Safe collaborative framework for compliant industrial manipulators

Marić, Bruno

Doctoral thesis / Disertacija

2024

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Bruno Marić

SAFE COLLABORATIVE FRAMEWORK FOR COMPLIANT INDUSTRIAL MANIPULATORS

DOCTORAL THESIS

Zagreb, 2024

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Bruno Marić

SAFE COLLABORATIVE FRAMEWORK FOR COMPLIANT INDUSTRIAL MANIPULATORS

DOCTORAL THESIS

Supervisor: Associate Professor Matko Orsag, PhD

Zagreb, 2024

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Bruno Marić

SIGURNO KOLABORATIVNO OKRUŽENJE ZA PODATNE INDUSTRIJSKE MANIPULATORE

DOKTORSKI RAD

Mentor: izv. prof. dr. sc. Matko Orsag

Zagreb, 2024.

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

Supervisor: Associate Professor Matko Orsag, PhD

Doctoral thesis has 125 pages

Dissertation No.: ____________________

About the Supervisor

Matko Orsag is an associate professor at the University of Zagreb Faculty of Electrical Engineering and Computing (UNIZG-FER). He has been involved as a researcher in various projects financed by the government and industry. In 2011/2012, he worked as a visiting researcher at the Drexel University, Philadelphia, USA as a recipient of the Fulbright exchange grant.

As a researcher, he participated in several national and international research projects in the field of robotics, control, and automation. Currently, he is working as a researcher in several EU projects: AeRoTwin, ENCORE, RoboCom++. He is the principal investigator of project SPECULARIA, funded by the Croatian Science Foundation, and project ENDORSE, funded through the H2020 framework. He authored and coauthored over 30 scientific and professional papers, including journal and conference papers, as well as a monograph and a book chapter in the field of unmanned aerial systems and robotics.

He is serving as an associate editor of AUTOMATIKA – Journal for Control, Measurement, Electronics, Computing, and Communications. He is a member of IEEE and euRobotics Aerial Robotics Topic Group. He is currently a co-chair of Croatian section of IEEE RAS.

He is involved as a reviewer in journals and at conferences, and as an editor and guest editor in several scientific journals (Automatika, JINT, etc.). He is a member of the IEEE, currently serving as the chairman of the IEEE Robotics and Automation Society, Section IEEE Croatia, and a member of topical group for aerial robotics. He is a member of Scientific and Expert Council of the Nikola Tesla Innovation Center (ICENT). In 2019, he received the young scientist award of the Croatian Technical Academy "Vera Johanides".

O mentoru

Matko Orsag izvanredni je profesor na Fakultetu Elektrotehnike i računarstva, Sveučilišta u Zagrebu (UNIZG-FER). Kao istraživač, uključen je u brojne projekte financirane državnim sredstvima te industrijske projekte. U razdoblju 2011./2012. kao dobitnik sredstava za razmjenu u sklopu programa Fulbright, radio je kao gostujući istraživač na Sveučilištu Drexel, Philadelphia, Sjedinjene Američke Države.

Kao istraživač sudjeluje u nekoliko nacionalnih i internacionalnih istraživačkih projekata u području robotike, upravljačkih algoritama te automatizacije. Trenutačno je zaposlen kao istraživač na nekoliko EU projekata: AeRoTwin, ENCORE, RoboCom++. Glavni je istraživač na projektu SPECULARIA koji financira Hrvatska Zaklada za Znanost te na projektu ENDORSE financiranom kroz program Obzor 2020. Kao autor i koautor objavio je preko 30 znanstvenih i profesionalnih članaka, uključujući članke u časopisima i na konferencijama, kao i monografiju te poglavlje u knjizi na temu bespilotnih letjelica i robotike.

Trenutno je urednik časopisa AUTOMATIKA - Journal for Control, Measurement, Electronics, Computing, and Communications. Član je IEEE te euRobotics Aerial Robotics Topic grupe kao i supredsjedatelj hrvatskog ogranka IEEE RAS.

Angažiran je kao recenzent u časopisima i na konferencijama, te kao urednik i gostujući urednik nekoliko časopisa (Automatika, JINT i dr.). Član je IEEE-a, trenutno je predsjedavajući ogranka Društva IEEE Robotics and Automation Society Sekcije IEEE Hrvatska i član je tematske grupe za zračnu robotiku. Član je Znanstvenog i stručnog vijeća Inovacijskog centra Nikola Tesla (ICENT). Godine 2019. dobio je nagradu za mladog znanstvenika Hrvatske tehničke akademije „Vera Johanides“.

Acknowledgements

I would like to begin by expressing my heartfelt gratitude to all those who supported me throughout the journey of completing this thesis. Then, I would like to express my deepest gratitude to my supervisor, associate professor Matko Orsag, for their invaluable guidance, unwavering support, and constructive criticism throughout the course of this research. Their expertise, encouragement, and dedication have been instrumental in shaping the development of this thesis.

I am deeply grateful to my dear colleague Frano Petric, whose insightful feedback, encouragement, and unwavering support have been invaluable throughout this journey. Their constructive criticism and willingness to engage in meaningful discussions have greatly enriched the quality of this work. Furthermore, I extend my appreciation to my colleagues at LARICS (Laboratory for Robotics and Intelligent Control Systems) for providing me with the essential resources, facilities, and conducive environment needed to conduct my research. The collaborative atmosphere and the collective efforts of the laboratory team have played a significant role in the success of this project.

I am profoundly grateful to my parents, Jadranka and Zeljko, for their unwavering support, guidance, and love. They have been my steadfast pillars of strength, nurturing my dreams and aspirations with boundless encouragement and care.

Above all, I want to emphasize my deepest appreciation to my beloved wife Antonija. Her unwavering support and understanding throughout this journey have been the cornerstone of my success. Her encouragement and steadfast belief in me have served as a constant source of inspiration, propelling me to overcome obstacles and strive for excellence. Her presence by my side has infused every step of this journey with meaning and fulfillment.

Thank you all for your support and belief in me.

Abstract

This thesis is primarily focused on developing a safe collaborative framework for compliant industrial manipulators. Industrial manipulators have been a cornerstone of modern industry for several decades, excelling in tasks that are repetitive, tedious, or hazardous to humans. They are characterized by their precision, repeatability, and high payload capacity. However, they have very limited flexibility in terms of programming and deployment. For this reason, they are typically found in large-scale industries. On the other hand, as industries evolve and labor shortages become more prevalent, there is a growing need for robots to enter smaller-scale operations and handle delicate tasks that require adaptability and compliance in direct interaction with the environment. This is where collaborative robots have emerged as a promising solution. While collaborative robots offer features that enable soft contact and safe human-robot interaction, off-the-shelf models often lack the precision, repeatability, payload capacity, and reach required by industrial applications. To bridge this this gap, the goal of this thesis is to develop a collaborative framework that combines the strengths of both industrial and collaborative manipulators. By integrating compliant behaviour into industrial manipulators, this thesis aims to enable safe interaction with humans and adaptability to dynamic environments, while retaining the precision, repeatability, and payload capacity essential for industrial tasks. Through this endeavor, the thesis seeks to enhance the versatility and applicability of manipulator systems across a wide range of industrial settings.

The thesis introduces three key components of the framework. The first component focuses on Compliance Control for industrial manipulators. Drawing inspiration from fundamental concepts such as impedance, admittance, and force control, a compliance control algorithm tailored for use with stiff joint-position controlled industrial manipulators was developed. This component specifically addresses compliant Cartesian motion, with particular attention given to understanding the interaction between the robot and environments of varying stiffness. The second component of the system deals with human-robot interaction, which is partially enabled by the first component. Here, different aspects of Programming-by-Demonstration (PbD) were explored, beginning with kinesthetic teaching. Additionally, a novel PbD concept using a virtual pen device based on a Motion Capture (mocap) system is introduced. The effectiveness and intuitiveness of the proposed PbD system was evaluated through a case study. Given the intended close collaboration between humans and machines in the proposed system, particular emphasis is placed on ensuring human safety. To address safety concerns, the third component of the system focuses on collision detection and analyzes the proposed system in terms of compliance with ISO technical specification for collaborative robotics. This component aims to ensure that the collaborative framework meets the necessary safety standards while maintaining efficiency and effectiveness in industrial applications.

As this framework is specifically tailored for industrial applications, a crucial aspect of the thesis involves the experimental validation of the developed system in real-case scenarios across different industrial settings. The first application involves the delicate grinding of composite materials used in aircraft manufacturing. While this process is currently partially automated using standard industrial manipulator equipped with active flanges for controlling contact force in one axis, the majority of the work is still carried out by human operators. Given the hazardous nature of grinding, particularly the production of fine dust, our primary objective was to minimize the time human operators spend in such environments. By incorporating active compliance control in all six degrees of freedom (6DOF) for autonomous grinding, and leveraging human skill through a programming-by-demonstration approach, we significantly increased the areas treated by the robot. In the second application, the system for deep-micro-hole drilling of moulds used in glass container manufacturing was utilized. Here, the focus was on improving operator comfort during skill demonstration, resulting in capturing the essence of the demonstration more effectively. Once again, the developed compliance control system, combined with skill replication, led to a substantial increase in the number of holes drilled by the robot compared to human operators. Lastly, the autonomous putty plastering application was addressed, which involves the manipulation of deformable objects due to the nature of the plastering knife. This application further demonstrates the versatility and effectiveness of the developed framework in handling diverse industrial tasks.

Contribution of this thesis can be decoupled in:

- *A compliant control system for an industrial manipulator based on real time forward dynamics computation for both soft and rigid body contact for industrial applications involving deep micro drilling, plastering and polishing* (Chapters 3, 4, 6)
- *Safe collaborative human-robot interface for industrial manipulators which enables programming by demonstration* (Chapter 5)

Keywords: Compliance and Impedance Control, Force Control, Contact Modeling, Collaborative Robots in Manufacturing, Human-Centered Robotics, Learning from Demonstration

Sigurno kolaborativno okruženje za podatne industrijske manipulatore

Glavni fokus ovog doktorskog rada je razvoj sigurnog kolaborativnog okruženja za podatne industrijske manipulatore. Industrijski manipulatori su temelj moderne industrije već nekoliko desetljeća, izvrsni su u obavljanju zadataka koji su ponavljajući, dosadni ili opasni za ljude. Karakteriziraju ih preciznost, izvrsna ponovljivost te visoka nosivosti i doseg. Međutim, njihova fleksibilnost u pogledu programiranja i implementacije je iznimno ograničena. Zbog navedenoga, obično su prisutni u industrijama velikog obujma. Međutim, razvojem industrije te sveprisutnim problemom nedostatka radne snage raste i potreba za robotima primjenjivim u industrijama manjeg obujma te primjenama koje uključuju delikatne zadatke i zahtijevaju prilagodljivost i podatnost u izravnoj interakciji s okolinom. S obzirom na navedeno, kolaborativni roboti postaju sve više popularni. Iako kolaborativni roboti nude značajke koje omogućuju mekani kontakt i sigurnu interakciju čovjeka i robota, modeli koji su trenutno dostupni na tržištu često nemaju preciznost, ponovljivost i nosivost potrebnu u industrijskim primjenama. Kako bi se premostila navedena praznina, cilj ovog doktorskog rada je razvoj kolaborativnog okruženja koji spaja najbolje značajke industrijskih i kolaborativnih manipulatora. Dodavanjem podatnosti industrijskim manipulatorima, cilj je omogućiti sigurnu interakciju s ljudima kao i mogućnost prilagodbe dinamičkim okolinama zadržavajući preciznost, ponovljivost i nosivost kao bitne faktore inustrijskih zadaća. Ovi napori nastoje proširiti i poboljšati primjenu manipulacijskih sustava u širokom rasponu industrijskih primjena.

Doktorski rad uvodi tri ključne komponente kolaborativnog okruženja. Prva komponenta se fokusira na podatno upravljanje industrijskim manipulatorima. Inspirirano baznim konceptima poput upravljanja impedancijom, admitancijom te silom, razvijen je sustav podatnog upravljanja za primjenu na industrijskim manipulatorima koji su kruto upravljani po poziciji zglobova. Posebna pažnja je posvećena podatnom kartezijskom gibanju, s naglaskom na razumijevanje interakcije između robota i okolina različitih krutosti. Druga komponenta sustava bavi se interakcijom između čovjeka i robota, što je djelomično omogućeno prvom komponentom sustava. Nadalje, obrađeni su različiti aspekti programiranja putem demonstracije, počevši s kinestetičkim učenjem. Predstavljen je novi koncept programiranja putem demonstracije pomoću virtualne olovke, uređaja temeljenog na sustavu za praćenje pokreta. Učinkovitost i intuitivnost predloženog sustava programiranja putem demonstracije ispitana je kroz studiju. Imajući na umu blisku interakciju čovjeka i robota kroz predloženo kolaborativno okruženje, poseban naglasak stavljen je na sigurnost čovjeka. Stoga se treća komponeta sustava bavi detekcijom kolizije i analizira predloženo okruženje u kontekstu usklađenosti s ISO tehničkom specifikacijom 15066 za kolaborativne robote. Ova komponenta ima cilj osigurati kolaborativno okruženje usklađeno sa sigurnosnim standardima, uz istovremeno održavanje učinkovitosti u industrijskim

primjenama.

S obzirom da je predloženo kolaborativno okruženje namijenjeno industrijskim primjenama, ključan aspekt ovog doktorskog rada je i eksperimentalna provjera razvijenog sustava u stvarnim scenarijima različitih industrijskih primjena. Prva primjena uključuje delikatno brušenje kompozitnih materijala koji se koriste u proizvodnji zrakoplova. Iako je ovaj proces trenutno djelomično automatiziran korištenjem standardnog industrijskog manipulatora opremljenog aktivnim alatom koji omogućava kontrolu sile kontakta u jednoj osi, većinu posla i dalje obavljaja čovjek. S obzirom da proces brušenja stvara sitnu prašinu koja je posebice opasna za čovjeka, jedan od glavnih ciljeva bio je smanjiti vrijeme koje čovjek provodi u takvom radnom okruženju. Integriranjem aktivne podatnosti u svih šest stupnjeva slobode robotskog sustava te korištenjem vještina čovjeka kroz programiranje demonstracijom, značajno je povećan udio površina obrađenih robotom. U drugoj primjeni, predloženo okruženje je korišteno u primjeni dubokog mikro bušenja kalupa koji se koriste u proizvodnji staklenih posuda. Fokus je bio unaprijediti komfor operatera tijekom demonstracije vještine, što je rezultiralo učinkovitijem snimanju same biti vještine čovjeka. Ponovno se pokazalo kako predloženo kolaborativno okruženje u kombinaciji s pristupom programiranja demonstracijom dovodi do značajnog povećanja broja probušenih rupica od strane robota u odnosu na čovjeka. Zaključno, razvijeno okruženje je primjenjeno na autonomni sustav gletanja, što uključuje manipulaciju deformabilnim objektom, s obzirom na prirodu alata za gletanje. Ovom primjenom dodatno je prikazana svestranost i učinkovitost razvijenog kolaborativnog okruženja za industrijske manipulatore. Doktorski rad je podijeljen u sedam poglavlja:

U prvom poglavlju dan je uvod doktorskog rada, počevši s klasifikacijom robotskih manipulatora te njihovim osnovnim primjenama. S obzirom na temu doktorskog rada, poseban naglasak stavljen je na klasifikaciju robotskih manipulatora, industrijske i kolaborativne. Za industrijske manipulatore, predstavljene su njihove prednosti, prvenstveno u pogledu nosivosti, dohvata, ponovljivosti te preciznosti. Također, dan je osvrt na način upravljanja industrijskim manipulatorima, koji su tipično upravljani po poziciji zglobova. Navedenim upravljanjem, industrijski manipulatori nemaju mogućnosti osjeta dodira s okolinom, zbog čega su uvijek ograđeni sigurnosnim ogradama kako bi se spriječio ulazak čovjeka u radno okruženje robota. Uz sve navedeno kao i činjenicu da su takvi robotski sustavi nefleksibilni u pogledu prilagodbe novim zadaćama, predstavljena je potreba te mogućnosti koje pružaju kolaborativni manipulatori. Činjenica da su kolaborativni manipulatori upravljani po momentu zglobova, što im omogućava osjećanje dodira s okolinom te otvara razne mogućnosti bliske i sigurne interakcije između čovjeka i robota. Jedan od primjera interakcije je i programiranje demonstracijom, gdje čovjek na jednostavan način pokazuje robotu kako odraditi određeni zadatak, što otvara razne mogućnosti primjene robota čak i u proizvodnim procesima malog obujma. Poglavlje završava adresiranjem značajne razlike između industrijskih i kolaborativnih manipulatora, posebice kada je riječ o preciznosti, snazi, ali također i o ekonomskoj opravdanosti korištenja kolaborativnih robota u modernoj industriji.

U drugom poglavlju definiran je problem kojim se ovaj doktorski rad bavi te tri komponente predloženog sigurnog kolaborativnog okruženja. Prva komponenta je podatno upravljanje robotom koji nema mogućnost osjeta dodira, što uključuje opremanje robota potrebnom senzorikom i razvoj algoritama podatnog upravljanja za industrijske manipulatore. Druga komponenta se odnosi na implementaciju programiranja demonstracijom, čiji je cilj jednostavno programiranje robota te bliska interakcija čovjeka i robota. Zaključno, treća komponenta adresira osiguranje sigurnosti čovjeka u bliskoj interakciji s robotom. Za svaku komponentu predloženog sustava predstavljen je pregled područja te detaljno obrazloženje kako izvorni znanstveni doprinos ovog doktorskog rada proširuje znanje znanstvene zajednice.

U trećem poglavlju predstavljena je prva komponenta predloženog kolaborativnog okuženja, uvođenjem podatnog upravljanja za industrijske manipulatore. Poglavlje započinje prikazom baznih algoritama podatnog upravljanja zasnovanih na upravljanju impedancije, admitancije i sile. Predstavljen je i algoritam podatnog upravljanja koji integrira tri bazna pristupa, dok se za mapiranje iz kartezijske domene alata u domenu zakreta zglobova koriste metoda simulacije unaprijedne dinamike ili metoda bazirana na Jakobijan matrici. Predstavljene su prednosti metode simulacije unaprijedne dinamike ponajprije u pogledu izbjegavanja problema singulariteta, međutim i ograničenja ove metode kada je riječ o ulasku u nulprostor robota. S druge strane, metoda bazirana na Jakobijan matrici je računalno manje zahtjevna, međutim posebnu pažnju je potrebno obratiti na problem singulariteta, posebice singulariteta inverza Jakobijan matrice. Dodatno, raspravlja se o posebnoj izvedenici upravljačkog algoritma razvijenog za kartezijski ograničena gibanja. U izvedenici algoritma sa simulacijom unaprijedne dinamike kartezijski ograničeno gibanje se postiže krutim ponašanjem impedancijskog filtra u željenoj osi. Ukoliko je potrebno postići potpuno ograničeno gibanje željenom osi, kao za primjer robotskog bušenja, razvijena je posebna izvedenica algoritma podatnog upravljanja koja ograničava gibanje robota u jednoj osi. Poglavlje završava analizom stabilnosti, gdje je interakcija robota i okoline modelirana pomoću masa, prigušnica i opruga. Analiza stabilnosti je provedena pomoću metode krivulje mjesta korijena. Analizom je pokazano kako je sustav stabilan ukoliko robot ima kruće i/ili sporije ponašanje od ponašanja okoline. Analiza je provjerena kroz simulacijsko okruženje te je eksperimentalno potvrđena.

