6\times N_J}$ is the Jacobian matrix. The Jacobian can be analytically computed for any configuration $\boldsymbol{q}$ and depends on the geometric parameters of a rigid body system, such as the lengths of the individual links and joint offsets. The pose of the end-effector combines its position and orientation into a single six-dimensional quantity. Consequently, the spatial velocity of the end-effector is constructed as follows:

$$\boldsymbol{v}_{ee} = \begin{bmatrix} \mathbf{v}_{ee} \\ \boldsymbol{\omega}_{ee} \end{bmatrix} \in \mathbb{R}^6, \tag{2.2}$$

where $\mathbf{v}_{ee} \in \mathbb{R}^3$ and $\boldsymbol{\omega}_{ee} \in \mathbb{R}^3$ are the linear and the angular velocity of the end-effector respectively. The vector $\boldsymbol{v}_{ee} \in \mathbb{R}^6$ is called a *spatial* vector, combining linear and angular motion into a single quantity [13].

Inverse instantaneous kinematics address the opposite problem: determining the joint velocity vector $\dot{\boldsymbol{q}}$ that corresponds to some spatial velocity of the end-effector $\boldsymbol{v}_{ee}$. This problem is significantly more complex than forward instantaneous kinematics, and does not generally

have a unique solution. The inverse instantaneous kinematics problem is closely related to task space control, discussed in the following sections.

2.2.2Task Space and Redundancy

The term *task space* refers to the vector space chosen to conveniently describe the desired motion of the system [9]. While the task space can correspond to the pose of the end-effector, it can also relate to any other quantity that has a corresponding Jacobian matrix. For example, the task space can correspond to the pose of any other link within the kinematic model, or to the distinct component of the link's pose, such as its position or orientation. Also, the task space can extend beyond the confines of the kinematic model to involve properties like the center of mass position [14], centroidal momentum [14], or the manipulability index [15,16].

A task space is defined with a corresponding task Jacobian, leading to an equation similar to (2.1):

$$\boldsymbol{v}_T = \boldsymbol{J}_T(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2.3}$$

where $\boldsymbol{v}_T \in \mathbb{R}^{N_T}$ is the task velocity, and $\boldsymbol{J}_T(\boldsymbol{q}) \in \mathbb{R}^{N_T \times N_J}$ is the task Jacobian. The dimensions of the matrix $\boldsymbol{J}_T$ and the vector $\boldsymbol{v}_T$ depend on the specific task, where N_T represents the number of degrees of freedom required to perform it.

As previously discussed, redundancy, like the task Jacobian, is a property of the specific task being performed. In fact, redundancy and the Jacobian matrix are closely related, where the dimensions and the rank of the Jacobian matrix determine whether the task is redundant. As mentioned in section2.1, a robot system is generally considered *intrinsically* redundant (redundant for the six-dimensional task of controlling the pose of the end-effector) if it has more than six actuated degrees of freedom, implying the condition for redundancy $N_J > N_T$. However, other factors have to be taken into account to truly determine whether a robot system exhibits redundancy when performing a certain task.

The Jacobian matrix depends on the specific task being performed, kinematic model of a robot and its current joint configuration $\boldsymbol{q}$. A robot system with a high number of actuated degrees of freedom, satisfying the condition $N_J > N_T$, is not necessarily redundant and can even be underactuated. To illustrate this, consider a simple scenario of a robot system with $N_J > 6$ linear joints, all controlling the position of the end-effector in the same direction. In this case, the robot system is clearly underactuated for the task of controlling the pose of the end-effector, even though $N_J > N_T$ holds. It is also possible for a system to exhibit redundancy or be fully actuated in one configuration while not being redundant or being underactuated in another. This loss of manipulability is typically observed when the robot approaches the edge of its workspace or reaches its joint limits. In general, the claim combining the task Jacobian and redundancy is that the system exhibits redundancy if the row rank of the task Jacobian is

higher than the number of degrees of freedom required to perform a specific task:

$$\text{rank}(\boldsymbol{J}_T) > N_T \tag{2.4}$$

The above condition implies a weaker condition mentioned before, that the number of actuated joints in a robot system has to be higher than N_T, since $\text{rank}(\boldsymbol{J}_T) \leq N_J$.

2.2.3Task Space Control

The goal of task space control is to determine the control commands necessary to achieve the desired task space motion. These control commands can take the form of joint torques, joint accelerations, or joint velocities, depending on the specific control strategy and the capabilities of the robot [17]. Many modern robots, including those used in this thesis, support direct joint velocity control. Therefore, in this thesis, the focus is on a task space motion control problem with velocity commands: selecting such a joint velocity vector $\dot{\boldsymbol{q}}$ that generates the desired task velocity $\boldsymbol{v}_T$. This can be seen as a generalization of the instantaneous inverse kinematics problem to an arbitrary task space.

In some cases, this can be accomplished by calculating the inverse of the Jacobian matrix and then multiplying it by the desired task space velocity:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_T^{-1}\boldsymbol{v}_T \tag{2.5}$$

The inverse of a matrix exists for square matrices of full rank. For the Jacobian matrix $\boldsymbol{J}_T(\boldsymbol{q}) \in \mathbb{R}^{N_T \times N_J}$ to be square, the number of joints in the robot system must be equal to the number of degrees of freedom required by the task. Even if this condition is met, the Jacobian inverse approach is rarely used due to the potential loss of rank in certain configurations, as discussed earlier.

The *pseudoinverse*, also known as the *Moore-Penrose inverse*, is a generalization of matrix inversion for non-square matrices, and is more commonly used to solve the velocity level task space control problem:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_T^{\dagger}\boldsymbol{v}_T \tag{2.6}$$

The pseudoinverse $\boldsymbol{A}^{\dagger}$ of a matrix $\boldsymbol{A}$ can be calculated using the following equation:

$$\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \tag{2.7}$$

In cases where equation (2.3) does not have a solution, meaning there is no vector $\dot{\boldsymbol{q}}$ that results in $\boldsymbol{v}_T$, the pseudoinverse based approach will compute a least-squares closest solution to the problem. On the other hand, if the system exhibits redundancy and there are multiple vectors $\dot{\boldsymbol{q}}$

for which equation (2.3) holds, the pseudoinverse results in a value of $\dot{\boldsymbol{q}}$ that satisfies equation (2.3) while having a minimal Euclidian norm. These properties, combined with the ability to compute the pseudoinverse analytically for any Jacobian, make this approach widely used for task space control.

One major drawback of this approach to task space control is its inability to constrain the values of $\dot{\boldsymbol{q}}$. Solving equation (2.6) can yield high values of $\dot{\boldsymbol{q}}$ that may not be achievable by the robot system. To address this issue, optimization-based task space control is often employed. The optimization-based approach to task space control is utilized throughout this thesis, and is explained in the following sections.

2.3Positional Kinematics

As mentioned in the previous section, *positional* kinematics deal with the positional aspects of both joint and end-effector motion of a rigid-body system. Positional kinematics focus on two fundamental problems: positional forward and positional inverse kinematics.

The positional forward kinematics problem is to determine the position and orientation of the end-effector of the robot system relative to its base, given the positions of all the joints [12]. Similar to its instantaneous counterpart discussed before, the positional forward kinematics problem has a unique solution, and can be solved analytically.

The positional inverse kinematics problem deals with the problem of determining joint positions that result in a desired position and orientation of the end-effector. The positional inverse kinematics are generally more complex than forward inverse kinematics, and do not necessarily have a unique solution. A desired pose of the end-effector can be infeasible, in which case there is no solution to the positional inverse kinematics problem. It is also possible for the inverse kinematics problem to have multiple solutions, which is often the case for intrinsically redundant robot systems.

As the problem of positional kinematics is fundamental to the field of robotics, a wide variety of methods has been developed and employed to address it over the years. For some robot systems, analytical closed-form solutions are possible, offering advantages in terms of time efficiency and numerical precision. On the other hand, numerical methods for positional inverse kinematics offer more flexibility and can be applied to any robot system.

2.3.1Iterative Methods for Positional Inverse Kinematics

Perhaps the most commonly used numerical methods for positional inverse kinematics are based on iteratively solving the instantaneous inverse kinematics problem, also called *iterative* methods.

As already mentioned, the goal of positional inverse kinematics is to select such a joint position vector $\boldsymbol{q}$ that results in the target pose of the end-effector $\boldsymbol{p}_{ee,t} \in \mathbb{R}^6$. The pose of the end-effector is a function of the joint position vector, and will be denoted as $\boldsymbol{p}_{ee}(\boldsymbol{q})$. The iterative methods are based on iteratively updating the value of $\boldsymbol{q}$, starting from its initial guess $\boldsymbol{q}_{initial}$, until the pose of the end-effector $\boldsymbol{p}_{ee}(\boldsymbol{q})$ reaches its target value $\boldsymbol{p}_{ee,t}$ [18].

The instantaneous kinematics equation can be rewritten to include the rate of change in the pose of the end-effector:

$$\dot{\boldsymbol{p}}_{ee} = \boldsymbol{J}_{ee}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2.8}$$

and an approximate relationship can be derived:

$$\Delta\boldsymbol{p}_{ee} \approx \boldsymbol{J}_{ee}(\boldsymbol{q})\Delta\boldsymbol{q}, \tag{2.9}$$

where $\Delta\boldsymbol{q}$ is the change in the joint angle vector, and $\Delta\boldsymbol{p}_{ee}$ is the change in end-effector pose. This equation is not exact as the Jacobian matrix depends (nonlinearly) on $\boldsymbol{q}$, but for sufficiently small values of $\Delta\boldsymbol{q}$ the equation (2.9) will hold.

The iterative inverse kinematics methods are based on iteratively updating the joint positions vector:

$$\boldsymbol{q} := \boldsymbol{q} + \Delta\boldsymbol{q}, \tag{2.10}$$

where $\Delta\boldsymbol{q}$ is calculated from the aforementioned approximation (2.9).

There are various approaches to iterative inverse kinematics that differ in their strategies for selecting $\Delta\boldsymbol{q}$ to update the joint angles [18]. One of the commonly used approaches is the pseudoinverse based method, corresponding to the pseudoinverse based task space control, and resulting in the following law for updating $\Delta\boldsymbol{q}$:

$$\Delta\boldsymbol{q} := \boldsymbol{J}_{ee}^{\dagger}(\boldsymbol{q})\Delta\boldsymbol{p}_{ee,d}, \tag{2.11}$$

where $\Delta\boldsymbol{p}_{ee,d}$ is the desired change in the pose of the end-effector.

One of the challenges encountered in this approach is to ensure a sufficiently small value of $\Delta\boldsymbol{p}_{ee,d}$ for the approximation (2.9) to be justified. The most obvious selection of $\Delta\boldsymbol{p}_{ee,d}$ is to set it to the difference between the target pose and the current pose:

$$\Delta\boldsymbol{p}_{ee,d} := \boldsymbol{p}_{ee,t} - \boldsymbol{p}_{ee}(\boldsymbol{q}), \tag{2.12}$$

also referred to as the pose error. For a large pose error the approximation will not hold, and the resulting $\Delta\dot{\boldsymbol{q}}$ might not decrease the pose error.

One of the techniques used to mitigate this problem is *error clamping*, having the effect of moving the target pose closer to the current end-effector pose [18]. Commonly, separate

maximum values are specified for the translation and the rotational components of the pose error. Instead of using the pose error as the desired change in the pose, its calculated as:

$$\Delta \boldsymbol{p}_{ee,d} := \text{clampMagnitude}(\boldsymbol{p}_{ee,t} - \boldsymbol{p}_{ee}(\boldsymbol{q})) \tag{2.13}$$

The *clampMagnitude* function limits both the translational and the rotational value of its input to their respective maximum values.

An example of the pseudoalgorithm for an iterative inverse kinematics solver is given in algorithm1. The pseudoinverse is applied iteratively to update $\boldsymbol{q}$, until the algorithm reaches its

$\boldsymbol{q} \leftarrow \boldsymbol{q}_{\text{initial}}$
$\boldsymbol{err} \leftarrow \boldsymbol{p}_{ee,t} - \boldsymbol{p}_{ee}(\boldsymbol{q})$
while $\|\boldsymbol{err}\| \geq \varepsilon_e$ **do**
  $\boldsymbol{J} \leftarrow \text{calculateJacobian}(\boldsymbol{q})$
  $\Delta \boldsymbol{p}_{ee,d} \leftarrow \text{clampMagnitude}(\boldsymbol{err})$
  $\Delta \boldsymbol{q} \leftarrow \boldsymbol{J}^{\dagger} \Delta \boldsymbol{p}_{ee,d}$
  $\boldsymbol{q} \leftarrow \boldsymbol{q} + \Delta \boldsymbol{q}$
  $\boldsymbol{err} \leftarrow \boldsymbol{p}_{ee,t} - \boldsymbol{p}_{ee}(\boldsymbol{q})$
end

Algorithm 1: Positional inverse kinematics solver using the pseudoinverse method.

stopping criterion, defined as the euclidean norm of the pose error reaching a predefined small value ε_e.

The strength of iterative positional inverse kinematics solvers is in the provided flexibility. These solvers can be applied to any robot system and different tasks, not limited to the end-effector pose. The algorithm discussed here can be extended to any task that has a corresponding Jacobian matrix.

In this thesis, an algorithm for prioritized positional inverse kinematics is presented, following similar principles to the ones shown here, and discussed in section3.3. The presented algorithm utilizes the same idea of iteratively updating the solution guess, with a more sophisticated strategy for selecting $\Delta \boldsymbol{q}$ based on prioritized task space control.

2.4Mathematical Optimization

Mathematical optimization, often simply referred to as optimization, is a field of mathematics focused on finding the best or optimal solution to a problem within a set of possible solutions. It usually involves minimizing or maximizing a real function by systematically selecting its input, while considering constraints.

A general optimization problem can be written as:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0,\ i = 1,\ldots,m \end{aligned} \tag{2.14}$$

where the vector $\boldsymbol{x} \in \mathbb{R}^n$ represents the optimization variable, function $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ is the objective function (also called the criterion function or the cost function), and $g_i(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ is the i-th inequality constraint function. The optimization problem given above can be read as: select such an optimization variable vector $\boldsymbol{x}$ that *minimizes* the objective function $f(\boldsymbol{x})$, *subject to* the constraint functions $g_i(\boldsymbol{x}) \leq 0$. Constraint functions are real functions of $\boldsymbol{x}$ that are required to be satisfied, effectively *constraining* the selection of $\boldsymbol{x}$ to vectors for which equations $g_i(\boldsymbol{x}) \leq 0$ hold.

Any $\boldsymbol{x}$ for which all the constraint functions are satisfied is referred to as *feasible*, and if such $\boldsymbol{x}$ exist, the overall optimization problem is said to be *feasible*. In contrast, if no feasible $\boldsymbol{x}$ exists, the optimization problem is said to be *infeasible*. The solution $\boldsymbol{x}^*$ is *optimal* if $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ holds for all feasible $\boldsymbol{x}$.

In the optimization problem formulation (2.14) the objective function is chosen to be minimized, but the formulation is general, and can be used to maximize an arbitrary function $f(\boldsymbol{x})$ by minimizing $-f(\boldsymbol{x})$. In addition to inequality constraints, the optimization problem can be formulated in such a way to explicitly include equality constraints:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0,\ i = 1,\ldots,m \\ & h_j(\boldsymbol{x}) = 0,\ j = 1,\ldots,p \end{aligned} \tag{2.15}$$

Here, equality constraints are explicitly defined as $h_j(\boldsymbol{x}) = 0$, while the previous formulation allows their implicit definition using two inequality constraints, since $h(\boldsymbol{x}) = 0$ is equivalent to $h(\boldsymbol{x}) \leq 0$ and $-h(\boldsymbol{x}) \leq 0$.

In a general optimization problem, the functions $f(\boldsymbol{x})$, $g_i(\boldsymbol{x})$ and $h_j(\boldsymbol{x})$ can take the form of arbitrary nonlinear functions that map from $\mathbb{R}^n$ to $\mathbb{R}$. The form of these functions determines the class of the optimization problem. An optimization problem belonging to a specific class

is usually solved more efficiently and precisely with a class-specific solver. Perhaps the most important distinction among different classes of optimization problems is between *convex* and *non-convex* problems [19].

A *convex* optimization problem is one in which the objective and constraint functions are convex, meaning they satisfy the inequality:

$$f_i(\alpha \boldsymbol{x} + \beta \boldsymbol{y}) \leq \alpha f_i(\boldsymbol{x}) + \beta f_i(\boldsymbol{y}) \tag{2.16}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$ [19].

An important property of all convex optimization problems is that their local optimum is guaranteed to be a global one. The theory of convex optimization is fairly complete, and there are methods to solve different subclasses of such optimization problems with guarantees on finite time execution and the correctness of the solution. For a detailed resource on theory of convex optimization the reader is referred to [19] by Boyd and Vandenberghe.

Various classes of optimization problems are encountered in both robotics and the broader field of optimal control. Convex optimization is widely applied in these domains, since the solving times are often required to be relatively low. A common form of a convex optimization problem encountered in these fields, and throughout this thesis, is a *quadratic program* (QP).

2.4.1 Quadratic Programming

A *quadratic program* is a convex optimization problem, where the criterion function is in the shape of a quadratic form, and both the equality and the inequality constraint are affine:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{q}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}_{eq}\,\boldsymbol{x} + \boldsymbol{b}_{eq} = \boldsymbol{0} \\ & \boldsymbol{A}_{ieq}\,\boldsymbol{x} + \boldsymbol{b}_{ieq} \preceq \boldsymbol{0} \end{aligned} \tag{2.17}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{q} \in \mathbb{R}^n$ are the quadratic form matrix and vector respectively, $\boldsymbol{A}_{ieq} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b}_{ieq} \in \mathbb{R}^m$ are the inequality constraint matrix and vector respectively, and $\boldsymbol{A}_{eq} \in \mathbb{R}^{p \times n}$ and $\boldsymbol{b}_{eq} \in \mathbb{R}^p$ are the equality constraint matrix and vector respectively. The symbol $\preceq$ represents element-wise comparison between two vectors, requiring every vector entry on the left side of the symbol to be less than or equal to the corresponding entry on the right side.

A formulation of a quadratic programming problem where the criterion function is written as a squared quadratic norm is commonly encountered in practice, also called the least squares

formulation:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & ||\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}||^2 \\ \text{s.t.} \quad & \boldsymbol{A}_{eq}\,\boldsymbol{x}+\boldsymbol{b}_{eq}=\boldsymbol{0} \\ & \boldsymbol{A}_{ieq}\,\boldsymbol{x}+\boldsymbol{b}_{ieq}\preceq\boldsymbol{0} \end{aligned} \tag{2.18}$$

The squared quadratic norm of a vector is defined as a sum of the squares of its individual components:

$$||\boldsymbol{v}||^2=\sum_{i=1}^{n} v_i^2=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} \tag{2.19}$$

This formulation of the criterion function can be expanded to the standard quadratic form as:

$$||\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}||^2=(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}-2\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b} \tag{2.20}$$

The term $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b}$ is a constant which does not depend on the optimization variable vector $\boldsymbol{x}$, and can therefore be disregarded from the criterion function. It is also worth noting that a sum of multiple squared quadratic norms is also a squared quadratic norm, and can be used as a criterion function in a quadratic program.

This form of a quadratic program criterion function can be directly utilized for optimization based task space control. As explained in previous sections, the velocity level task space control problem is to select such a joint velocity vector $\dot{\boldsymbol{q}}$ that results in the task velocity $\boldsymbol{v}_T$, if $\boldsymbol{v}_T=\boldsymbol{J}_T\dot{\boldsymbol{q}}$. A quadratic program can be utilized to solve this problem, while enforcing to joint velocity limits:

$$\begin{aligned} \min_{\dot{\boldsymbol{q}}} \quad & ||\boldsymbol{J}_T\dot{\boldsymbol{q}}-\boldsymbol{v}_T||^2 \\ \text{s.t.} \quad & \underline{\dot{\boldsymbol{q}}}\preceq\dot{\boldsymbol{q}}\preceq\overline{\dot{\boldsymbol{q}}} \end{aligned} \tag{2.21}$$

Here, vectors $\underline{\dot{\boldsymbol{q}}}$ and $\overline{\dot{\boldsymbol{q}}}$ represent the lower and the upper bound on joint velocities, respectively. Inequality constraints formulated in this specific way are referred to as *bound* constraints. The quadratic program given above can be used as a base optimization problem for velocity level task space control, while the cost function can be modified and additional constraints can be added to incorporate the needs of a specific task.

Quadratic programming is widely used in robotics and optimal control due to its reliable and efficient solving techniques, in addition to its complete theory. Due to both the development of solving methods and the rise in computing power of contemporary machines, quadratic programming solvers are able to handle large, high-dimensional problems efficiently, with a guarantee on the correctness of the solution.

2.5Prioritized Optimization

Prioritized optimization can be conceptualized as a distinctive category of mathematical optimization. A standard optimization problem deals with selecting an optimization variable vector $\boldsymbol{x}$ that minimizes a single criterion function $f(\boldsymbol{x})$. However, optimization problems encountered in practice often have multiple optimization objectives. This challenge is commonly addressed by combining multiple criterion functions into a single function, using a weighted sum of the individual criterion functions as the overall criterion function:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}) + \ldots + w_N f_N(\boldsymbol{x}) \\ \text{s.t.} \quad & g_j(\boldsymbol{x}) \leq 0, \ j = 1, \ldots, m \end{aligned} \tag{2.22}$$

where N is the number of separate optimization objectives.

The weight w_i assigned to a certain objective function $f_i(\boldsymbol{x})$ corresponds to its importance in the overall optimization problem. Increasing the weight w_i results in the solution being closer to the optimum of $f_i(\boldsymbol{x})$. However, generally, the minimum of the weighted sum does not correspond to the minimum of any of the objectives, but presents a compromise between them.

The idea behind prioritized optimization is to replace the commonly used objective weighting approach with hard priorities that are guaranteed to be satisfied. The prioritized optimization problem can be written as:

$$\begin{aligned} h_i = \min_{\boldsymbol{x}} \quad & f_i(\boldsymbol{x}) \\ \text{s.t.} \quad & f_l(\boldsymbol{x}) = h_l, \ \forall l < i \\ & g_j(\boldsymbol{x}) \leq 0, \ j = 1, \ldots, m \end{aligned} \tag{2.23}$$

Here, $h_i \in \mathbb{R}$ represents the optimal value of the objective function of the i-th priority, with the addition of prioritization constraints $f_l(\boldsymbol{x}) = h_l, \ \forall l < i$. The objective functions $f_i(\boldsymbol{x})$ have decreasing priorities, $f_1(\boldsymbol{x})$ having the highest priority, and $f_N(\boldsymbol{x})$ representing an objective function with the lowest priority. Consider a case with two prioritized objective functions, $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$. The optimization problem (2.23) is solved for the first priority $f_1(\boldsymbol{x})$, and its minimum h_1 is found. For the first priority there are no prioritization constraints, and the problem is solved as a standard optimization problem:

$$\begin{aligned} h_1 = \min_{\boldsymbol{x}} \quad & f_1(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, \ i = 1, \ldots, m \end{aligned} \tag{2.24}$$

For the second objective, the optimization problem has an additional prioritization constraint:

$$\begin{aligned} h_2 = \min_{\boldsymbol{x}} \quad & f_2(\boldsymbol{x}) \\ \text{s.t.} \quad & f_1(\boldsymbol{x}) = h_1 \\ & g_i(\boldsymbol{x}) \leq 0,\ i = 1,\ldots,m \end{aligned} \tag{2.25}$$

The solution to the second priority is found subject to a constraint $f_1(\boldsymbol{x}) = h_1$, constraining the first priority objective function to remain in its optimum.

Prioritized optimization relies on redundancy of higher priority objectives to achieve the prioritization. An objective function is said to be redundant if there are multiple inputs $\boldsymbol{x}$ that minimize the function and result in $f(\boldsymbol{x}) = h$. If the function $f_1(\boldsymbol{x})$ has multiple solutions, the one that minimizes $f_2(\boldsymbol{x})$ is selected between them. This logic is extended to enable an arbitrary number of priorities N. Once a priority with a unique solution (a non-redundant priority) is reached, the prioritized optimization is finished, as the value of $\boldsymbol{x}$ minimizing this priority is also the solution to the entire prioritized optimization problem.

2.5.1Prioritized Quadratic Programming

The term *prioritized quadratic programming* is used to refer to a prioritized optimization problem where all of the objective functions are quadratic forms, and the constraints are affine:

$$\begin{aligned} h_i = \min_{\boldsymbol{x}} \quad & E_i(\boldsymbol{x}) \\ \text{s.t.} \quad & E_l(\boldsymbol{x}) = h_l,\ \forall l < i \\ & \boldsymbol{A}_{eq}\,\boldsymbol{x} + \boldsymbol{b}_{eq} = \boldsymbol{0} \\ & \boldsymbol{A}_{ieq}\,\boldsymbol{x} + \boldsymbol{b}_{ieq} \preceq \boldsymbol{0} \end{aligned} \tag{2.26}$$

Symbol E_i denotes a quadratic criterion function of the i-th priority, which can be written as a squared quadratic norm:

$$E_i(\boldsymbol{x}) = ||\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i||^2 \tag{2.27}$$

As explained in the previous section, prioritized optimization exploits the redundancies in the higher priority objective functions, and resolves those redundancies in such a way to optimize the lower priorities. It is therefore worthwhile to investigate the circumstances under which an objective function of a quadratic form exhibits redundancy.