U četvrtom poglavlju fokus je na drugoj komponenti predloženog kolaborativnog okruženja, naglašavajući blisku interakciju čovjeka i robota, osobito u paradigmi programiranja putem demonstracije. U početku je obrađen kinestetički pristup, koji se bazira na razvijenom algoritmu podatnog upravljanja, koji je posebno parametriran kako bi operateru omogućio vođenje robota držanjem robota za alat. Nakon toga slijedi predstavljanje novog pristupa programiranju demonstracijom koristeći sustav praćenja alata. Praćenje alata je ostvareno komercijalno dostupnim sustavom baziranim na infracrvenim kamerama te markerima koji se postavljaju na objekt za praćenje. Kako bi omogućili jednostavno i intuitivno programiranje demonstracijom, predstavljena je virtualna olovka, uređaj koji se sastoji od markera za praćenje smještenih na vrhu, tijela s integriranim ugradbenim elektroničkim sustavom za funkcije komunikacije te vrha koji dolazi u dvije izvedbe. Kako bi se omogućila puna funkcionalnost virtualne olovke, predstavljeni su algoritmi kalibracije koji omogućavaju brzo i efikasno određivanje transformacija, od centroida markera za praćenje do vrha olovke, za oba tipa vrha, točkasti i linijski. Metoda kalibracije je potvrđena kroz simulacijsko okruženje te eksperimentalno. Prikazane su razne primjene predstavljene virtualne olovke, počevši s virtualnim mjerenjem i markiranjem točaka i putanja, sve do mogućnosti skeniranja objekata jednostavnim šaranjem proizvoljnog oblika po površini predmeta. Također, u sklopu ovog poglavlja predstavljen je univerzalni algoritam lokalizacije, koji omogućava lokalizaciju poznatih 3D objekata neovisno o izvoru snimljenog oblaka točaka. Ergonomski aspekt i intuitivnost oba pristupa programiranja demonstracijom ispitani su kroz studiju o korisničkom iskustvu.

U petom poglavlju dan je pregled završne komponente kolaborativnog okruženja, što uključuje sigurnosni aspekt predloženog okruženja. Poglavlje započinje pregledom kolaborativnih načina rada koje prepoznaje ISO tehnička specifikacija 15066, koja se bavi sigurnosnim problemima povezanim uz kolaborativne robote. Nadalje, predstavljen je poboljšani algoritam detekcije kolizije koji se bazira na biblioteci "GPU-Voxels". Predloženo poboljšanje sadrži metodu kalibracije senzora za praćenje okoline robota te algoritam detekcije kolizije baziran na dinamičkom alociranju opisanog prostora gibanja robota. Dinamičkim alociranjem prostora u kojemu će se robot gibati u ovisnosti o brzini gibanja robota, suprotno statičkom gdje se alocira prostor kompletne trajektorije, ostvaruje se veća iskoristivost radnog prostora robota. Valjanost i učinkovitost predloženog algoritma je eksperimentalno potvrđena. Zaključno, poglavlje pokazuje sukladnost predloženog kolaborativnog okruženja za industrijske manipulatore sa svim zahtjevima kolaborativnih načina rada definiranih u tehničkoj specifikaciji.

U šestom poglavlju , uzevši u obzir primarni fokus ovog doktorskog rada na industrijskim primjenama, predstavljena su tri različita industrijska scenarija predloženog kolaborativnog okruženja. U uvodu poglavlja predstavljen je eksperimentalni postav koji uključuje industrijski manipulator *KUKA KR10* upravljan pomoću upravljačke jedinice *KUKA KR C4 Compact* te opremljen senzorom sile s osjetom sile i momenta u šest stupnjeva slobode. Upravljačka jedinica je opremljena *Robot Sensor Interface (RSI)* sučeljem koje omogućava vanjsku kontrolu robota preko računala te operativnog sustava *Robot Operating System (ROS)*. ROS integracija se pokazala ključnom za jednostavno i učinkovito korištenje svih komponenata predstavljenog kolaborativnog okruženja za industrijske manipulatore.

Prvi industrijski scenarij je delikatno brušenje kompozitnih materijala koji se koriste u proizvodnji zrakoplova, odnosno panela smještenih u unutrašnjosti zrakoplova, razvijen u sklopu projekta ENDORSE (*Effective Robotic GriNDing of Surface Areas through HORSE framework)*. Iako je proces brušenja trenutno djelomično robotiziran koristeći standardni predprogramirani industrijski manipulator opremljen s aktivnom flandžom s kontrolom sile u jednoj osi, značajan dio proizvodnje se i dalje oslanja na iskusne radnike. Uzimajući u obzir zdravstvenu štetnost procesa brušenja, glavni cilj je bio minimizirati vrijeme koje radnici provode u štetnoj okolini. Uzevši u obzir problematiku planiranja brušenja kompleksnih 3D površina te činjenicu da se proces brušenja većinom oslanja na iskustvo i vještine radnika, cilj je bio razviti robotski sustav koji će omogućiti prijenos znanja i iskustva radnika kroz blisku interakciju robota i čovjeka. S obzirom na zahtjeve procesa brušenja u kontekstu mase alata te potrebnog dohvata, za robotsko brušenje odabran je industrijski manipulator, opremljen senzorom sile. Manipulator je upravljan pomoću predloženog algoritma podatnog upravljanja s dodatnom funkcijom koja omogućava kinestetičko učenje. Kako bi se smanjio utjecaj mase alata te vibracija koje alat stvara, poseban naglasak je stavljen na obradu signala senzora sile. Zaključno, eksperimentalno je potvrđena uspješnost predloženog robotskog sustava, gdje je razina robotske obrade pojedinih pripravaka podignuta do preko 90%.

Drugi industrijski scenarij je duboko mikrobušenje kalupa za proizvodnju staklenih posuda. Iako je proizvodnja kalupa gotovo u potpunosti robotizirana i automatizira, postupak dubokog mikrobušenja i dalje u potpunosti ovisi o iskustvu i vještinama malog broja radnika. Uzmajući u obzir ograničenja kinestetičkog pristupa programiranju demonstracijom iz prethodnog primjera kao i rezultate provedene studije predstavljene u ranijim poglavljima, u ovom slučaju fokus je stavljen na snimanju demonstracije iskusnih operatera u njima poznatom radnom okruženju. U uvodu je predstavljena problematika bušenja, što uključuje tehnologiju bušenja te kinematiku koja opisuje trajektoriju bušenja. Nadalje, fokus je stavljen na snimanje demonstracije, što uključuje razvoj postava za snimanje, skeniranje i lokalizaciju pripravaka unutar radnog okruženja radnika. Nakon provedenog snimanja, snimljene demonstracije su obrađene te pripremljene za reprodukciju na robotu. Za reprodukciju je korišten algoritam *Dynamci Motion Primitives (DMP)*, dok je za kontrolu pozicije i sile bušenja korišten predloženi algoritam podatnog upravljanja u jednoj osi gibanja. S obzirom na visoke brzine posmaka koje se ostvaruju pri bušenju te ograničenja dostupnog upravljačkog sustava *KUKA KR C4*, prilikom bušenja pojavio se problem odstupanja vrha svrdla od željenog pravca bušenja za gotovo $1mm$. Kako bi se ograničilo neželjeno odstupanje, provedena je analiza niske razine upravljanja te je uvrđeno da različita dinamička ponašanja zglobova robota utječu na neželjeno odstupanje vrha svrdla. Sintezom novog upravljačkog sustava niske razine, odnosno upravljačkog sustava svakog pojedinog zgloba, postignuto je sumjerljivo dinamičko ponašanje te je neželjeno odstupanje vrha svrdla od pravca bušenja svedeno na razinu polovine promjera svrdla odnosno $\pm 0.3mm$. U eksperimentalnoj provjeri, predloženi robotski sustav je nadmašio rezultate operatera za sve bušene kalupe, odnosno za pojedine kalupe je postignut do čak 8 puta veći broj probušenih rupica. Također, snimljene demonstracije bušenja jednog kalupa uspješno su iskorištene za bušenje drugih kalupa sličnih svojstava.

Na kraju, prikazano je robotsko gletanje, fokusirajući se na domenu manipulacije deformabilnim objektima. S obzirom na iskustvo operatera, za gletanje je potrebno osigurati kontrolu kuta između špahtle i predmeta te kontaktnu silu. Imajući na umu deformabilnu prirodu špahtle za gletanje, posebna pažnja je stavljena na estimaciju oblika špahtle. Estimacija oblika špahtle je provedena pomoću vizualnih tragova postavljenih na unutarnji dio špahtle te kamere postavljene u smjeru vizualnih tragova. Kada je riječ o upravljanju robotskim gletanjem, razvijen je upravljački sustav za osiguravanje željenog kuta između špahtle i pripravka. S druge strane, za kontrolu sile dodira korišten je ranije prikazan algoritam podatnog upravljanja. U eksperimentalnoj provjeri prikazana je uspješnost predloženog sustava upravljanja.

U sedmom poglavlju dan je zaključak ovog doktorskog rada.

Znanstveni doprinos ovog doktorskog rada je moguće raščlaniti kako slijedi:

- *Sustav podatnoga upravljanja industrijskim manipulatorom zasnovan na izračunu unaprijedne dinamike u stvarnom vremenu za kontakt s mekim i krutim tijelom za industrijske primjene koje uključuju duboko mikro bušenje, gletanje i poliranje* (Poglavlja 3, 4, 6)
- *Sigurno kolaborativno sučelje čovjeka i industrijskog manipulatora koje omogućava programiranje demonstracijom* (Poglavlje 5)

Ključne riječi: Podatno i impedantno upravljanje, Upravljanje silom, Modeliranje kontakta, Kolaborativni roboti u proizvodnji, Robotika usmjerena na čovjeka, Učenje putem demonstracije

Contents

CHAPTER 1

Introduction

Robot manipulators are highly versatile machines designed to perform a wide range of tasks across various industries. When it comes to term manipulator, it refers to a mechanical device or machine designed to manipulate or move objects in a controlled manner. They typically consist of interconnected rigid links and joints, allowing them to move and manipulate objects in three-dimensional space with precision. They can be found in various forms, from robotic arms used in manufacturing and industrial settings, agriculture up to medical robots. Robot manipulators have a rich history dating back to the early 20th century, with significant advancements made over the years. Originally developed for tasks deemed too dangerous or tedious for humans, they have since become integral part of modern industry.

When discussing robot manipulators, the typical assumption is Serial Manipulators [1], which consist of a series of interconnected links and joints arranged in a chain-like form. These manipulators are commonly found in manufacturing settings. Additionally, there are Parallel Manipulators [2, 3], which feature multiple chains of links connected to a common base. In contrast, Selective Compliance Assembly Robot Arm (SCARA) Manipulators [4, 5], known for their parallel axes, are frequently utilized in assembly operations due to their speed and precision in planar movement. In terms of mobility, robot manipulators typically fall into two categories. Stationary manipulators are fixed in place and do not move relative to their surroundings. In contrast, mobile manipulators are often mounted on linear axes or mobile platforms, enabling them to navigate and operate within their environment.

Robot manipulators play a crucial role in various industries and applications. Primarily associated with industrial use, they are extensively employed in manufacturing processes such as welding, assembly, painting, and material handling. For instance, in the automotive industry, manipulators are integral to tasks ranging from the initial handling of metal sheets to shaping,

(a) Industrial: Fanuc R-2000 [1] (b) Collaborative: Universal Robots UR20 [2]

Figure 1.1: Robot manipulators

welding, painting, and beyond. However, their utility extends far beyond manufacturing. Manipulators are also prevalent in sectors such as agriculture, logistics, and hospitality. Moreover, advancements in technology have facilitated the integration of robots into medical applications. They are used in different medical procedures such as radiotherapy, telesurgery [6] or robotic-assisted surgical procedures, showcasing the versatility and adaptability of robot manipulators across diverse domains.

Another classification of robot manipulators, which is important for this thesis is Industrial vs Collaborative manipulators, as shown in Fig. 1.1. Industrial manipulators have become the workhorse of modern industry and advanced manufacturing plants worldwide. Typical industrial applications for robots include repetitive or hazardous tasks that require considerable precision. These robots are stiff joint-position controlled, which enables precision and repeatability, but also impressive payload and reach. Considering lack of sensing capabilities, but also achievable speed and power, these machines are always caged under safety fences, strictly restricting human operators to enter robot's workspace during operation. It is important to note that industrial manipulators are typically pre-programmed, with very limited flexibility to adjust motions in real-time, which requires them to have strictly structured environment. Any changes in robot motions or environment requires engineers to reprogram the robot. Considering all of these, together with costs of purchasing, deploying and maintaining, the use of industrial

---

[1] https://www.fanucamerica.com/products/robots/series

[2] https://www.universal-robots.com/products/ur20-robot/

manipulators are only used in large-scale productions.

In a rapidly evolving industrial environment the main challenges engineers usually face are the flexibility, the reliability and the safety of robotic solutions. These challenges have opened up a new research field in robotics, Collaborative manipulators. Their main property stems from fact they are soft joint-torque controlled. This enables the robot to sens the contact with the environment with it's whole kinematic chain. The sensing capability enables human-robot interaction (HRI), where human and robot can safely work together. One of main HRI possibilities is utilized through Programming-by-Demonstration, where human operator can easily demonstrate the task to the robot. All of these plays crucial role in system flexibility, opening up possibility to utilize manipulation tasks even in small-scale industries. However, it soon became apparent that collaborative robots lack the necessary accuracy and power as well as economic justification for a truly disruptive impact on modern factories.

1.1 Contribution

The contribution of this thesis can be decoupled in:

- *A compliant control system for an industrial manipulator based on real time forward dynamics computation for both soft and rigid body contact for industrial applications involving deep micro drilling, plastering and polishing* (Chapters 3, 4, and 6)
- *Safe collaborative human-robot interface for industrial manipulators which enables programming by demonstration* (Chapter 5)

1.2 Thesis Outline

This thesis is organized in seven chapters, as follows:

Chapter 2 : The chapter outlines the problem addressed within this thesis, providing an overview of related work and detailing how the contribution proposed by this thesis expand the scientific body of knowledge.

Chapter 3 : As a first component of proposed complaint framework, this chapter introduces compliant control for industrial manipulators. It commences by outlining fundamental compliance control algorithms based on Impedance, Admittance, and Force control. Subsequently, it presents a compliant control algorithm that integrates these three fundamental approaches and maps Cartesian motion and compliance into the joint state using both Jacobian-based and forward dynamics simulation methods. Additionally, it discusses a specialized version of the controller tailored for hard position-constrained Cartesian motion.

Chapter 4 : This chapter focuses on the second component of the proposed compliant framework, emphasizing close human-robot interaction, particularly in the Programming by Demonstration (PbD) paradigm. Initially, kinesthetic teaching was explored using an industrial manipulator, followed by the introduction of programming-by-demonstration based on a tool tracking system. The ergonomic aspects and user-friendliness of both approaches are examined through a user-experience survey.

Chapter 5 : In its final component, this chapter delves into the safety aspect of the proposed collaborative framework. It examines safety according to the technical specification ISO/TS 15066, which addresses safety concerns related to collaborative manipulators. The chapter demonstrates that the proposed framework aligns with all collaborative operation requirements outlined in the technical specification.

Chapter 6 : Given the primary focus of this thesis on industrial applications of the proposed collaborative framework, this chapter presents three distinct scenarios. It begins with the industrial scenario of delicate grinding of composite materials used in aircraft manufacturing. It then transitions to deep-micro-hole drilling in the glass container mould industry. Finally, it explores robotic putty plastering, focusing on the domain of deformable object manipulation.

Chapter 7 : At the conclusion, the thesis provides a summary of the presented work.

CHAPTER 2

Problem Description and Related Work

In the introduction, an overview of robot manipulators was provided, highlighting two main types of manipulators: standard industrial and collaborative. Each type has its distinct advantages. Industrial manipulators are prevalent in various industries due to their precision, repeatability, high payload, and extensive reach. However, they typically require significant expertise in programming, with limited opportunities for adjustments without the involvement of robotic engineers. As a result, they are primarily suitable for large-scale productions. However, as industries evolve, there is a growing demand for robots to be used in smaller-scale productions, requiring greater flexibility. This need is fulfilled by collaborative robots [7]. In the realm of collaborative robotics, it typically refers to manipulators that are torque-controlled and capable of sensing contact with the environment through residual joint torques. Standard use cases for this type of manipulators are simple pick-and-place tasks as a result of Programming by Demonstration [8, 9, 10], where a high degree of collaboration is required. However, off-the-shelf collaborative manipulators often lack the payload, reach, precision, and repeatability typically required in industrial applications, creating a gap between the two types of manipulators. For this reason, industrial manipulators can also be used collaboratively when integrated with modern computer vision and advanced force control algorithms.

The first challenge tackled in this thesis involves enabling compliant control of a robot that lacks sensing capabilities. This begins with equipping the industrial manipulator with appropriate force/torque sensing capabilities. Subsequently, the focus shifts to developing a compliant control algorithm to facilitate compliant behaviour of the industrial manipulator. Notably, stability analysis becomes crucial given the wide range of stiffnesses present in different environments. The second challenge addressed by this thesis is the implementation of Programming-by-Demonstration. This is achieved through human-robot interaction and the introduction of a novel concept for tracking demonstrators' motion. Moreover, given the close collaboration between humans and robots, special attention is dedicated to ensuring human safety. This encompasses collision detection, along with an assessment of the proposed collaborative system in accordance with the available ISO technical specifications for collaborative robotics.

Compliant Robot Control

When discussing compliance, Vanderborght et al. [11] distinguishes between two types: passive and active compliance. Passive compliance involves elements that achieve compliance through mechanical means, including both fixed stiffness mechanical components and systems capable of adjusting their compliance through controlled mechanical reconfiguration. On the other hand, active compliance [12] is driven by software control. Schutter et al. [13] defines compliant robot control as the modification or even re-planning of robot motion based on real-time sensor feedback. This approach enables robots to sense and compensate for direct contact with the environment, rather than being solely stiff and position-controlled. The relationship between robot motion and environmental contact force is typically modeled using impedance-admittance relations [14, 15]. Various control approaches are employed based on the type of environment encountered. These include advanced impedance-admittance controllers [16], hybrid position-force control, hybrid impedance control [17], and position-based and direct force control [18, 19]. Another example of advanced compliant control is Forward Dynamics Compliance Control (FDCC), introduced by Scherzinger et al. [20], which serves as inspiration for the compliant control algorithm derived in this thesis. The algorithm integrates three main control concepts: Impedance, Admittance, and Force Control, ensuring compliant behaviour for both open and closed loop motions. This thesis embraces this approach and tailors it to suit Cartesian motion while ensuring stable interaction with environments across a wide spectrum of stiffness levels, spanning from soft and deformable to completely rigid ones.