A quadratic form $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x}$ can exhibit various configurations determined by the definiteness of the matrix $\boldsymbol{Q}$. A visualization of different configurations of a two-dimensional quadratic form is given in figure2.2. For a positive definite matrix $\boldsymbol{Q}$ it can be said that $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} > 0$ for all $x \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$. The level curves of such a quadratic form are shaped as ellipsoids, and minimizing

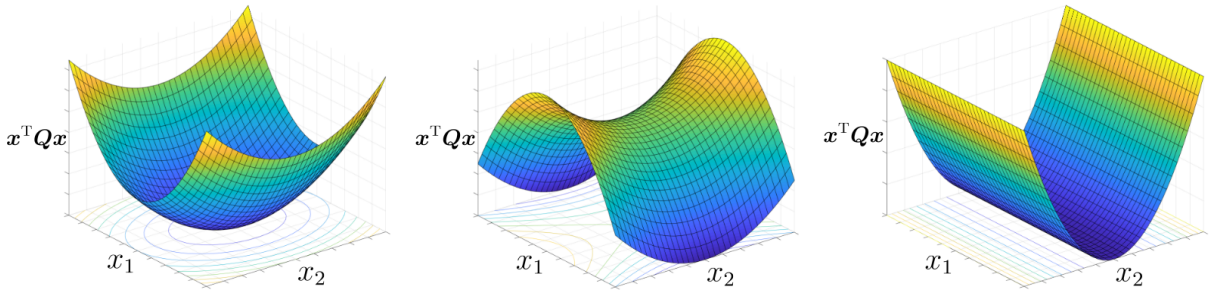

Figure 2.2: Different configurations of two-dimensional quadratic forms determined by the definiteness of the matrix $\boldsymbol{Q}$. The definiteness of matrix $\boldsymbol{Q}$ dictates the shape of the level curves and the number of minimums a quadratic form has.

a positive definite quadratic form will yield a unique solution. Indefinite quadratic form are the ones that do not exhibit a consistent sign pattern for $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}$ for all $x \in \mathbb{R}^n \setminus \{\mathbf{0}\}$. An indefinite quadratic form has a minimum of $-\infty$, and is usually not encountered in optimization problems. For a positive semi-definite matrix $\boldsymbol{Q}$, it follows that $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} \geq 0$ for all $x \in \mathbb{R}^n \setminus \{\mathbf{0}\}$. The level curves of a positive semi-definite quadratic form are shaped as straight lines, resulting in multiple minimums, as shown in figure2.2. An important property of the redundancy exhibited by minimizing a positive semi-definite function, is that all the solutions $\boldsymbol{x}^*$ can be parameterized as an affine function of some parameter $\boldsymbol{w}$:

$$\boldsymbol{x}^*(\boldsymbol{w}) = \boldsymbol{A}_*\boldsymbol{w} + \boldsymbol{b}_* \tag{2.28}$$

As previously discussed, the complexity of an optimization problem in terms of its execution time and the correctness of the solution depends on the nature of its objective function and constraint functions. The formulation of the prioritized quadratic program given in (2.26) has non-affine equality constraints $E_l(\boldsymbol{x}) = h_l$, $\forall l < i$, since the cost function E_l is quadratic. This disqualifies the given formulation of the optimization problem as a quadratic program. However, it is still possible to solve a prioritized quadratic program using N quadratic programs, by transforming the vector space of the optimization variable between the priorities. This is done by exploiting the already mentioned ability to parametrize all the solutions to a criterion function of a semi-definite quadratic form.

Algorithms for unconstrained and constrained versions of prioritized quadratic optimization are given by de Lasa et. al. in [20] and [21] respectively. Unconstrained prioritized quadratic optimization refers to the optimization problem with no equality or inequality constraints, except for the constraints posed by the prioritization itself. A pseudoalgorithm for unconstrained prioritized quadratic optimization by de Lasa et. al. [20] is given in Algorithm2. The algorithm solves N unconstrained quadratic optimization problems, using the Moore-Penrose pseudoinverse. Vector $\overline{\boldsymbol{d}}$ stores a solution to the current priority, while the matrix $\overline{\boldsymbol{C}}$ transforms the

$\overline{\boldsymbol{C}} \leftarrow \boldsymbol{I}, \overline{\boldsymbol{d}} \leftarrow \boldsymbol{0}$
for $i \leftarrow 1$ **to** N **do**
  $\overline{\boldsymbol{A}}_i \leftarrow \boldsymbol{A}_i \overline{\boldsymbol{C}}$
  $\overline{\boldsymbol{b}}_i \leftarrow \boldsymbol{b}_i - \boldsymbol{A}_i \overline{\boldsymbol{d}}$
  $\overline{\boldsymbol{d}} \leftarrow \overline{\boldsymbol{d}} + \overline{\boldsymbol{C}}\, \overline{\boldsymbol{A}}_i^{\dagger} \overline{\boldsymbol{b}}_i$
  if $\overline{\boldsymbol{A}}_i$ *is full rank* **then**
    return $\overline{\boldsymbol{d}}$
  end
  $\overline{\boldsymbol{C}} \leftarrow \overline{\boldsymbol{C}}\, \text{null}(\overline{\boldsymbol{A}}_i)$
end
return $\overline{\boldsymbol{d}}$

Algorithm 2: Unconstrained quadratic prioritized solver pseudoalgorithm by De Lasa et. al. [20].

problem into the nullspace of all the higher priorities. The algorithm is finished once all N priorities are solved, or once the transformed matrix $\overline{\boldsymbol{A}}_i$ is of full rank. If the matrix $\overline{\boldsymbol{A}}_i$ is of full rank, the prioritized optimization problem has a unique solution at the i-th priority.

The algorithm for constrained prioritized quadratic optimization by de Lasa et.al. [21] functions in a similar way (Algorithm3). Here, instead of solving N unconstrained quadratic op-

$\overline{\boldsymbol{C}} \leftarrow \boldsymbol{I}, \overline{\boldsymbol{d}} \leftarrow \boldsymbol{0}$
for $i \leftarrow 1$ **to** N **do**
  $\overline{\boldsymbol{A}}_i \leftarrow \boldsymbol{A}_i \overline{\boldsymbol{C}}$
  $\overline{\boldsymbol{b}}_i \leftarrow \boldsymbol{b}_i - \boldsymbol{A}_i \overline{\boldsymbol{d}}$
  $\boldsymbol{d}_i \leftarrow \arg\min_{\boldsymbol{w}} \; ||\overline{\boldsymbol{A}}_i \boldsymbol{w} - \overline{\boldsymbol{b}}_i||^2$
    s.t. $\boldsymbol{A}_{eq} \overline{\boldsymbol{C}} \boldsymbol{w} + \boldsymbol{A}_{eq} \overline{\boldsymbol{d}} + \boldsymbol{b}_{eq} = \boldsymbol{0}$
    $\boldsymbol{A}_{ieq} \overline{\boldsymbol{C}} \boldsymbol{w} + \boldsymbol{A}_{ieq} \overline{\boldsymbol{d}} + \boldsymbol{b}_{ieq} \preceq \boldsymbol{0}$
  if *problem is infeasible* **then**
    return $\overline{\boldsymbol{d}}$
  end
  $\overline{\boldsymbol{d}} \leftarrow \overline{\boldsymbol{d}} + \overline{\boldsymbol{C}} \boldsymbol{d}_i$
  if $\overline{\boldsymbol{A}}_i$ *is full rank* **then**
    return $\overline{\boldsymbol{d}}$
  end
  $\overline{\boldsymbol{C}} \leftarrow \overline{\boldsymbol{C}}\, \text{null}(\overline{\boldsymbol{A}}_i)$
end
return $\overline{\boldsymbol{d}}$

Algorithm 3: Constrained quadratic prioritized solver pseudoalgorithm by De Lasa et. al. [21].

timization problem using a pseudoinverse, N constrained quadratic programming problems are solved. In addition to transforming the $\boldsymbol{A}_i$ matrices to the null space of previous priorities, equality and inequality constraints posed by the problem are also transformed into the men-

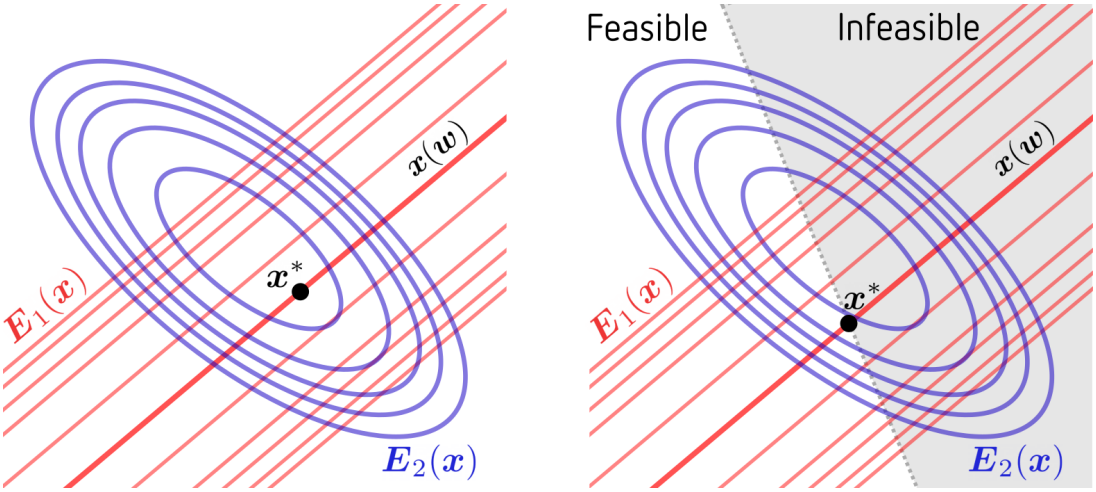

Figure 2.3: Visualization of prioritized optimization with quadratic criterion functions.

tioned space. There is an additional stopping criterion for the algorithm, ending the solver once the transformed quadratic programming problem is infeasible. This stopping criterion is inserted before the solution update $\overline{\boldsymbol{d}} \leftarrow \overline{\boldsymbol{d}} + \overline{\boldsymbol{C}}\boldsymbol{d}_i$ since this means there is no solution for the current priority, and the solution of the previous priority is returned as the solution of the overall prioritized optimization problem.

A visualization of both the unconstrained and the constrained version of the prioritized quadratic optimization problem with two priorities is given in Fig.2.3. The objective function of the first priority $E_1(\boldsymbol{x})$ is positive semi-definite, and its level curves are visualized as red lines. The first priority objective function has multiple minimums, shown as a thick red line, which can be parameterized as $\boldsymbol{x}(\boldsymbol{w})$. The solution to the prioritized optimization problem is given by transforming a second priority objective function $E_1(\boldsymbol{x})$ to the nullspace of the first problem, $E_1(\boldsymbol{w})$, and minimizing that function. This vector space transformation is used to implicitly constrain the second optimization problem to the space of all the solutions of the first one. Solution is given as a point on the line $\boldsymbol{x}(\boldsymbol{w})$ that minimizes the second criterion function E_2.

Prioritized optimization is especially useful for redundancy resolution. Prioritized optimization allows the primary task (the one for which the system exhibits redundancy) to be prioritized over other criteria that serve to resolve the redundancy. There can be an arbitrary number of priorities between those redundancy-resolving criteria, allowing multiple levels of redundancy resolution, while respecting different constraints posed by the tasks.

2.6Model Predictive Control

Model predictive control (MPC) represents a type of predictive optimal control, aiming to determine a sequence of control inputs that result in optimal system performance during some time

horizon. This approach involves employing a mathematical *model* of the process to *predict* the future behaviour of its output $\boldsymbol{y}(t)$, and state $\boldsymbol{x}(t)$, for some control input sequence $\boldsymbol{u}(t)$.

While a general (nonlinear) model predictive control approach might involve a nonlinear system model and a general nonlinear objective function, in the context of this thesis, the emphasis will be on *linear* MPC. In the realm of linear MPC, the attention is directed towards linear control systems.

The state-space system dynamics equation for a discrete linear control system can be written as follows:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k), \tag{2.29}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^{n_x}$ is the system state vector, and $\boldsymbol{u}(k) \in \mathbb{R}^{n_u}$ is the control input vector in control step $k \in \mathbb{N}$. The system dynamics are fully described with matrices $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$ and $\boldsymbol{B} \in \mathbb{R}^{n_x \times n_u}$, called the state matrix and the input matrix respectively. The equation for calculating the output of such a system can be written as:

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k), \tag{2.30}$$

where $\boldsymbol{y}(k) \in \mathbb{R}^{n_y}$ is the output vector in control step $k \in \mathbb{N}$. Matrices $\boldsymbol{C} \in \mathbb{R}^{n_y \times n_x}$ and $\boldsymbol{D} \in \mathbb{R}^{n_y \times n_u}$ are called the output matrix and the feedforward matrix respectively. If the system does not have direct feedforward of the control inputs to its output, $\boldsymbol{D}$ is a zero matrix.

The MPC algorithm predicts system output during some time horizon of $N_h \in \mathbb{N}$ control steps. Let us define the vectors that store the control input, system state and system outputs during the entire horizon as:

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}(0) \\ \boldsymbol{u}(1) \\ \vdots \\ \boldsymbol{u}(N_h-1) \end{bmatrix} \in \mathbb{R}^{N_h \cdot n_u}, \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}(0) \\ \boldsymbol{x}(1) \\ \vdots \\ \boldsymbol{x}(N_h-1) \end{bmatrix} \in \mathbb{R}^{N_h \cdot n_x}, \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}(0) \\ \boldsymbol{y}(1) \\ \vdots \\ \boldsymbol{y}(N_h-1) \end{bmatrix} \in \mathbb{R}^{N_h \cdot n_y} \tag{2.31}$$

Then, the MPC optimization problem can be written as follows:

$$\begin{aligned} \min_{\boldsymbol{U}} \quad & J(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{Y}) \\ \text{s.t.} \quad & \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ & \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 \end{aligned} \tag{2.32}$$

Function $J(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{Y})$ represents a cost function optimized by the MPC algorithm, which is generally a function of system outputs, states and control inputs. The optimization problem selects

a sequence of control inputs $(\boldsymbol{u}(0),\ \boldsymbol{u}(1),\ \ldots,\ \boldsymbol{u}(N_h-1))$ that minimize this function, while respecting the system dynamics. The constraint $\boldsymbol{x}(0)=\boldsymbol{x}_0$ sets the currently measured state of the system as the initial state of the system.

Solving the optimization problem (2.32) gives an optimal sequence of control inputs, that results in optimal behaviour of the system according to the given model. To increase the robustness of the control method, and to allow for some inaccuracy in system modeling, MPC is based on using only the first control input of the optimal sequence $\boldsymbol{u}(0)$ as the system input. After applying the first control input to the system, a new initial state $\boldsymbol{x}_0$ is measured and the optimization problem (2.32) is solved again. This results in constantly moving the predictive horizon forward in time, which is why model predictive control is also referred to as *receding horizon control*.

Criterion functions for linear MPC are most often quadratic, which results in the optimization problem being a quadratic program. An MPC algorithm used to optimally follow a sequence of changing desired output values $\boldsymbol{Y}_d$ is referred to as *reference tracking* MPC. A common optimization problem used for reference tracking MPC is given as follows:

$$\begin{aligned} \min_{\boldsymbol{U}} \quad & Q||\boldsymbol{Y}-\boldsymbol{Y}_d||^2+R||\boldsymbol{U}||^2 \\ \text{s.t.} \quad & \boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}\boldsymbol{u}(k) \\ & \boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k) \\ & \boldsymbol{x}(0)=\boldsymbol{x}_0 \end{aligned} \tag{2.33}$$

Here, the first part of the criterion function $Q||\boldsymbol{Y}-\boldsymbol{Y}_d||^2$ serves to minimize the difference between the desired reference and the predicted system output, where Q is a scalar weight value. The second part of the criterion function $R||\boldsymbol{U}||^2$ serves to minimize the control inputs, with R as the associated scalar weight value. The selection of weight parameters Q and R is used to tune the trade-off between the reference tracking error and the aggressiveness of control inputs. For example, if R is set to zero, the optimization problem will select such control inputs that result in ideal reference tracking. However, this will result in large values of control inputs, which the system might not be able to produce. One way to combat this issue is to introduce additional bound constraints for the control inputs:

$$\underline{\boldsymbol{u}} \preceq \boldsymbol{u}(k) \preceq \overline{\boldsymbol{u}} \tag{2.34}$$

The optimization problem (2.33) is a simple solution for reference tracking MPC, and requires only two parameters to be tuned. However, this simplicity comes with the cost of limited flexibility. If the optimal control problem at hand is redundant, and there are multiple ways control inputs can produce the same output, the cost function can be modified to introduce more

versatility regarding the tuning process. A version of reference tracking MPC used in this thesis solves a following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{U}} \quad & ||\boldsymbol{Y} - \boldsymbol{Y}_d||^2 + ||\boldsymbol{W}_u \boldsymbol{U}||^2 + ||\boldsymbol{W}_x \boldsymbol{X}||^2 \\ \text{s.t.} \quad & \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ & \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ & \underline{\boldsymbol{x}} \preceq \boldsymbol{x}(k) \preceq \overline{\boldsymbol{x}} \\ & \underline{\boldsymbol{u}} \preceq \boldsymbol{u}(k) \preceq \overline{\boldsymbol{u}} \end{aligned} \tag{2.35}$$

There are three parts to the criterion function. The first one is the same as in (2.33) and serves to minimize the reference tracking error. The second part of the criterion function $||\boldsymbol{W}_u \boldsymbol{U}||^2$ is similar to the part of the criterion function associated with the control inputs in a previous example, with the difference of the weighting being done by a matrix $\boldsymbol{W}_u$ instead of a scalar R. This is used to have separate weights for separate control inputs, which allows for more flexibility in the tuning process. The third part of the criterion function $||\boldsymbol{W}_x \boldsymbol{X}||^2$ minimizes a linear function of all the process states. Additional bound constraints are introduced to limit the control inputs and system states at each control step.

CHAPTER 3

Vineyard Spraying with a Mobile Manipulator

3.1Introduction

Agricultural spraying refers to the process of applying various substances, such as pesticides, herbicides, fungicides and fertilizers to crops. There are two widely used contemporary approaches to vineyard spraying: manual and mechanized spraying. Manual spraying usually involves a human operator carrying a tank of the spraying agent and controlling its application using a *spray wand*: a nozzle attached at the end of a lightweight pole. A nozzle is commonly mounted at an angle to the pole, allowing the operator to control both the position and the direction of the nozzle with minimal movement at the base of the pole. This allows the operator to target both high and low areas of the foliage with the spraying agent, and apply the spraying agent at the desired angle. However, manual spraying is generally labor-intensive and time-consuming, which has driven the development of contemporary mechanization for agricultural spraying.

The widely used contemporary mechanization for the vineyard spraying task usually involves mounting a trailer with atomizers or multiple nozzles to a tractor or some other form of a human-operated vehicle (Fig.3.1). These machines are designed with time-efficiency in mind, processing a large area of a vineyard in a short amount of time. Precise targeting of the spraying agent is sacrificed in order to achieve this, and large amounts of it end up in the soil, presenting an ecological and sustainability concern.

The proposed robotic system for viticultural spraying uses a spray wand for manual spraying, mounted on a mobile manipulator, aiming is to achieve targeted application of the spraying

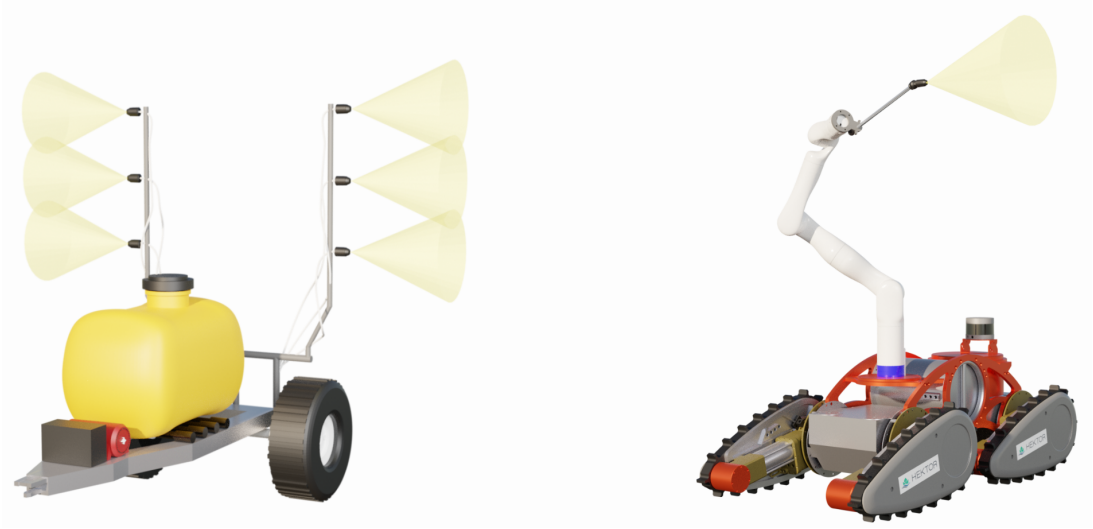

Figure 3.1: The image on the left depicts a contemporary agricultural spraying machine, in a form of a trailer with multiple spray nozzles on both sides. This kind of machine is usually attached on a human-operated vehicle. The image on the right depicts the proposed robotic solution for agricultural spraying. A spray wand, often used for manual vineyard spraying, is mounted as the end-effector of a mobile manipulator, allowing for precise targeting of the spraying agent.

agent without the need for manual labour (Fig.3.1). While the time-efficiency of this approach cannot reach the one of contemporary mechanization, the introduction of autonomy to the system has the potential to mitigate this concern.

In this chapter, two variants of the viticultural spraying task are considered: application of the protective substance to the entire grapevine canopy, and its application to a specific area of the plant (for example a single disease-ridden leaf or a cluster of grapes). These two tasks will be referred to as *continuous* and *selective* spraying, respectively.

During continuous spraying, position of the spraying agent is controlled both by the mobile vehicle and the robot arm. This introduces a redundancy, where the desired motion of the nozzle can be achieved in multiple ways: either by movement of the mobile base, or that of the robot arm. This redundancy is resolved in a predictive way, using task space model predictive control (MPC) to anticipate changes in the reference spraying agent trajectory, and optimize mobile vehicle and robot arm movement accordingly. MPC provides the task space commands for the robot arm, and an additional task space control algorithm is used to select its low-level controls.

Control of the robot arm itself for the task of continuous spraying is also a redundant task. As already discussed, the spraying nozzle used in this thesis is an axis-symmetric tool (Fig.2.1), meaning its orientation around its approach axis does not impact the application of the spraying agent. The task of controlling the spraying nozzle is a 3T2R task, where three translational and two rotational degrees of freedom affect task execution. This redundancy is resolved using prioritized task space control, handling the control of the spraying nozzle as a prioritized 3T2R task. Linear velocity of the spraying agent is prioritized over the orientation of the nozzle,

since applying the agent at the commanded area of the plant is generally more important than applying it with some exact orientation.

The proposed solution to the selective spraying task follows similar reasoning, resolving the redundancy posed by the axis-symmetry of the nozzle using the same prioritization between the tasks, but in joint position rather than joint velocity space. Prioritized positional inverse kinematics solver is presented, using iterative prioritized task space control to select joint positions that optimize a number of prioritized positional tasks.

The methods discussed in this chapter are previously discussed in two papers: one dealing with task space model predictive control [8], and another one addressing constrained prioritized 3T2R task control for vineyard spraying [22].

3.1.1Related Work

In recent years, significant research efforts have been devoted to the development of robotic systems for various viticultural tasks, with research focusing on areas such as navigation, monitoring, spraying, harvesting, and more [23,24,25,26,27,28,29,30,31]. However, interest in this research field dates back much earlier. As early as 1995., Monta et al. [32] presented their work on a multipurpose robot for grapevine harvesting, berry thinning, spraying, and bagging.

There is a significant amount of research focusing on the development and utilization of robotic technology for vineyard spraying [23,25,33,34,35,36,37]. In [33], Berenstein et al. present computer vision algorithms for grape cluster and foliage detection, which they utilize for autonomous vineyard spraying. They use a mobile vehicle with multiple sprayer nozzles mounted at different heights, similar to the one shown on the left side of Fig.3.1. However, instead of constantly spraying from all the nozzles, the visual feedback is used to select which nozzles should be active at which point in time. Oberti et al. [23,34] mounted a precision-nozzle on a robot manipulator, and use it for selective spraying, to spray specific, disease-ridden spots of the grapevine. Again, the focus is on the detection algorithms developed for disease detection in multispectral images, rather than on control of the robot manipulator.

There is a number trajectory planning algorithms developed for mobile manipulation [38, 39,40,41,42]. These algorithms are often designed to be as general as possible, capable of performing a variety of different tasks while considering obstacles, dynamics and stability. The vineyard is a semi-structured environment, the grapevines usually being planted in fairly straight rows. Area between the rows is obstacle-free, and, during the vineyard spraying task, mobile base is constrained to move in a straight line between the rows. Therefore, for this specific use case, a linear MPC algorithm is presented, considering only the motion of the mobile base along the row, and two-dimensional manipulator tool motion. This simplification results in an optimization problem that can be solved in real-time even for large prediction horizons.

In the research focusing on utilization of mobile manipulation for paint spraying [43,44,45],

a lawnmower trajectory is often used as a tool position reference to ensure uniform coverage, an approach adopted in the presented work.

As previously mentioned, a nozzle used for viticultural spraying in this thesis is an axis-symmetric tool, introducing functional redundancy to the spraying problem. There is extensive research on different approaches to resolving functional redundancies in robot manipulation [10,11,46,47,48]. Some of the research in the field of robotic paint spraying suggests that achieving commanded position of the spraying agent during the task is more important than having ideal nozzle orientation [48,49,50]. According to the findings presented in the work of From et al. [50], the linear velocity of the paint gun holds greater significance than its orientation in ensuring uniform paint coating. The presented research is motivated by the hypothesis that the same is the case for agricultural spraying, and perhaps to an even greater degree. This is primarily due to the fact that the spraying agent used in agricultural applications is typically less dense than the paint used in spray painting, and human operators tend to be less careful in their handling of nozzle orientation during agricultural spraying tasks compared to their counterparts in paint spraying applications.

This insight is addressed by introducing a prioritization between the translational and rotational components of the 3T2R task. The constrained prioritized quadratic programming algorithm presented in [21], and discussed previously in section2.5.1, is used to enforce this prioritization in the velocity space, for the task of continuous spraying. Similar approaches to task prioritization for robot arm control are sometimes referred to as prioritized velocity space inverse kinematics, prioritized instantaneous inverse kinematics or just prioritized inverse kinematics in the literature [51,52,53,54].

A solution to the problem of selective spraying is also presented, handling it as a prioritized positional inverse kinematics problem. A numerical, iterative solver to this problem is presented, similar to the one discussed in section2.3.1. The prioritized quadratic programming problem with the same prioritization as for continuous spraying is iteratively solved to update the joint position vector, until a solution is reached.

3.2Continuous Spraying

The continuous spraying task involves controlling the position and the orientation of the spraying nozzle continuously, to achieve sufficient coverage of the entire plant canopy. The position of the spraying nozzle is controlled both by the robot arm and the mobile base, while its orientation is controlled solely by the robot arm.

This section is divided into two subsections. Subsection3.2.1describes the task space model predictive control algorithm used to calculate coordinated motion actions for the mobile base and the robot arm that result in desired motion of the spraying nozzle. Here, the motion actions for the robot arm are selected in a task space rather than joint space. Subsection3.2.2presents a prioritized optimization based task space control algorithm used to select the low-level joint velocity commands for the robot arm, while also taking into account the orientation and the axis-symmetry of the nozzle.