While traditional robotic tasks predominantly involve manipulating rigid objects with rigid end effectors, the research field of deformable object manipulation is raising interest thanks to the advancements in sensing and computational capabilities. Robotic plastering, which employs a flexible robotic tool, illustrates one such application. Other examples include manipulating textiles and clothing, manufacturing flexible parts and cables [21], agricultural robotics [22], medical applications [23], and even artistic endeavors [24]. Unlike rigid bodies where the control objective typically revolves around positioning the manipulated objects, manipulating deformable objects or tools presents a more nuanced challenge due to the influence of physical interactions on their shape. A comprehensive review of deformation sensing, modeling, and control across various applications is presented in [21]. Historically, much of the existing work in this area relied on analytical model fitting based on visual and force feedback until recent advancements in deep learning. In [25], a deep learning-based control feedback for manipulating a deformable object in an industrial setting was introduced. Similar to other research in this field, this work selects a set of deformation features to enable the control system to account for tool deformation. Whereas traditional approaches typically model position and shape information independently in the deformation features [23], in this thesis, these aspects are coupled to represent the combined effect of position control and force application on the flexible robot tool.

Programming-by-Demonstration (PbD)

Over the past three decades, Programming by Demonstration (PbD) has gained substantial traction in both academic research and industrial applications. As described in [26], PbD typically encompasses two primary phases: learning and representation. In the learning phase, demonstrations are collected and segmented into action representations, forming an assembly tree that outlines the sequence of actions. Various methods, such as kinesthetic teaching [27] or teleoperation [28], can be employed during this phase. Subsequently, the representation phase involves mapping robot movements and executing them. A range of representation approaches are available, including probabilistic models [8], data-driven AI-based models [29], and the increasingly prevalent Dynamic Movement Primitives (DMP) [30]. This thesis contributes to the PbD domain by presenting and systematically comparing various demonstration techniques to identify the most effective approach.

When it comes to interesting PbD applications, Wang et al. [31] have introduced a novel approach for insertion tasks, where the robot can accurately replicate precise insertion maneuvers by passively observing a human perform the task once. Visual servoing is employed to enable tracking of the human hand. This thesis, instead of visual servoing, utilizes and evaluates a motion capture system to track a virtual marker, facilitating the recording and subsequent processing of the recorded motion. This introduces flexibility and allows for collecting demonstrations from different operators. In another study by Ajaykumar et al. [32], the proposed PbD system incorporates various modalities often used by humans when naturally communicating physical tasks or missions, such as gaze and speech. Raising caution, the authors report that employing multimodal PbD may lead to overtrust and automation bias in the long run. However, it is worthwhile to explore Human-Robot Interaction (HRI) through PbD using different modalities of kinesthetic teaching and human demonstration, while also integrating other modalities synchronized with virtual marker motion, such as force or human pose measurements. Additionally, in the work by Steinmetz et al. [33], a task-level PbD approach is proposed, which could prove beneficial for the integration of collaborative robots in small and medium-sized enterprises (SMEs). A key distinction from the proposed approach is that they utilize kinesthetic teaching coupled with an online semantic skill recognition algorithm, enabling untrained operators to demonstrate certain tasks, whereas the aim of this thesis is to capture the expertise of a skilled task performer.

Safety issue

Modern industrial manipulators are large and powerful machines that cover a large workspace, and special attention must be paid to human safety. Suppose one uses a standard pre-programming control approach where the robot system is not able to understand its environment and adapt its own behaviour. In that case, the robot must be limited by a safety fence that allows the robot to move only when the human operator is at a safe distance from the robot. This approach significantly reduces production efficiency, especially in industries where some operations are still performed manually by a human operator. Addressing this problem shifted the research field in the last decade towards collaborative robotic systems, where the robot can safely share a workspace with human operators.

In collaborative robotic manufacturing environments, ensuring safety is paramount. Unlike traditional industrial settings where humans are confined to specific safe zones, close collaboration with robots requires humans to enter the robot's workspace. Consequently, it becomes impractical to restrict humans to predefined safe regions. Instead, collision detection algorithms [34], [35] have been developed to detect and prevent collisions between robots and human operators or the surrounding environment. This thesis introduces a fast online collision detection algorithm, leveraging the GPU-Voxels library [36, 37] and RGB-D camera sensor data. The proposed algorithms and methodologies adhere to the requirements outlined in the technical specification ISO/TS 15066 [38, 39] for collaborative robots.

CHAPTER 3

Compliant Control of Industrial manipulator

The paradigm of Compliant Robot Control is defined as a control strategy through which one can achieve compliant behaviour of the robot. In [13], Compliant Control refers to robot motion wherein the end-effector trajectory is dynamically modified or generated based on real-time sensor information. This implies that robots are not solely rigid and position-controlled. Instead, they are controlled in a manner that enables them to perceive and adjust for direct contact with the environment. Presently, two main types of robot manipulators are prevalent in the market. Firstly, traditional Industrial Manipulators that are widely used in the industry. These manipulators are characterized by stiff joint-position control, providing substantial reach and payload capacity, as well as exceptional precision and repeatability. However, despite being a staple of modern industry, particularly in large-scale production, these manipulators lack sensing capabilities, making them unable to perceive contact and interact safely with their environment. On the other hand, collaborative manipulators are gaining popularity especially in small-scale manufacturing due to their ability to sense contact and interact safely with the environment. It is noteworthy that off-the-shelf collaborative manipulators still lag behind their industrial counterparts in terms of payload, reach, precision, and repeatability.

Given these considerations, this chapter focuses on enhancing industrial manipulators by equipping them with sensing capabilities and developing control algorithms to introduce compliance. The goal is to bridge the gap between the precision and payload capabilities of traditional industrial manipulators and the safety and adaptability offered by collaborative manipulators in collaborative environments.

3.1 Fundamentals of Compliant Control

Industrial manipulators have been integral machines in various industries for the past few decades. Typically employed in large-scale production lines, they specialize in a limited set of operations. These manipulators can also be found in dedicated working cells where they undertake hazardous or repetitive tasks. Additionally, they serve other machines such as CNC milling machines. These machines vary significantly in terms of size, reach, and payload, spanning from a range of $1m$ with a $5kg$ payload to $3-4m$ with payloads up to $300kg$.

Despite their diversity in size and capabilities, a common characteristic of all industrial manipulators is their control method. They are stiff joint-position controlled. While this design ensures exceptional precision and repeatability, it restricts their capacity to sense interactions with the environment. To enable safe interaction with the environment, where contact force must be considered, robots need to be equipped with force/torque sensors and controlled in a manner that facilitates secure contact, essentially making the robot compliant. This section will delve into the challenges and solutions associated with achieving compliant control on industrial manipulators.

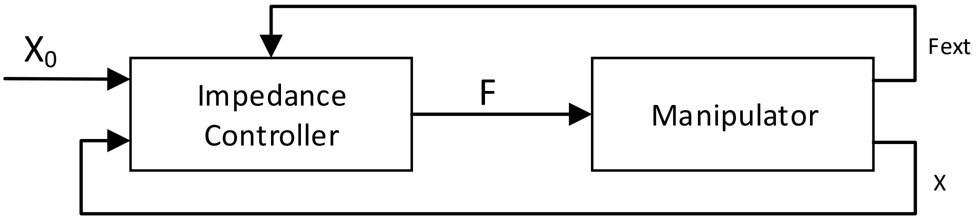
Impedance and Admittance Control

The most common fundamental Compliance Control algorithm is Impedance Control. It was first introduced by Hogan [14]. The main idea of the control concept is that physical interaction between the manipulator and the environment must complement each other. An origin of such an approach is in the investigation of a human arm interacting with the environment, [40]. For different types of environment, the arm muscles have different impedance behaviour. Such a control strategy can be divided into two categories, Passive and Active Impedance Control. The Passive Impedance Control includes the end-effector equipped with a mechanical device composed of springs, that produce fixed impedance behaviour with the environment. On the other hand Active Impedance Control assumes control of the robot motion in such a behaviour that produces a virtual spring system. This way, the robot is not only designed to track the required motion, but also to change it's impedance in order to handle the contact with different types of the environment. Standard implementation of impedance control is given by the following filter equation:

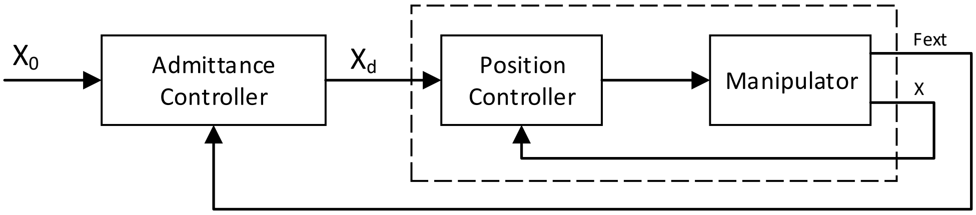
$$F(t) = m \cdot [\ddot{x}(t) - \ddot{x}_d(t)] + d \cdot [\dot{x}(t) - \dot{x}_d(t)] + c \cdot [x(t) - x_d(t)], \tag{3.1}$$

where $F(t)$ is force output, $x(t)$ system position and $x_d(t)$ system desired position.

Although the same filter is applied on the motion of the robot in both cases, in impedance control shown in Fig. 3.1a, controller behaves like mechanical impedance, while the environment is admittance. In this case the process is force controlled, meaning that the input of the



(a) Impedance Control assumes control of the robot motion in such a behaviour that produces a virtual spring system. This way, the robot is not only designed to track the required motion, but also to change its impedance in order to handle the contact with different types of the environment



(b) In Admittance Control, the robot is position-controlled and behaves like a mechanical impedance, while the controller is designed to be admittance. The inner position-control loop is usually of PID form.

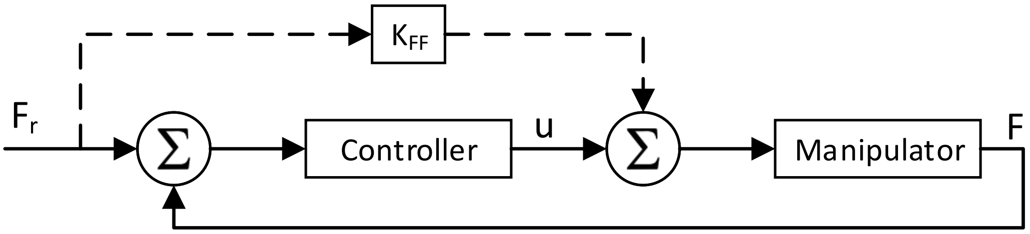
Figure 3.1: Impedance-Admittance control schematics

impedance controller is the motion, and the output is the controlled force. On the other hand, in admittance control shown in Fig. 3.1b, the process is position-controlled, the environment is considered as impedance, while the controller is designed to behave like a mechanical admittance. The choice whether to use impedance or admittance control usually depends on the stiffness range of the environment. As presented in [16], impedance provides better performance on the stiff environment, and admittance provides better performance in softer environment. To summarize, the performance of the impedance and the admittance controller in terms of accuracy and robustness is complementary [41].

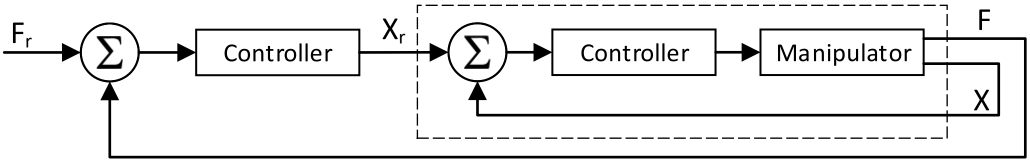
Force Control

Another aspect of compliant motion is Force Control. It can be classified in two categories, Implicit and Explicit Force Control [42]. Implicit force control is position-based controller, without force feedback, where the gain of the controller K_p predefines the scaling of the relationship between the position and force that determines the stiffness of the manipulator. The more common form of force control is the explicit form, implemented as force-based and position-based, respectively [18]. The force-based force control shown in Fig. 3.2a is given as a control system that compares the reference and the measured value of the force to provide a control signal directly to the process. The controller is usually in a standard linear PID form, with or without feed-forward signal. Different forms and synthesis methods for PID controllers with experimental results are given in [18], [19].

On the other hand, position-based force control, shown in Fig. 3.2b, includes an inner



(a) Force Control compares reference and measured value of the force and provides a control signal directly to the process. The controller is usually in standard linear PID form, with or without feedforward signal.



(b) Position-based Force Control includes an inner position control loop, while the outer loop provides reference to the inner loop through the admittance filter.

Figure 3.2: Force-Control versions

position control loop, while the outer loop provides a reference to the inner loop through an admittance filter. The reason to use a position-based approach is a practical one because most of the commercial manipulators provide joint position interfaces and do not allow direct control of actuator torques. Although the control is quite similar to impedance control, one can only control the contact force, not the position.

3.2 Cartesian Compliance Control

In addition to the challenges mentioned earlier, achieving compliant behavior in robots is often intricately linked to the Cartesian motions of the robot's tool. Consider the scenario of robot grinding, where the robot must adhere to a grinding trajectory in Cartesian space while concurrently controlling the contact force. Similar example includes the scenario of robotic drilling, where the robot's motion needs to be constrained along the drilling axis. In both scenarios, apart from the Cartesian motion, another imperative is to exhibit compliant behaviour. Given that simultaneous tracking of motion and explicit force extends beyond the fundamental algorithms, this section presents an advanced compliance control algorithm addressing the complexity associated with mapping Cartesian motion into joint motion, which stems from the inherent non-linearity and redundancy in the relationship between Cartesian and joint coordinates, leading to various singularity problems. Additionally, this chapter addresses the issue of environmental stiffness and its impact on the stability of the compliant system.

3.2.1 Controller base

Drawing the inspiration from [20], this work employs the compliance control architecture illustrated in Fig. 3.4. The control schematic combines fundamental control concepts, ensuring Cartesian compliant behaviour for both open and closed-loop motions. Given that the proposed controller operates in Cartesian space, special attention is required to transform Cartesian motion into joint motion $\mathbf{q} = f(\mathbf{X})$. Subsequent subsections will detail various approaches to address this transformation challenge. The components of the compliance controller are:

1. **Impedance control**, which is slightly different compared to a standard implementation. In this case, it represents the compliance behaviour of each individual component, i.e. spring, damper, and mass. The difference stems from the fact that the resulting motion from the forward dynamics simulation is carried out without feedback from the physical robot. The resulting Impedance force is used for open-loop, trajectory following motion.
2. **Admittance control** which combines virtual impedance control with sensor measurements from a physical robot. Whenever the robot is in contact with the environment the control loop is closed.
3. **Force control**, which is very similar to the admittance control component, but includes the desired contact force. In this part, it is important to substitute the part of the spring force co-linear with the desired force [43], [20]. Consider an example where the manipulator needs to apply exactly 40N in its approach axis to the treated object, with a localization error of $5mm$. When the stiffness of the impedance control is set to $1000\frac{N}{m}$ this results in $5N$ force of impedance control. The final value of the applied force would be $35N$ or $45N$, depending on the error direction.

6-DOF Controller base

The controller depicted in Fig. 3.3 is designed for 6-DOF cartesian control, indicating that each degree of freedom has its controller. The force contributions from the Impedance filter $\mathbf{F}_{imp} \in \mathbb{R}^{6\times1}$, force/torque sensor feedback $\mathbf{F}_{sens} \in \mathbb{R}^{6\times1}$, and the desired force wrench $\mathbf{F}_d \in \mathbb{R}^{6\times1}$ are combined into the net force $\mathbf{F}_{net} \in \mathbb{R}^{6\times1}$. This net force is controlled to zero using a PD controller. The control output of the PD controller $\mathbf{F}_{ctrl} \in \mathbb{R}^{6\times1}$ is then applied to a virtual mass, resulting in its virtual motion. The simulated motion is subsequently commanded to the robot as the joint command $\mathbf{q}$ while simultaneously calculating the Cartesian acceleration $\ddot{\mathbf{X}} \in \mathbb{R}^{6\times1}$, velocity $\dot{\mathbf{X}} \in \mathbb{R}^{6\times1}$, and position $\mathbf{X} \in \mathbb{R}^{6\times1}$ of the end-effector using Forward Kinematics (FK). The position state vector $\mathbf{X} = \begin{bmatrix}\mathbf{X}_p & \mathbf{X}_o\end{bmatrix}^T$ is comprised of Cartesian position $\mathbf{X}_p \in \mathbb{R}^{3\times1}$ and orientation $\mathbf{X}_o \in \mathbb{R}^{3\times1}$ expressed in Euler angle notation. The impedance filter is defined as:

$$\mathbf{F}_{imp} = \mathbf{K}_r \cdot (\mathbf{X}_d - \mathbf{X}) - \mathbf{b}_r \cdot \dot{\mathbf{X}} - \mathbf{I}_r \cdot \ddot{\mathbf{X}}, \tag{3.2}$$

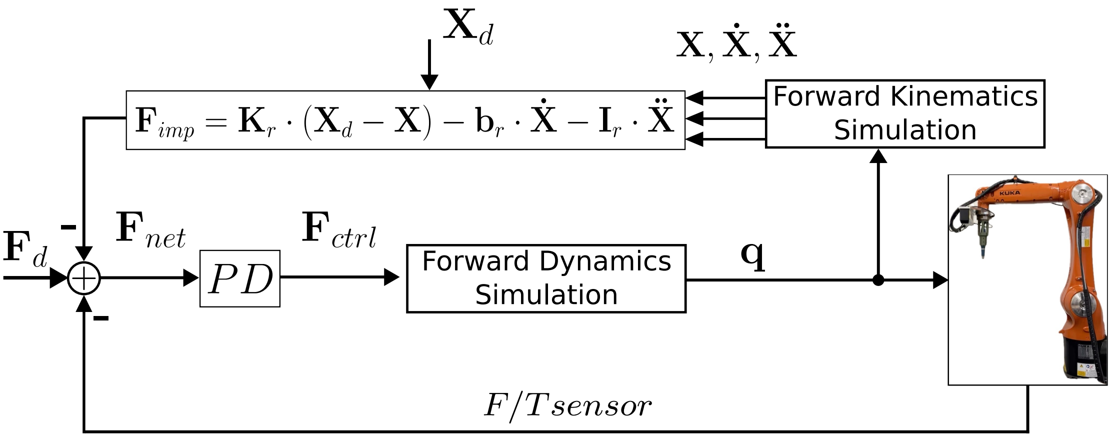


Figure 3.3: Forward Dynamics Compliance Control schematics.

where $\mathbf{K}_r \in \mathbb{R}^{6\times6}$ is 6-DOF stiffness, $\mathbf{b}_r \in \mathbb{R}^{6\times6}$ is 6-DOF damping and $\mathbf{I}_r \in \mathbb{R}^{6\times6}$ is 6-DOF inertia of virtual impedance filter. The stiffness $\mathbf{K}_r$ is defined as:

$$\mathbf{K}_p = \begin{bmatrix} \mathbf{K}_{r,p} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{r,o} \end{bmatrix}, \mathbf{K}_{r,p} = \begin{bmatrix} K_{r,px} & 0 & 0 \\ 0 & K_{r,py} & 0 \\ 0 & 0 & K_{r,pz} \end{bmatrix}, \mathbf{K}_{r,o} = \begin{bmatrix} K_{r,ox} & 0 & 0 \\ 0 & K_{r,oy} & 0 \\ 0 & 0 & K_{r,oz} \end{bmatrix}, \tag{3.3}$$

the damping $\mathbf{b}_r$ is:

$$\mathbf{b}_r = \begin{bmatrix} \mathbf{b}_{r,p} & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_{r,o} \end{bmatrix}, \mathbf{b}_{r,p} = \begin{bmatrix} b_{r,px} & 0 & 0 \\ 0 & b_{r,py} & 0 \\ 0 & 0 & b_{r,pz} \end{bmatrix}, \mathbf{b}_{r,o} = \begin{bmatrix} b_{r,ox} & 0 & 0 \\ 0 & b_{r,oy} & 0 \\ 0 & 0 & b_{r,oz} \end{bmatrix}, \tag{3.4}$$

while the inertia matrix is defined as:

$$\mathbf{I}_r = \begin{bmatrix} \mathbf{m} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathbf{m} = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & m_z \end{bmatrix}, \mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}. \tag{3.5}$$

3.2.2 Virtual Robot Motion Dynamics

When it comes to the motion of the robot, the most fundamental approach in determining the required forces and torques of the robot actuators to exert required force and compliance to the environment begins by deriving the dynamic equations of the manipulator either using *Newton-Euler* or *Lagrange* method. As defined in [44] the motion for an manipulator with n serially linked joints can be defined as:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{G}(\mathbf{q})=\tau+\mathbf{J}(\mathbf{q})^T\mathbf{F}_{ext}, \tag{3.6}$$

where $\mathbf{q}\in\mathbb{R}^{n\times 1}$ represents the joint-state vector, its first derivative $\dot{\mathbf{q}}\in\mathbb{R}^{n\times 1}$ corresponds to joint velocity, and the second derivative $\ddot{\mathbf{q}}\in\mathbb{R}^{n\times 1}$ denotes joint acceleration. The manipulator inertia matrix is denoted by $\mathbf{D}(\mathbf{q})\in\mathbb{R}^{n\times n}$. The term $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ represents the velocity coupling vector, encompassing centrifugal and Coriolis terms. Simultaneously, $\mathbf{G}(\mathbf{q})$ signifies the gravitational force vector. On the other hand, the vector of torques across the joints is denoted by $\tau\in\mathbb{R}^{n\times 1}$, the Jacobian matrix in joint position $\mathbf{q}$ is $\mathbf{J}(\mathbf{q})\in\mathbf{R}^{6\times n}$, and $\mathbf{F}_{ext}\in\mathbf{R}^{6\times 1}$ represents the vector of external forces acting on the robot's end-effector.