3.2.1Task Space Model Predictive Control

The MPC algorithm described in this section aims to solve the following problem: given a description of a row of grapevines, select coordinated mobile vehicle and robot arm commands that result in satisfactory canopy coverage, while aiming to minimize spraying agent waste. One of the controller objectives is to enable the mobile manipulator to adapt to different row structures, accelerate in areas without grapevines and slow down in areas with largest foliage heights. The grapevine plants are usually planted in structured rows, the obstacle-free area between the rows being fairly narrow. The limited width of this area significantly constrains the movement of the vehicle. Therefore, in order to simplify the MPC algorithm and the corresponding optimization problem, a simplifying assumption is made. It is assumed that the mobile base is able to travel at a constant distance to a row of grapevines, and that its velocity along the row can be controlled separately to this assumption.

A coordinate frame is defined at a fixed distance to the spray nozzle, referred to as the *spray frame* and denoted as L_S (Fig.3.2). Using the assumption that the mobile base is at a fixed distance to the vines, the goal of the MPC becomes to control the two-dimensional position of the L_S frame, with respect to a fixed ground-level global frame denoted as L_G. The vertical position of the L_S frame with respect to L_G is denoted as $p_{S,z}$, and its position along the row is denoted as $p_{S,x}$ (Fig.3.2). The vertical position of the L_S frame ($p_{S,z}$) is controlled solely by the robot arm, while its position along the row of grapevines ($p_{S,x}$) is the sum of mobile base and robot arm components of that position, $p_{B,x}$ and $p_{A,x}$ respectively:

$$p_{S,x} = p_{B,x} + p_{A,x} \tag{3.1}$$

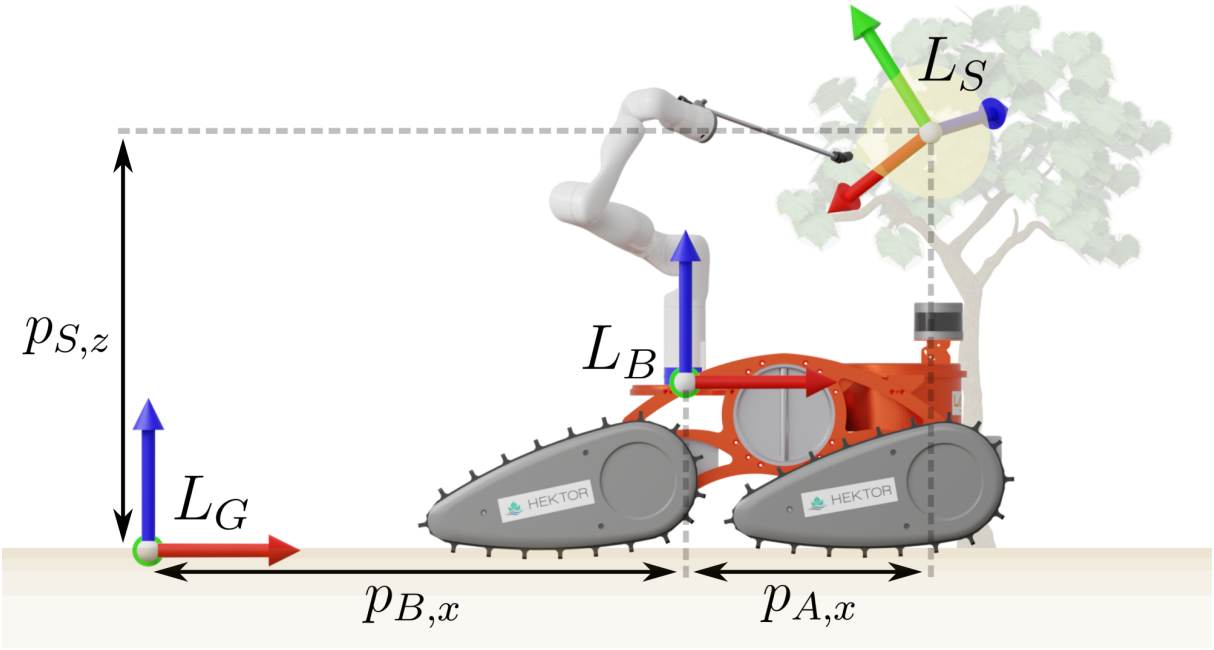

Figure 3.2: Three coordinate frames defined for the task space model predictive control algorithm: a global frame at the ground level L_G, mobile base frame L_B and the spray frame L_S. The x, y and z axes of the coordinate frames are represented with red, green and blue arrows respectively.

The redundancy in the control of $p_{S,x}$ is clearly visible from equation3.1. The desired movement of the spray frame along the row of grapevines can be produced in multiple ways, by mobile base or robot arm movement in that direction, or a combination of the two. The MPC algorithm serves to predictively resolve this redundancy, using knowledge of the future changes in reference trajectory to select the optimal controls for the mobile base and the robot arm. Considering task space rather than joint space motion commands for the robot arm significantly simplifies the MPC problem, allowing the use of a linear system model.

The overall control diagram used for continuous spraying is shown in Fig.3.3. A two-dimensional reference trajectory for the spray frame is used as an input for a reference tracking MPC solver. The MPC solver resolves the already discussed redundancy, and selects the coordinated motion of the mobile base and the robot arm that track this reference. The output of the MPC solver is the mobile base velocity along the row, and the two-dimensional task space velocity for the robot arm.

The reference spray frame trajectory is generated from the canopy description, with the aim of providing full coverage of the canopy area. The canopy description consists of its upper and lower boundary, shown with a blue and a red line respectively (Fig.3.4). The spray frame reference is a lawnmower trajectory, traversing the area between the provided canopy boundaries. The *Height Offset* and *Spraying Width* parameters shape the trajectory, depending on the width of the spraying agent produced by the nozzle, and its distance to the vines. A constant velocity piecewise linear functions are used in between the lawnmower pattern vertices.

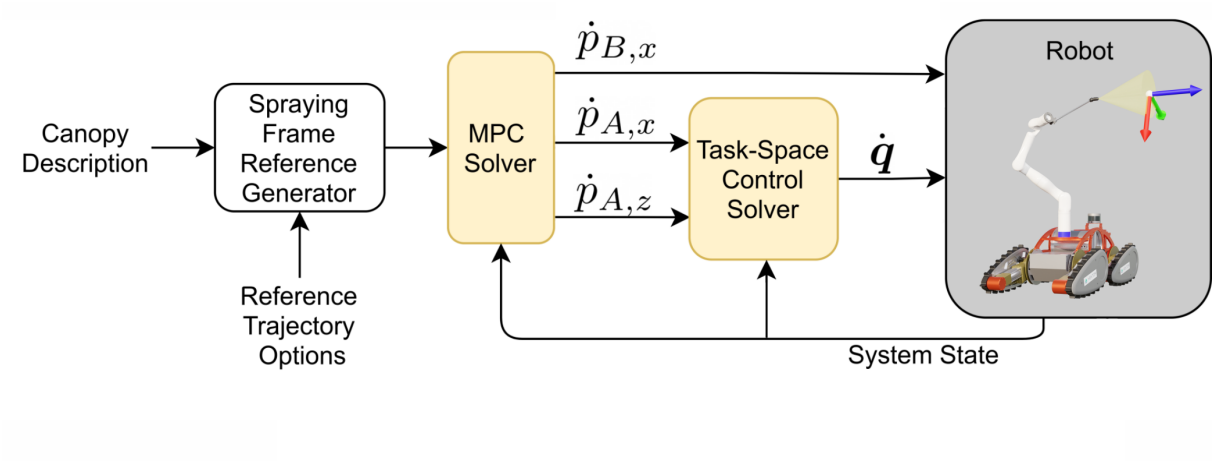

Figure 3.3: Overall system control diagram. The reference trajectory for the spray frame is generated based on the canopy description. This trajectory is used as an input to the MPC solver. The MPC solver selects the velocity of the mobile base along the row of vines, denonted as $\dot{p}_{B,x}$, and the task space velocities for the robot arm $\dot{p}_{A,x}$ and $\dot{p}_{A,z}$. The task space control solver is used to select the joint velocity commands sent to the robot arm $\dot{\boldsymbol{q}}$. Copyright [8]CC BY 4.0.

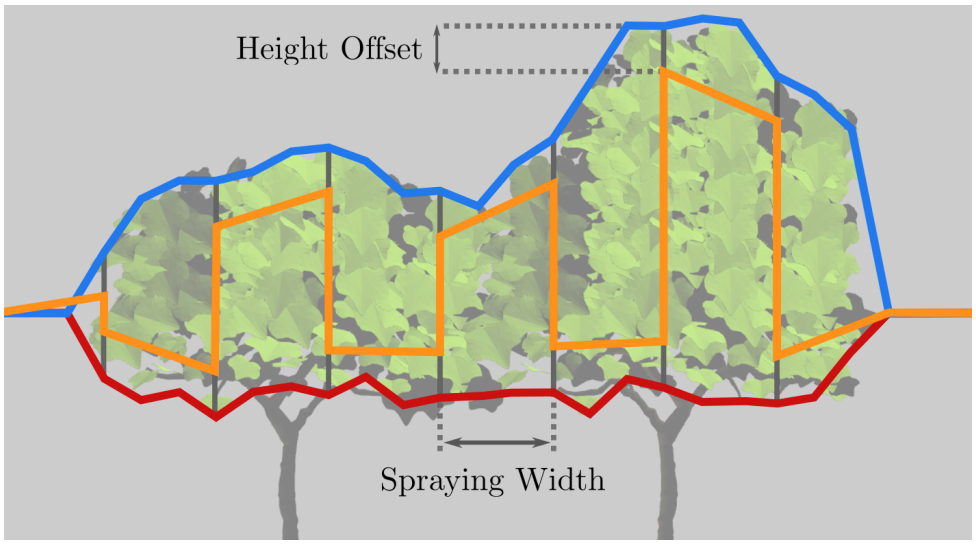

Figure 3.4: Orange line represents the lawnmower trajectory used as the spray frame reference. The upper and lower boundary of the canopy description are represented by a blue and a red line respectively. Copyright [8]CC BY 4.0.

3.2.1.1System Model and MPC Optimization Problem

A reference tracking linear MPC is used to select the trajectories for the mobile base and the robot arm that follow a two-dimensional spray frame reference, by solving an optimization problem described in this section. The optimization is performed over a prediction horizon N_h, with a fixed discretization time step T. Although the velocities of the mobile base and robot arm are used to control the robot (as seen in Fig.3.3), the model used in the MPC algorithm considers acceleration control inputs to achieve the desired behavior.
A linear system model used by the MPC has the following states and inputs:

$$\boldsymbol{x}(k) = \begin{bmatrix} p_{B,x}(k) \\ p_{A,x}(k) \\ p_{S,z}(k) \\ \dot{p}_{B,x}(k) \\ \dot{p}_{A,x}(k) \\ \dot{p}_{S,z}(k) \end{bmatrix} \in \mathbb{R}^6 \quad \boldsymbol{u}(k) = \begin{bmatrix} \ddot{p}_{B,x}(k) \\ \ddot{p}_{A,x}(k) \\ \ddot{p}_{S,z}(k) \end{bmatrix} \in \mathbb{R}^3, \tag{3.2}$$

where $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$ are the system state and control input vector respectively, at the time step k.
The output of the system is the two-dimensional position of the spray frame L_S, relative to the global frame L_G:

$$\boldsymbol{y}(k) = \begin{bmatrix} p_{S,x}(k) \\ p_{S,z}(k) \end{bmatrix} = \begin{bmatrix} p_{B,x}(k) + p_{A,x}(k) \\ p_{S,z}(k) \end{bmatrix} \in \mathbb{R}^2, \tag{3.3}$$

where $\boldsymbol{y}(k)$ is the system output in the k-th step.
The dynamics of the discrete linear system used by the MPC can be written as follows:

$$\boldsymbol{x}(k+1) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x}(k) + \underbrace{\begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & \frac{T^2}{2} \\ T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{u}(k) \tag{3.4}$$

$$\boldsymbol{y}(k) = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{C}} \boldsymbol{x}(k) \tag{3.5}$$

The optimization problem used for reference tracking MPC is a QP problem, with a criterion function formulated as previously discussed in section2.6:

$$\begin{aligned} \min_{\boldsymbol{U}} \quad & W_y||\boldsymbol{Y}-\boldsymbol{Y}_d||^2+||\boldsymbol{W}_u\boldsymbol{U}||^2+||\boldsymbol{W}_x\boldsymbol{X}||^2 \\ \text{s.t.} \quad & \boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}\boldsymbol{u}(k) \\ & \boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k) \\ & \boldsymbol{x}(0)=\boldsymbol{x}_0 \\ & \underline{\boldsymbol{x}} \preceq \boldsymbol{x}(k) \preceq \overline{\boldsymbol{x}} \\ & \underline{\boldsymbol{u}} \preceq \boldsymbol{u}(k) \preceq \overline{\boldsymbol{u}} \end{aligned} \tag{3.6}$$

where $\boldsymbol{Y}_d$ is the system output reference and W_y is the reference tracking weight. $\boldsymbol{W}_u$ and $\boldsymbol{W}_x$ are control input and system state weight matrices, respectively. For more details on the used form of the criterion function the reader is referred to section2.6.

The reference tracking part of the criterion function ($W_y||\boldsymbol{Y}-\boldsymbol{Y}_d||^2$) ensures that the system output follows the reference lawnmower trajectory, shown in Fig.3.4. The second part of the criterion function ($||\boldsymbol{W}_u\boldsymbol{U}||^2$) is employed to minimize a general linear function of the system inputs. For the specific use case of continuous spraying, only a single control input is minimized, corresponding to the acceleration of the vehicle along the row. The control input weight matrix used for this purpose is as follows:

$$\boldsymbol{W}_u=\begin{bmatrix} \boldsymbol{w}_u & 0 & \cdots & 0 \\ 0 & \boldsymbol{w}_u & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{w}_u \end{bmatrix}, \quad \boldsymbol{w}_u=\begin{bmatrix} w_{\ddot{p}_{B,x}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.7}$$

where the scalar $w_{\ddot{p}_{B,x}}$ represent the mobile base acceleration weight.

The third and final part of the criterion function ($||\boldsymbol{W}_x\boldsymbol{X}||^2$) minimizes a linear function of the system state vector. In this specific use case, the displacement of the robot arm in the x direction is minimized. This is achieved with a weight matrix of the following form:

$$\boldsymbol{W}_x=\begin{bmatrix} \boldsymbol{w}_x & 0 & \cdots & 0 \\ 0 & \boldsymbol{w}_x & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{w}_x \end{bmatrix}, \quad \boldsymbol{w}_x=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{p_{A,x}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

To summarize, the criterion function of the MPC optimization problem minimizes three distinct criteria: the reference tracking error, mobile base acceleration and robot arm displacement. The reasoning behind this choice of the criterion function is to resolve the redundancy exhibited by the system in a specific way, where the mobile base is responsible for gradual, global changes in the reference trajectory, and the robot arm is responsible for rapid, local ones. To achieve this behaviour, the optimization problem has to include an appropriate set of constraints, and the weights used in the criterion function have to be selected in a specific way.

3.2.1.2MPC Constraints and Parameter Tuning

The performance of the MPC algorithm and its ability to achieve the described behaviour depend on the constraints within the optimization problem and the tuning of the weights present in the criterion function.

The MPC optimization problem (3.6) can include bound constraints on any of the system states and control inputs. For the presented use case of continuous spraying, the general state and control input bound constraints can be written as:

$$\begin{bmatrix} -p_{B,x}^{\max} \\ -p_{A,x}^{\max} \\ -p_{S,z}^{\max} \\ -\dot{p}_{B,x}^{\max} \\ -\dot{p}_{A,x}^{\max} \\ -\dot{p}_{S,z}^{\max} \end{bmatrix} \preceq \boldsymbol{x}(k) \preceq \begin{bmatrix} p_{B,x}^{\max} \\ p_{A,x}^{\max} \\ p_{S,z}^{\max} \\ \dot{p}_{B,x}^{\max} \\ \dot{p}_{A,x}^{\max} \\ \dot{p}_{S,z}^{\max} \end{bmatrix} \tag{3.9}$$

$$\begin{bmatrix} -\ddot{p}_{B,x}^{\max} \\ -\ddot{p}_{A,x}^{\max} \\ -\ddot{p}_{S,z}^{\max} \end{bmatrix} \preceq \boldsymbol{u}(k) \preceq \begin{bmatrix} \ddot{p}_{B,x}^{\max} \\ \ddot{p}_{A,x}^{\max} \\ \ddot{p}_{S,z}^{\max} \end{bmatrix} \tag{3.10}$$

While these constraints are easily incorporated in the MPC optimization problem, the number of inequality constraints has a significant effect on its execution time. Therefore, it is generally desirable to reduce the number of constraints as much as possible, while achieving the desired behaviour. To achieve the desired behaviour where the robot arm is responsible for local movement of the spray frame in the x direction, its position is constrained to remain in between some bounds $-p_{A,x}^{\max} \leq p_{A,x} \leq p_{A,x}^{\max}$.

The bound constraints on control inputs have a lesser effect on the time required to solve the optimization problem. The limit on the acceleration of the mobile base $\ddot{p}_{B,x}^{\max}$ is known, and is directly enforced as $-\ddot{p}_{B,x}^{\max} \leq \ddot{p}_{B,x} \leq \ddot{p}_{B,x}^{\max}$. The maximum values of the robot arm accelerations in the task space $\ddot{p}_{A,x}^{\max}$ and $\ddot{p}_{S,z}^{\max}$ are determined experimentally. While the velocity and acceleration limits of each joint are known, the maximum acceleration the robot arm can produce in

the task space depends on the current arm configuration. Therefore, the task space acceleration constraints used in the MPC optimization problem are determined experimentally, and the joint space constraints are enforced at the level of the task space control algorithm (Section3.2.2).

While the constraints themselves are sufficient to find feasible solutions to the MPC problem, tuning the weights of individual objectives of the optimization problem (3.6) affects the balance between mobile base and robot arm motion used for redundancy resolution. The weights minimizing arm displacement $p_{A,x}$ and mobile base acceleration $\ddot{p}_{B,x}$ are tuned by trial and error to achieve the desired effect where the mobile base is responsible for slower, global changes in the reference trajectory and the arm is responsible for faster, local changes. Extreme examples of possible results of the weight tuning process are shown in Fig.3.5, with the corresponding criterion function weights presented in Table3.1.

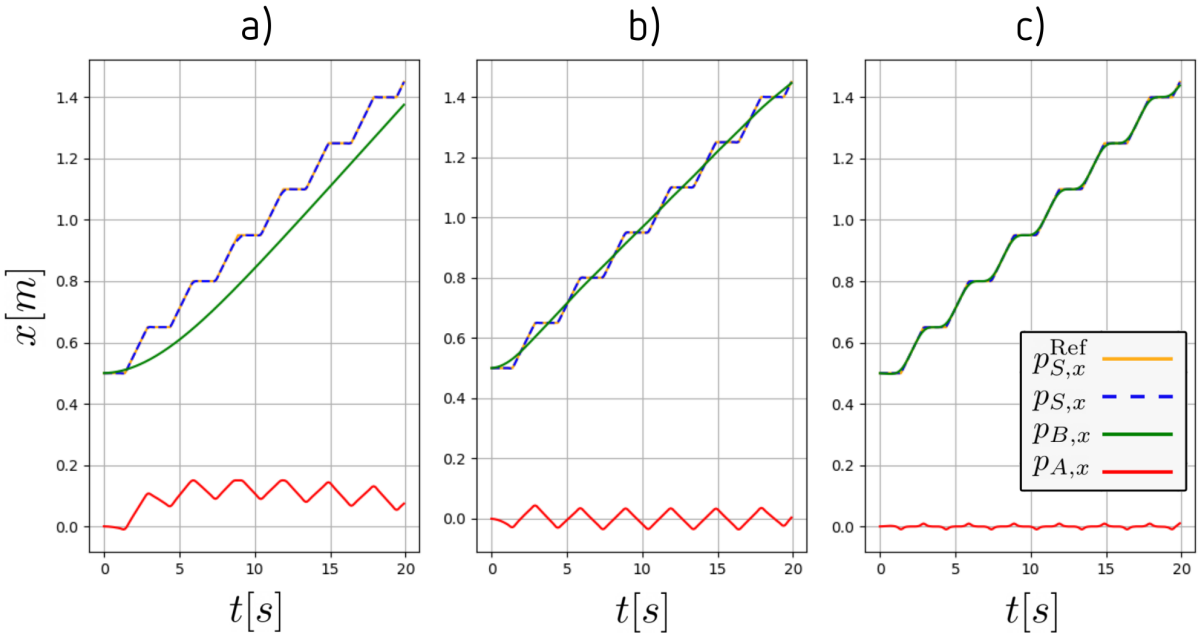

Figure 3.5: The trajectories of the mobile base and the robot arm in the x direction corresponding to the criterion function weights given in Table3.1. Spray frame trajectory $p_{S,x}$ is a sum of $p_{A,x}$ and $p_{B,x}$. In all the examples reference tracking is achieved, with extreme examples of varying redundancy resolution behaviours.

Table 3.1: Weights used in the MPC criterion function corresponding to the examples shown in Fig.3.5.

Fig.3.5	*a)*	*b)*	*c)*
W_y	800.0	800.0	800.0
$w_{\ddot{p}_{B,x}}$	80.0	1.5	1.0
$w_{p_{A,x}}$	1.0	1.0	10.0

The behaviour of the system depends on the ratios between the different optimization weights rather than their exact value. In the examples shown in Fig.3.5, the system assumes an initial state with zero velocity. Large values of the weight corresponding to the vehicle acceleration

$w_{\ddot{p}_{B,x}}$ result in large arm displacement while the base accelerates (Fig.3.5a)). Lowering the value of $w_{p_{A,x}}$ has a similar effect. Large arm displacement weight $w_{p_{A,x}}$ (Fig.3.5 c)) leads to the effect where the mobile base alone tracks the reference with large accelerations, while the robot arm displacement remains close to zero. The desired behaviour is shown in Fig.3.5b), where the base follows the general shape of the reference trajectory with low accelerations, and the robot arm moves rapidly with small displacements.

3.2.2Prioritized Task Space Control

Prioritized task space control is utilized to select the joint velocity commands that result in the desired task space motion provided by the MPC (Fig.3.3). In addition to following the linear velocity of the spray frame commanded by the MPC solver, the selection of joint velocity commands also considers the orientation of the nozzle. The nozzle is symmetric around the z axis of the spray frame, and its orientation around this axis does not impact the application of the spraying agent. Therefore, the task space control algorithm only needs to consider two components of the spray frame's orientation, a characteristic of 3T2R tasks.

As already mentioned, the area affected by the spraying agent is considered to be of larger importance than the exact direction of the nozzle [48,49,50]. Therefore, the translational component of the 3T2R task is chosen to be prioritized over its rotational component. This is achieved using prioritized quadratic programming, explained in detail in section2.5.1. Prioritized optimization is utilized to achieve task space control handling multiple, conflicting tasks with clear priorities, also referred to as prioritized task space control. The control algorithm aims to achieve the commanded orientation of the spray frame, but its translation is considered to be of a higher priority.

Priorities used for the continuous spraying are, with descending priorities:

- Translational part of the 3T2R task
- Rotational part of the 3T2R task
- Desired joint positions

The cost function of the first priority has the following form:

$$E_1(\dot{\boldsymbol{q}}) = ||\boldsymbol{J}_T\dot{\boldsymbol{q}} - \mathbf{v}_c||^2 \tag{3.11}$$

where $\mathbf{v_c}$ is the commanded linear velocity of the spray frame, and $\boldsymbol{J}_T$ is the translational part of the spray frame Jacobian. The commanded linear velocity $\boldsymbol{v_c}$ is constructed using the output

of the MPC solver and the assumption that the distance from the robot to the vines is constant:

$$\boldsymbol{v_c} = \begin{bmatrix} \dot{p}^c_{A,x} \\ K_{P,y}(p^c_{A,y} - p_{A,y}) \\ \dot{p}^c_{A,z} \end{bmatrix} \tag{3.12}$$

Since the MPC algorithm considers a two-dimensional task space, it only provides two components of the linear spray frame velocity, $\dot{p}^c_{A,x}$ and $\dot{p}^c_{A,z}$. The y component of the commanded spray frame velocity is calculated using a proportional controller, where $p^c_{A,y}$ and pA,y denote the commanded and the currently measured position of the spray frame in the y direction, respectively.

Generally, the task of following the commanded linear velocity of the spray frame is redundant, and there are multiple joint velocity vectors $\dot{\boldsymbol{q}}$ that achieve it. The criterion function of the second priority deals with the rotational part of the 3T2R task:

$$E_2(\dot{\boldsymbol{q}}) = ||\boldsymbol{J}^L_{R,x}\dot{\boldsymbol{q}} - \omega^L_{c,x}||^2 + ||\boldsymbol{J}^L_{R,y}\dot{\boldsymbol{q}} - \omega^L_{c,y}||^2 \tag{3.13}$$

where $\omega^L_{c,x}$ and $\omega^L_{c,y}$ are commanded angular velocities of the spray frame around its local x and y axes respectively, and $\boldsymbol{J}^L_{R,x}$ and $\boldsymbol{J}^L_{R,y}$ are the corresponding Jacobian matrices. As already mentioned, the angular velocity around the local z axis of the spray frame does not affect task execution, and is not directly controlled. The use of angular velocities around local axes results in a more complex calculation of their commanded values, explained in section3.2.2.1.

The final priority, used to resolve any redundancy remaining after minimizing the first two priorities, favors such joint velocities $\dot{\boldsymbol{q}}$ that move the arm towards a desired configuration:

$$E_3(\dot{\boldsymbol{q}}) = ||\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_c||^2 \tag{3.14}$$

The commanded joint velocities $\dot{\boldsymbol{q}}_c$ that drive the robot arm towards a desired pose $\boldsymbol{q}_d$ are selected by a proportional controller:

$$\dot{\boldsymbol{q}}_c = K_{P,q}(\boldsymbol{q}_d - \boldsymbol{q}) \tag{3.15}$$

where $K_{P,q}$ is the controller gain and $\boldsymbol{q}$ is a current joint position vector.
Inequality constraints are used to enforce joint velocity and acceleration limits at the task space control level:

$$\underline{\dot{\boldsymbol{q}}} \preceq \dot{\boldsymbol{q}} \preceq \overline{\dot{\boldsymbol{q}}} \tag{3.16}$$

$$\underline{\ddot{\boldsymbol{q}}} \preceq \ddot{\boldsymbol{q}} \preceq \overline{\ddot{\boldsymbol{q}}} \tag{3.17}$$

Since the prioritized task space control problem deals with joint velocities, equation (3.17) is

replaced with the one in the velocity space:

$$\dot{\boldsymbol{q}}_P + \underline{\ddot{\boldsymbol{q}}}\Delta t \preceq \dot{\boldsymbol{q}} \preceq \dot{\boldsymbol{q}}_P + \overline{\ddot{\boldsymbol{q}}}\Delta t \tag{3.18}$$

where Δt is the control time step for the task space control (not necessarily the same as MPC control step T), and $\dot{\boldsymbol{q}}_P$ are joint velocities in the previous time step.