The presented approach plays a crucial role in determining and controlling joint torques, particularly in the context of collaborative robots. However, the primary focus of this thesis lies in developing a compliance control system for an industrial manipulator with stiff-joint position control. In this scenario, the robot's joints can be observed as black boxes employing a low-level closed-loop control system that enables fast and accurate joint position reference tracking. With this in mind, the gravitational term from Eq. 3.6 can be neglected and treated as a disturbance of the low-level closed-loop control system. In addition to that if one assumes a fully underactuated (floating) system, and control force wrench $\mathbf{F}_{ctrl}$ acting on the end-effector, Eq. 3.6 transforms into:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}=\mathbf{J}^T\mathbf{F}_{ctrl}.. \tag{3.7}$$

With all this in mind, particularly with the fast low-level control capable of compensating for disturbances and providing accurate joint position reference tracking, the dynamics of the robot's end-effector can be simulated by concentrating the entire robot dynamics at its end effector. This entails assigning negligible mass to all links in the kinematic chain, from the base to the flange, except for the last link, the end effector, where the entire mass is concentrated. Through this approach, the mass and inertia of the end effector can be flexibly chosen to align with a particular use case. Furthermore, this methodology facilitates a manipulator-independent solution, allowing easy adaptation to various manipulators. This ensures consistent compliant Cartesian behaviour, requiring only adjustments to the robot's kinematics. With all this in mind, the Forward Dynamics (FD) calculation can be defined as a function of the kinematics model, joint state, and control force $\mathbf{F}_{ctrl}$:

$$\ddot{\mathbf{q}}=f_{FD}(model,\mathbf{q},\dot{\mathbf{q}},\mathbf{F}_{ctrl}). \tag{3.8}$$

In terms of implementation, simulating the forward dynamics of the kinematic chain, along with its mass and inertial properties, is achieved using available solvers such as the Rigid Body

Dynamics Library (RBDL) by Felis [45], the Kinematics and Dynamics Library (KDL) [46] [1], or others. The forward dynamics approach boasts a significant advantage in effectively avoiding the singularity issue. However, particular attention must be dedicated to poses near a null-space, primarily due to the omission of the mass of the kinematic chain, excluding the end-effector. This concern arises in situations where the slight inertia of the links impacts the swift movement of the joint in the null-space.

Jacobian approach

A simplified version of resolving the forward dynamics problem defined in Eq. 3.8 is shown in Fig. 3.4. Here the dynamics, in one axis, is simulated with $\ddot{X} = \frac{1}{M}F_{ctrl}$, while the velocity $\dot{X}$ and position X are achieved by integrating the $\ddot{X}$ and $\dot{X}$ respectively. Still the issue of mapping the Cartesian motion $\mathbf{X}$ into the joint motion $\mathbf{q}$ is present. Here, the mapping function using the Jacobian matrix is presented.

First, let us define the Jacobian matrix as the relation between infinitesimally small change of the joint angles to infinitesimally small linear and angular motion of the tool. Observed in short time period dt, one can write:

$$\frac{d\mathbf{X}}{dt} = \begin{bmatrix} \frac{d\mathbf{X}_p}{dt} \\ \frac{d\mathbf{X}_o}{dt} \end{bmatrix} = \mathbf{J}(\mathbf{q}) \cdot \frac{d\mathbf{q}}{dt}, \tag{3.9}$$

where $\mathbf{X} \in \mathbb{R}^{6\times 1}$ is Cartesian pose of the end-effector, $\mathbf{q} \in \mathbb{R}^{n\times 1}$ joint-state vector and $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6\times n}$ Jacobian matrix. Using the aforementioned equation, the discrete form of joint vector $\mathbf{q}(k) \in \mathbb{R}^{n\times 1}$ calculation is derived as:

$$\mathbf{q}(k) = \mathbf{q}(k-1) + \mathbf{J}(\mathbf{q}(k-1))^{-1} \cdot \dot{\mathbf{X}}(k) \cdot \Delta T, \tag{3.10}$$

where $\mathbf{q}(k) \in \mathbb{R}^{n\times 1}$ is calculated joint state vector in step k, while $\mathbf{q}(k-1) \in \mathbb{R}^{n\times 1}$ corresponds to previous step. Vector $\dot{\mathbf{X}}(k)$ presents Cartesian velocity, obtained through control schematics shown in Fig. 3.4. When employing the Jacobian-based approach, special attention must be given to potential Jacobian singularities, primarily arising from the singularity of the Jacobian inverse.

3.2.3 Cartesian motion

When it comes to ensuring precise tracking of the desired motion in Cartesian space, the impedance filter $\mathbf{F}_{imp}$ defined in Eq. 3.2 plays a crucial role. The tracking of the desired motion is executed through the spring component of the Impedance filter, generating a motion wrench

---

[1] `https://www.orocos.org/kdl.html`

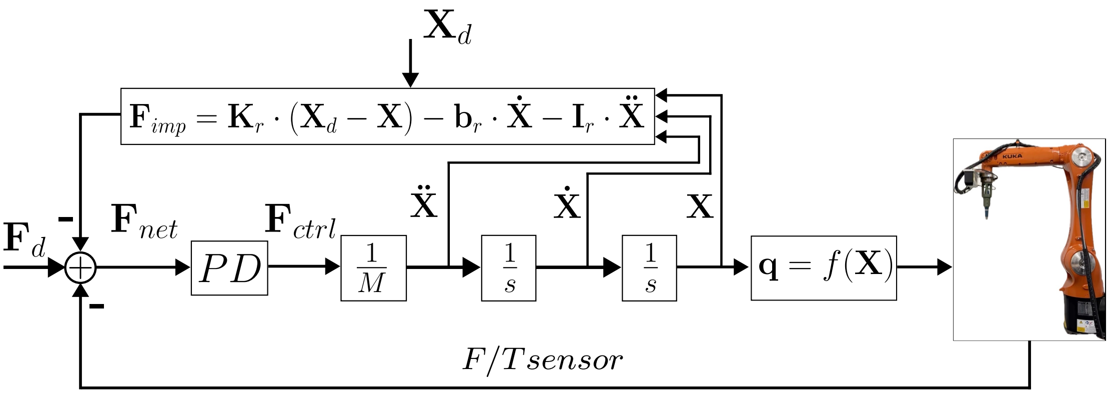


Figure 3.4: Advanced Compliance Control system schematics. The system enables compliance control of the robot both for Cartesian trajectory execution and explicit Force Control.

$\mathbf{F}_{sp}$ acting on the end-effector. This is achieved based on the spring behaviour modeled as:

$$\mathbf{F}_{sp} = \mathbf{K}_r(\mathbf{X}_d - \mathbf{X}) = \mathbf{K}_r \cdot \Delta\mathbf{X}, \tag{3.11}$$

where $\Delta\mathbf{X} \in \mathbb{R}^{6\times 1}$ is motion error, defined as:

$$\Delta\mathbf{X} = \begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{r} \end{bmatrix}, \tag{3.12}$$

with $\Delta\mathbf{p} \in \mathbb{R}^{3\times 1}$ being positional component and $\Delta\mathbf{r} \in \mathbb{R}^{3\times 1}$ being orientation component of motion error. Representation of the positional component is trivial, while the rotational component can be expressed as $\Delta\mathbf{r} = \hat{\omega} \cdot \Theta$, where $\hat{\omega} \in \mathbb{R}^{3\times 1}$ corresponds to unit rotation vector and value of angular error $\Theta \in \mathbb{R}$. In other words, it represents the vector of rotation and angle of rotation that will move the robot from the current towards the desired pose. To calculate $\hat{\omega}$ and Θ, firstly the motion error in a form of transformation matrix is calculated as follows:

$$\Delta\mathbf{T} = \mathbf{T}_e^{-1} \cdot \mathbf{T}_t = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.13}$$

where $\mathbf{T}_e$ represents transformation matrix of current global pose of the robot, $\mathbf{T}_t$ corresponds to target pose transformation matrix. In addition, $\Delta\mathbf{T}$ can be decoupled as $\Delta\mathbf{R} \in \mathbb{R}^{3\times 3}$ rotation and $\Delta\mathbf{p} \in \mathbb{R}^{3\times 1}$ translation component of pose correction towards the desired pose. As the rotation $\Delta\mathbf{R} \in \mathbb{R}^{3\times 3}$ represents an SO(3) rotation, the angular error value and rotation vector are computed following the equation derived in [44]:

$$\cos\Theta = \frac{1}{2}(\mathrm{Tr}(\Delta\mathbf{R}) - 1), \tag{3.14}$$

$$\hat{\omega} = \frac{1}{2sin(\Theta)}(\Delta\mathbf{R} - \Delta\mathbf{R}^T). \tag{3.15}$$

This calculation becomes invalid when $\Delta\mathbf{R}$ is a symmetric matrix. This can happen if the tool is already in the target orientation, resulting in $\Delta\mathbf{R} \approx \mathbb{I}$, or if it's rotated by 180°. While the second scenario is unlikely due to small angular motion, special attention must be paid to handling singularities when the tool orientation closely approaches the target orientation. In such cases, the rotation angle becomes zero, and the rotation axis is chosen to be the z axis. An effective approach for handling such calculations can be found in the "Kinematics and Dynamics Library (KDL)" [46].

3.2.4 Position-constrained Cartesian motion

In previous subsections, the algorithm for Cartesian compliance based on forward dynamics simulation was introduced, which ensures compliant motion across all 6 degrees of freedom (DOF). By varying the stiffness across different degrees of freedom, the robot's motion can be selectively constrained. An example of such a scenario is robotic grinding, where the robot must precisely follow a lawnmower-like grinding path on an object's surface while maintaining the soft behaviour in the contact axis. Conversely, there are applications where motion in certain axes needs to be tightly constrained. An example of such an application is drilling, where the drill tool must solely move along its axial approach axis. Any non-axial motions could potentially result in damage to the drill bit.

To develop a solution for drilling, a single-degree-of-freedom (DOF) Drill Controller was derived, inspired by control systems shown in Fig. 3.3 and the Jacobian-based forward kinematics approach shown in Fig. 3.4. This controller constrains the tool motion along the approach axis $\mathbf{z}_{tool} \in \mathbb{R}^3$, where the control value d denotes the offset from the initial tool position $d_{ref} \in \mathbb{R}$. The control system is illustrated in Fig. 3.5, with d_{ref} and d representing the reference (drilling depth) and control value, respectively. Employing the impedance filter and sensor feedback, the system promptly responds to external forces F_{sens} and maintains the desired motion d_{ref}. Subsequently, after computing the PD control value, forward dynamics motion is simulated using the virtual mass coefficient M and translated into the robot joint space through motion error calculation and the Jacobian matrix.

While the impedance controller is a single DOF equation, the computed control value d has to initiate the movement of the 6-DOF robot manipulator. The new global position of the tool $\mathbf{T}_t$ is calculated using Eq. 3.16, where $\mathbf{T}_{e_0}$ is the global initial position of the tool:

$$\mathbf{T}_t = \begin{bmatrix} \mathbf{R}_t & \mathbf{p}_t \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{T}_{e_0} \begin{bmatrix} \mathbf{1} & d \cdot \hat{\mathbf{z}}_0 \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3.16}$$

After calculating the new target global position of the tool $\mathbf{T}_t$ and observing the current

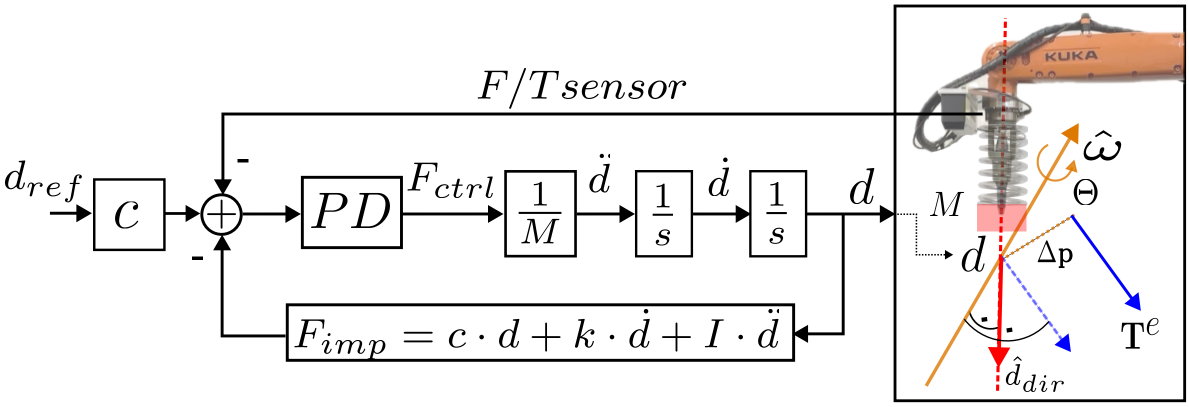


Figure 3.5: Single DOF drill controller designed to achieve compliant behaviour constrained to the drilling axis.

tool position $\mathbf{T}_e$, the Cartesian motion correction $\Delta\mathbf{T}$ of the tool towards the new position is calculated using Eq. 3.13. Subsequently, the motion error $\Delta\mathbf{X}$ towards the desired pose is calculated using Eqs. 3.12 and 3.13, which include the unit rotation vector $\hat{\omega}$ from Eq. 3.14 and the rotation angle Θ from Eq. 3.15. After computing the motion error $\Delta\mathbf{X}$, the new joint position reference is determined as follows:

$$\mathbf{q}_{ref} = \mathbf{f}(\mathbf{q}_e, d) = \mathbf{q}_e + \mathbf{J}(\mathbf{q}_e)^{-1} \cdot \Delta\mathbf{E}(\mathbf{q}_e, d), \tag{3.17}$$

where $\mathbf{q}_{ref} \in \mathbb{R}^{6\times1}$ and $\mathbf{q}_e \in \mathbb{R}^{6\times1}$ represent the new joint reference and actual joint position, $\mathbf{J}(\mathbf{q}_e) \in \mathbb{R}^{6\times6}$ stands for the Jacobian matrix in the current joint configuration, and $\Delta\mathbf{E}(\mathbf{q}_e, d) \in \mathbb{R}^{6\times1}$ signifies the calculated motion error as a function of the current joint configuration and the computed drilling depth reference.

3.3 Stability Analysis

The previous two sections provided an overview of the fundamentals of compliance control and introduced the Cartesian compliance controller for industrial manipulators. This section shifts focus to the issue of control stability of the proposed controller when in contact with different environments. It begins by defining the mechanics of physical interaction between the robot and its environment. Building upon the fundamental principles of impedance and admittance control, the robot's behaviour is characterized as a virtual spring-damping-mass system. To detect any contact with the environment, a force sensor is crucial. Given that the stiffness of the force sensor typically exceeds that of both the robot and the diverse environmental stiffness, it becomes imperative to incorporate the sensor model into the overall model. Furthermore, modeling efforts extend to encompass the varying stiffness levels across different application contexts.

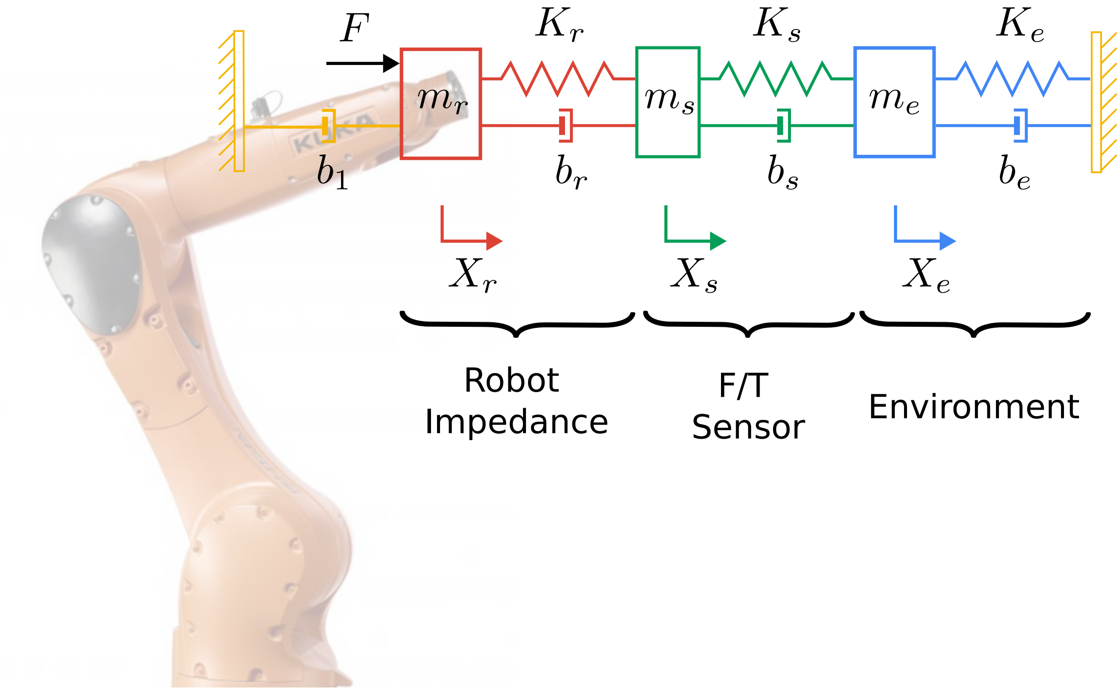


Figure 3.6: Simplified single-degree-of-freedom (DOF) model illustrating the interaction among a robot, force sensor, and environment.