To summarize, the prioritized optimization approach to spray frame control leads to the following behaviour: if both the commanded linear and angular velocities of the spray frame are simultaneously achievable, the prioritization between these tasks is not activated. Both tasks are achieved, and a single solution is selected, achieving the commanded 3T2R velocity while minimizing the final priority. However, if the 3T2R task velocity is not feasible in its entirety, its linear component is prioritized over its rotational component.

Whether the 3T2R task velocity is feasible depends on the current configuration of the robot arm and various constraints that can be incorporated into the optimization problem, as discussed in section3.2.3.

3.2.2.1Rotational Component of the 3T2R Task

The local angular velocity commands of the spray frame $\omega^L_{c,x}$ and $\omega^L_{c,y}$ are calculated based on the error between the desired and current approach axis orientation. If $\boldsymbol{app}_z$ and $\boldsymbol{app}_{z,d}$ are the current and the desired approach axis vector, respectively, the angular approach axis error err_α is calculated as the shortest angular distance between them:

$$err_\alpha = \arccos(\boldsymbol{app}_z \cdot \boldsymbol{app}_{d,z}) \tag{3.19}$$

The axis around which the angular error acts is calculated as:

$$\boldsymbol{err}_{axis} = \boldsymbol{app}_z \times \boldsymbol{app}_{d,z} \tag{3.20}$$

Assuming vectors $\boldsymbol{app}_z$ and $\boldsymbol{app}_{z,d}$ are defined with respect to some base frame, the angular error vector in the local spray frame is:

$$\boldsymbol{\alpha}^L_{err} = {}_L\boldsymbol{R}^B(err_\alpha \cdot \boldsymbol{err}_{axis}), \tag{3.21}$$

where ${}_L\boldsymbol{R}^B \in \mathbb{R}^{3\times3}$ is the corresponding rotation matrix.

If the z axis of the frame is considered its approach axis, the z component of $\boldsymbol{\alpha}_{err}^{L}$ will always be zero, and commanded local angular velocities can be calculated as:

$$\boldsymbol{\omega}_{c}^{L} = K_{P,\omega}\boldsymbol{\alpha}_{err}^{L} = \begin{bmatrix} \omega_{c,x}^{L} \\ \omega_{c,y}^{L} \\ 0 \end{bmatrix} \tag{3.22}$$

where $K_{P,\omega}$ is the associated proportional controller gain.

3.2.3Results and Discussion

The first experiment described in this section demonstrates continuous spraying of a single row of grapevines using a mobile manipulator (Fig.3.6). Footage of the experiment can be seen in the accompanying video*. In this experiment, a previous version of task space control algorithm is used, controlling only the position of the spray frame and disregarding its orientation. The reasoning behind this decision was that, in order to achieve large linear velocities of the spray frame and reach high and low areas of the foliage, it is not possible to fully control the orientation of the spray frame. As demonstrated through a second set of experiments discussed in this section, this problem is successfully addressed by the prioritized 3T2R task control algorithm discussed in this thesis.

Figure 3.6: Mobile manipulator performing a continuous spraying experiment in a vineyard. A single row of grapevines is sprayed using the presented task space MPC based method. Copyright [8]CC BY 4.0.

*https://www.youtube.com/watch?v=BDO7qQldmyQ

Figure 3.7: The canopy description used for the continuous spraying experiment in a vineyard, selected by an operator using a simple graphical user interface. Copyright [8]CC BY 4.0.

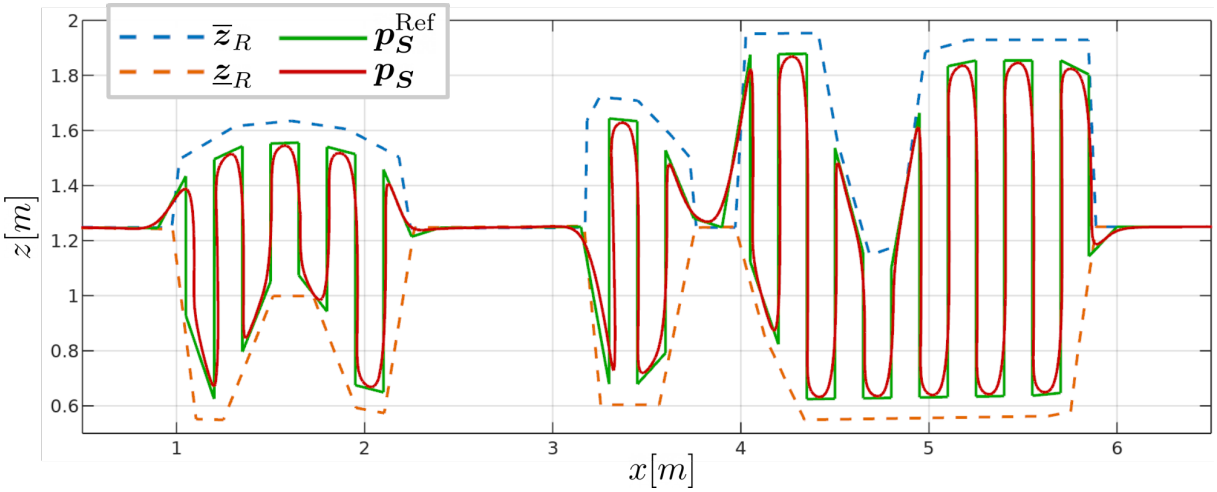

Figure 3.8: The lawnmower reference trajectory and reference tracking during the continuous spraying experiment in the vineyard. The reference trajectory generated based on the row description is denoted as $\boldsymbol{p}_S^{\text{Ref}}$, where $\overline{z}_R$ and $\underline{z}_R$ represent the upper and lower foliage boundaries respectively. The trajectory of the spray frame during the experiment $\boldsymbol{p}_S$ is represented by a red line. Copyright [8]CC BY 4.0.

The canopy description used to generate the reference spray frame trajectory is shown in Fig.3.7. For the purpose of the experiment, the description was manually selected by a human operator through a simple graphical user interface (GUI). A canopy segmentation algorithm could be utilized to generate this description autonomously using visual feedback, while the vehicle travels along the row of grapevines. Here, the emphasis is on the control aspect of the problem, demonstrating the ability of the system to perform continuous spraying on an uneven canopy, using its description.

A graph demonstrating the spray frame trajectory tracking during the experiment is given in Fig.3.8. Here, the position of the spray frame is calculated based on the encoder feedback from the robot arm, and the odometry of the mobile base. Because of the existence of acceleration constraints on control inputs, there is a trade-off between the reference tracking error and the

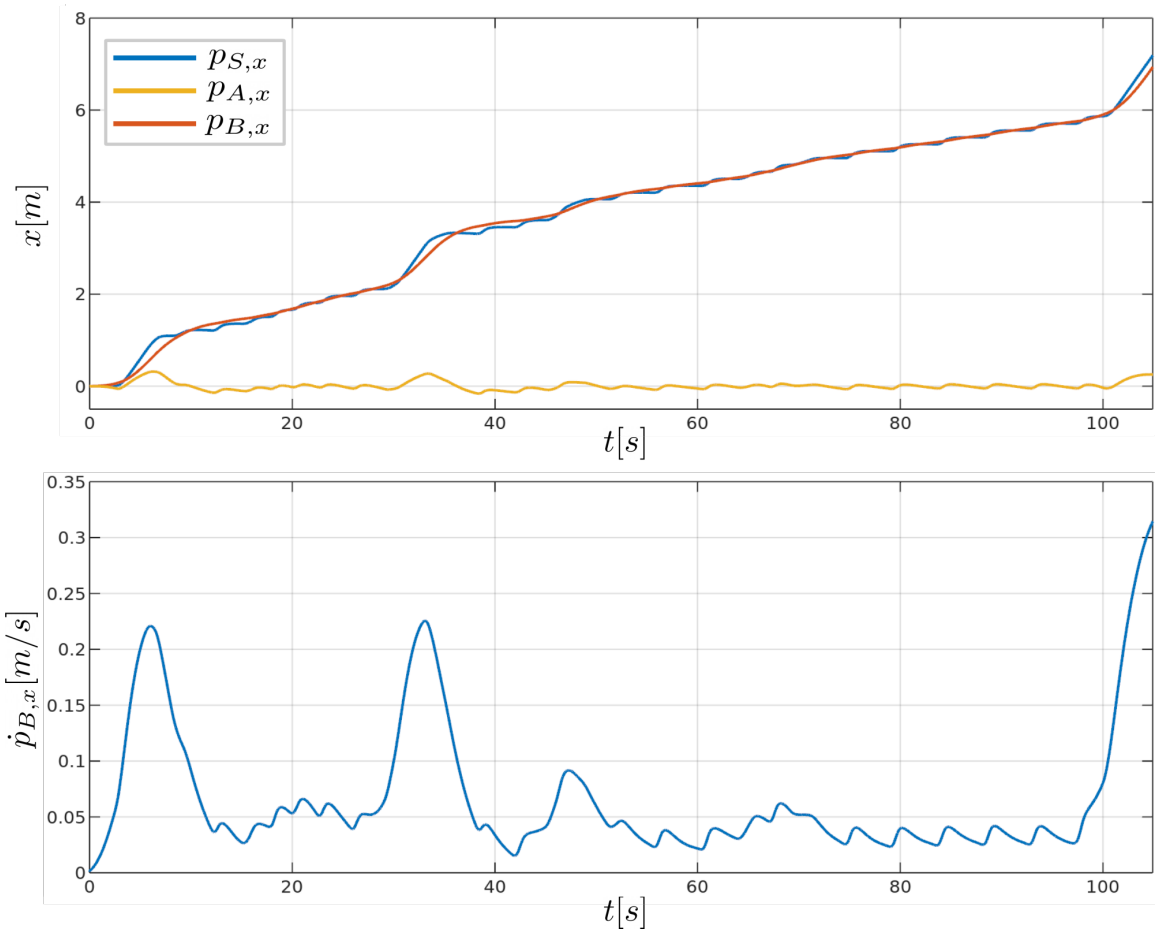

Figure 3.9: The upper diagram shows the x component of the reference trajectory $p_{S,x}$, along with the robot arm and mobile base components, $p_{A,x}$ and $p_{B,x}$ respectively. The bottom graph shows the forward velocity of the vehicle during the experiment. Copyright [8]CC BY 4.0.

velocity of the reference spray frame trajectory. Reducing this velocity would result in slower task execution, but would also reduce the reference tracking error.

A non-uniform canopy shape used in the experiment demonstrates the adaptability of the presented approach. The velocity of the vehicle adapts to the row description by automatically accelerating in areas without vines and decelerating to the lowest velocities in areas with the highest vines (Fig.3.9).

The second set of experiments demonstrate the utility of the presented prioritized task space control scheme for continuous spraying. The implementation of this control scheme is based on the constrained prioritized task space control algorithm by de Lasa et al. [21]. This algorithm is implemented in C++ using the OSQP (Operator Splitting Quadratic Program) quadratic programming solver [55], and is available on GitHub[†]. The implementation is based on the *Eigen* linear algebra library, and allows for definition of a set of prioritized quadratic tasks and linear constraints. Unconstrained version of the prioritized quadratic optimization algorithm by de Lasa et al. [20] is also implemented in the same GitHub library.

[†]https://github.com/ivatavuk/ptsc_eigen

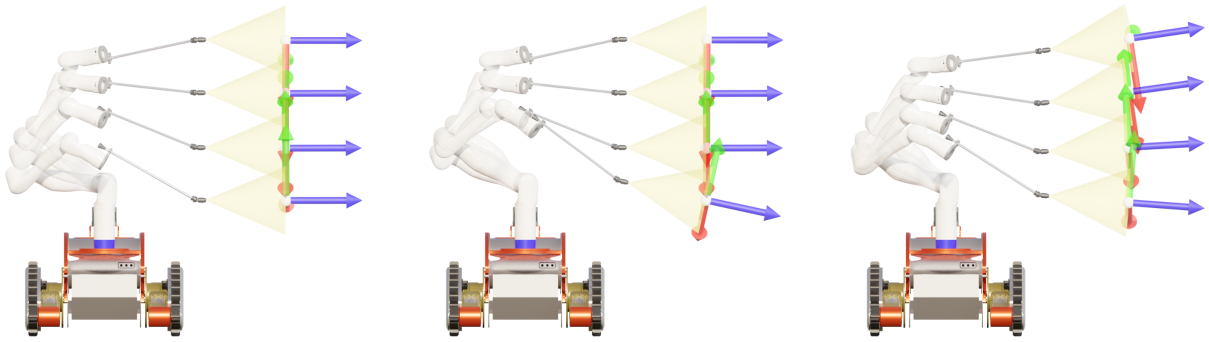

Figure 3.10: Prioritized 3T2R task control demonstration for three continuous spraying examples. The examples vary based on the commanded linear velocity of the spray frame, and constraints used in the optimization problem, and demonstrate, from left to right: slow spraying, positionally constrained slow spraying, fast spraying.

As already mentioned, depending on the velocity of the reference lawnmower trajectory, and the high and low limits of the canopy, it might not always be possible to achieve spraying with the desired orientation of the spray frame. The prioritization between the translational and rotational components of the spray frame control task was tested on three continuous spraying examples, with different commanded linear velocities and constraints (Fig.3.10). Footage of these experiments can be seen in the accompanying video[‡].

The rotation of the spray nozzle around its approach axis does not affect task execution, and, as the task is handled as a 3T2R control task, the spray frame rotates freely around its approach axis in all the examples (Fig.3.11). First of the three examples uses a low value of the commanded linear velocity of the spray frame (0.2 m/s), resulting in both the linear and rotational components of the 3T2R task being feasible during the entire trajectory. The 3T2R task is followed in its entirety as a result, and, if there are multiple joint velocities that achieve it, the third priority (Eq. (3.14)) is minimized to select a single solution between them.

The second example uses the same value of the commanded linear velocity (0.2 m/s), but with an addition of a positional constraint on a nozzle height. The nozzle is constrained to remain above the height of 0.3 m from the robot arm base. The inclusion of such a constraint is motivated by the inability of the spray frame to reach high or low areas of the canopy using the ideal orientation of the nozzle. During the lower segment of the trajectory, this positional constraint on nozzle height becomes active, which results in the prioritization between the translational and rotational component of the 3T2R task being noticeable (Fig.3.10). The solver selects a solution sacrificing ideal spray frame orientation in order to achieve the commanded linear velocity of the spray frame.

The third and final example uses a large commanded linear velocity of the spray frame (0.8 m/s), which results in joint velocity and acceleration constraints being active throughout the execution of the trajectory. As a consequence, the 3T2R task is not achievable for the most part

[‡] https://www.youtube.com/watch?v=FRdmGsSCAh4

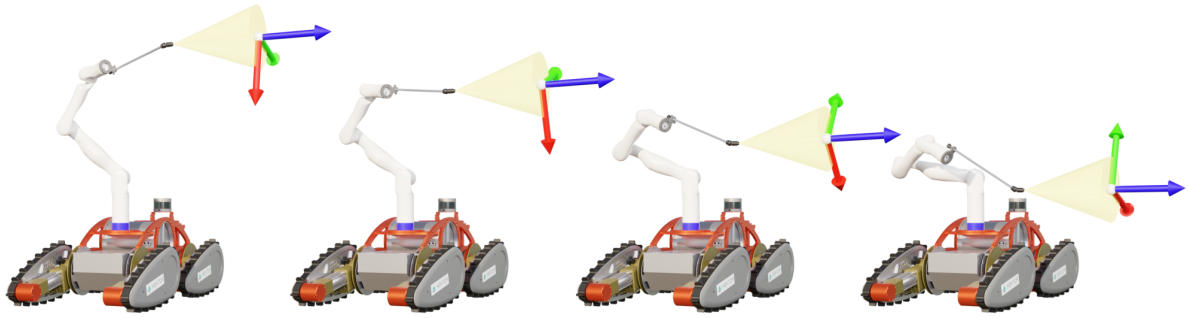

Figure 3.11: As a consequence of treating the control of the spray frame as a 3T2R task, the spray frame freely rotates around its approach axis during all the examples. If the 3T2R task is feasible in its entirety, the third priority of the prioritized task space control problem selects this rotation.

of the trajectory, and the third priority of the prioritized task space control scheme is disregarded. To combat this issue, in this example the prioritized optimization problem is modified to include only two priorities. The first priority is the same as in the previous examples (Eq.3.11), while the second priority is a weighted combination of the rotational component of the 3T2R task and desired joint movement:

$$E_2(\dot{\boldsymbol{q}}) = ||\boldsymbol{J}^L_{R,x}\dot{\boldsymbol{q}} - \omega^L_{x,d}||^2 + ||\boldsymbol{J}^L_{R,y}\dot{\boldsymbol{q}} - \omega^L_{y,d}||^2 + w||\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d||^2 \tag{3.23}$$

This results in the commanded linear velocity of the spray frame being followed, while its orientation is sacrificed in order to achieve this.

In summary, the utility of the presented method resides in the prioritization between the translational and rotational components of the 3T2R tasks. This prioritization addresses the challenge of not being able to achieve high spray frame linear velocities, or high and low areas of the canopy, with an ideal nozzle orientation. When this limitation is active, the presented method prioritizes achieving the commanded linear velocity over the nozzle orientation. The nozzle orientation achieved as a result of the prioritized optimization can be considered optimal under the circumstances.

3.3Selective Spraying

The term *selective spraying* is used to describe the task of spraying a specific part of a plant, for example a single leaf or a cluster of grapes. The application of robotic technology for selective spraying has the potential to be combined with advanced sensing techniques, utilizing multi-spectral cameras and other sensors to detect specific disease-ridden areas of the plants. This section discusses the proposed solution to the selective spraying problem from a manipulation perspective, assuming the desired position and orientation of the spray frame are known, and focusing on selecting the joint positions that achieve them.

The objective is to include the previously discussed prioritization between the translational and rotational components of the 3T2R task to selective spraying, performing the prioritization at the position rather than velocity level. A method for prioritized positional inverse kinematics is presented, inspired by the iterative positional inverse kinematics methods described in section 2.3.1.

3.3.1Prioritized Positional Inverse Kinematics

The standard positional inverse kinematics problem deals with selecting such joint positions $\boldsymbol{q}$ that result in the commanded pose of the end-effector. The goal of prioritized positional inverse kinematics is to expand the aforementioned problem with multiple, potentially conflicting positional tasks. Similarly to prioritized task space velocity control, the idea is to allow multiple task variants with hard priorities that are guaranteed to be satisfied.

To gain a better understanding of the underlying idea, consider an example where the desired end-effector pose can be achieved in multiple ways by a robot arm (Fig.3.12). Prioritized positional inverse kinematics allow for the introduction of a secondary positional task to fully constrain the inverse kinematics problem. In the concrete example shown in Fig.3.12, the position of the elbow frame is commanded as a lower priority task. Because of the hard prioritization of the tasks, different commanded elbow positions result in different joint positions, but do not affect the pose of the end-effector. The examples shown in Fig.3.12demonstrate the utility of prioritized inverse kinematics for redundancy resolution on a positional level. It is worth noting that the commanded elbow position is not achieved in either of the examples shown in Fig.3.12, but the solutions selected by the solver minimize the elbow position error, while respecting all the higher priorities.

In the case the commanded end-effector pose in infeasible by the robot arm, prioritized positional inverse kinematics can be used to prioritize its translational over its rotational component. The introduction of this prioritization does not change the outcome of the examples shown in Fig.3.12, since both the position and the orientation commands for the end-effector are feasible. However, if the commanded end-effector position is feasible but its pose is not fea-

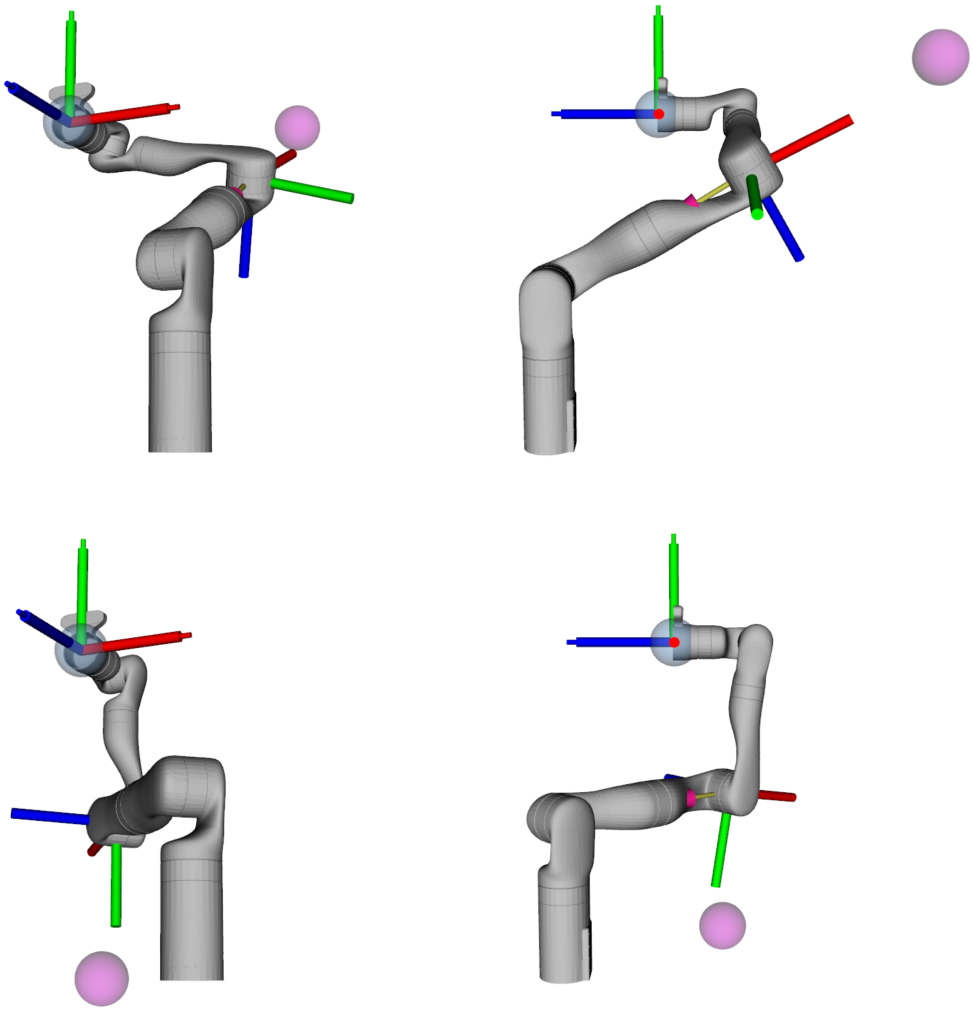

Figure 3.12: Prioritized positional inverse kinematics can be used for redundancy resolution at a positional level. The pose of the end-effector is prioritized over the elbow frame position, whose desired value is shown with a purple sphere. Out of the multiple solutions achieving desired the end-effector position, a single one is selected minimizing the elbow position error.

sible in its entirety, the solver selects the solution reaching the commanded end-effector position while minimizing its orientation error, shown in Fig.3.13. In this case, the solver completely disregards the commanded elbow position, since there is a unique solution that minimizes the end-effector orientation error, while remaining in its commanded position.

The final case that might be considered is the one where neither the position or the orientation of the end-effector are feasible, shown in Fig.3.14. In this case the solution to the first priority (e.g. minimizing the end-effector position error) has a unique solution, achieved by the robot arm "reaching" towards the commended position as much as possible, and all the other priorities are disregarded.

It is worth mentioning that all the previously discussed behaviours are achieved using an identical set of task priorities. The strength of prioritized positional inverse kinematics is that it performs redundancy resolution between different positional tasks based on the feasibility of

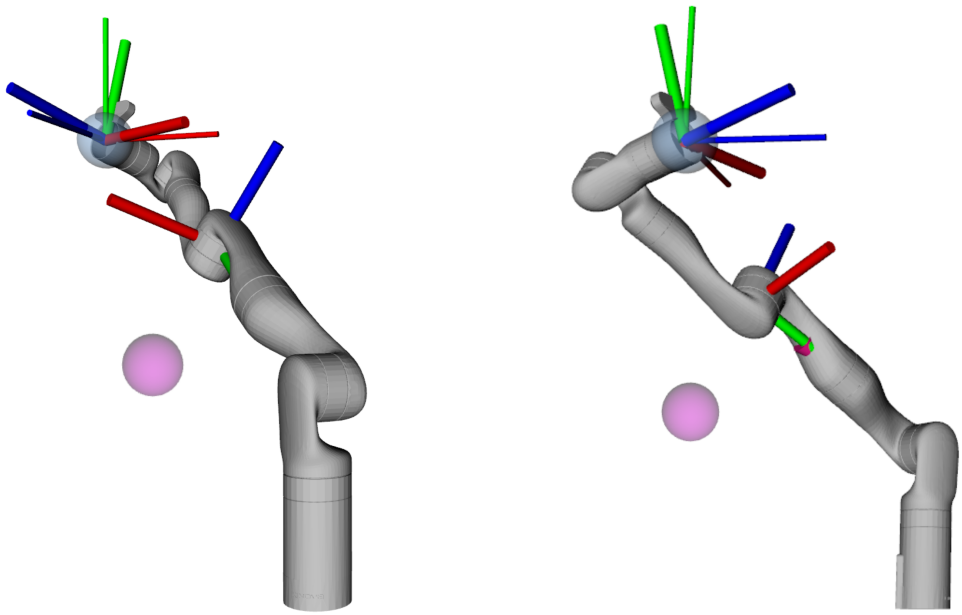

Figure 3.13: Prioritizing end-effector position over its orientation affects the solution if the desired end-effector pose is infeasible. The elbow position task is given third priority, and is disregarded by the solution in order to minimize the end-effector orientation error.

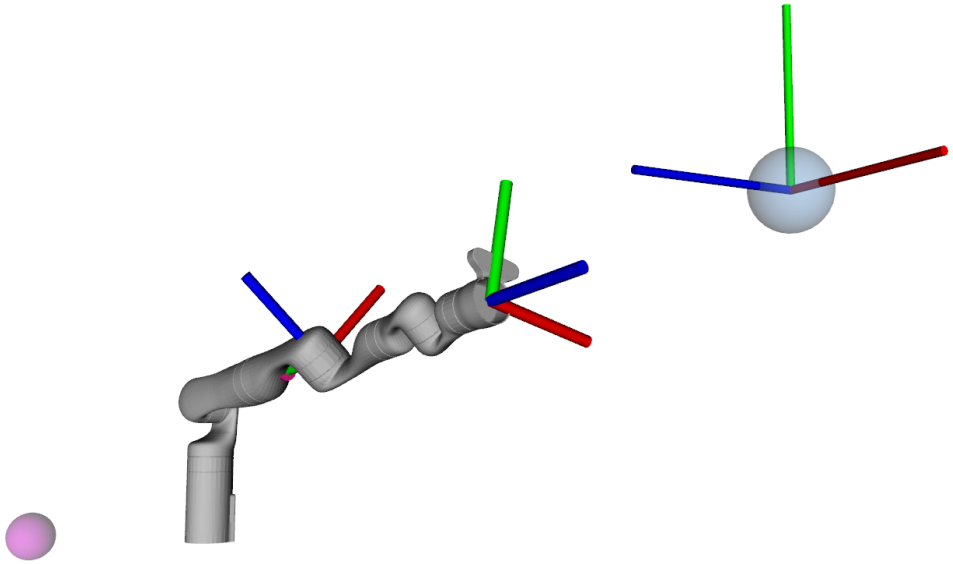

Figure 3.14: In the case none of the priorities are feasible, a solution is found minimizing the first priority, or, more specifically, the end-effector position error.

the commanded task values.