3.3.1 Robot Tool Dynamics Model

The interaction model, between the robot's tool and environment, as shown in Fig. 3.6 can be conceptualized as a system of serially connected springs, damping elements and masses [47, 48]. It is crucial to note that, at this stage, the modeling is conducted in a single degree of freedom (1-DOF) Cartesian space, although the same applies in other axes. The mentioned interaction is defined with the following equations of motion:

$$m_r \cdot \ddot{x}_r = -K_r \cdot (x_r - x_s) - b_r \cdot (\dot{x}_r - \dot{x}_s) - b_1 \cdot \dot{x}_r + F, \tag{3.18}$$

$$m_s \cdot \ddot{x}_s = -K_r \cdot (x_s - x_r) - K_s \cdot (x_s - x_e) - b_r \cdot (\dot{x}_s - \dot{x}_r) - b_s \cdot (\dot{x}_s - \dot{x}_e), \tag{3.19}$$

$$m_e \cdot \ddot{x}_e = -K_e \cdot x_e - K_s \cdot (x_e - x_s) - b_e \cdot \dot{x}_e - b_s \cdot (\dot{x}_e - \dot{x}_s), \tag{3.20}$$

where x_r, x_s, and x_e represent the system state variables, namely the positions of the robot, sensor, and environment, respectively. The parameters of the dynamic system are detailed in Tab. 3.1.

In the following step *Laplace* transformation is conducted to the Eqs. 3.18, 3.19, and 3.20, resulting with following equations:

$$[m_r \cdot s^2 + (b_r + b_1) \cdot s + K_r] \cdot X_r(s) = [b_r \cdot s + K_r] \cdot X_s(s) + F(s) \tag{3.21}$$

$$[m_s \cdot s^2 + (b_r + b_s) \cdot s + K_r + K_s] \cdot X_s(s) = [b_r \cdot s + K_r] \cdot X_r(s) + [b_s \cdot s + K_s] \cdot X_e(s) \tag{3.22}$$

Parameter	Description
m_r	Virtual mass of the robot
m_s	Mass of the force/torque sensor
m_e	Mass of the environment
K_r	Virtual stiffness of the robot
K_s	Stiffness of the force/torque sensor
K_e	Stiffness of the environment
b_r	Virtual damping of the robot
b_s	Damping of the force/torque sensor
b_e	Damping of the environment
b_1	Robot damping

Table 3.1: 3-mass dynamics model parameters

$$[m_e \cdot s^2 + (b_e + b_s) \cdot s + K_e + K_s] \cdot X_e(s) = [b_s \cdot s + K_s] \cdot X_s(s) \tag{3.23}$$

The system is resolved firstly by including $X_e(s)$ from Eq. 3.23 into the Eq. 3.22. Following that, the $X_s(s)$ from Eq. 3.22 is included into the Eq. 3.21. As the final result, the motions of the system are defined with following relations:

$$\frac{X_r(s)}{F(s)} = \frac{N_4(s)}{D_6(s)} \tag{3.24}$$

$$\frac{X_s(s)}{F(s)} = \frac{N_3(s)}{D_6(s)} \tag{3.25}$$

$$\frac{X_e(s)}{F(s)} = \frac{N_2(s)}{D_6(s)} \tag{3.26}$$

where $D_6(s)$ represent characteristic polynomial of the system defined in Eq. 3.27.

$$\begin{aligned} D_6(s) = [m_r \cdot s^2 + (b_r + b_1) \cdot s + K_r] \cdot [m_s \cdot s^2 + (b_r + b_s) \cdot s + K_r + K_s] \cdot \\ [m_e \cdot s^2 + (b_e + b_s) \cdot s + K_e + K_s] - \\ [m_r \cdot s^2 + (b_r + b_1) \cdot s + K_r] \cdot (b_s \cdot s + K_s)^2 - \\ [m_e \cdot s^2 + (b_e + b_s) \cdot s + K_e + K_s] \cdot (b_r \cdot s + K_r)^2 \end{aligned} \tag{3.27}$$

$$\begin{aligned} N_4(s) = [m_s \cdot s^2 + (b_r + b_s) \cdot s + K_r + K_s] \cdot [m_e \cdot s^2 + (b_e + b_s) \cdot s + K_e + K_s] - \\ (b_s \cdot s + K_s)^2 \end{aligned} \tag{3.28}$$

$$N_3(s) = (b_r \cdot s + K_r) \cdot [m_e \cdot s^2 + (b_e + b_s) \cdot s + K_e + K_s] \tag{3.29}$$

$$N_2(s) = (b_s \cdot s + K_s) \cdot (b_r \cdot s + K_r) \tag{3.30}$$

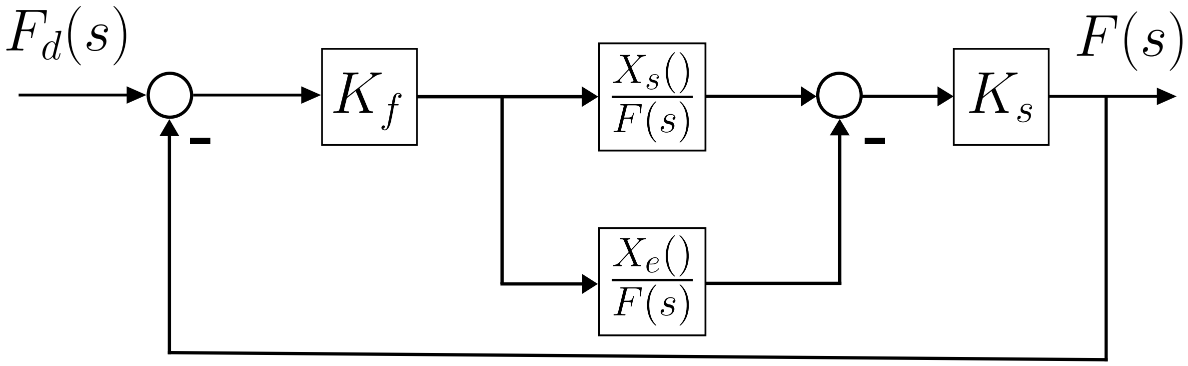


Figure 3.7: Illustrating the control system schematic used for stability analysis via root locus. The proportional gain K_f is systematically increased to observe the system's behaviour.

Since the force sensor reading is defined with:

$$F_s(s) = K_s \cdot (x_s - x_e), \tag{3.31}$$

a simple force control law can be defined as:

$$F(s) = K_f \cdot (F_d(s) - F_s(s)), \tag{3.32}$$

where K_f is the proportional controller gain, and $F_d(s)$ force reference. Schematics of the system is shown in Fig. 3.7. To examine the stability of the system, root locus analysis was conducted. In the first step, open-loop transfer function is derived as:

$$G_{ol}(s) = K_s \cdot \frac{X_s(s) - X_e(s)}{F(s)} = K_s \cdot \frac{N_3(s) - N_2(s)}{D(s)} \tag{3.33}$$

Root locus stability analysis

For the open-loop transfer function (Eq. 3.33), root locus analysis is conducted using the *MATLAB* environment. The analysis, depicted in Fig. 3.8, illustrates how the system evolves as the proportional controller gain K_f increases. The root locus plot reveals the motion of six poles and three zeros across the plane. In this analysis, two real poles are accompanied by two pairs of complex conjugate poles. Notably, as the gain K_f rises, one pair of complex conjugate poles drives the system towards instability. It is apparent that the system remains conditionally stable for gains K_f below a certain threshold, denoted as $K_{f,\max}$.

Given that the force/torque sensor typically exhibits significantly higher stiffness compared to the environment, the influence of the sensor stiffness on the system's stability was examined. In this analysis, the impedance characteristics of the robot and environment were maintained while varying the stiffness of the force/torque sensor. Initially, a sensor stiffness much lower

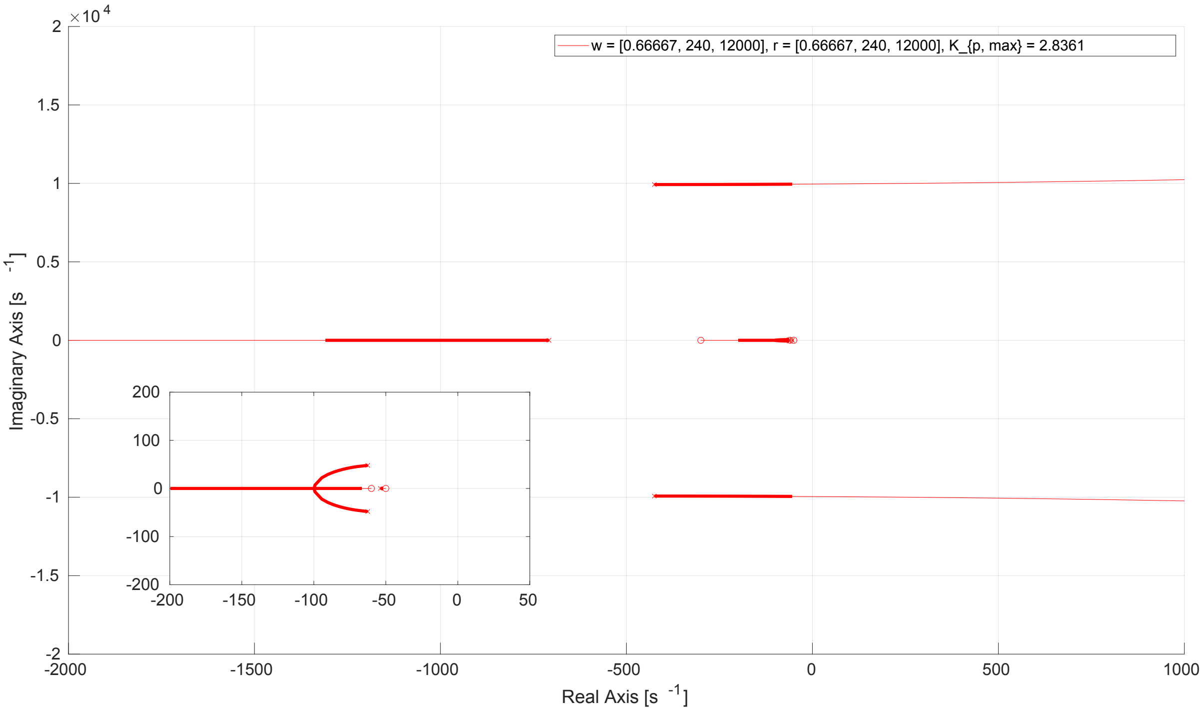


Figure 3.8: Root locus analysis for freely chosen and matching impedance behaviour of the robot and environment. Tick line showcases the movement of the poles through the plane until unstable behaviour.

than the typical environmental stiffness encountered was considered, then matched with the environmental stiffness, and finally, the realistic stiffness of a commercially available force/torque sensor designed for industrial applications was utilized. The root locus analysis corresponding to these scenarios is depicted in Fig. 3.9. It is evident that the realistic sensor stiffness drives the poles of the system into the left half-plane, resulting in a significantly faster response. However, two complex conjugate poles exhibit greater dispersion along the imaginary axis, leading to a more oscillatory response.

The stiffness of the force sensor remains constant, representing a fixed parameter associated with sensor type. Conversely, the robot operates within various scenarios, each characterized by a distinct environmental stiffness range. Although both parameters are beyond direct control, the behaviour of the robot can be controlled. The focus here is on examining how the impedance of the robot contributes to system stability. Firstly, the process for deriving the impedance filter parameters is outlined. Given that the filter can be observed as a PT2 system, the parameters are obtained by combining the PT2 unit gain function with the desired impedance stiffness, as described in the following equation:

$$G_{imp}(s) = \frac{1}{K_e \cdot [\frac{m_e}{K_e} \cdot s^2 + \frac{b_e}{K_e} \cdot s + 1]} = \frac{1}{K_e} \cdot G_{PT2}(s), \tag{3.34}$$

where $G_{PT2}(s)$ represents the PT2 transfer function with freely chosen qualitative parameters, such as overshoot, settling time, and response time. Additionally, there is the option to select

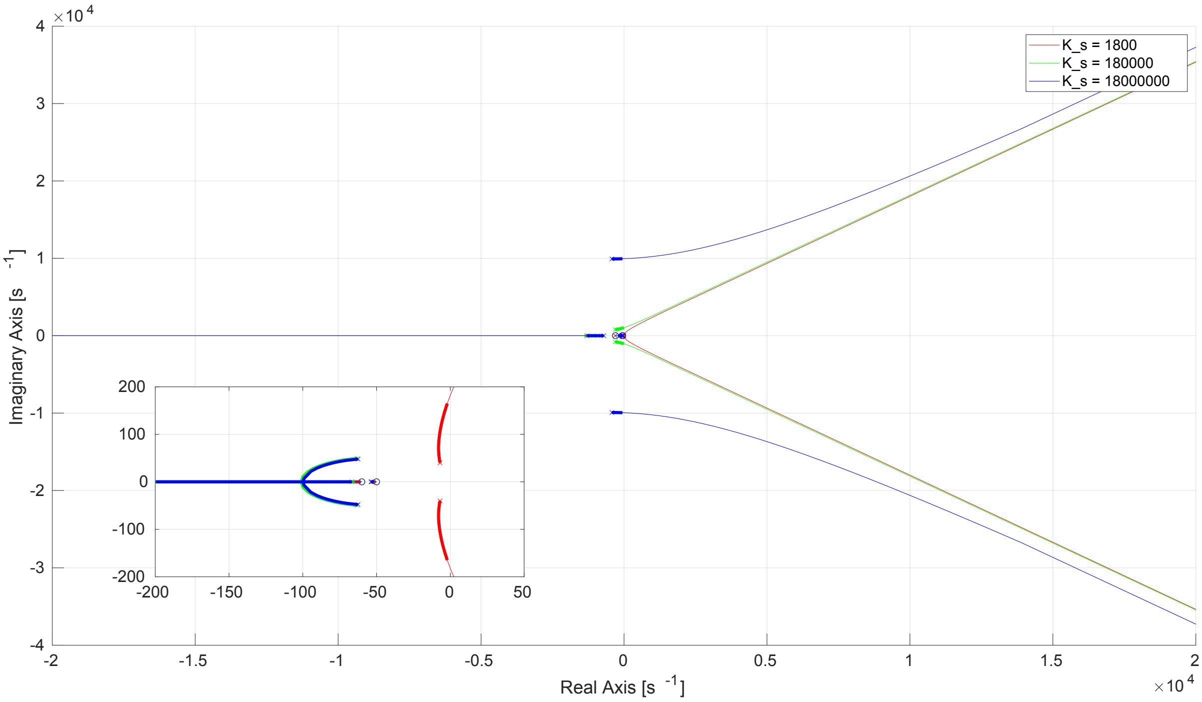


Figure 3.9: Root locus analysis of force sensor stiffness impact on stability of the system. For a single impedance of the robot and environment, force sensor stiffness has been varied, ranging from values under typical environmental stiffness to typical environmental stiffness and higher stiffness fitting the stiffness of commercially available industrial force/torque sensors.

the function with either oscillatory or an aperiodic response. Step responses for chosen PT2 transfer functions used in this analysis are shown in Fig. 3.10.

Initially, the effect of increasing the robot's stiffness while maintaining a consistent response time for the impedance filter is examined. The root locus analysis depicted in Fig. 3.11a illustrates that a robot with higher stiffness than the environment exhibits slower and less oscillatory behaviour, accompanied by a broader stable gain range. Subsequently, the robot's stiffness was maintained equivalent to that of the environment, while varying the robot's response time relative to the environmental response time. Analysis, as shown in Fig. 3.11b, highlights a slower and less oscillatory response when the robot's dynamics are slower compared to those of the environment.