3.3.1.1Solver Implementation

The implementation of the prioritized positional inverse kinematics solver is inspired by standard iterative inverse kinematics methods, discussed in section2.3.1. These methods are based on iteratively solving the velocity level inverse kinematics problem, to achieve the commanded end-effector pose. In order to introduce the ability of handling multiple, potentially conflicting

positional tasks with different priorities, the presented solver replaces the standard velocity level problem with a constrained prioritized task space control problem.

The pseudoalgorithm for the prioritized inverse kinematics is given in Algorithm4. The algorithm shares multiple similarities with Algorithm1: both require an initial guess for joint positions $\boldsymbol{q}_{initial}$, both iteratively update the solution guess using the Jacobian matrix and the task error, and both use the error magnitude as the stopping criteria. The main difference lies in the way the solution step $\Delta\boldsymbol{q}$ is updated at each iteration of the algorithm, replacing the Jacobian pseudoinverse with a prioritized task space control algorithm.

$\boldsymbol{q} \leftarrow \boldsymbol{q}_{\text{initial}}$
tasks $\leftarrow []$
while $\sum \|\boldsymbol{err}_i\| \geq \varepsilon_e$ ***and*** $\sum \|\nabla \boldsymbol{err}_i\| \geq \varepsilon_\nabla$ **do**
  for $i \leftarrow 0$ **to** N **do**
    $\boldsymbol{J}_i \leftarrow \text{getTaskJacobian}(\boldsymbol{q}, \text{tasktype}_i)$
    $\boldsymbol{err}_i \leftarrow \text{getTaskError}(\boldsymbol{q}, \text{tasktype}_i)$
    $\nabla \boldsymbol{err}_i \leftarrow \text{updateGradient}(\boldsymbol{err}_i)$
    $\boldsymbol{err}_i \leftarrow \text{clampTaskError}(\boldsymbol{err}_i, \text{tasktype}_i)$
    tasks.insert$(\boldsymbol{J}_i, \boldsymbol{err}_i)$
  end
  $\Delta\boldsymbol{q} \leftarrow \text{solvePTSC}(\text{tasks}, \text{constraints})$
  $\boldsymbol{q} \leftarrow \boldsymbol{q} + \Delta\boldsymbol{q}$
  tasks.clear()
end

Algorithm 4: Positional prioritized inverse kinematics solver.

Algorithm4has an additional **for** loop, since there are N prioritized tasks instead of a single task considered by standard inverse kinematics. Task errors and Jacobians are calculated for each priority, based on the current joint position guess $\boldsymbol{q}$ and the specific type of the task. Another crucial difference between the algorithms is the inclusion of an additional stopping criterion, based on task error gradients. As already discussed in the previous section, the prioritized positional inverse kinematics problem often includes lower priority tasks which are infeasible. Therefore, as the solver approaches the solution, all the task errors do not necessarily approach zero. Error gradients are calculated as a difference between the task error in current and previous iteration of the algorithm. The error gradient serves as the measure of how much the last iteration of the algorithm improved the error of a specific task. A sum of all the gradients approaching zero is interpreted as reaching the solution to the overall prioritized inverse kinematics problem, as additional solver iterations do not improve the solution guess. The task error calculation and clamping depends on the type of the specific task, and follows similar principles to the ones discussed in section2.3.1. The solution update $\Delta\boldsymbol{q}$ is the solution to the prioritized task space control problem, constructed from task Jacobians, clamped errors and constraints bounding the solution.

The prioritized task space control problem used to calculate $\Delta \boldsymbol{q}$ is as follows:

$$\begin{aligned} h_i = \min_{\Delta \boldsymbol{q}} \quad & ||\boldsymbol{J}_i \Delta \boldsymbol{q} - \boldsymbol{err}_i||^2 \\ \text{s.t.} \quad & ||\boldsymbol{J}_l \Delta \boldsymbol{q} - \boldsymbol{err}_l||^2 = h_l, \ \forall l < i \\ & -\Delta \boldsymbol{q}_{lim} \preceq \Delta \boldsymbol{q} \preceq \Delta \boldsymbol{q}_{lim} \end{aligned} \tag{3.24}$$

Here, $\boldsymbol{J}_i$ and $\boldsymbol{err}_i$ are the Jacobian matrix and the clamped error for a task of the i-th priority, respectively, and $\Delta \boldsymbol{q}_{lim} > 0$ is a limit on the absolute value of the solution step $\Delta \boldsymbol{q}$.

The implementation of the presented algorithm in C++, *PikRos* (*Prioritized Inverse Kinematics ROS library*) is available on GitHub[§]. A widely used open-source library for robot manipulation *MoveIt* [56] is used to calculate the Jacobians of the specified links, which must be present in the *URDF* file of the *MoveIt* planning group. The implemented algorithm follows the general idea shown in the pseudoalgorithm4, while introducing additional implementation details and features such as various user defined parameters, solution polishing and iteration visualization for debugging purposes.

The *MoveIt* library allows for the calculation of a Jacobian corresponding to a pose of a specific link in the kinematic chain. The Jacobians obtained using *MoveIt* are modified to support any of the following tasks:

- Link pose task
- Link position task
- Link orientation task
- Link approach axis direction task

These task types correspond to the $tasktype_i$ variable in the pseudoalgorithm4. The link position and orientation task Jacobians correspond to the first and last three rows of the standard, pose Jacobian. The link approach axis direction task refers to the direction of a single axis of a link's coordinate frame, which corresponds to the rotational component of the 3T2R task. Jacobian and the error for the approach axis direction task type are calculated as described in section3.2.2.1.

The solution polishing feature is based on the idea of using larger steps $\Delta \boldsymbol{q}$ at the start of the algorithm, and smaller steps once the vicinity of the solution is reached. The current guess being near to the solution is detected as the sum of the error gradient norms reaching a certain threshold $\sum \|\nabla \boldsymbol{err}_i\| \leq \varepsilon_\nabla^p$. The value ε_∇^p is implemented as a user defined parameter, and $\varepsilon_\nabla^p > \varepsilon_\nabla$ has to hold for the solution polishing to be triggered. Once the vicinity of the solution is reached, $\Delta \boldsymbol{q}_{lim}$ is replaced with a more restricting limit $\Delta \boldsymbol{q}_{lim}^p$.

[§] https://github.com/ivatavuk/pik_ros

The library allows for the solver to be tuned using following user defined parameters:

- Joint step limit (scalar value used as each entry of the $\Delta \boldsymbol{q}_{lim}$ vector, in radians)
- Linear clamp magnitude (for tasks associated with link position, in meters)
- Angular clamp magnitude (for tasks associated with link orientation, in radians)
- Error norm threshold (ε_e)
- Error gradient norm threshold (ε_∇)
- Error gradient norm threshold for polishing (ε_∇^p)
- Polishing joint step limit (scalar value used as each entry of the $\Delta \boldsymbol{q}_{lim}^p$ vector, in radians)
- Use constrained optimization (if *false*, $\Delta \boldsymbol{q}_{lim}$ and $\Delta \boldsymbol{q}_{lim}^p$ are not enforced during optimization)
- Use solution polishing (*true* or *false*)
- Run in debug mode (if *true*, the robot movement is displayed for every iteration of the solver)
- Solver timeout (maximum allowed execution time in seconds)
- Maximum number of iterations (integer value)

3.3.2Prioritized Selective Spraying

The presented prioritized positional inverse kinematics method fits the requirements of the task of selective spraying, where a specific area of the plant needs to be sprayed with a mobile manipulator. The position of that area is assumed to be known, as is the desired orientation of the nozzle. One example of such a selective spraying problem would be to select the joint positions required to spray the lower surface of a single disease ridden leaf. While both the position and the desired orientation of the nozzle are commanded, the prioritization between these components used for continuous spraying remains for this use case.
Therefore, the prioritized positional inverse kinematics algorithm is used with the following task priorities:

- Spray frame position task
- Spray frame approach axis direction task
- Elbow frame position task

The commanded elbow frame position is used to fully constrain the solution in cases where the 3T2R task can be achieved in multiple ways, similar to the examples in Fig.3.12.

Four different examples of selective spraying are shown in Fig.3.15, with the desired values for all the tasks given in Table3.2. In the first two examples the desired values of the full 3T2R task are feasible, and the elbow position task fully constrains the problem. In the third example, the 3T2R task is not feasible in its entirety, resulting in a solution with the correct spray frame position, and minimal approach axis direction error for the commanded spray frame position. This example demonstrates the utility of the prioritization between the translational and the

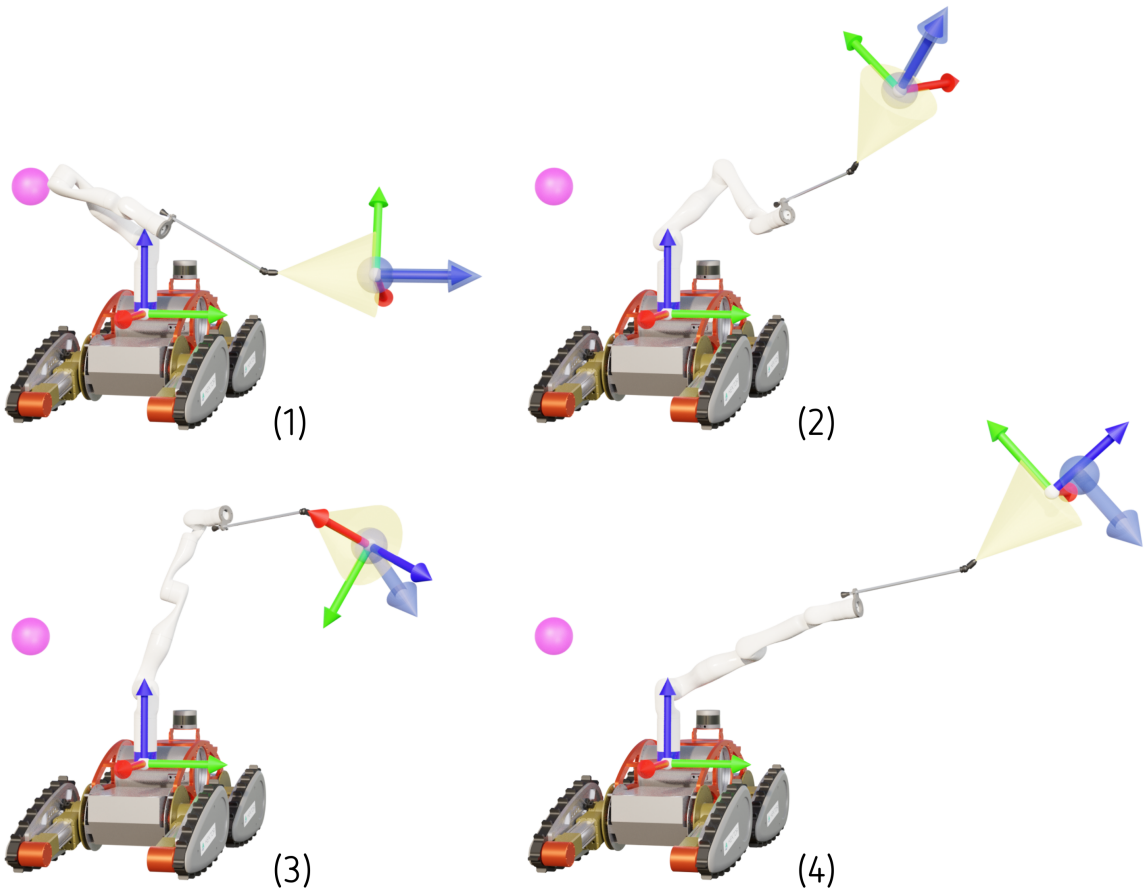

Figure 3.15: Prioritized positional inverse kinematics examples for the task of selective spraying. The desired position and approach axis direction of the spray frame are represented with a transparent blue sphere and arrow respectively, and the transparent purple sphere represents the desired elbow position.

Example	Spray frame position [m]	Spray frame approach axis vector	Elbow position [m]
1	[0.4 1.0 0.2]	[0 1 0]	[0.0 -0.5 0.5]
2	[0.4 1.0 0.8]	[0.511 0.511 0.69]	[0.0 -0.5 0.5]
3	[0.4 1.0 0.8]	[0.577 0.577 -0.577]	[0.0 -0.5 0.5]
4	[0.5 1.6 1.0]	[0.577 0.577 -0.577]	[0.0 -0.5 0.5]

Table 3.2: Desired values for prioritized tasks used in the examples.

rotational components of the 3T2R task: the desired part of the plant is sprayed, at an angle as close as possible to the commanded one. In the final example neither of the commanded tasks are feasible, corresponding to the previously discussed example shown in Fig.3.14. This behaviour where the first priority has a non-zero error at the solution is not useful for the use case of selective spraying, and can be treated as the selective spraying problem having no solution.

The solver performance was evaluated for the presented examples, with the results given in Tab.3.3. All experiments were conducted on a 2 $.2GHz$ Intel Core i7 processor. It can be noticed that the execution time is higher for later examples (examples 3 and 4), which is due

Example	Task 1 err [m]	Task 2 err [rad]	Task 3 err [m]	Time [ms]
1	0.00020	0.00019	0.3525	11.09
2	0.00054	0.00067	0.7222	19.86
3	0.00247	0.2052	0.6253	30.07
4	0.10456	1.359	0.87901	84.78

Table 3.3: Task errors and calculation time for the examples.

to the solution in these examples being close to singularity. This is even more noticable if solution polishing is not used, which can result in oscillatory behaviour of the solver around the singularity.

The same solver parameters are used in all the examples:

- Joint step limit = 10 [$^\circ$]
- Linear clamp magnitude = 0 .3 [m]
- Angular clamp magnitude = 30 [$^\circ$]
- Error norm threshold = 1 $\times 10^{-3}$
- Error gradient norm threshold = 1 $\times 10^{-3}$
- Error gradient norm threshold for polishing = 1 $\times 10^{-2}$
- Polishing joint step limit = 3 [$^\circ$]
- Use constrained optimization = *true*
- Use solution polishing = *true*
- Run in debug mode = *false*
- Solver timeout = ∞ [s]
- Maximum number of iterations = ∞

All three tasks are not feasible in any of the given examples, so the solver considers the problem solved once task error gradients reach a specified threshold. For most prioritized inverse kinematics applications the same would be the case, as the main strength of this approach is its ability to handle conflicting, infeasible tasks with clearly defined priorities.

In summary, the examples discussed in this section demonstrate the utility of prioritized positional inverse kinematics for the task of selective spraying. The reason why the nozzle is usually mounted at an angle to the pole of the spray wand is also made obvious by the examples. The angled nozzle allows for a wider range of achievable 3T2R poses, and allows for much more flexibility than would be possible with the nozzle pointing in the direction of the pole. In the third example, the solver determines that the optimal strategy to achieve the commanded positional 3T2R task is to rotate the base of the pole upside down. Achieving such angles of the spray nozzle would not be possible without the angled nozzle.

The presented prioritized inverse kinematics framework is planned to be expanded to accommodate additional task types, such as preferred joint positions and manipulability maximization

tasks. Future enhancements may also include positional constraints and voxel-based obstacle avoidance. This method is applicable to various robotic tasks, being particularly interesting for high dimensional floating base robot systems, which could have a large number of prioritized conflicting tasks.

3.4Summary

This chapter focuses on methodologies and algorithms developed for robotic vineyard spraying using a mobile manipulator. The scenario considered in this thesis revolves around a mobile manipulator equipped with a spray wand as its end-effector. Both the continuous and selective variant of the vineyard spraying task are discussed. The continuous spraying method presented in the thesis utilizes reference tracking model predictive control to resolve the redundancy between the mobile base and robot arm movement during the task. Additionally, the axis-symmetry of the spraying nozzle used in this thesis introduces another form of redundancy to the robot arm control problem. The problem of controlling the spraying nozzle with the manipulator is approached as prioritized 3T2R task control, prioritizing its linear movement over its angular component. This prioritization is also applied to positional inverse kinematics, corresponding to the selective spraying task. This chapter discusses the first contribution of this thesis:

- *Task space model predictive control based method for vineyard spraying with a mobile manipulator*

Associated Publications

Parts of this chapter have already been published by the author in the following article:

•Vatavuk, I., Vasiljevi ć G., and Kovačić, Z. (2022). Task Space Model Predictive Control for Vineyard Spraying with a Mobile Manipulator. Agriculture 2022, 12, 381. https://doi.org/10.3390/agriculture12030381 (IF = 3.6, Q1)

Additionally, some of the methodologies and results presented in this chapter are not currently published but are available as a preprint:

•Vatavuk, I., and Kova čić, Z. (2023). Constrained Prioritized 3T2R Task Control for Robotic Agricultural Spraying, available at: https://arxiv.org/abs/2307.00302

CHAPTER 4

Viticultural Suckering with a Mobile Manipulator

4.1Introduction

Modern vineyards use various plant training methods, referring to the methods of physical manipulation of a plant's form [57]. Plant training achieves multiple objectives such as optimizing leaf exposure to sunlight, allowing for better and more efficient disease control, and making manual work easier for the human workers [57]. Many contemporary plant training techniques rely on grapevine trellis that consist of multiple wires and posts which are used to train the plant. In most training methods, the fruit bearing vines are kept above a horizontal wire installed at a certain height, and all the growth emerging bellow it is removed.

This process is known as *suckering*, removing shoots, buds and suckers that emerge on that trunk area of the grapevine plant. The idea is to remove the unwanted growth that competes with the main, fruit-bearing vines for nutrients and water, potentially affecting the quality of the grape crop.

Performing the suckering task manually usually requires the worker to assume a bent position and remove the buds by hand (Fig.4.1). This method is tedious and time consuming, especially for larger vineyards. A number of hand-held suckering tools are developed with the idea of removing the need for the worker to crouch or bend to perform the suckering task manually (Fig.4.2a)). A curved brush is mounted at the end of a long handle, allowing the worker to use the brush bristles to remove the growth while standing up.

In addition to manual methods, there are mechanized solutions available for the viticultural suckering task. Most contemporary approaches to this mechanization use a form of a motorized

Figure 4.1: A human worker performing the suckering task. All the shoots and buds on the trunk are removed by hand, requiring the worker to assume a bent position. Copyright [58]CC BY 4.0.

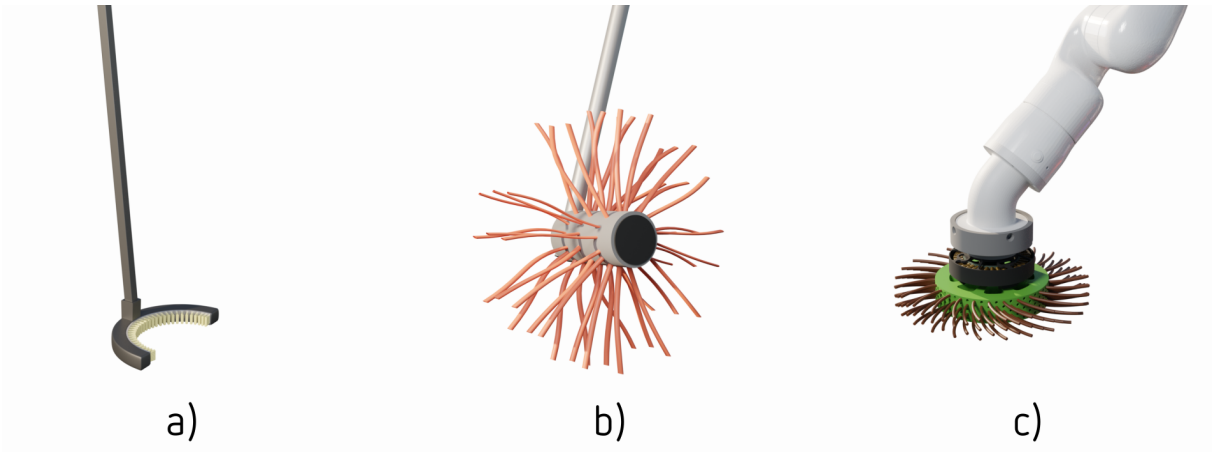

Figure 4.2: The design of different tools used for the suckering task in viticulture. The figure on the left shows a handheld tool used for manual suckering, consisting of a curved area with brush bristles at the end of a long handle. The figure in the middle shows a motorized flailing mechanism, commonly used in contemporary mechanization, consisting of long rubber strips actuated by a motor. The figure on the right shows the actuated, circular, brush-shaped robotic tool discussed in this thesis.

flailing mechanism, mounted on a tractor or some other human-operated vehicle (Fig.4.2b)). These mechanisms operate by continuously spinning the flails, which, upon impact with the plant, remove the unwanted growth. Such machinery sacrifices precision for fast execution of the suckering task. Multiple flailing mechanisms can be mounted on both sides of a vehicle, and a single traversal between the rows of plants will remove most, although usually not all, of the growth. In our previous work [59], a robotic suckering tool based on direct drive actuation with a functionality similar to that of a flailing mechanism is presented.

The objective driving the development of the suckering method discussed in this thesis is to provide the precision of manual suckering to an autonomous mobile manipulator. The aim is to remove the need for time-consuming and tedious manual work, without sacrificing precision. The proposed solution consists of a novel motorized brush-shaped suckering tool mounted as the end-effector of the robot arm (4.2c)), previously discussed in an article dedicated to its development [58]. The design of this tool combines different properties of both of the tools in Figures4.2a) and4.2b), combining a motorized, constantly spinning approach used by the flailing mechanism, with brush bristles instead of rubber flails.

This chapter is divided into two main sections, one discussing the development and properties of the proposed suckering tool (section4.2), and one dedicated to the control approach used for the suckering task (section4.3). The proposed tool design utilizes the direct drive design philosophy, allowing the actuator driving the tool to be used as a torque sensor. The torque sensing ability of the tool is utilized both to avoid damaging the plant and to provide the robot arm with tactile feedback. The latter is achieved using an estimated relationship between the torque exerted by the tool and the overlap between the brush bristles and the plant trunk.

The ability of the presented tool to be used as a tactile sensor is exploited to achieve compliant robot arm control. Compliant robot arm control is used for tactile exploration of the plant, allowing the suckering task to be performed without previously detecting the shape of a plant trunk. Like the nozzle used for the previously discussed task of vineyard spraying, the presented suckering tool is also axis-symmetric. Similar principles are applied to control the resulting 3T2R task, and resolve the redundancy caused by the tool's axis-symmetry. Finally, the torque sensing ability of the presented tool is also utilized for tactile mapping, using multiple occupancy grid maps to estimate the shape of the trunk during the suckering procedure.

4.1.1Related Work

Prior research has examined various methods for applying robotic technology to suckering and shoot thinning tasks [59,60,61,62,63,64,65]. Majeed et al. [61,62] report on computer vision algorithms used to detect vine cordon shapes, and on the use of a flailing mechanism for robotic shoot thinning. In [64], Martelloni et al. propose the use of flaming as an alternative to purely mechanical suckering methods. In [65], Polić et al. present a tactile plant exploration algorithm, using joint torque measurements to estimate the tactile feedback acting on the robot end-effector. This tactile feedback is also used to construct an estimate of the plant shape, and its reuse for future procedures is discussed. A possible use case of the presented tactile exploration algorithm for vine suckering is discussed.

The investigation of direct drive actuation and its application in robotics has been the focus of continuous research [66,67,68,69]. The direct drive design philosophy is predicated on the absence of mechanical reduction between the rotor and the mounting shaft of the actuator. This

lack of mechanical reduction results in maximal backdrivability of the actuator, enabling precise torque control with high transparency [70,71]. Due to the lack of gearing-related friction or losses, it is possible to achieve precise and high-bandwidth torque sensing through motor current measurements [70,71]. The driving force behind the development of direct drive actuation is its ability to offer transparency and high-bandwidth torque sensing without the need for external torque sensors. Interest in direct drive actuation emerged as early as 1983, with its early application in the field of robotics documented by Asada et al. [66]. The absence of mechanical reduction required the motor itself to generate a significant amount of torque, which presented a challenge in the early stages of research. This eventually led to the development of quasi direct drive actuation, a concept frequently applied in legged robots [71,72,73,74]. The quasi direct drive design philosophy aims to achieve torque transparency comparable to a direct drive actuator while producing higher torques, using a single-stage transmission with a low reduction ratio. In [73], Seok et al. examine the correlation between the geometry of a brushless DC motor and its torque density. These principles guided the selection of the brushless motor actuating the suckering tool presented in this thesis. The utilization of both direct drive and quasi direct drive actuation for the design of robotic tools and grippers is an active area of research [69,75,76]. These tools have the ability to provide precise tactile feedback through their high bandwidth torque sensing and transparency, eliminating the need for external sensors.

The integration of tactile perception for robotic exploration, mapping and shape estimation is a long-lasting research endeavour [77,78,79,80,81,82,83]. This pursuit is driven by the aspiration to replicate the human ability to employ the sense of touch for accomplishing these tasks. The approach to robotic suckering adopted in this thesis follows similar principles, relying on the tactile sensing ability of the presented suckering tool for plant trunk exploration. This allows for the traversal of differently shaped trunks, without the need to detect their shape prior to the procedure. Also, the use of tactile feedback for plant shape estimation is discussed.

4.2Direct Drive Brush-Shaped Suckering Tool

This chapter focuses on the development and properties of a robotic suckering tool, which takes the form of a motorized circular brush. This specialized tool is used as the end-effector for the robot manipulator, enabling it to perform the suckering task (Fig.4.3). It is designed with compliant task execution in mind, allowing the mobile manipulator to perform the suckering task without requiring prior detection of the plant trunk's shape. The remainder of this chapter presents the design, as well as control and sensing capabilities of the developed tool.

Figure 4.3: Mobile manipulator equipped with a specialized brush-shaped suckering tool in a vineyard. Copyright [58]CC BY 4.0.

As already mentioned, the presented suckering tool follows the direct drive design philosophy, having no mechanical transmission between the rotor and the mounting shaft of the actuator. This design allows for maximum *backdrivability* of the actuator, referring to the ability to transparently transmit forces from the output shaft of the actuator to its rotor. The lack of gearing related friction and energy losses allows for precise and high bandwidth torque sensing, using the motor current measurements and the relationship:

$$\tau = K_t I, \tag{4.1}$$

where τ is the torque, I is the current and K_t is the torque constant of the motor.