3.3.2 Simulation and Experimental Validation

Building upon the stability analysis of the general three-mass model discussed earlier, now the focus shifts to a real-world scenario involving a robot controlled by the controller proposed in Sec. 3.2. In this setup, a position-constrained controller confined to a single degree of freedom (1-DOF) is utilized. Considering that the robot's motion, denoted by the offset from the initial position X_r, incorporates modeled impedance behaviour, the next step was to model the remainder of the interaction, including the force sensor and environment. Assuming that the

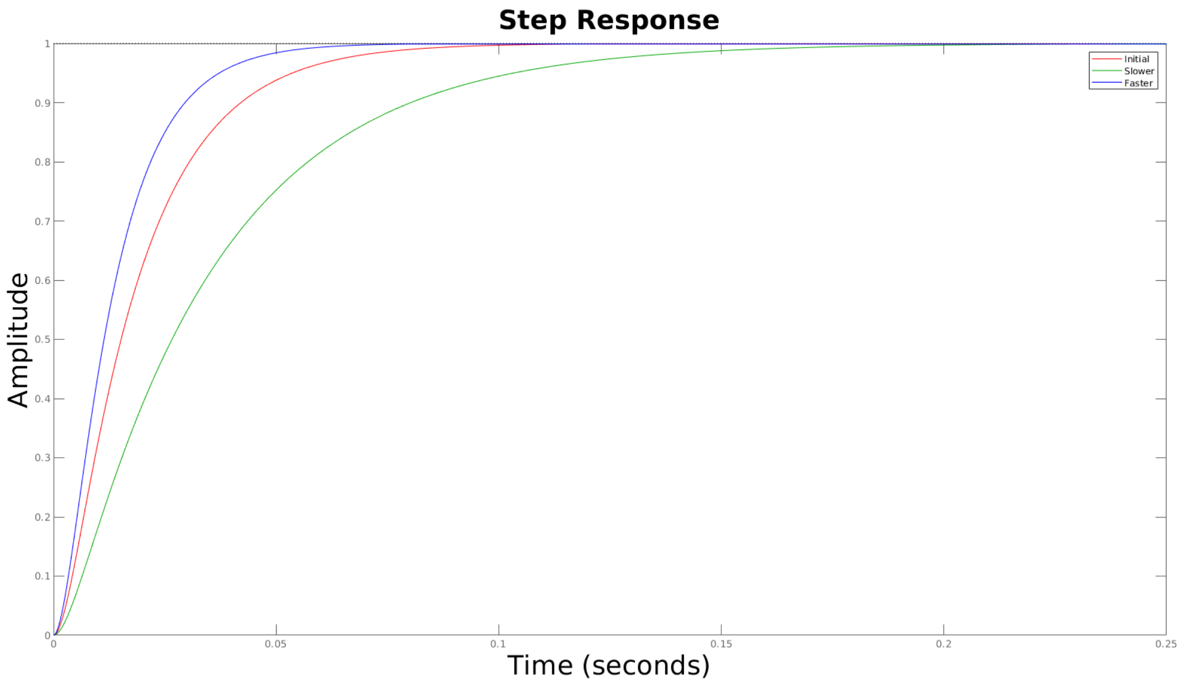


Figure 3.10: Three different impedance response functions used in analysis.

robot motion X_r acts as a control value driving the system, an adjusted model illustrated in Fig. 3.12 is derived. The motion of this adjusted model can be described as follows:

$$m_e \ddot{x}_e = -K_s \cdot (x_e - x_r) - b_s \cdot (\dot{x}_e - \dot{x}_r) - K_e \cdot x_e - b_e \cdot \dot{x}_e. \tag{3.35}$$

Applying *Laplace* transformation to previous equation results in environment motion:

$$X_e(s) = \frac{b_s \cdot s + K_s}{m_e \cdot s^2 + (b_s + b_e) \cdot s + K_s + K_e} X_r(s). \tag{3.36}$$

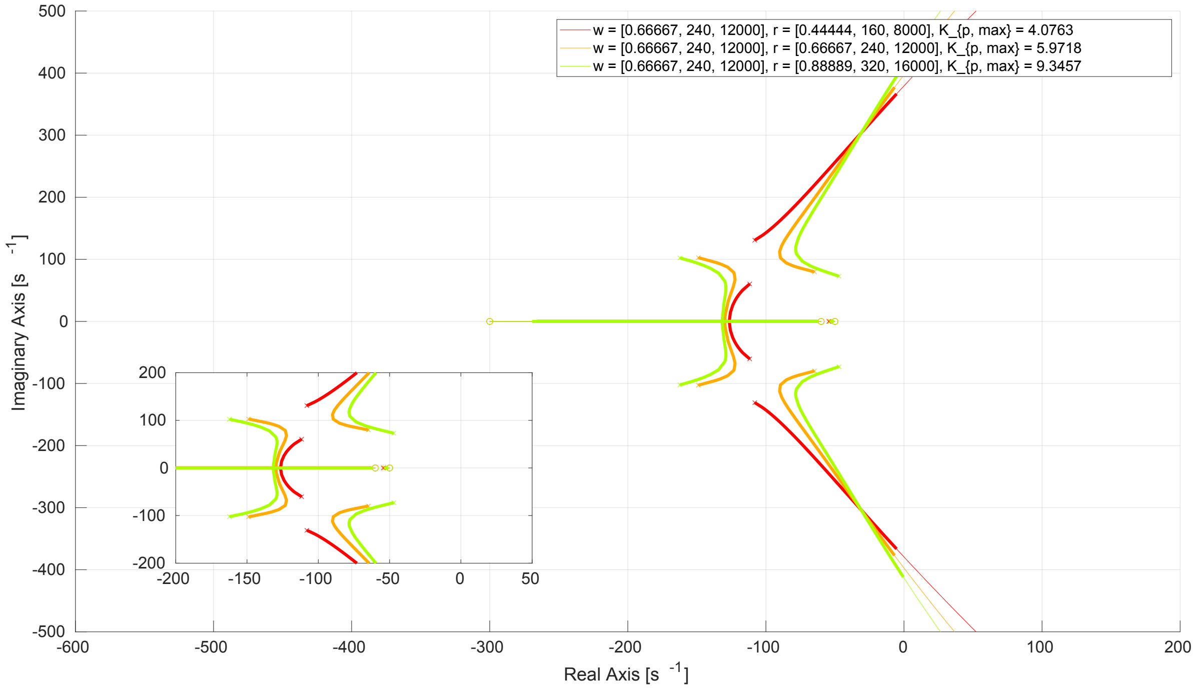
Finally the force sensor reading is defined as:

$$F_s(s) = K_s \cdot (X_e(s) - X_r(s)) = -K_s \cdot \frac{m_e \cdot s^2 + b_e \cdot s + K_e}{m_s \cdot s^2 + (b_s + b_e)s + K_s + Ke} \cdot X_r(s). \tag{3.37}$$

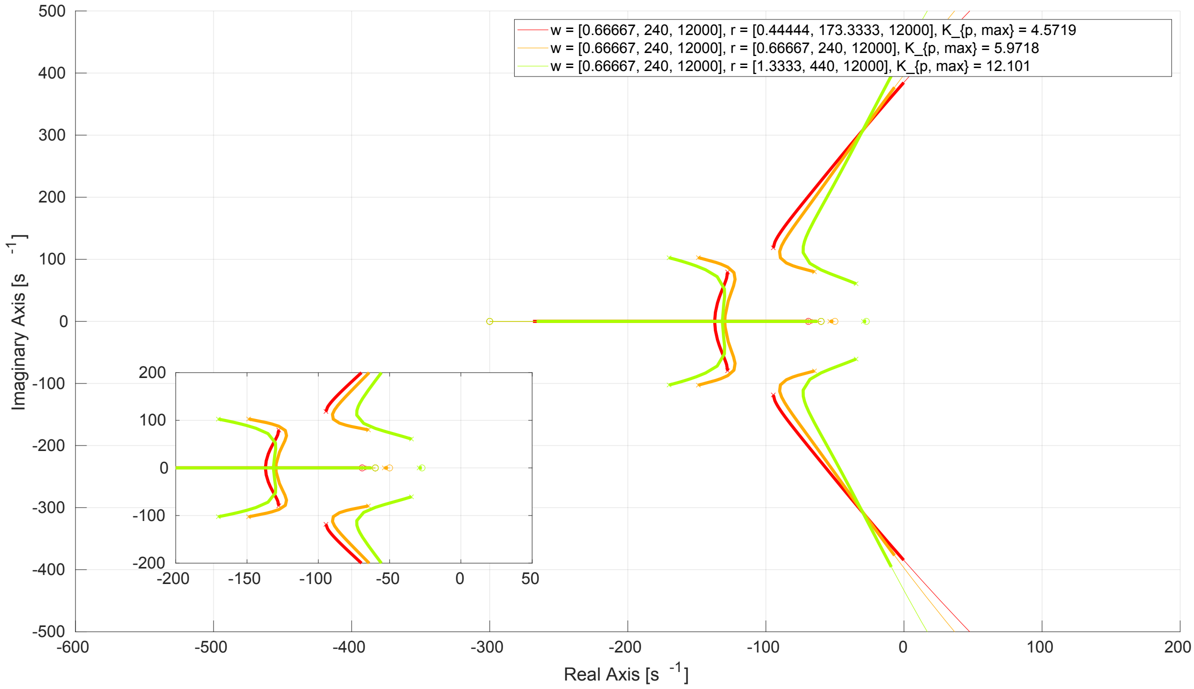
With the previously discussed transfer function, it is undeniable that the force/torque sensor characteristics play a significant role in the system.

Experimental validation

To validate the previously derived model of interaction among the robot, sensor, and environment, experiments in both a simulation environment and a real robot system were conducted. The simulation model was built using *MATLAB Simulink*, while for the real robot system a *KUKA KR10* robot equipped with the position-constrained controller was employed. These experiments aim to explore the scenarios previously analyzed through root locus analysis.



(a) Varying levels of stiffness



(b) Varying response time

Figure 3.11: Root locus analysis of the three-mass dynamic model with different characteristics of the robot and the environment.

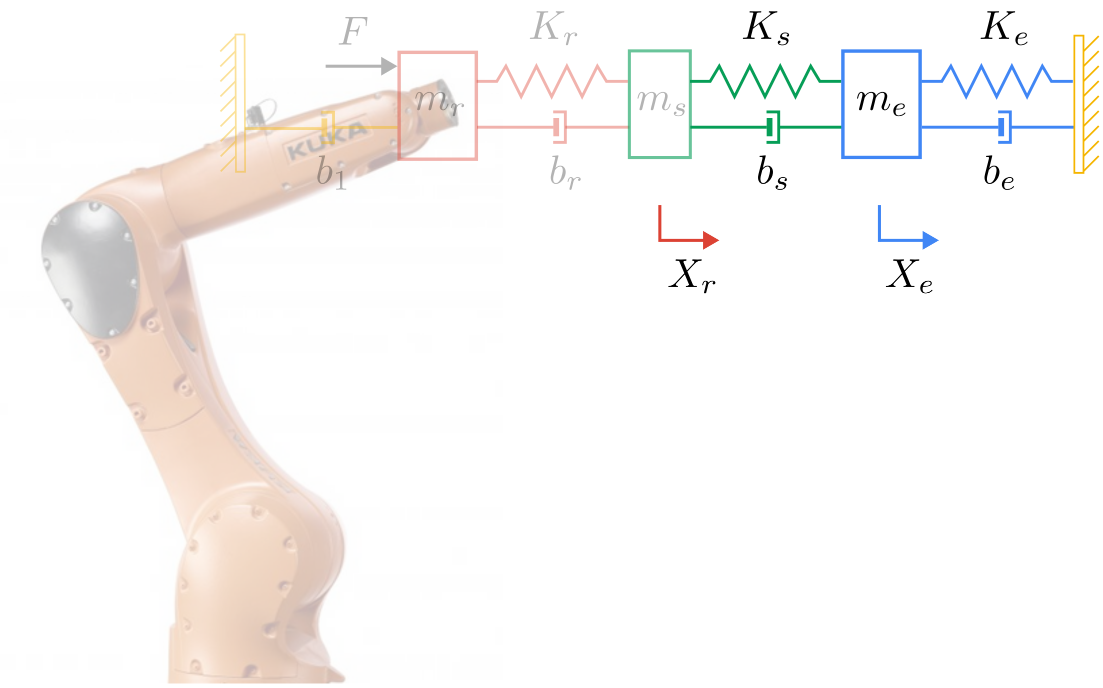


Figure 3.12: Adjusted model depicting the interaction between a Cartesian Compliance-controlled robot and its environment. This model is pivotal for stability analysis, conducted in both simulated environments and real robotic systems.

To manipulate the stiffness of the environment, a specialized tool capable of accommodating varying numbers of identical springs in parallel was developed. By adjusting the number of springs placed in the environment simulator tool, the environmental stiffness can be varied. The experimental setup is illustrated in Fig. 3.13.

In the experiment, firstly the stiffness of the robot relative to the environment is varied. The left plots in Fig. 3.14 illustrate that when the robot stiffness K_r is lower than the environmental stiffness, the response becomes more oscillatory with higher amplitude compared to the initial matching stiffness. Conversely, when the robot stiffness is higher, the system stabilizes, resulting in a less oscillatory behavior. This observation holds true for environments with both two and four springs placed in the environment simulator tool. The right side of the same Fig. 3.14 presents the results of varying the response time of the robot impedance. A slower impedance leads to a less oscillatory response, while a faster robot behavior results in oscillations with higher amplitudes and longer calming periods compared to the initial system. Additionally, the simulation results with the behavior of the actual robot are compared. The robot behavior is depicted by dashed lines, whereas the simulated behaviour is shown with solid lines.

Building upon the previous experiment, the stability of the system was examined when the robot has lower stiffness than the environment, but with a slower impedance filter response. The results of this experiment are depicted in Fig. 3.15. First, the robot impedance was matched to the impedance of a two-spring system, while the environment had four springs. This led to a highly oscillatory response, as depicted in the figure. Next, the response time was increased

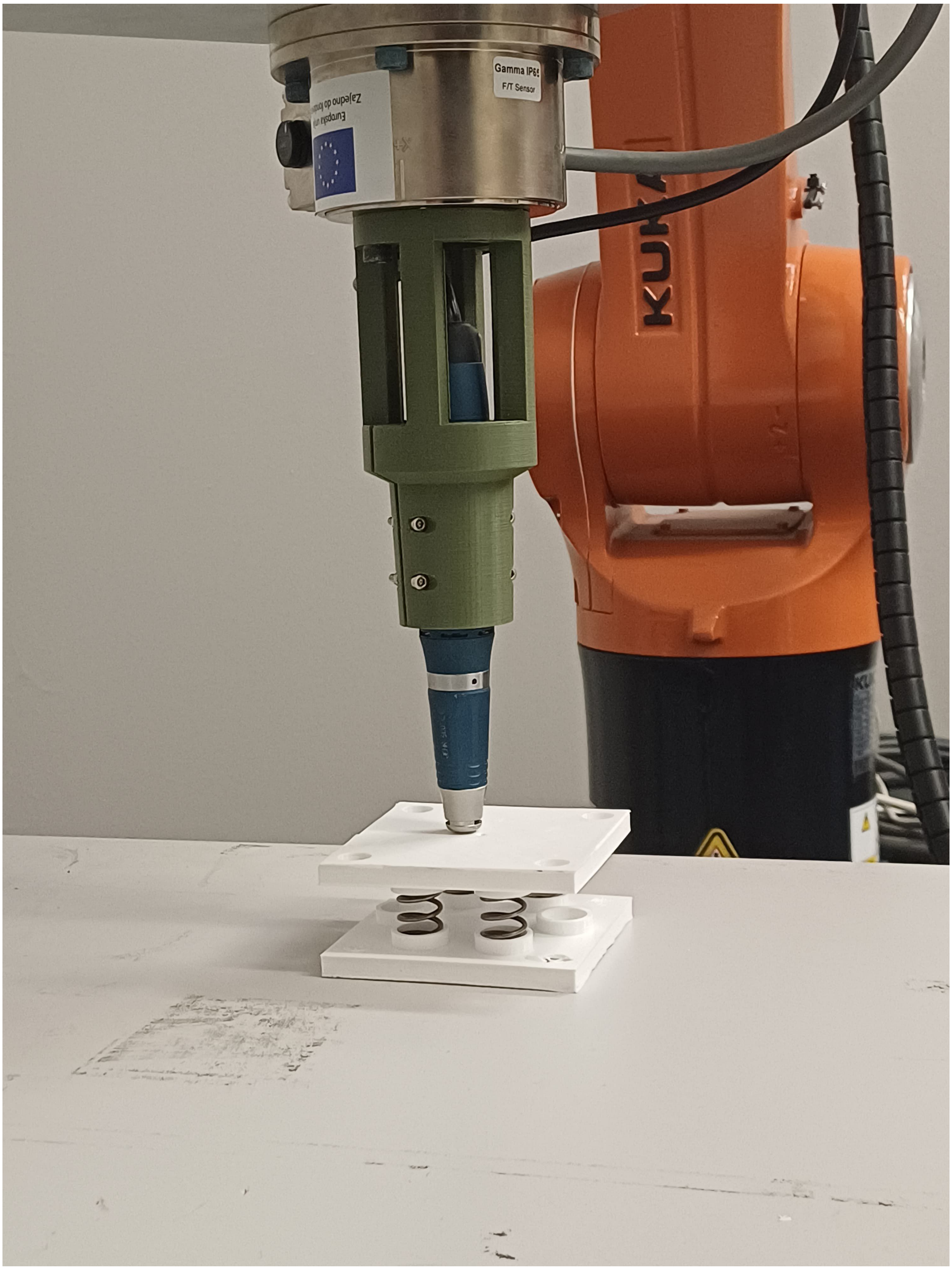
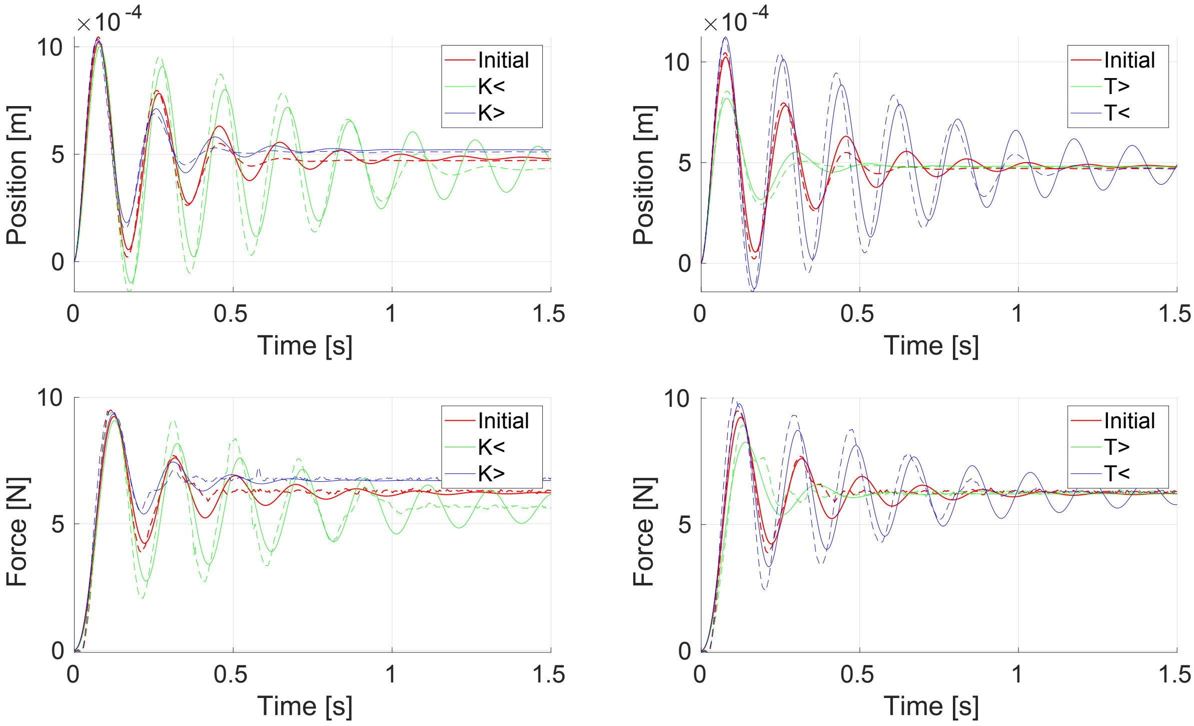
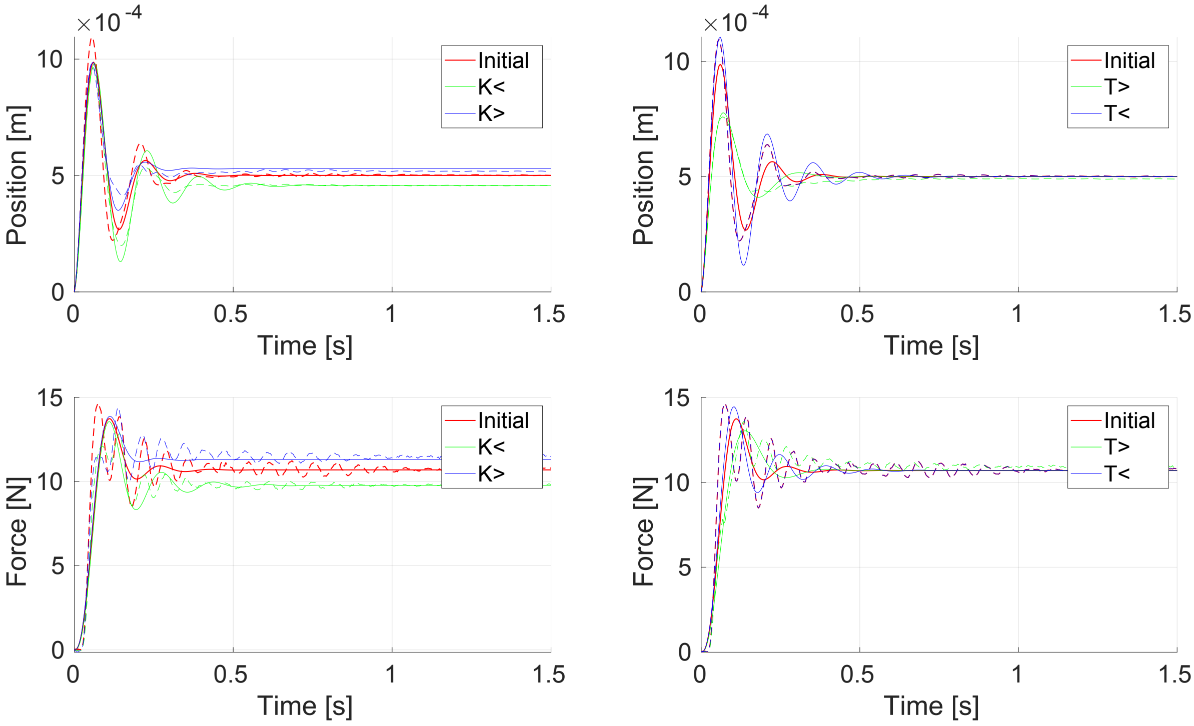


Figure 3.13: A specialized tool was developed to manipulate the stiffness of the environment. This tool accommodates a varying number of identical springs placed in parallel. By adjusting the number of springs placed in the tool, the environmental stiffness can be varied.