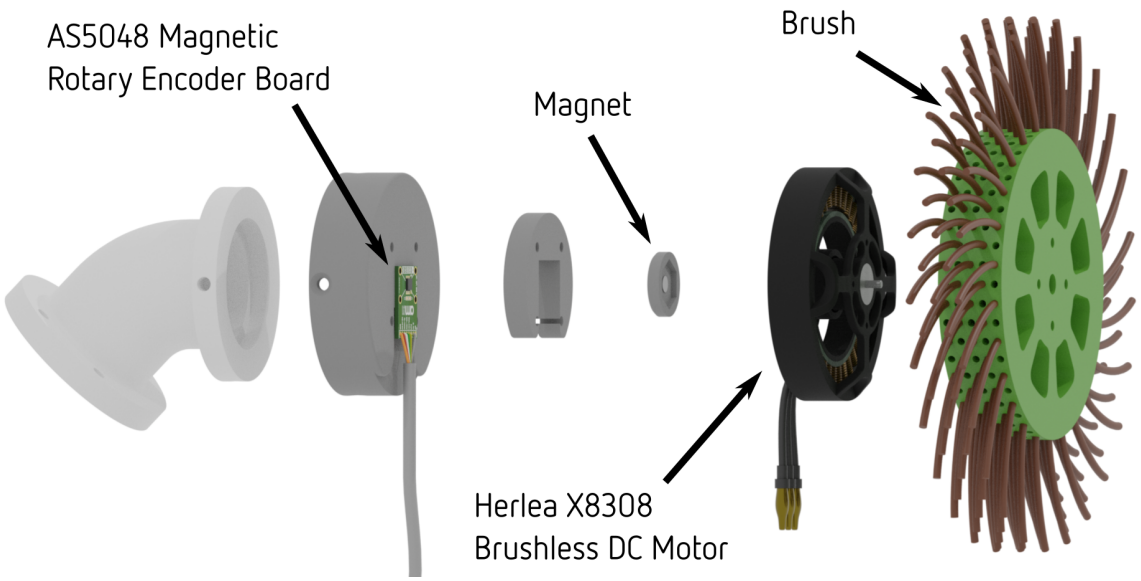

Figure 4.4: Exploded view of the direct drive brush-shaped suckering tool. The tool is actuated by a high torque density brushless DC motor. The lack of mechanical reduction allows for the use of current measurements from the motor for precise torque sensing.

In the context of robotic suckering, the torque sensing ability of the actuator driving the suckering tool is useful for three reasons:

- limiting the amount of torque exerted by the tool to avoid damaging on the plant
- using torque measurements as tactile feedback for compliant robot arm control
- using torque measurements as feedback for tactile mapping of the plant trunk's shape

An exploded view of the suckering tool design is shown in Fig.4.4. The tool consists of a circular brush mounted on a brushless DC motor and a magnetic encoder. The circular brush features flexible plastic bristles, which are responsible for removing the growth from the plant. The choice of the Herlea X8308 brushless DC motor for actuating the tool is driven by its favorable geometrical properties: it has a small length and a large radius, resulting in high torque density [73]. Torque density is defined as the ratio between the output torque of a motor and its mass. Given the lack of mechanical reduction in the actuation system, the high torque density of the motor is essential for the tool to generate sufficient torque. Both the motor and encoder are connected to a high-performance brushless motor control board ODrive, which is utilized for control and current sensing purposes.

The motor is controlled in the torque control mode using velocity limiting as described in the ODrive documentation [84]. This results in the behaviour where the circular brush is spinning at the velocity limit while the resistance torque from the obstacle is smaller than the commanded torque, and comes to a stop if the resistance torque felt by the tool is greater then or equal to the commanded torque. As already mentioned, this stopping behaviour serves as a safety feature, aiming to avoid damaging the plant. Consider the case when the tool spins at its velocity limit. The amount of torque necessary to achieve that velocity depends on the resistance between the

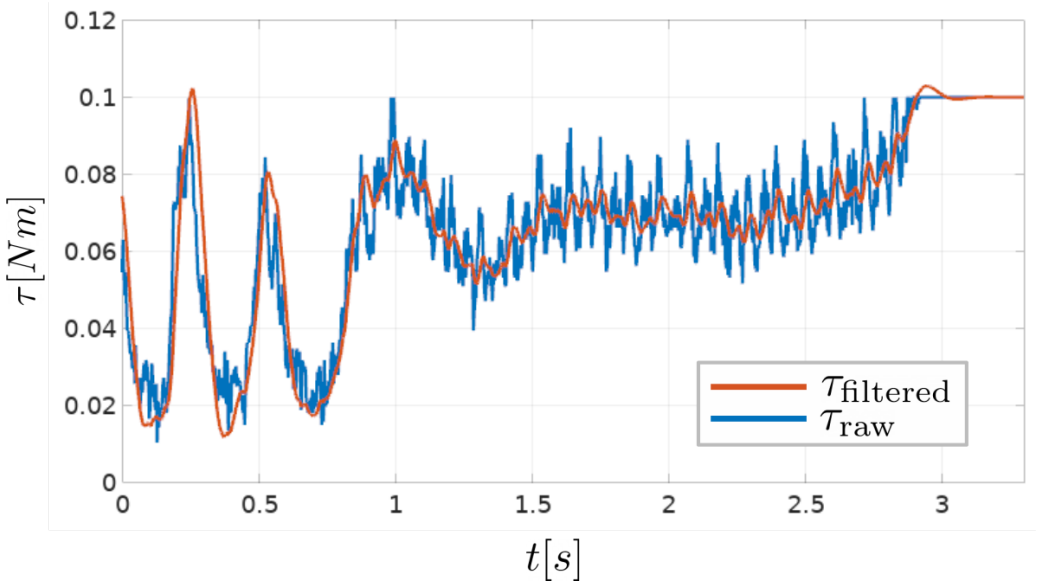

Figure 4.5: Torque measurements acquired by the suckering tool in torque control mode with velocity limiting. Raw torque signal calculated from current measurements is denoted as τ_{raw}, and the filtered torque signal as (τ_{filtered}). Copyright [58]CC BY 4.0.

brush bristles and the obstacle, and the tool acts as a sensor, measuring the resistance torque (Fig.4.5).

The amount of resistance torque sensed by the tool depends on the overlap between the brush bristles and the obstacle. This relationship and its use for tactile compliant control of the robot arm are discussed in the rest of this chapter.

4.2.1Torque-Overlap Relationship

This section discusses a correlation of the torque exerted by the suckering tool and the overlap between brush bristles and the obstacle (Fig.4.6). The initial hypothesis is that increasing this overlap produces increased resistance, resulting in higher values of torque produced by the motor. The focus of this section is on the experimental investigation into the relationship of these two quantities.

An experiment is performed, slowly moving the robot arm equipped with a brush-shaped tool in a single direction, and linearly increasing obstacle overlap (Fig.4.6). Once the torque exerted by the motor reaches its maximum specified value the experiment is stopped. The motor current and robot arm encoder measurements are used to calculate the torque and overlap values during the experiment, and their measured relationship is shown with a blue line in Fig. 4.7. The acquired experimental results suggest that this relationship can be approximated with a quadratic function:

$$O(\tau) = a\tau^2 + b\tau + c \tag{4.2}$$

where $O(\tau)$ is the estimated obstacle overlap, and a, b and c are quadratic function parameters.

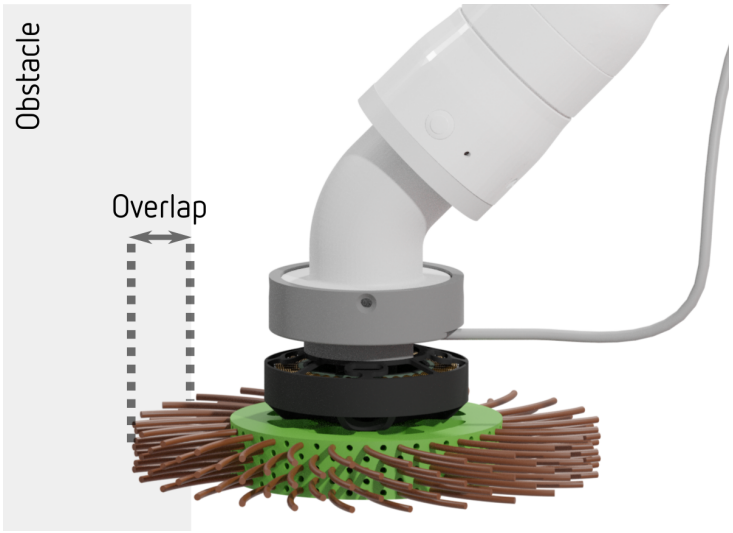

Figure 4.6: The torque exerted by the brush depends on the overlap between the brush bristles and the obstacle. The relationship between those values is estimated in order to simplify tactile control explained in the following section.

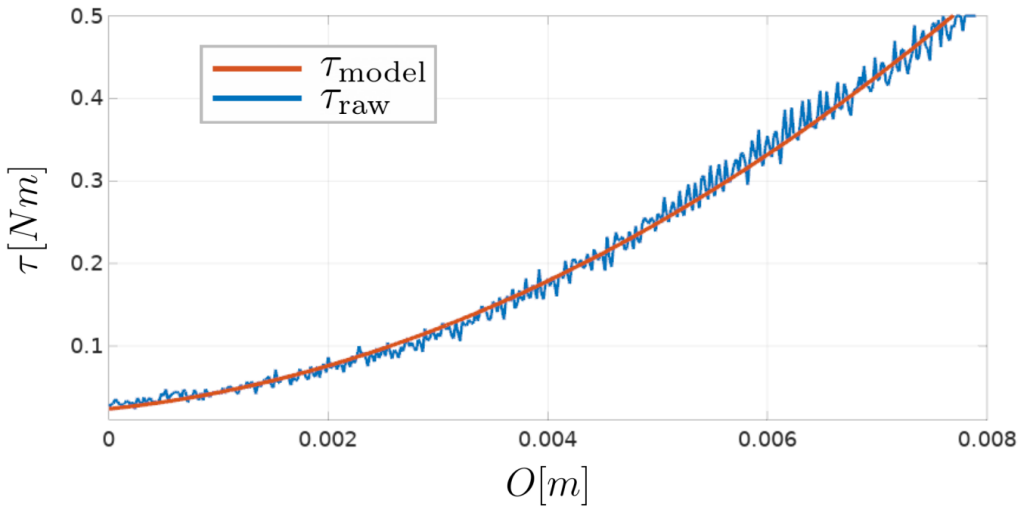

Figure 4.7: Experimental results suggest that the relationship between obstacle overlap and resistance torque resembles a quadratic function. Unfiltered torque measurements are denoted as τ_{raw} and the fitted quadratic function model as τ_{model}.

Least squares method is used to select these parameters that minimize the error between the quadratic approximation and real-world data. Multiple brush designs were used during the development of the tool, the particular one presented in this thesis having fitted parameter values of $a = 6281.9, b = 13.626$ and $c = 0.0238$.

This estimated relationship is used to transform the torque measured by the tool into obstacle overlap feedback, to simplify the tuning of the controllers used for tactile exploration of the grapevine trunk, further elaborated in the following sections.

4.3Prioritized Task Space Control based Suckering

This section focuses on the control methodology developed for the suckering task, utilizing the previously presented suckering tool. The control algorithm used for suckering is based on prioritized optimization, which is used, among other reasons, to exploit the functional redundancy introduced by the axis-symmetry of the tool (Fig.4.8).

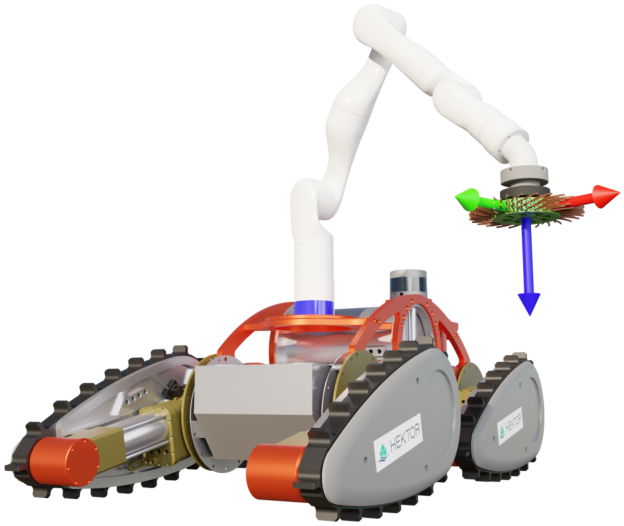

Figure 4.8: To perform the suckering task, the mobile manipulator is equipped with a direct drive brush-shaped suckering tool discussed in previous section. The tool is axis-symmetric, introducing functional redundancy into the control problem.

The torque sensing capability of the brush-shaped tool is utilized for tactile trunk exploration and mapping. The objective is to enable the tool to navigate and process as much of the trunk area as possible, relying solely on the tactile feedback. While certain general assumptions about the trunk's shape are considered, the method is designed to accommodate irregular and uneven trunks with significant curvatures. The tactile exploration algorithm aims to control the tool's movement along the trunk, while keeping the amount of overlap with the brush bristles constant. A prioritized task space control algorithm is utilized to prioritize between different components of the tool's velocity.

The torque measuring ability of the suckering tool is also utilized for tactile mapping, utilizing a probabilistic model of the tool. Multiple occupancy grid maps are constructed during the exploration, and are used to estimate the shape of the trunk.

4.3.1State Machine

As already mentioned, the presented control method has the objective of processing as much of the trunk area as possible using the suckering tool. The control algorithm is based on a high-level state machine that switches the robot arm between different work modes. The goal

is to produce a behavior similar to the lawnmower trajectory used in the spraying algorithm described in Chapter3. This lawnmower behaviour is based on switching between stages of sideways and vertical movement of the suckering tool, as shown in Fig.4.9.

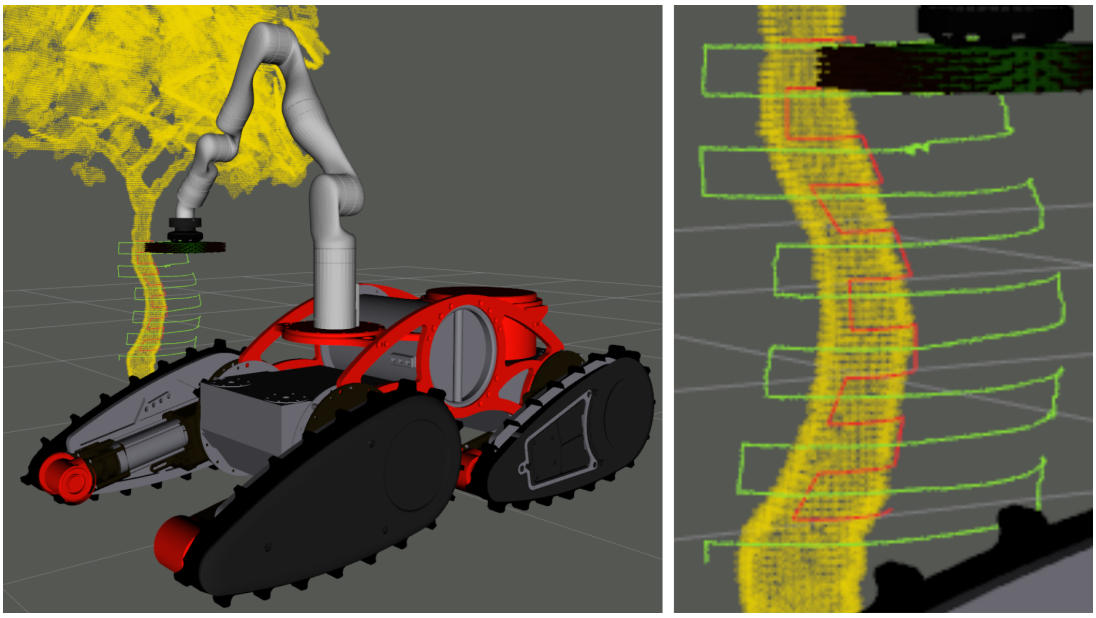

Figure 4.9: The processing of the trunk area is achieved using a state machine switching between different control modes for the robot arm, resulting in a lawnmower shaped trajectory. The green path corresponds to the position of the suckering tool, and the red path represents the point of contact with the trunk.

A visualization of the state machine used to achieve this behaviour is shown in Fig.4.10. The initial assumption is that prior to running the state machine the robot arm is positioned directly in front of the trunk. If this initial assumption is met, the control algorithm allows the execution of the entire algorithm based solely on the tactile feedback provided by the developed suckering tool.

First, the robot arm is commanded to move forward until contact with the trunk is made, detected as the estimated overlap exceeding a prescribed threshold. Once contact with the trunk is made, the arm is commanded to move sideways. It is worth mentioning that this movement is not purely sideways with respect to the base coordinate frame but is tangential to the detected obstacle, as a result of the tool remaining in contact with the trunk at all times. Since the shape of a trunk in a horizontal plane exhibits a quasi-circular shape, the obstacle tangent angle constantly changes during this stage. Once this tangent angle exceeds a certain threshold, the robot arm is commanded to move downwards by a specified amount, while maintaining contact with the trunk. To ensure full vertical coverage of the trunk, this specified height must be lower than the height of the brushing tool itself. After this movement is completed, if the global height limit has not been reached, the arm is once again commanded to move sideways, but in an opposite direction, and the entire process is repeated. Once the global height limit is reached, the suckering task is considered to be finished.

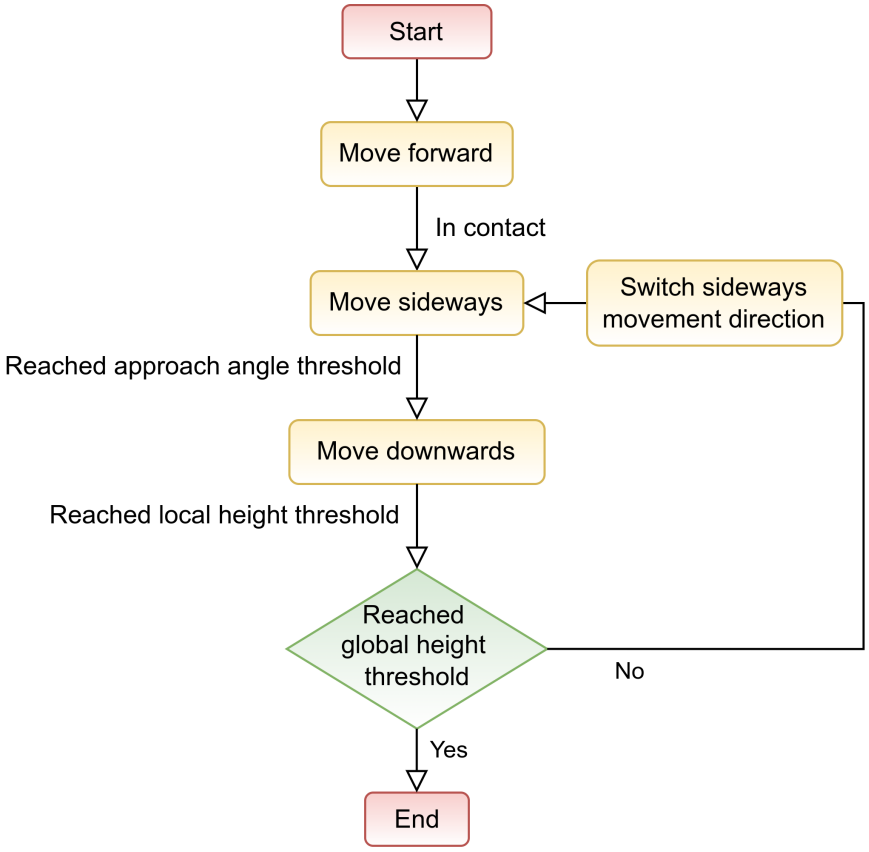

Figure 4.10: A diagram of the state machine used to execute the suckering task. To achieve the lawnmower shaped trajectory, robot arm is commanded to switch between sideways and downwards movement modes.

The main control challenge is to satisfy the requirement to remain in contact with the trunk at all times, while moving in a specified fashion. This is achieved using prioritized task space control, presented in the following section.

4.3.2Prioritized Task Space Control for Tactile Exploration

A custom task space is defined for the purpose of tactile trunk exploration, referred to as *exploration* space, with the corresponding exploration velocity $\boldsymbol{v}_E$ and exploration Jacobian matrix $\boldsymbol{J}_E$. Two coordinate frames are defined, the end-effector coordinate frame and the exploration coordinate frame, as seen in Fig.4.11. The end-effector frame is fixed with respect to the final link of the robot arm. It is positioned in the center of the suckering tool, and both its position and orientation are fully determined by the current joint positions of the robot arm. For the purpose of the tactile exploration task, the exploration coordinate frame is defined as the end-effector frame rotated around its z axis. The exploration frame is rotated for the amount α_O, aligning its y axis with the estimated obstacle tangent.

The path of the tool during the tactile exploration, shown with a green line in Fig.4.9, is used to estimate the angle α_O. The latest segment of the tool path is used to numerically

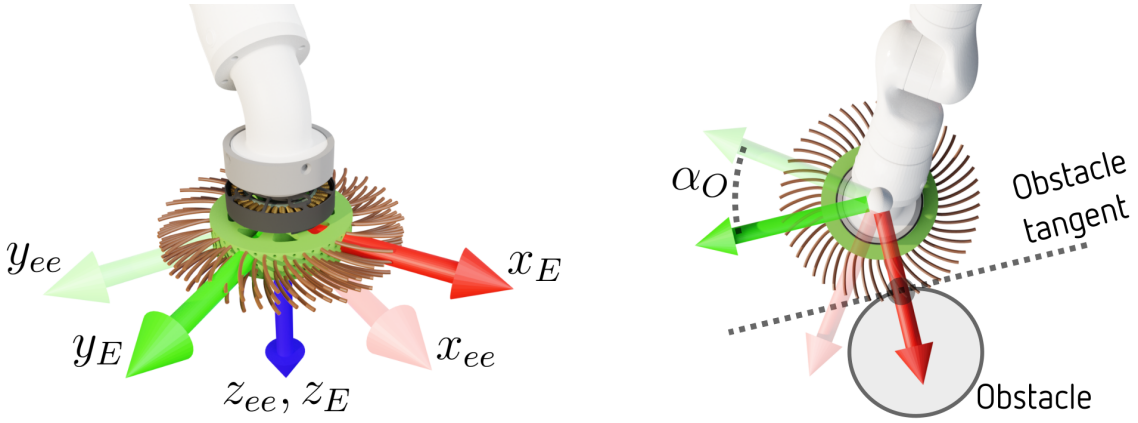

Figure 4.11: Coordinate systems used for the task of tactile trunk exploration. Exploration coordinate frame rotates with respect to the final link of the robot arm. The y coordinate of the exploration frame is parallel to the estimated obstacle tangent, and its x coordinate is directed towards the obstacle.

calculate its tangent and α_O. This angle constantly changes during the *Move sideways* stages of the tactile exploration process, resulting in the task space that is not a part of the kinematic chain. The obstacle angle α_O is assumed to remain constant during the *Move downwards* state.

The axis-symmetry of the suckering tool introduces a functional redundancy into the tactile exploration problem. The angle around the z axis of the tool does not affect task execution and is disregarded, resulting in a five dimensional exploration velocity:

$$\boldsymbol{v}_E = \begin{bmatrix} v_{E,x} \\ v_{E,y} \\ v_{E,z} \\ \omega_{E,x} \\ \omega_{E,y} \end{bmatrix} = \begin{bmatrix} v_O \\ v_{tan} \\ v_{E,z} \\ \omega_{E,x} \\ \omega_{E,y} \end{bmatrix} \in \mathbb{R}^5 \tag{4.3}$$

The x and y direction components of the exploration velocity are given more descriptive labels, v_O and v_{tan} respectively. Velocity v_O denotes the tool's velocity approaching the obstacle, and v_{tan} its velocity tangential to the obstacle (Fig.4.11).

The task space control problem used for tactile exploration, similarly to the one used for continuous spraying, utilizes task prioritization for multi-stage redundancy resolution. One of the roles of the prioritization is to separate the bristle overlap control from the rest of the exploration velocity. As already mentioned, the tactile exploration algorithm aims to keep a constant amount overlap between the bristles of the suckering tool and the plant trunk while traversing it. The component of the exploration velocity controlling overlap v_O is given a lower priority than the rest of the exploration velocity. The reasoning behind this is to allow for aggressive overlap controller design, while ensuring the correct angle, height and tangential velocity of the tool.

Therefore, the cost function of the first priority regards the entire exploration velocity except for the obstacle approach velocity. The tangential and vertical linear velocities are considered, along with rotational velocities around the x and y axes of the exploration frame. The cost function of the first priority can be written as:

$$E_1(\dot{\boldsymbol{q}}) = ||\boldsymbol{J}_1\dot{\boldsymbol{q}} - \boldsymbol{v}_1||^2, \tag{4.4}$$

where:

$$\boldsymbol{v}_1 = \begin{bmatrix} v_{tan} \\ v_{E,z} \\ \omega_{E,x} \\ \omega_{E,y} \end{bmatrix} \in \mathbb{R}^4, \; \boldsymbol{J}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{J}_E \in \mathbb{R}^{4\times N_J}, \tag{4.5}$$

The cost function of the second priority, referring to the obstacle approach component of the exploration velocity, has the following form:

$$E_2(\dot{\boldsymbol{q}}) = ||\boldsymbol{J}_2\dot{\boldsymbol{q}} - v_O||^2, \tag{4.6}$$

where $\boldsymbol{J}_2$ is the second row of the exploration Jacobian:

$$\boldsymbol{J}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{J}_E \in \mathbb{R}^{1\times N_J}, \tag{4.7}$$

The final priority, which resolves any redundancy remaining after minimizing the first two priorities, favors such joint velocities $\dot{\boldsymbol{q}}$ that move the arm towards a desired configuration:

$$E_3(\dot{\boldsymbol{q}}) = ||\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_c||^2, \tag{4.8}$$

where the commanded joint velocities $\dot{\boldsymbol{q}}_c$ that drive the robot arm towards a desired pose $\boldsymbol{q}_d$ are selected by a proportional controller:

$$\dot{\boldsymbol{q}}_c = K_{P,q}(\boldsymbol{q}_d - \boldsymbol{q}) \tag{4.9}$$

Here, $K_{P,q}$ is the controller gain and $\boldsymbol{q}$ is the current joint position vector. Similarly to the continuous spraying task space control problem, inequality constraints are used to enforce joint velocity and acceleration limits:

$$\underline{\dot{\boldsymbol{q}}} \preceq \dot{\boldsymbol{q}} \preceq \overline{\dot{\boldsymbol{q}}} \tag{4.10}$$

$$\dot{\boldsymbol{q}}_P + \underline{\ddot{\boldsymbol{q}}}\Delta t \preceq \dot{\boldsymbol{q}} \preceq \dot{\boldsymbol{q}}_P + \overline{\ddot{\boldsymbol{q}}}\Delta t \tag{4.11}$$

where Δt is the control time step, and $\dot{\boldsymbol{q}}_P$ are joint velocities in the previous time step.