(a) 2-spring environment



(b) 4-spring environment

Figure 3.14: Effect of varying robot stiffness compared to the environment on the left, and varying robot response time on the right. Subfigure a) corresponds to an environment stiffness encompassing two springs, while Subfigure b) includes four springs.

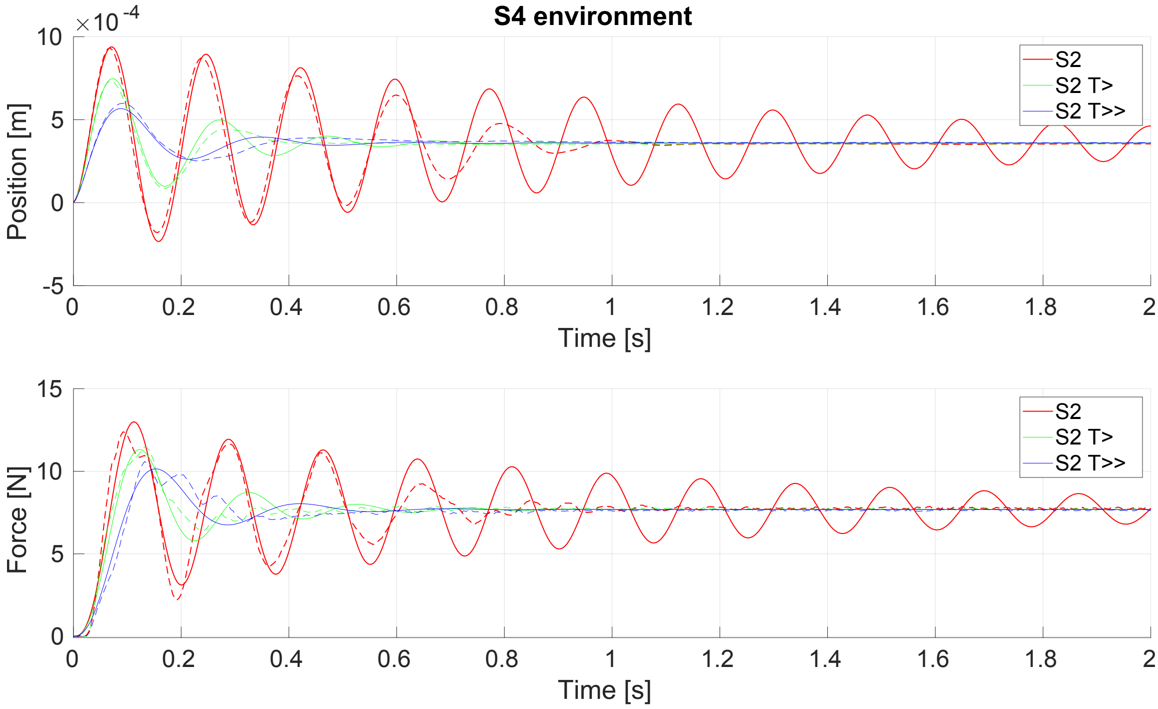


Figure 3.15: Illustrating the impact of a slower controller compared to the environment, despite the controller having significantly lower stiffness than the environment.

in two steps, resulting in a significant reduction in oscillatory behaviour. These experiments confirm the validity of the stability analysis and derived motion models. Moreover, they provide detailed insights into controlling the system across various environments, ranging from soft to stiff. Later, in Ch. 6, the compliance control system is tested in various environments, including soft autonomous plastering, grinding of composite materials, and stiff deep-micro-hole drilling of cast iron moulds.

CHAPTER 4

Programming-by-Demonstration

With the increasing adoption of robot manipulators in factory settings, standard industrial manipulators are renowned for their high payloads, extensive reach, and precision. However, their stiff, joint position-controlled nature lacks sensing capabilities, resulting in confinement within safety fences, restricting human interaction. These machines are predominantly found in large-scale manufacturing due to complex programming and setup requirements. As industries evolve, there is a growing demand for more flexible robot manipulators to enter small-scale manufacturing and collaborate closely with humans. Sensing capabilities are essential for compliant and safe interactions with human operators and the environment, allowing sharing the workspace and close collaboration between the human and the robot. This also opens-up different possibilities in Programming-by-Demonstration (PbD) of specific tasks by the operator. However, putting the operator in close contact with the machine, even if compliant and safe, may affect their comfort and influence the demonstration of skills. Therefore, alongside the kinesthetic approach for skill demonstration, a novel concept employing specially developed tools and a motion capture system is presented.

This chapter presents a collaborative manipulation principles, exploring PbD through kinesthetic guiding of the robot's end-effector and a novel human-robot interface framework based on a motion capture system. The chapter addresses skill reproduction issues and presents the results of a user-experience survey on two PbD approaches.

4.1 Kinesthetic Teaching

Kinesthetic teaching, also known as manual guidance or lead-through programming, is a method employed in collaborative robotics to instruct robots through physical manipulation by human

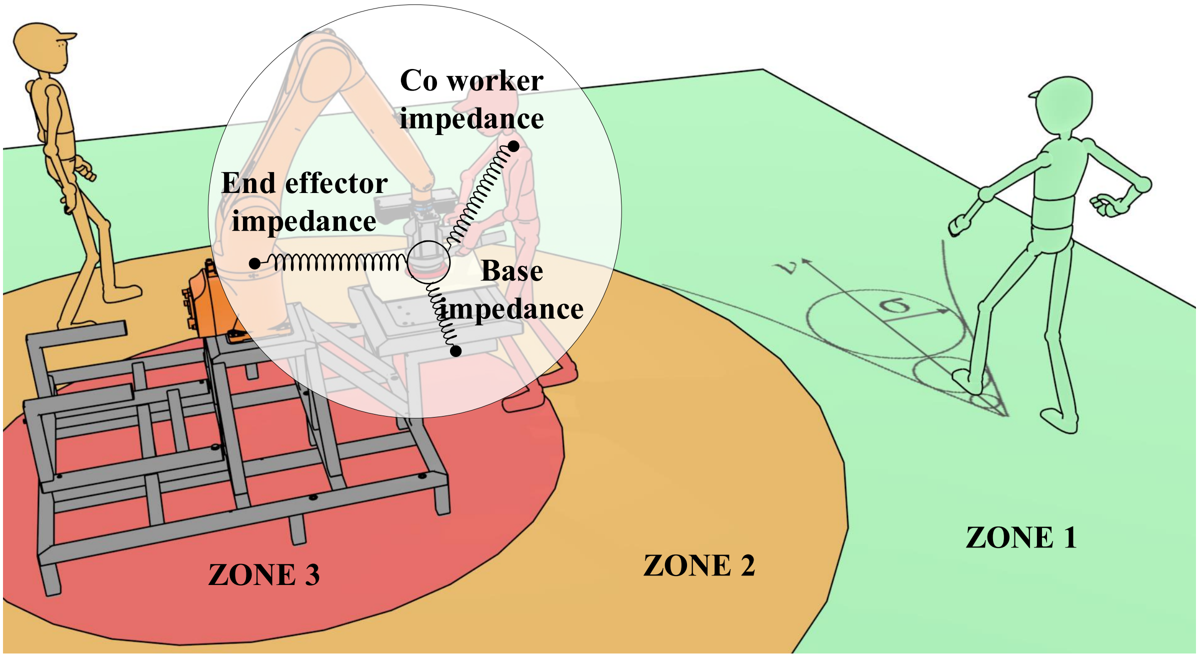


Figure 4.1: The schematic view of the collaboration between the robot and human operator. The robot becomes compliant once the operator picks up the tool and presses a button to take control of the robot end-effector. While in teaching mode, robot compliance is modified in order to match compliance of the human arm and the treated object.

operators. Unlike traditional programming methods that involve writing code, kinesthetic teaching allows operators to physically guide the robot's end-effector. This approach offers several advantages. Firstly, it simplifies the programming and deployment of robot manipulators, as operators do not need to possess advanced robot programming skills. Instead, they can rely on their expertise in performing specific tasks and physically and intuitively teach the robot to do the same. This opens up new possibilities for deploying robots even in small-scale industries, as the robots can be deployed quickly and without the need for robotic engineers to program them. Operators can quickly adjust robot motions based on real-time feedback, refining trajectories or task sequences on the fly. Overall, kinesthetic teaching represents a powerful approach to paradigm of Programming-by-Demonstration, enabling intuitive, flexible, and collaborative interactions between humans and machines.

PbD using Collaborative Manipulator

The kinesthetic approach in PbD became possible with development of collaborative manipulators. Such manipulators, with example shown in Fig. 4.2 are joint-torque controlled and compliant, meaning they move freely in response to external forces acting on the robot's body. In addition, it ensures safety during the human-robot interaction. Having this in mind, collaborative manipulators are perfectly suited for utilizing in kinesthetic teaching approach, but face significant problem in industrial applications where high precision, repeatability, payload and reach are required.

(a) Franka Emika Panda [3]   (b) KUKA iiwa [4]

Figure 4.2: Collaborative robot manipulators

PbD using Standard Industrial Manipulator

Unlike collaborative robots, standard industrial manipulators are typically stiff joint-position controlled and lack the ability to sense contact with the environment, rendering them incapable of kinesthetic teaching. However, industrial manipulators excel in precision, repeatability, payload capacity, and reach—attributes that often render the use of collaborative robots impractical in industrial settings. Building upon the proposed framework for the collaborative use of industrial manipulators, one of key components is enabling kinesthetic teaching with these robots. The effectiveness of this system is aimed to be demonstrated in an industrial scenario involving delicate grinding, where the robot can be guided by operators in performing grinding tasks on delicate surfaces. The concept of the Human-Machine Interface (HMI) is illustrated in Fig. 4.1. To enable this, a control system was built and tuned, as presented in Sec. 3.2, consisting of a double set of impedance-based controllers, which is able to switch its compliance behaviour and operate in autonomous and manual control mode.

When the system enters the teaching mode, impedance of the system is switched to a more user-intuitive compliance that will allow the user to move the sanding tool easily. Simultaneously, the position correction of the desired end-effector position is performed following the friction-less motion equation $\mathbf{F}_{sens} = \mathbf{I}_r \cdot \ddot{\mathbf{X}}$, where $\mathbf{F}_{sens} \in \mathbb{R}^{6\times1}$ is the sensor force reading vector, $\ddot{\mathbf{X}} \in \mathbb{R}^{6\times1}$ is end-effector acceleration vector (Eq. 3.2), while $\mathbf{I}_r \in \mathbb{R}^{6\times6}$ is chosen tool inertia matrix (Eq. 3.5). Now, position and orientation correction, driven by kinesthetic teaching, are calculated by double integration of the motion correction, for each vector element $i \in [1,\dots,6]$

---

[3] https://interestingengineering.com/innovation/franka-emika-a-robot-that-can-replicate-itself

[4] https://robotsguide.com/robots/lbriiwa

according to:

$$\Delta X_i = \frac{1}{2}\frac{F_{sens,i}}{I_{r,i}}\Delta T^2, \tag{4.1}$$

where $I_{r,i}$ is i-th diagonal element of inertia matrix and ΔT sampling time of the control. The calculated vector $\Delta\mathbf{X}$, similarly to the Eq. 3.11, contains values of rotation and translation in all 6-DOF in the end-effector frame. Using the given values, transformation matrix $\Delta\mathbf{T}_{actual}^{desired}$ is defined. The orientation part of the transformation is derived using a standard *roll-pitch-yaw* notation. Since the calculated transformation matrix $\Delta\mathbf{T}_{actual}^{desired}$ represents the correction movement in actual robot position frame, it is used to transform the new desired position to the robot base frame. New desired position and orientation of the robot tool are calculated using:

$$\mathbf{T}_{base}^{desired} = \Delta\mathbf{T}_{actual}^{desired} \cdot \mathbf{T}_{base}^{actual} \tag{4.2}$$

To sum up, demonstration mode is run using impedance behaviour of the controller, which pushes the robot to the desired position. The movement is limited to acceleration limits and impedance controller parameters. Actual pose of the robot tool during demonstration mode can be influenced with the forces generated from a human as well as the contact forces. This can result in pose difference between the actual and the desired pose of the robot. For this reason, actual pose of the robot is utilized for PbD purposes.

4.2 Tool Motion Tracking

While the off-the-shelf available cobots perform well in tasks like pick-and-place operations, their limitations in precision, payload, and reach hinder their efficacy in standard industrial applications such as spot or arc welding, painting, etc. Manually demonstrating accurate robot poses can be tedious, slow, or imprecise, presenting challenges, especially in small-scale businesses and small batch productions requiring complex tasks. In this context, a method that enables safer deployment of robot manipulators in PbD tasks is proposed. The method is simple and intuitive for the operator, and the setup is quick since the fast calibration procedure can be conducted online. The setup is designed in such a way that it minimizes the direct contact of the human operator with the robot. Instead, it relies on visual communication using a motion capture system and a set of specifically designed tools.

The next section outlines specifically designed human-robot interaction tools tailored for measurement or demonstration tasks. The tools are designed so that their deployment is intuitive, yet they enable the operator to demonstrate complex manipulation tasks to the robot. Two tools are presented, one as a pointer, and another as a line pointer. Additionally, the proposed system is expanded to track any other tool used for specific industrial applications, such as the drilling machine. The idea behind this concept is to expand it to so called manipulation primitives, as described in [49], where each tool can be used to demonstrate a specific part of a complex manipulation task. The calibration of the instruments is done through very fast numerical optimization enabled by preprocessing of calibration data. This is extremely important since the envisioned scenario involves the technology that requires the operator to calibrate the tool each time the system is turned on.

Motion Capture System - OptiTrack

The OptiTrack system [5] is a leading commercially available motion capture system designed not only for tracking object movements but also for precisely capturing human body dynamics, facilitating the generation of lifelike animations and the acquisition of data for diverse applications. Utilizing a network of cameras and specialized markers (fiducials), as illustrated in Fig. 4.3c, this system meticulously tracks motion within a 3D space. These markers, affixed to either the subject's body or objects, reflect emitted infrared light captured by the cameras, enabling exceptionally accurate and real-time motion tracking.

The infrared (IR) cameras are typically strategically positioned around the designated capture area, referred to as the *OptiTrack arena*, depicted in Fig. 4.3a. This arrangement ensures comprehensive coverage of the workspace by multiple cameras within their respective fields of view. Precisely calibrated, these cameras detect the positions of passive reflective markers. However, for more straightforward setups, *OptiTrack* offers streamlined alternatives like the *OptiTrack Duo* or *OptiTrack Trio*, as shown in Fig. 4.3b. These systems incorporate two or three cameras integrated into a single unit, pre-calibrated at the factory. This plug-and-play design facilitates ease of use, particularly suitable for smaller working areas or on-site motion capture requirements.

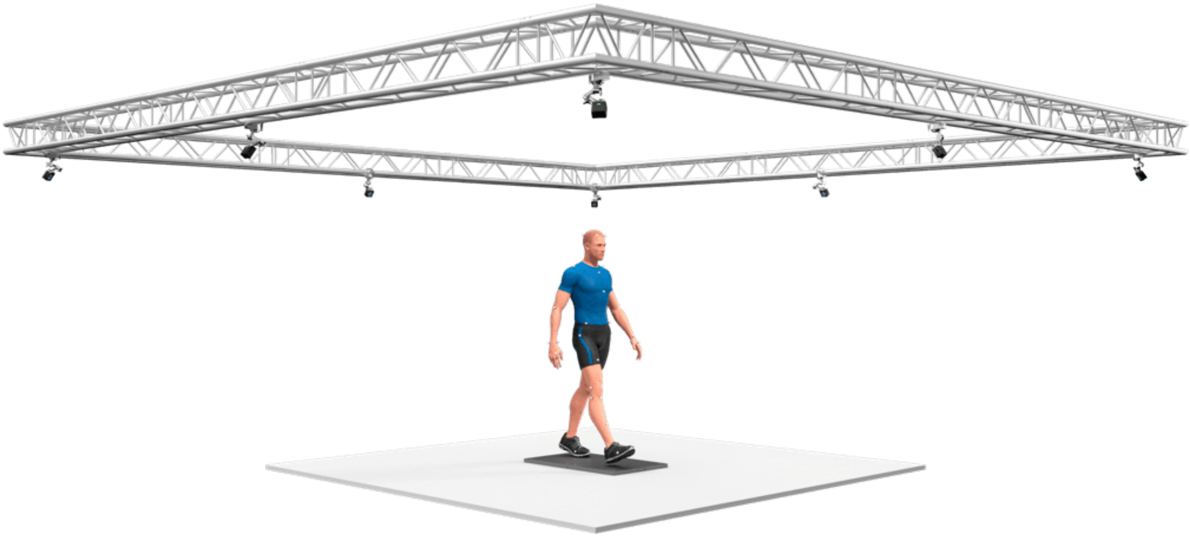
In tracking the pose of a rigid body, passive reflective markers are affixed to the object being tracked. In a 3D space, a minimum of three markers are necessary to establish the plane through vector relationships, defining the object's position and orientation. However, utilizing four or more markers is recommended whenever possible. Additional markers yield more coordinates for computing both positions and orientations of the rigid body, enhancing overall tracking stability and reducing vulnerability to marker occlusions. Crucially, markers should be asymmetrically placed to ensure clear orientation distinction. Avoiding symmetrical configurations—such as squares or equilateral triangles—is essential. Symmetrical arrangements can hinder asset identification and potentially cause the rigid body assets to flip during capture, disrupting the tracking process. In addition, when tracking multiple objects, it is beneficial to use a distinctive arrangement of markers between each rigid body.