The commanded overlap velocity v_O is the result of a proportional overlap controller:

$$v_O = K_{P,O}(O_d - O(\tau)), \tag{4.12}$$

where O_d is the desired amount of overlap and $O(\tau)$ is the current overlap estimate, calculated using equation (4.2). $K_{P,O}$ represents the gain of the proportional controller.

One of the challenges of the presented method is the choice of this controller gain. While the tool is able to estimate the amount of overlap with the obstacle, this estimate is saturated, and has values from 0 to 8 millimeters (as seen in Fig.4.7). The prioritization between the components of the exploration velocity allows for aggressive overlap controller design, since correct orientation and height of the tool are guaranteed by the higher priority.

The commanded tangential velocity v_{tan} depends on the current stage of the exploration process. In the *Move downwards* stage the tangential velocity is commanded to be zero, and in the *Move sideways* stage it is calculated as:

$$v_{tan} = movement\_direction \cdot |v_{tan}|, \tag{4.13}$$

where $|v_{tan}|$ is the absolute value of tangential velocity set as a parameter by the user, and $movement\_direction$ is either 1 or -1 based on the current movement direction, and changing after each *Move downwards* state as seen in Fig.4.10.

The commanded vertical velocity $v_{E,z}$ also depends on the current stage of the exploration process, being set to zero during the *Move sideways* state, and having the user defined value $|v_{E,z}|$ during the *Move downwards* state. Commanded angular velocity values are calculated as for the continuous spraying task, discussed in section3.2.2.1.

4.3.3Trunk Shape Estimation using Tactile Mapping

Trunk shape estimation is performed utilizing the torque sensing ability of the developed suckering tool as feedback for tactile mapping. Multiple two dimensional occupancy grid maps are gathered at different heights, each map relating to a single *Move sideways* state (Fig.4.10). The main challenge presented by this task is to estimate the obstacle position at a point in time based solely on tool's movement and torque measurements.

Occupancy grid maps address the problem of generating consistent maps from noisy and uncertain measurement data, under the assumption that the sensors pose is known [85]. A map is represented by an evenly spaced two-dimensional grid with values $0 \leq m^{i,j} \leq 1$, where $m^{i,j}$ represents the probability of the map cell at coordinates i, j being occupied by an obstacle. The map is initially set to $m^{i,j} = 0.5$ for all map cells, representing an unobserved point in space, neither free nor occupied.

The probability values of an occupancy grid map are updated according to the Bayes rule, and the calculations can be performed with the probability representation of the map. However, the *log-odds* representation of occupancy is often used to improve execution time and avoid numerical instabilities for probabilities near zero or one [85]. The *odds* of the event x is defined as its probability divided by the probability of its negate, and the *log-odds* refers to the logarithm of this expression:

$$l(x) = \log(odds(x)) = \log\left(\frac{p(x)}{1-p(x)}\right), \tag{4.14}$$

where $0 \leq p(x) \leq 1$ is the probability of the event x, and $-\infty \leq l(x) \leq \infty$ is its log odds.

The recursive equations used to update the log-odds representation of the occupancy grid map are as follows:

$$l_k^{i,j} = l_{k-1}^{i,j} + \log\frac{S_k^{i,j}}{1-S_k^{i,j}} \tag{4.15}$$

Here $l_k^{i,j}$ represents the log-odds representation of the map cell $m^{i,j}$, at the k-th iteration of the mapping algorithm. The value $S_k^{i,j}$ represents the corresponding value of the probabilistic sensor model. Probabilistic sensor model represents the sensor measurement as the probability of the obstacle being at a certain point in space. The main challenge tackled in this section is determining the probabilistic sensor model of the developed suckering tool. The probability representation of the occupancy grid map can be retrieved from the log-odds notation as follows:

$$m^{i,j} = 1 - \frac{1}{1+e^{l^{i,j}}} \tag{4.16}$$

Even though the overlap between the brush and the obstacle can be estimated from torque measurements (as discussed in section4.2.1), the torque measurements cannot be used to determine the angle at which the obstacle is contacted, due to the axis-symmetry of the tool. Additionally, the overlap estimate might not be ideal due to the noise or delay present in torque measurements. Both of these issues are addressed using a probabilistic sensor model of the suckering tool, also referred to as the *inverse sensor model* [85].

This model uses torque measurement and tool movement data to calculate a probability of the presence of an obstacle at a certain point in space. Both the overlap and the obstacle angle are handled as probabilistic variables with normal distributions. Mean value of the overlap depends solely on the current torque measurement and is calculated using the relationship (4.2), while the mean value of the obstacle angle is chosen as the angle of the obstacle tangent, shown as α_O in Fig.4.11.

The probability density function of the normal distribution is given by:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{4.17}$$

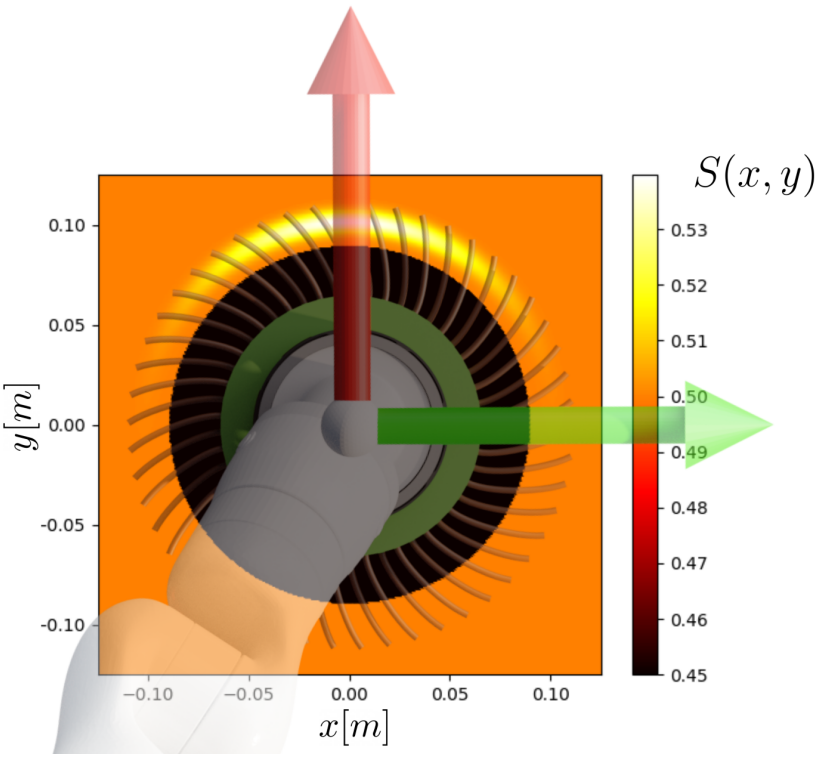

Figure 4.12: Probabilistic sensor model of the suckering tool incorporates two normal distributions, one for the obstacle angle and one for the obstacle overlap.

where $p(x)$ is the probability density function of a random variable x, μ is the mean and σ is the standard deviation. Two normal distributions are used in the probabilistic sensor model, the overlap distribution $p_O(O(\tau))$ and the obstacle angle distribution $p_\alpha(\alpha_O)$. The mean values used for these distributions represent their expected value, and are calculated as follows:

$$\mu_O = a\tau^2 + b\tau + c, \quad \mu_\alpha = \alpha_O \tag{4.18}$$

The corresponding standard deviations σ_O and σ_α represent the trust in the correctness of the overlap and angle measurement respectively, and are chosen experimentally.

The probabilistic sensor model is represented by the probability of the obstacle being measured at a certain two-dimensional point in space:

$$S(x,y) = \begin{cases} 0.45, & \text{if } r(x,y) \leq \underline{r} \\ p_O \cdot p_\alpha, & \text{if } \underline{r} \leq r(x,y) \leq \overline{r} \text{ and } \underline{\alpha} \leq \alpha(x,y) \leq \overline{\alpha} \\ 0.5, & \text{else} \end{cases} \tag{4.19}$$

Here, $S(x,y)$ represents the probability of an obstacle being at the coordinates x,y. A visualization of the probabilistic sensor model is shown in Fig.4.12. The first segment of the sensor model represents the point (x,y) located inside the brush, with a small radius for which p_O is

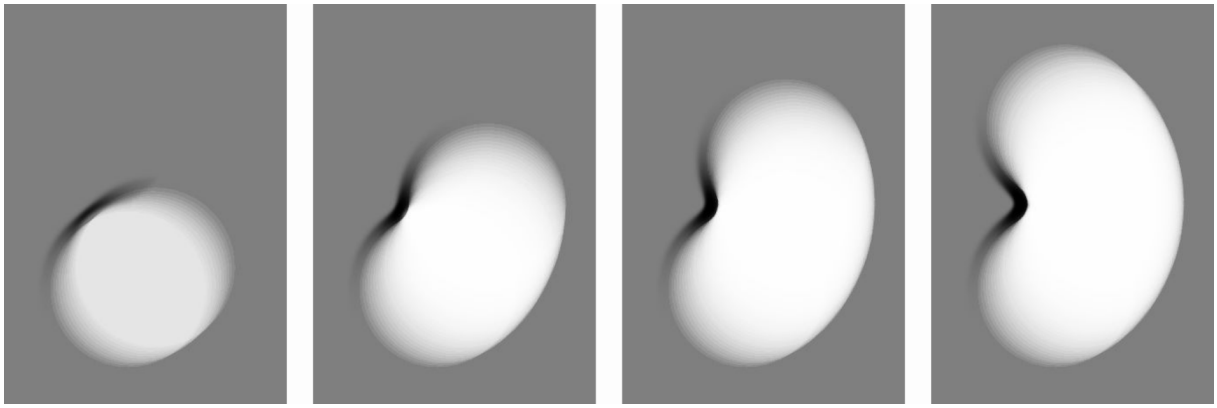

Figure 4.13: A two-dimensional occupancy grid map is built at the constant tool height using the Bayes rule and the inverse sensor model. Dark pixels represent obstacles and the white pixels represent obstacle free space, while the grey area is not observed by the sensor.

negligible. Since the points inside the brush itself cannot be occupied by an obstacle, these points are given a probability $S(x,y) \leq 0.5$, and are shown as the black area in Fig.4.12. The second segment of $S(x,y)$ represents the points in space for which neither of the normal distributions yields negligible values, and the combination of the p_O and p_α is used as the overall probability of the obstacle being present. The product of p_O and p_α yields probability values of $S(x,y) \geq 0.5$, resulting in highest probability of the obstacle being present at the mean of both the overlap and obstacle angle distributions. The final segment of $S(x,y)$ corresponds to unobserved space.

The limits used in the sensor model are:

$$\overline{r} = \mu_O + 3\sigma_O, \quad \underline{r} = \mu_O - 3\sigma_O \tag{4.20}$$

$$\overline{\alpha} = \mu_\alpha + 3\sigma_\alpha, \quad \underline{\alpha} = \mu_\alpha - 3\sigma_\alpha \tag{4.21}$$

A visualization of the process of building a map for a single *Move sideways* state can be seen in Fig.4.13. The sensor model is used to recursively update the occupancy grid map, resulting in the obstacle being represented with sufficiently dark parts of the map.

4.3.4Results and Discussion

The experimental evaluation of the presented suckering method commenced in a controlled setup is depicted in Fig.4.14. A grapevine trunk with prominent curves and bumps was fixed to the ground next to the mobile manipulator. This trunk was intentionally selected to deviate from the conventional straight structure, showcasing the flexibility of the presented method.

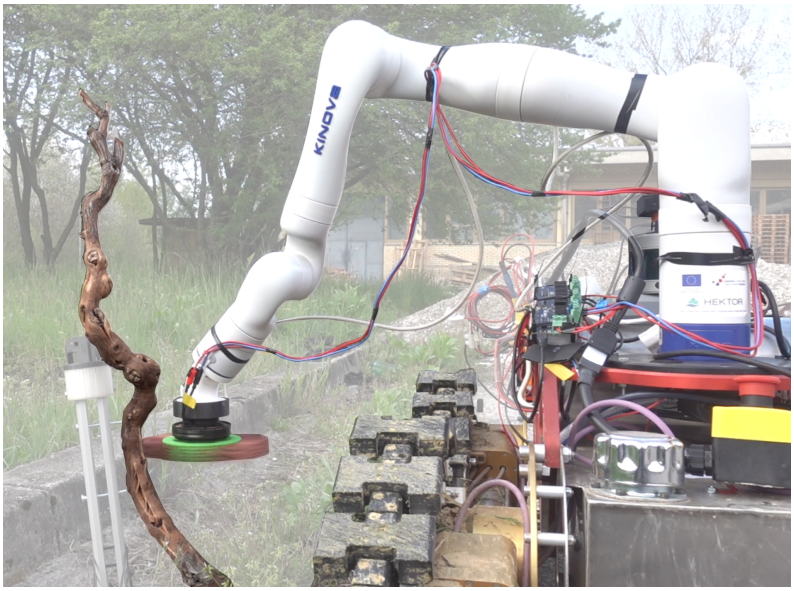

Figure 4.14: Experimental setup for testing the trunk shape following and tactile mapping. A grapevine trunk with an unconventional shape was deliberately chosen to demonstrate the utility of tactile exploration and mapping for a wide range of trunk shapes.

As previously explained, once contact with the plant is established, the lawnmower behaviour consisting of multiple *Move sideways* and *Move downwards* states is initialized. Throughout the *Move sideways* states, the tool maintains a constant height, and online occupancy grid mapping is executed estimating the shape of the trunk in real-time (Fig.4.15).

Once a particular *Move sideways* state finishes, an acquired occupancy grid map corresponding to that state is stored for future use. This process is repeated for every *Move sideways* state, and once the entire algorithm is finished, the stored occupancy grid maps are layered at their respective tool heights to estimate the overall shape of the trunk (Fig.4.16).

The shape of the trunk estimated by the presented method, and the path of the suckering tool during this experiment are shown in Fig.4.17. The three-dimensional path data is fitted to two images of the trunk used in the experiment, demonstrating the resemblance of the estimation and the real trunk shape.

Additional experiments were conducted in actual vineyards, demonstrating the ability of the presented method to successfully remove the growth from the grapevine plant trunk. Some of the results can be seen in the accompanying video [*].

[*] https://www.youtube.com/watch?v=y9Cca1SuyCU

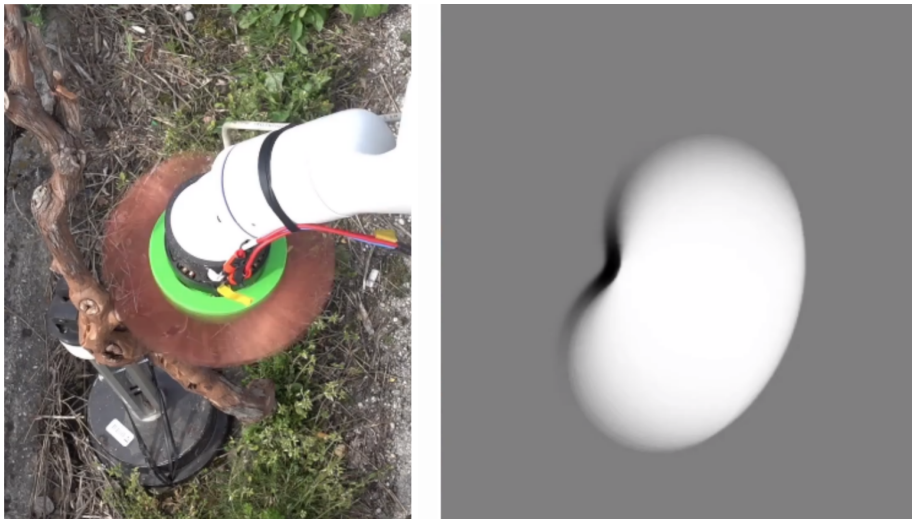

Figure 4.15: Occupancy grid mapping at a constant tool height during the suckering experiment. The map is built online, being constantly updated during the experiment.

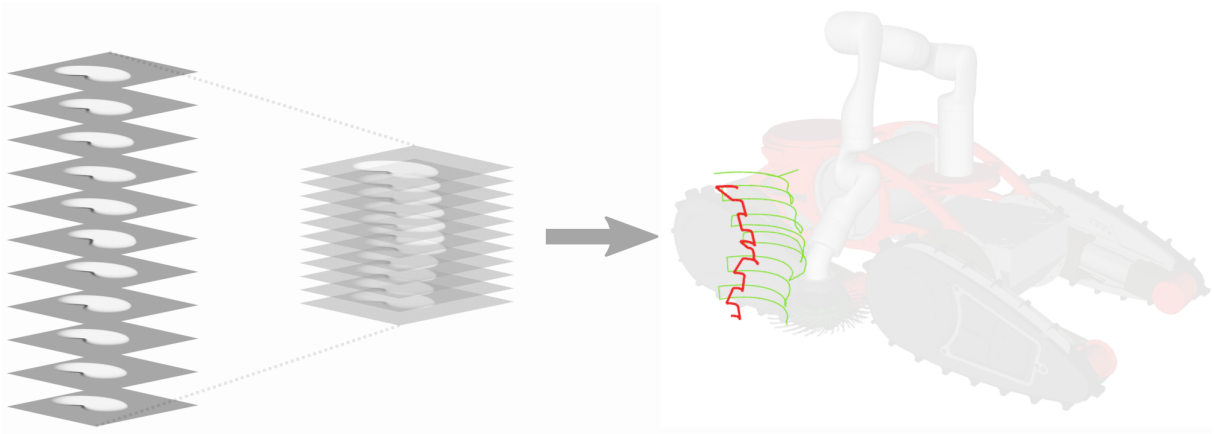

Figure 4.16: Once the suckering experiment is finished, all the occupancy grid maps built at different heights are accumulated to estimate the overall shape of the trunk.

The presented method demonstrated satisfactory performance in real-world conditions, successfully executing the suckering task in most cases. One of the drawbacks of the presented method is its inability to process the entire area of the trunk. The robot arm is not able to reach the opposite side of the trunk without collisions, which is why the tangent angle limits are set in such a way to conservatively constrain the amount of the trunk processed by the tool. This issue could be combated to some extent by including collision awareness to the control algorithm, allowing the tool to reach more of the trunk area. Although the method is expected to perform well in various vineyards, additional obstacles that are sometimes found on grapevine trunks (such as wires or water hoses) can disrupt its execution. Also, the trunk of the grapevine plant is not completely rigid, which can result in the oscillatory behaviour of the system, usually more pronounced at the upper part of the trunk. This has to be taken into account when tuning the overlap controller, and the desired tangential and vertical velocity values. Successful removal of the growth also highly depends on its stage of development, the younger buds being easier

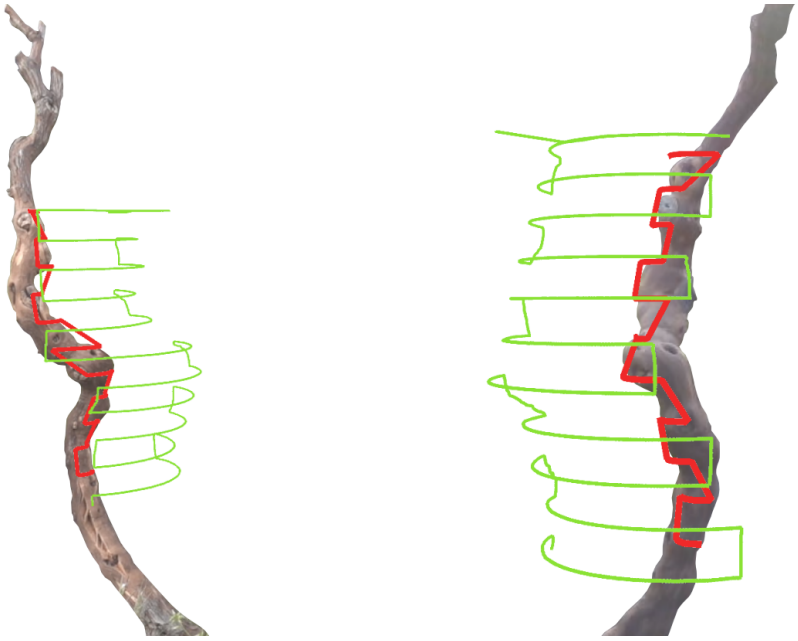

Figure 4.17: The green line represents the path of the tool during the experiment, and the red line represents the shape of the trunk estimated using tactile mapping. The estimated trunk shape is fitted on images of the actual trunk used in the experiment.

to removal than the older, more developed ones. Apart from that, the length, thickness, flexibility and sharpness of the brush bristles also has a profound effect on successfulness of growth removal.

4.4Summary

This chapter is dedicated to the application of a mobile manipulator in the context of robotic viticultural suckering. The chapter is divided into two parts: one addressing the development of the robotic tool for the suckering task, and the other covering the control methodology and algorithms developed for this task. The robotic suckering tool is designed as a circular brush-shaped tool, utilizing the principle of direct drive actuation to achieve precise torque feedback. The use of these torque measurements for tactile sensing is discussed. The circular suckering tool is axis-symmetric, introducing a functional redundancy into the robot arm control problem. Prioritized task space control is used to resolve this redundancy, while prioritizing between the different velocity components of the tool. The tactile sensing capability of the suckering tool is used to achieve tactile plant shape exploration and mapping. Two contributions of this thesis related to the viticultural suckering task are discussed in the chapter:

- *A direct drive brush-shaped vine suckering robotic tool with torque sensing*
- *Prioritized task space control based method for compliant vine suckering using a direct drive brush-shaped robotic tool*

Associated Publications

Parts of this chapter have already been published by the author in the following article:

•Vatavuk, I., Stuhne, D., Vasiljevi ć, G., and Kovačić, Z. (2023). Direct Drive Brush-Shaped Tool with Torque Sensing Capability for Compliant Robotic Vine Suckering. Sensors, 23, 1195. https://doi.org/10.3390/s23031195 (IF = 3.9, Q2)

CHAPTER 5

Conclusion

The aim of the research presented in this thesis is to develop control methods for mobile manipulation in a viticultural scenario. Tasks of spraying and suckering in viticulture are considered, both of which require the robot system to manipulate its environment. The common theme for all the developed control algorithms is the use of optimization and optimal control to resolve different kinds of redundancies that arise in the execution of mentioned tasks. The tools used for both of the manipulation tasks are axis-symmetric, introducing functional redundancy to both control problems. The main contributions of the thesis are laid out as follows.

Task space model predictive control based method for vineyard spraying with a mobile manipulator

The first contribution is focused on the task of vineyard spraying. The first kind of redundancy tackled as a part of this task is the redundancy in control of the spraying nozzle during continuous spraying, which is controlled both by the robot arm and the mobile base. Task space model predictive control is used to select coordinated control inputs for the robot arm and the mobile base that follow the reference while predictively resolving the redundancy. A function of the acceleration of the mobile base and the displacement of the robot arm is minimized as the criterion function of the model predictive control optimization problem. This results in a behaviour where the robot arm is responsible for fast, local movement of the nozzle, and the mobile base is responsible for slower movement, tracking the global trend in change of the reference trajectory.

The model predictive control algorithm selects the desired task space movement of the robot arm, which does not correspond to the actual low-level commands used in its control. The task

space control of the robot arm, selecting the low-level joint velocity commands, is itself a redundant problem. Axis-symmetry of the spray nozzle introduces a functional redundancy to the problem, which is resolved using prioritized task space control. The translational velocity of the spray frame is prioritized over its orientational velocity, due to its larger importance for even spray coverage. The methodology and implications of including various constraints to the prioritized task space control are discussed. Inspired by the velocity level prioritization between the task components used in continuous spraying, a positional version of this prioritization is utilized for the task of selective spraying. A prioritized positional inverse kinematics algorithm is developed for this task, iteratively solving the velocity level problem. The developed algorithm allows for positional inverse kinematics of an arbitrary robotic system with multiple tasks with clearly defined priorities.

A direct drive brush-shaped vine suckering robotic tool with torque sensing

The second contribution is focused on the task of viticultural suckering. The goal of the suckering procedure is to remove all the unwanted growth beneath a certain height of the trunk of the grapevine plant. A brush-shaped robot tool relying on direct drive design philosophy is developed for this task. The direct drive design is based on having no mechanical reduction between the motor shaft and the actuator output. The lack of mechanical reduction allows for precise and high-bandwidth torque sensing, based on the motor current measurements. Brushless DC electric motor with high torque density is chosen to allow sufficient torque output without any reduction. The design objectives and decisions are discussed, along with the details on tools torque sensing ability. An experiment is conducted to estimate the relationship between the measured torque and the overlap between the brush bristles and the obstacle. The data gathered during this experiment is used to approximate a mathematical relationship between the two values. Using estimated obstacle overlap value as control feedback instead of torque measurements is advantageous for the suckering control method, which relies on constant overlap control. Like the nozzle used for vineyard spraying, the developed tool is also an axis-symmetric tool, which introduces the same kind of functional redundancy. The probabilistic model of the tool is presented, using torque measurements and brush movement data to estimate both the overlap and the angle at which the obstacle is contacted.

Prioritized task space control based method for compliant vine suckering using a direct drive brush-shaped robotic tool

The final contribution is focused on the control method developed for the suckering task, utilizing the previously described robotic tool and its torque sensing ability. The method is based

on prioritized task space control, using prioritized optimization to resolve the functional redundancy posed by the axis-symmetry of the tool, and assign priorities to different components of tool's velocity. The component of the tool velocity approaching the obstacle is given a lower priority than the rest of the translational velocity together with the direction of the brush. The reasoning behind this is to allow the obstacle overlap controller to be as aggressive as possible without affecting other components of the brush pose.

The high-level goal of the control method is to execute a lawnmower trajectory, and cover a large portion of the trunk area while maintaining contact with the tool. A state machine is used to switch between different robot arm control modes, following the logic resulting in lawnmower behaviour. Overlap estimation from the suckering tool's torque measurements is used to maintain the overlap as close as possible to its desired value throughout the trajectory. The probabilistic sensor model of the developed suckering tool is used for the probabilistic mapping algorithm, allowing the method to be used to estimate the overall shape of the trunk. Estimated trunk shape can be reused for future executions of the suckering procedures, reducing execution time.

Bibliography

[1]Azam-Ali, S. N., The Ninth Revolution. WORLD SCIENTIFIC, 2021, available at: https://www.worldscientific.com/doi/abs/10.1142/12262

[2]Oliveira, L. F. P., Moreira, A. P., Silva, M. F., "Advances in agriculture robotics: A state-of-the-art review and challenges ahead", Robotics, Vol. 10, No. 2, 2021, available at:https://www.mdpi.com/2218-6581/10/2/52

[3]"Hektor project website", Available:hektor.fer.hr.

[4]Gori čanec, J., Kapetanović, N., Vatavuk, I., Hrabar, I., Vasiljević, G., Gledec, G., Stuhne, D., Bogdan, S., Orsag, M., Petrović, T., Mišković, N., Kovačić, Z., Kurtela, A., Bolotin, J., Kožul, V., Glavić, N., Antolović, N., Anić, M., Kozina, B., Cukon, M., "Heterogeneous autonomous robotic system in viticulture and mariculture - project overview", in 2021 16th International Conference on Telecommunications (ConTEL), 2021, pp. 181-188.

[5]Hrabar, I., Goricanec, J., Kovacic, Z., "Towards autonomous navigation of a mobile robot in a steep slope vineyard", in 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE, Sep. 2021, available at: https://doi.org/10.23919/mipro52101.2021.9596997

[6]Hrabar, I., Kova čić, Z., "Localization of mobile manipulator in vineyards for autonomous task execution", Machines, Vol. 11, No. 4, Mar. 2023, pp. 414, available at: https://doi.org/10.3390/machines11040414

[7]Hrabar, I., Vasiljevi ć, G., Kovačić, Z., "Estimation of the energy consumption of an all-terrain mobile manipulator for operations in steep vineyards", Electronics, Vol. 11, No. 2, 2022, available at:https://www.mdpi.com/2079-9292/11/2/217

[8]Vatavuk, I., Vasiljevi ć, G., Kovačić, Z., "Task space model predictive control for vineyard spraying with a mobile manipulator", Agriculture, Vol. 12, No. 3, 2022, available at: https://www.mdpi.com/2077-0472/12/3/381

[9]Lee, D., Baeg, M. G., "Redundant robots", in Springer Handbook of Robotics, Siciliano, B., Khatib, O., (ur.). Springer, 2016, pp. 221–242.

[10]Schappler, M., Tappe, S., Ortmaier, T., "Resolution of functional redundancy for 3t2r robot tasks using two sets of reciprocal euler angles", in Advances in Mechanism and Machine Science. Springer International Publishing, 2019, pp. 1701–1710, available at: https://doi.org/10.1007/978-3-030-20131-9_168

[11]Žlajpah, L., "On orientation control of functional redundant robots", in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 2475-2482.

[12]Waldron, K. J., Schmiedeler, J., "Kinematics", in Springer Handbook of Robotics, Siciliano, B., Khatib, O., (ur.). Springer, 2016, pp. 11–35.

[13]Featherstone, R., Rigid Body Dynamics Algorithms. Springer US, 2008, available at: https://doi.org/10.1007/978-1-4899-7560-7

[14]Wensing, P. M., Orin, D. E., "Generation of dynamic humanoid behaviors through task-space control with conic optimization", in 2013 IEEE International Conference on Robotics and Automation. IEEE, May 2013, available at:https://doi.org/10.1109/icra.2013.6631008

[15]Marani, G., Kim, J., Yuh, J., Chung, W. K., "A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators", in Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Vol. 2, 2002, pp. 1973-1978 vol.2.

[16]Kurtz, V., Wensing, P. M., Lin, H., "Control barrier functions for singularity avoidance in passivity-based manipulator control", in 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 6125-6130.

[17]Lynch, K. M., Park, F. C., Modern Robotics: Mechanics, Planning, and Control, 1st ed. USA: Cambridge University Press, 2017.

[18]Buss, S., "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods", IEEE Transactions in Robotics and Automation, Vol. 17, 05 2004.

[19]Boyd, S., Vandenberghe, L., Convex Optimization. Cambridge University Press, 2004.

[20]de Lasa, M., Hertzmann, A., "Prioritized optimization for task-space control", in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, Oct. 2009, available at:https://doi.org/10.1109/iros.2009.5354341

[21]de Lasa, M., Mordatch, I., Hertzmann, A., "Feature-based locomotion controllers", ACM Transactions on Graphics, Vol. 29, No. 4, Jul. 2010, pp. 1–10, available at: https://doi.org/10.1145/1778765.1781157

[22]Vatavuk, I., Kova čić, Z., "Constrained prioritized 3t2r task control for robotic agricultural spraying", available at:https://arxiv.org/abs/2307.003022023.

[23]Oberti, R., Marchi, M., Tirelli, P., Calcante, A., Iriti, M., Tona, E., Ho čevar, M., Baur, J., Pfaff, J., Schütz, C., Ulbrich, H., "Selective spraying of grapevines for disease control using a modular agricultural robot", Biosystems Engineering, Vol. 146, Jun. 2016, pp. 203–215, available at:https://doi.org/10.1016/j.biosystemseng.2015.12.004

[24]Botterill, T., Paulin, S., Green, R., Williams, S., Lin, J., Saxton, V., Mills, S., Chen, X., Corbett-Davies, S., "A robot system for pruning grape vines", Journal of Field Robotics, Vol. 34, No. 6, Oct. 2016, pp. 1100–1122, available at:https://doi.org/10.1002/rob.21680

[25]Berenstein, R., "The use of agricultural robots in crop spraying/fertilizer applications", in Robotics and automation for improving agriculture. Burleigh Dodds Science Publishing, Jun. 2019, pp. 109–136, available at:https://doi.org/10.19103/as.2019.0056.10

[26]Santos, L., Santos, F., Mendes, J., Costa, P., Lima, J., Reis, R., Shinde, P., "Path planning aware of robot's center of mass for steep slope vineyards", Robotica, Vol. 38, No. 4, Jul. 2019, pp. 684–698, available at:https://doi.org/10.1017/s0263574719000961

[27]de Aguiar, A. S. P., dos Santos, F. B. N., dos Santos, L. C. F., de Jesus Filipe, V. M., de Sousa, A. J. M., "Vineyard trunk detection using deep learning – an experimental device benchmark", Computers and Electronics in Agriculture, Vol. 175, Aug. 2020, pp. 105535, available at:https://doi.org/10.1016/j.compag.2020.105535

[28]Roure, F., Moreno, G., Soler, M., Faconti, D., Serrano, D., Astolfi, P., Bardaro, G., Gabrielli, A., Bascetta, L., Matteucci, M., "Grape: Ground robot for vineyard monitoring and protection", 01 2018, pp. 249-260.

[29]Kerkech, M., Hafiane, A., Canals, R., "Vine disease detection in uav multispectral images using optimized image registration and deep learning segmentation approach", Computers and Electronics in Agriculture, Vol. 174, 04 2020.

[30]Bouloumpasi, E., Theocharis, S., Karampatea, A., Pavlidis, S., Mamalis, S., Koundouras, S., Merou, T., Vrochidou, E., Pachidis, T., Manios, M., Papakostas, G., Kaburlasos, V., "Exploration of viticultural tasks to be performed by an autonomous robot: Possibilities and limitations", 10 2020.

[31]Vrochidou, E., Tziridis, K., Nikolaou, A., Kalampokas, T., Papakostas, G. A., Pachidis, T. P., Mamalis, S., Koundouras, S., Kaburlasos, V. G., "An autonomous grape-harvester robot: Integrated system architecture", Electronics, Vol. 10, No. 9, Apr. 2021, pp. 1056, available at:https://doi.org/10.3390/electronics10091056

[32]Monta, M., Kondo, N., Shibano, Y., "Agricultural robot in grape production system", in Proceedings of 1995 IEEE International Conference on Robotics and Automation, Vol. 3, 1995, pp. 2504-2509 vol.3.

[33]Berenstein, R., Shahar, O. B., Shapiro, A., Edan, Y., "Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer", Intelligent Service Robotics, Vol. 3, No. 4, Sep. 2010, pp. 233–243, available at:https://doi.org/10.1007/s11370-010-0078-z

[34]Oberti, R., Marchi, M., Tirelli, P., Calcante, A., Iriti, M., Ho čevar, M., Baur, J., Pfaff, J., Schütz, C., Ulbrich, H., "Selective spraying of grapevine's diseases by a modular agricultural robot", Journal of Agricultural Engineering, Vol. 44, No. 2s, Sep. 2013, available at:https://doi.org/10.4081/jae.2013.271

[35]de Soto, M. G., Emmi, L., Perez-Ruiz, M., Aguera, J., de Santos, P. G., "Autonomous systems for precise spraying – evaluation of a robotised patch sprayer", Biosystems Engineering, Vol. 146, Jun. 2016, pp. 165–182, available at: https://doi.org/10.1016/j.biosystemseng.2015.12.018

[36]Adamides, G., "Agricultural robots in targeted spraying: A mini state-of-the-art review", Robotics & Automation Engineering Journal, Vol. 2, No. 2, Feb. 2017, available at: https://doi.org/10.19080/raej.2018.02.555581

[37]Adamides, G., Katsanos, C., Constantinou, I., Christou, G., Xenos, M., Hadzilacos, T., Edan, Y., "Design and development of a semi-autonomous agricultural vineyard sprayer: Human-robot interaction aspects", Journal of Field Robotics, Vol. 34, No. 8, May 2017, pp. 1407–1426, available at:https://doi.org/10.1002/rob.21721

[38]Furuno, S., Yamamoto, M., Mohri, A., "Trajectory planning of mobile manipulator with stability considerations", in 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422). IEEE, available at:https://doi.org/10.1109/robot.2003.1242116

[39]Pankert, J., Hutter, M., "Perceptive model predictive control for continuous mobile manipulation", IEEE Robotics and Automation Letters, Vol. 5, No. 4, 2020, pp. 6177-6184.

[40]Avanzini, G. B., Zanchettin, A. M., Rocco, P., "Constrained model predictive control for mobile robotic manipulators", Robotica, Vol. 36, No. 1, Apr. 2017, pp. 19–38, available at:https://doi.org/10.1017/s0263574717000133

[41]Minniti, M. V., Farshidian, F., Grandia, R., Hutter, M., "Whole-body MPC for a dynamically stable mobile manipulator", IEEE Robotics and Automation Letters, Vol. 4, No. 4, Oct. 2019, pp. 3687–3694, available at:https://doi.org/10.1109/lra.2019.2927955

[42]Korayem, M., Nazemizadeh, M., Azimirad, V., "Optimal trajectory planning of wheeled mobile manipulators in cluttered environments using potential functions", Scientia Iranica, Vol. 18, No. 5, Oct. 2011, pp. 1138–1147, available at: https://doi.org/10.1016/j.scient.2011.08.026

[43]Ren, S., Xie, Y., Yang, X., Xu, J., Wang, G., Chen, K., "A method for optimizing the base position of mobile painting manipulators", IEEE Transactions on Automation Science and Engineering, Vol. 14, No. 1, Jan. 2017, pp. 370–375, available at: https://doi.org/10.1109/tase.2016.2612694

[44]Liu, G., Sun, X., Liu, Y., Liu, T., Li, C., Zhang, X., "Automatic spraying motion planning of a shotcrete manipulator", Intelligent Service Robotics, Feb. 2021, available at:https://doi.org/10.1007/s11370-021-00348-9

[45]Wang, G., Yu, Q., Ren, T., Hua, X., Chen, K., "Task planning for mobile painting manipulators based on manipulating space", Assembly Automation, Vol. 38, No. 1, Feb. 2018, pp. 57–66, available at:https://doi.org/10.1108/aa-04-2017-044

[46]From, P. J., Gravdahl, J. T., "General solutions to functional and kinematic redundancy", in 2007 46th IEEE Conference on Decision and Control. IEEE, 2007, available at: https://doi.org/10.1109/cdc.2007.4434442

[47]Žlajpah, L., Müller, A., "A task space decomposition algorithm for the inverse kinematics of functionally redundant manipulators", in 2021 20th International Conference on Advanced Robotics (ICAR), 2021, pp. 1048-1053.

[48]Zanchettin, A. M., Rocco, P., "On the use of functional redundancy in industrial robotic manipulators for optimal spray painting", IFAC Proceedings Volumes, Vol. 44, No. 1, Jan. 2011, pp. 11 495–11 500, available at:https://doi.org/10.3182/20110828-6-it-1002.00687

[49]From, P. J., Gravdahl, J. T., "A real-time algorithm for determining the optimal paint gun orientation in spray paint applications", IEEE Transactions on Automation Science and Engineering, Vol. 7, No. 4, Oct. 2010, pp. 803–816, available at: https://doi.org/10.1109/tase.2009.2033567

[50]From, P. J., Gunnar, J., Gravdahl, J. T., "Optimal paint gun orientation in spray paint applications—experimental results", IEEE Transactions on Automation Science and Engineering, Vol. 8, No. 2, Apr. 2011, pp. 438–442, available at: https://doi.org/10.1109/tase.2010.2089450

[51]Chiacchio, P., Chiaverini, S., Sciavicco, L., Siciliano, B., "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy", The International Journal of Robotics Research, Vol. 10, No. 4, Aug. 1991, pp. 410–425, available at:https://doi.org/10.1177/027836499101000409

[52]Lillo, P. D., Chiaverini, S., Antonelli, G., "Handling robot constraints within a set-based multi-task priority inverse kinematics framework", in 2019 International Conference on Robotics and Automation (ICRA). IEEE, May 2019, available at: https://doi.org/10.1109/icra.2019.8793625

[53]Moe, S., Antonelli, G., Teel, A. R., Pettersen, K. Y., Schrimpf, J., "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results", Frontiers in Robotics and AI, Vol. 3, Apr. 2016, available at:https://doi.org/10.3389/frobt.2016.00016

[54]ik An, S., Lee, D., "Prioritized inverse kinematics: Generalization", IEEE Robotics and Automation Letters, Vol. 4, No. 4, Oct. 2019, pp. 3537–3544, available at: https://doi.org/10.1109/lra.2019.2927945

[55]Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S., "OSQP: an operator splitting solver for quadratic programs", Mathematical Programming Computation, Vol. 12, No. 4, 2020, pp. 637–672, available at:https://doi.org/10.1007/s12532-020-00179-2

[56]Coleman, D. T., Sucan, I. A., Chitta, S., Correll, N., "Reducing the barrier to entry of complex robotic software: a moveit! case study", 2014, available at: https://aisberg.unibg.it//handle/10446/87657

[57]Reynolds, A. G., Heuvel, J. E. V., "Influence of grapevine training systems on vine growth and fruit composition: A review", American Journal of Enology and Viticulture, Vol. 60, No. 3, Sep. 2009, pp. 251–268, available at:https://doi.org/10.5344/ajev.2009.60.3.251

[58]Vatavuk, I., Stuhne, D., Vasiljevi ć, G., Kovačić, Z., "Direct drive brush-shaped tool with torque sensing capability for compliant robotic vine suckering", Sensors, Vol. 23, No. 3, Jan. 2023, pp. 1195, available at:https://doi.org/10.3390/s23031195

[59]Stuhne, D., Vatavuk, I., Hrabar, I., Vasiljevi ć, G., Kovačić, Z., "Automated suckering of vines with a mobile robot and a torque-controlled suckering tool", in 2022 International Conference on Smart Systems and Technologies (SST), 2022, pp. 349-354.

[60]Dokoozlian, N., "THE EVOLUTION OF MECHANIZED VINEYARD PRODUCTION SYSTEMS IN CALIFORNIA", Acta Horticulturae, No. 978, mar 2013, pp. 265–278, available at:https://doi.org/10.17660%2Factahortic.2013.978.31

[61]Majeed, Y., Karkee, M., Zhang, Q., Fu, L., Whiting, M. D., "Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks", Computers and Electronics in Agriculture, Vol. 171, Apr. 2020, pp. 105308, available at:https://doi.org/10.1016/j.compag.2020.105308

[62]Majeed, Y., Karkee, M., Zhang, Q., "Estimating the trajectories of vine cordons in full foliage canopies for automated green shoot thinning in vineyards", Computers and Electronics in Agriculture, Vol. 176, Sep. 2020, pp. 105671, available at: https://doi.org/10.1016/j.compag.2020.105671

[63]Majeed, Y., Karkee, M., Zhang, Q., Fu, L., Whiting, M. D., "Development and performance evaluation of a machine vision system and an integrated prototype for automated green shoot thinning in vineyards", Journal of Field Robotics, Vol. 38, No. 6, Jan. 2021, pp. 898–916, available at:https://doi.org/10.1002/rob.22013

[64]Martelloni, L., Raffaelli, M., Frasconi, C., Fontanelli, M., Peruzzi, A., D'Onofrio, C., "Using flaming as an alternative method to vine suckering", Agronomy, Vol. 9, No. 3, Mar. 2019, pp. 147, available at:https://doi.org/10.3390/agronomy9030147

[65]Polic, M., Car, M., Petric, F., Orsag, M., "Compliant plant exploration for agricultural procedures with a collaborative robot", IEEE Robotics and Automation Letters, Vol. 6, No. 2, 2021, pp. 2768-2774.

[66]Asada, H., Kanade, T., Takeyama, I., "Control of a direct-drive arm", Journal of Dynamic Systems, Measurement, and Control, Vol. 105, No. 3, Sep. 1983, pp. 136–142, available at:https://doi.org/10.1115/1.3140645

[67]Ebner, M., Wallace, R., "A direct-drive hand: design, modeling and control", in Proceedings of 1995 IEEE International Conference on Robotics and Automation. IEEE, available at:https://doi.org/10.1109/robot.1995.525514

[68]Her, M.-G., Hsu, K.-S., Lan, T.-S., Karkoub, M., Journal of Intelligent and Robotic Systems, Vol. 35, No. 3, 2002, pp. 247–264, available at:https: //doi.org/10.1023%2Fa%3A1021156903545

[69]Bhatia, A., Johnson, A., Mason, M. T., “Direct drive hands: Force-motion transparency in gripper design”, in Robotics: Science and Systems XV. Robotics: Science and Systems Foundation, Jun. 2019, available at:https://doi.org/10.15607/rss.2019.xv.053

[70]Iqbal, J., Tsagarakis, N. G., Caldwell, D. G., “A human hand compatible optimised exoskeleton system”, in 2010 IEEE International Conference on Robotics and Biomimetics, 2010, pp. 685-690.

[71]Seok, S., Wang, A., Chuah, M. Y., Otten, D., Lang, J., Kim, S., “Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot”, in 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 3307-3312.

[72]Wensing, P. M., Wang, A., Seok, S., Otten, D., Lang, J., Kim, S., “Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots”, IEEE Transactions on Robotics, Vol. 33, No. 3, 2017, pp. 509-522.

[73]Seok, S., Wang, A., Chuah, M. Y., Hyun, D. J., Lee, J., Otten, D. M., Lang, J. H., Kim, S., “Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot”, IEEE/ASME Transactions on Mechatronics, Vol. 20, No. 3, 2015, pp. 1117-1129.

[74]Katz, B., “A low cost modular actuator for dynamic robots”, Doktorski rad, 01 2018.

[75]SaLoutos, A., Stanger-Jones, E., Kim, S., “Fast reflexive grasping with a proprioceptive teleoperation platform”, in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, Oct. 2022, available at:https://doi.org/10.1109/iros47612.2022.9981383

[76]Ostyn, F., Vanderborght, B., Crevecoeur, G., “Design and control of a quasi-direct drive robotic gripper for collision tolerant picking at high speed”, IEEE Robotics and Automation Letters, Vol. 7, No. 3, jul 2022, pp. 7692–7699, available at: https://doi.org/10.1109%2Flra.2022.3184777

[77]Roberts, K., “Robot active touch exploration: constraints and strategies”, in Proceedings., IEEE International Conference on Robotics and Automation. IEEE Comput. Soc. Press, available at:https://doi.org/10.1109/robot.1990.126118

[78]Ahmad, S., Lee, C. N., “Shape recovery from robot contour-tracking with force feedback”, Advanced Robotics, Vol. 5, No. 3, Jan. 1990, pp. 257–273, available at: https://doi.org/10.1163/156855391x00197

[79]Doulgeri, Z., Karayiannidis, Y., "Force/position regulation for a robot in compliant contact using adaptive surface slope identification", IEEE Transactions on Automatic Control, Vol. 53, No. 9, Oct. 2008, pp. 2116–2122, available at:https://doi.org/10.1109/tac.2008.930183

[80]Lepora, N. F., Aquilina, K., Cramphorn, L., "Exploratory tactile servoing with active touch", IEEE Robotics and Automation Letters, Vol. 2, No. 2, Apr. 2017, pp. 1156–1163, available at:https://doi.org/10.1109/lra.2017.2662071

[81]Rosales, C., Spinelli, F., Gabiccini, M., Zito, C., Wyatt, J. L., "GPAtlasRRT: A local tactile exploration planner for recovering the shape of novel objects", International Journal of Humanoid Robotics, Vol. 15, No. 01, Feb. 2018, pp. 1850014, available at: https://doi.org/10.1142/s0219843618500147

[82]Bauza, M., Canal, O., Rodriguez, A., "Tactile mapping and localization from high-resolution tactile imprints", in 2019 International Conference on Robotics and Automation (ICRA). IEEE, May 2019, available at:https://doi.org/10.1109/icra.2019.8794298

[83]Polic, M., Krajacic, I., Lepora, N., Orsag, M., "Convolutional autoencoder for feature extraction in tactile sensing", IEEE Robotics and Automation Letters, Vol. 4, No. 4, Oct. 2019, pp. 3671–3678, available at:https://doi.org/10.1109/lra.2019.2927950

[84]"Odrive website documentation", Available: https://docs.odriverobotics.com/v/latest/manual/control.htmlt control.

[85]Thrun, S., Burgard, W., Fox, D., Probabilistic robotics. Cambridge, Mass.: MIT Press, 2005, available at:http://www.amazon.de/gp/product/0262201623/102-8479661-9831324?v=glance&n=283155&n=507846&s=books&v=glance

Biography

Ivo Vatavuk is a researcher at the Laboratory for Robotics and Intelligent Control Systems within the University of Zagreb Faculty of Electrical Engineering and Computing (UNIZG-FER). He graduated and received his master's degree in 2018 at the same institution with a thesis "Use of Centroidal Dynamics for Automatic Generation of Humanoid Whole-Body Movement during Task Execution". He joined LARICS later that year.

His research interests are the application of optimization in robotics, optimal control, mobile manipulation and legged robots. As a Ph.D. student, he participated in the ERL competition (European Robotics League), MBZIRC2020 (Mohamed Bin Zayed International Robotics Challenge), HEKTOR project (Heterogeneous Autonomous Robotic System in Viticulture and Mariculture) and the simulation phase of MBZIRC2023. So far, he has authored and co-authored 8 research papers, and one book chapter.

List of publications

Book chapters

1. Arbanas, B., Petric, F., Batinovi ć, A., Polić, M., Vatavuk, I., Marković, L., Car, M., Hrabar, I., Ivanović, A., and Bogdan, S. (2022). From ERL to MBZIRC: Development of An Aerial-Ground Robotic Team for Search and Rescue. In Automation and Control. Automation and Control-Theories and Applications. IntechOpen. https://doi.org/10.5772/intechopen.99210

Journal papers

1. Vatavuk, I., Poli ć, M., Hrabar, I., Petric, F., Orsag, M., and Bogdan, S. (2022). Autonomous, Mobile Manipulation in a Wall-building Scenario: Team LARICS at MBZIRC 2020. Field Robotics, 2, 201–221. https://doi.org/10.55417/fr.2022008

2.Kapetanovi ć, N., Goričanec, J., Vatavuk, I., Hrabar, I., Stuhne, D., Vasiljević, G., Kovačić, Z., Mišković, N., Antolović, N., Anić, M., and Kozina, B. (2022). Heterogeneous Autonomous Robotic System in Viticulture and Mariculture: Vehicles Development and Systems Integration. Sensors, 22, 2961. https://doi.org/10.3390/s22082961 (IF = 3.9, Q2)

3.Vatavuk, I., Vasiljevi ć G., and Kovačić, Z. (2022). Task Space Model Predictive Control for Vineyard Spraying with a Mobile Manipulator. Agriculture 2022, 12, 381. https://doi.org/10.3390/agriculture12030381 (IF = 3.6, Q1)

4.Vatavuk, I., Stuhne, D., Vasiljevi ć, G., and Kovačić, Z. (2023). Direct Drive Brush-Shaped Tool with Torque Sensing Capability for Compliant Robotic Vine Suckering. Sensors, 23, 1195. https://doi.org/10.3390/s23031195 (IF = 3.9, Q2)

Conference papers

1.Gori čanec J., Kapetanović N., Vatavuk I., Hrabar I., Vasiljević G., Gledec G., Stuhne D., Bogdan S., Orsag M., Petrović T., Mišković N., Kovačić Z., Kurtela A., Bolotin J., Kožul V., Glavić N., Antolović N., Anić M., Kozina B., and Cukon M. (2021). Heterogeneous autonomous robotic system in viticulture and mariculture-project overview. 16th International Conference on Telecommunications (ConTEL), 181-188. https://doi.org/10.23919/ConTEL52528.2021.9495969

2.Stuhne, D., Vatavuk, I., Hrabar, I., Vasiljevi ć, G., and Kovačić, Z. (2022). Automated Suckering of Vines with a Mobile Robot and a Torque-Controlled Suckering Tool. International Conference on Smart Systems and Technologies (SST), 349-354. https://doi.org/10.1109/SST55530.2022.9954661

3.Vatavuk, I., and Kova čić, Z. (2021). Precise Jump Planning using Centroidal Dynamics based Bilevel Optimization. International Conference on Robotics and Automation (ICRA), 3026-3032. https://doi.org/10.1109/ICRA48506.2021.9561101

4.Zori ć, F., Križmančić, M., Vatavuk, I., and Orsag, M. (2023). Intermittent remote visual servoing in a heterogeneous robotic team. European Control Conference (ECC), 1-6. https://doi.org/10.23919/ECC57647.2023.10178239

Životopis

Ivo Vatavuk je istraživač u Laboratoriju za Robotiku i Inteligentne Sustave Upravljanja u sklopu Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu (UNIZG-FER). Diplomirao je 2018. godine na istom fakultetu diplomskim radom naslova "Korištenje centroidalne dinamike za automatsko generiranje pokreta cijelog tijela humanoidnog robota pri izvođenju zadatka". Iste godine se pridružuje LARICS-u.

Njegove interesne sfere su primjena optimizacije u robotici, optimalno upravljanje, mobilna manipulacija te hodajući roboti. Kao doktorski student sudjelovao je na natjecanjimu ERL (European Robotics League), MBZIRC2020 (Mohamed Bin Zayed International Robotics Challenge), projektu HEKTOR (Heterogeneous autonomous robotic system in viticulture and mariculture), te simulacijskoj fazi natjecanja MBZIRC2023. Dosad je autor ili koautor 8 znanstvenih radova i jednog poglavlja knjige.