---

[5] https://optitrack.com/

[6] https://optitrack.com/cameras/v120-trio/indepth.html

[7] https://abeer-cs491.weebly.com/system-confuguration.html



(a) Optitrack arena [5]

(b) Optitrack trio [6]

(c) Optitrack markers and camera [7]

Figure 4.3: Optitrack mocap system

Figure 4.4: The functionalities of a wireless joystick and a virtual marker were combined to design a prototype of an active virtual pen that can be used to program and deploy robotic systems.

4.2.1 Virtual pen

Our primary objective was to create a tool that facilitates effortless demonstration while keeping the operator within their comfort zone. To achieve this, the idea of developing a pen-like device for swift and accurate 3D pose tracking was conceived, providing an intuitive interface for streamlined functionality. Having these capabilities this device becomes active virtual pen. The pen is comprised of three main components: the modular tip, the body, and a set of passive IR reflective fiducials. In order to enable the trigger function for automatic storage of measured points, the pen's body was designed to include a button, feedback LEDs, battery and electronics with wireless communication functionality. The trigger button is placed on the bottom close to the tip, so the button can be triggered while holding the pen in the usual way. Special care was given to ensure that the pen accommodates the demonstrator's hand ergonomic. This makes it easy to precisely point and draw in 3D space.

Electronic components were put together using an *Arduino*-based *D1 Mini ESP32* microcontroller with *Bluetooth* and *Wi-Fi* functionalities. A Bluetooth connection was opted for to conserve energy. The electronic device is equipped with a *18650 Lithium-Ion* battery, along with a charging module. The device features an on/off switch and a single button used for the snapshot function. It is worth noting that more buttons can be easily added to the system. Additionally, two LEDs are integrated to provide feedback to the user. The LEDs offer information such as the status of the Bluetooth connection, readiness of the device for measurement, button

Specification	Value
Battery capacity:	2.600mAh
Weight:	105 g (body of the marker)
Estimated battery time:	approx 20h
Wireless range:	up to 30m

Table 4.1: Virtual pen specification

press acknowledgment, and charging status, among others. Communication is established using Bluetooth, which offers a 30 m range and efficient means of data transfer between the active pen and the computer. On the computer side, a Bluetooth serial connection and a dedicated driver are used to enable interfacing with the ROS environment. Technical parameters of the developed virtual pen are provided in Tab. 4.1.

4.2.1.1 Point-shaped virtual pen

The main idea of the point-shaped virtual pen is to measure and/or mark the global pose of a specific point in the global reference frame. The operator can place the tip of the virtual pen to a chosen point in the global frame, and virtually mark it as if it were a pencil. The other end of the pen is equipped with a set of markers. The camera system records the position and the orientation of the virtual rigid body formed with these markers. The point chosen and marked by the operator can then be measured and used in robot manipulation.

The *Optitrack* system measures the position and the orientation of the markers' centroid $\mathbf{T}_0^{\mathrm{I}}$. The global position of the point of interest (the tip position) is given with transformation $\mathbf{T}_0^{\mathrm{P}}$:

$$\mathbf{T}_0^{\mathrm{P}} = \mathbf{T}_0^{\mathrm{I}} \cdot \mathbf{T}_{\mathrm{I}}^{\mathrm{P}}, \tag{4.3}$$

where $\mathbf{T}_{\mathrm{I}}^{\mathrm{P}}$ is transformation from the centroid of the marker rigid body to the point instrument tip, obtained through the calibration procedure. During the calibration, the tip of point-shaped virtual pen is anchored at a fixed point in the global reference frame $\mathbf{T}_0^{\mathrm{P}}$, while the other end of the pen (namely, the markers) moves through space. The camera system records the markers' position and orientation, and generates a calibration dataset. This procedure is shown in Fig. 4.5. An auxiliary calibration tool is designed for easier conducting of the calibration procedure, allowing us to keep the tip fixed while rotating the markers.

The transformation $\mathbf{T}_{\mathrm{I}}^{\mathrm{P}}$ is be obtained such that all of the recorded dataset points $\mathbf{T}_{0,i}^{\mathrm{I}}$ project towards the same point (the tip) through this rigid transformation. In other words, the objective of the optimization is to minimize the dissipation of the transformed points $\mathbf{T}_{0,i}^{\mathrm{P}}$. With this approach, the exact position of the point-instrument tip $\mathbf{T}_0^{\mathrm{P}}$ can be left unknown, as long as it is fixed. This holds under assumption that the instrument body is rigid (not flexible). In this step of calibration, the orientation of the tip of the marker $\mathbf{T}_0^{\mathrm{P}}$ is irrelevant (only the position of

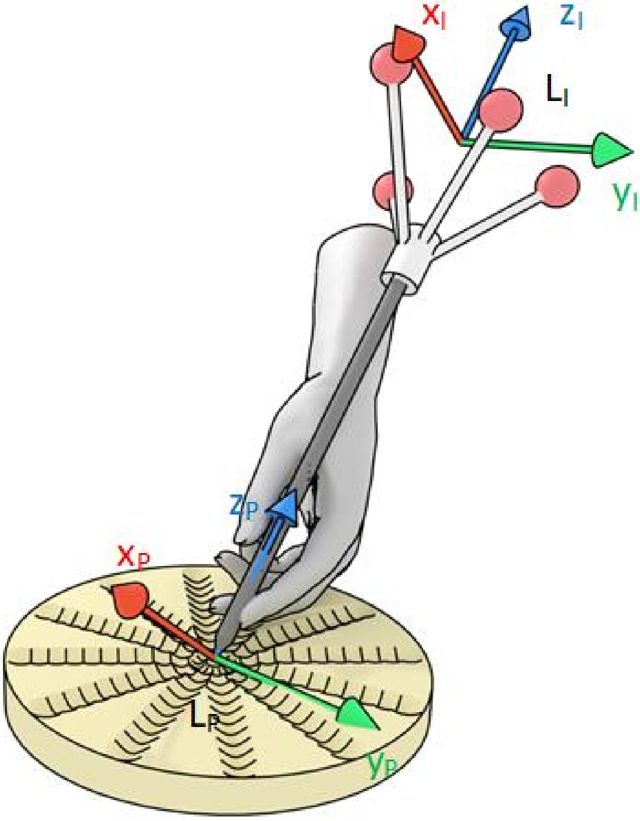


Figure 4.5: Point-shaped pen calibration procedure. The pin of the instrument is fixed in the central position of the calibration tool, while upper part of the pen moves. The global pose of the centroid of the Optitrack markers L_I is measured. Calibration procedure derives transformation $\mathbf{T}_{\mathrm{I}}^{\mathrm{P}}$ in order to obtain the global pose of the pen's tip $\mathbf{T}_{0}^{\mathrm{P}}$.

the marker tip is concerned), the rotation part of the transformation $\mathbf{T}_{\mathrm{I}}^{\mathrm{P}}$ is assumed as identity matrix, while translation part is in the scope of calibration:

$$\mathbf{P}^{*} = \arg\min_{\mathbf{P}} \sum_{i=1}^{\mathrm{N}} \sum_{j=1}^{\mathrm{N}} ||\mathbf{T}_{\mathrm{O},i}^{\mathrm{I}} \cdot \mathbf{T}_{\mathrm{I}}^{\mathrm{P}} - T_{\mathrm{O},j}^{\mathrm{I}} \cdot \mathbf{T}_{\mathrm{I}}^{\mathrm{P}}||, \tag{4.4}$$

where $\mathbf{P}^{*}$ is optimal translation part of the transformation $\mathbf{T}_{\mathrm{I}}^{\mathrm{P}}$. The transformation obtained through this optimization is further used in the experiments, such that each recorded marker position is transformed into the global tip position, i.e. the global position of the actual point of interest.

The point-shaped virtual pen can be calibrated so that both position and orientation of the pen are measured. This kind of calibration is useful in many different applications where both position and orientation of the robot manipulator have to be referenced. One example is a PbD robot drilling system. The point-shaped virtual pen can demonstrate both the point and the approach vector for the drilling tool. To provide a better understanding of the proposed concept, a robotic drilling system is derived, where the point-instrument is used as the Human-Machine

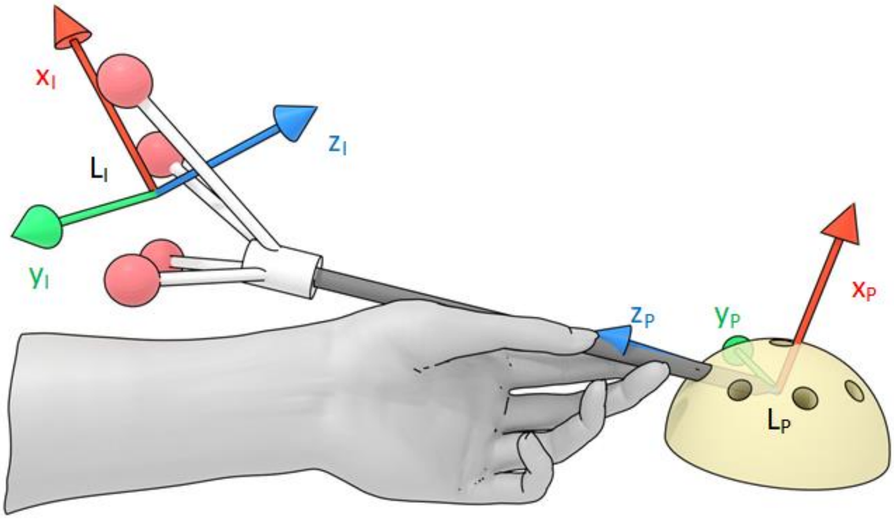


Figure 4.6: The angle calibration tool, designed with holes of a priori known inclination w.r.t. the camera system reference plane. Pose of the point-shaped virtual pen is measure for each hole of the calibration tool. The optimization process aims to re-calibrate point-shaped pen transformation to so that the measured vector $\mathbf{z}_p$ and reference vector $\mathbf{z}_{ref}$ are aligned.

Interface (HMI). In this scenario, human operator uses the point-instrument to specify point and approach vector for drilling tool. After the teaching phase, the robot starts drilling in the same position and same approach axis. More details about this use case can be found in Sec. 6.2.

Since the rotation part of the transformation matrix is crucial in such applications, a special calibration tool is designed as shown in Fig. 4.6. The tool is designed with holes of a priori known inclination with respect to the camera system reference plane. During the dataset collection process, the point-shaped virtual marker's position and orientation is recorded while the tip is placed into each hole of the tool.

The goal of the optimization process is to modify the transformation $\mathbf{T}_{\mathrm{I}}^{\mathrm{P}}$, so that the measured vector $\mathbf{z}_p$ and reference vector $\mathbf{z}_{ref}$ are aligned. The vectors $\mathbf{x}_p$ and $\mathbf{y}_p$ are out of scope of this optimization due to the axial-symmetric design of the marker's tip. Since the rotation of the rigid body not moving in space is a part of SO(3) group, at least 3 variables are needed to define a rotation. As a basis for defining rotation, yaw-pitch-roll Euler notation is used, resulting in three angles of rotation Ψ^*, Θ^*, Φ^* that are within the scope of the optimization. The optimization problem is defined as follows:

$$[\Psi^*,\Theta^*,\Phi^*] = \arg\max_{[\Psi,\Theta,\Phi]} \sum_{i=1}^{\mathrm{r}} \sum_{j=1}^{\mathrm{N}} \left\| \frac{\mathbf{z}_{ref,i} \cdot \mathbf{z}_{i,j}^{[\Psi,\Theta,\Phi]}}{||\mathbf{z}_{ref,i}|| \cdot ||\mathbf{z}_{i,j}^{[\Psi,\Theta,\Phi]}||} \right\|, \tag{4.5}$$

where the goal is to maximize the scalar product of the measured vector $\mathbf{z}_p$ and reference vector $\mathbf{z}_{ref}$ resulting in these two vectors to be aligned.

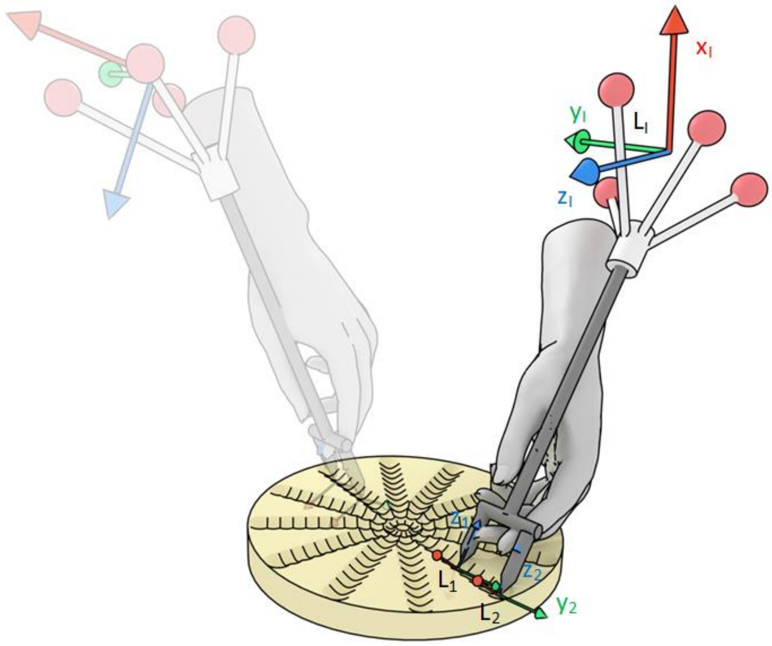


Figure 4.7: Presenting the line-shaped pen calibration experiment, where the line-instrument is placed in 9 positions, equally distributed on the line. The dataset is used to optimize transformations of the pin in order to get best possible distance between the pins and line direction measurement. The calibration tool's radius is 10 cm, with 7 lines placed circumferential. In each line there are holes with distance 1 cm between the holes.

4.2.1.2 Line-shaped virtual pen

The line-shaped virtual pen, like the point-shaped, has a set of markers on the free end, while the other end has two tips, as opposed to the point-shaped which only has one. The two tips, denoted *master* and *slave*, are used as two points that span a line through the space. The instrument is designed in such a way that the distance between the tips is 3 cm. The calibration procedure of the line-shaped pen is conducted in two phases. The first phase is similar to the point-shaped pen calibration, yielding independent position transformations for both master $\mathbf{T}_{\mathrm{I}}^{\mathrm{P}_1}$ and slave $\mathbf{T}_{\mathrm{I}}^{\mathrm{P}_2}$ pins w.r.t. the markers' centroid. In both phases the rotation part of the transforms are set to identity matrix, and only the translations $\mathbf{P}_1$ and $\mathbf{P}_2$ are optimized.

The second phase of the optimization re-optimizes the line-shaped pen transformation with the help of another auxiliary calibration tool, shown in Fig. 4.7. The tool is designed with holes arranged along a line, such that the holes are placed at 1 cm distance. This way, the pen tips can fit into two holes at 3 cm distance, with two unoccupied holes between them on the same line.

The first measurement in the second phase of the optimization is taken while the pen tips are placed in the first and the third position along a chosen line. Then, the marker position is recorded while the instrument is moved along the line one position at a time. The optimization is conducted as a multi-criteria optimization with two parts: *Alpha* and *Beta*. The cost function of the optimization is given in Eq. 4.6. The minimization of the cost function is again conducted with Nelder-Mead method, with initial values of the optimized parameters $\mathbf{P}_1^*, \mathbf{P}_2^*$ obtained from the first phase.

$$(\mathbf{P}_1^{**},\mathbf{P}_2^{**}) = \underset{\mathbf{P}_1,\mathbf{P}_2}{\arg\min}\left\{\alpha \cdot f_{c1}(\mathbf{P}_1,\mathbf{P}_2) + \beta \cdot f_{c2}(\mathbf{P}_1,\mathbf{P}_2)\right\}, \tag{4.6}$$

where $\mathbf{P}_1^{**}$ and $\mathbf{P}_2^{**}$ are optimal translations of the pen tip after second phase of the optimization process.

The *Alpha*-criterion relies on the geometric property of the calibration tool in further optimizing the translation transformation. Since the pen is fixed at discrete positions on the tool, the position of the slave tip in i-th recording should correspond to the position of the master tip in the $(i+3)$-rd recording. The cost function of the *Alpha*-criterion thus evaluates this distance as:

$$f_{c1}(\mathbf{P}_1^{**},\mathbf{P}_2^{**}) = \sum_{i=1}^{\mathrm{r}-3}\sum_{j=1}^{\mathrm{N}}\sum_{k=1}^{\mathrm{N}} ||\mathbf{T}_{0,j,i}^{\mathbf{I}} \cdot \mathbf{T}_{\mathbf{I}}^{\mathbf{P}_2} - \mathbf{T}_{0,k,i+3}^{\mathbf{I}} \cdot \mathbf{T}_{\mathbf{I}}^{\mathbf{P}_1}||, \tag{4.7}$$

where index i denotes the measurement position on the calibration tool. The global position of the slave pin $\mathbf{T}_0^{\mathbf{P}_2}$ at tool position i should correspond to the global position of master pin $\mathbf{T}_0^{\mathbf{P}_1}$ at tool position $i+3$. Since multiple recordings (N) are collected for each position on the tool, all of them are mutually compared with j and k indices that denote each $\mathbf{T}_0^{\mathbf{I}}$ measurement sample.

The *Beta*-criterion evaluates the orientation of the measured line. The objective is to align the direction vectors of the measured lines that the master and the slave pin span, through re-optimization of the translation transform. The measured directions are fit towards the mean direction vector across the measurements. The cost function is defined as the sum of scalar products of the measured vectors with the average line vector as:

$$f_{c2}(\mathbf{P}_1^{**},\mathbf{P}_2^{**}) = \sum_{i=1}^{\mathrm{r}}\sum_{j=1}^{\mathrm{N}} ||\mathbf{v}_{avg} \cdot \mathbf{v}_{meas}(\mathbf{T}_{0,j,i}^{\mathbf{I}}, \mathbf{T}_{\mathbf{I}}^{\mathbf{P}_1}, \mathbf{T}_{\mathbf{I}}^{\mathbf{P}_2})||. \tag{4.8}$$

4.2.2 Experimental validation

The evaluation of the proposed measurement procedure is conducted in three phases. First, the calibration method is evaluated in the simulation setup against ground-truth. Second, a real-world experiment is conducted, where known distances (using millimeter paper) are measured with a 3D printed point-instrument. Finally, programming-by-demonstration experiments are conducted, in which the method is evaluated in a typical deployment setup.

4.2.2.1 Calibration results

Point-shaped virtual pen

The results of the conducted point-shaped pen optimization process are shown in Fig. 4.8

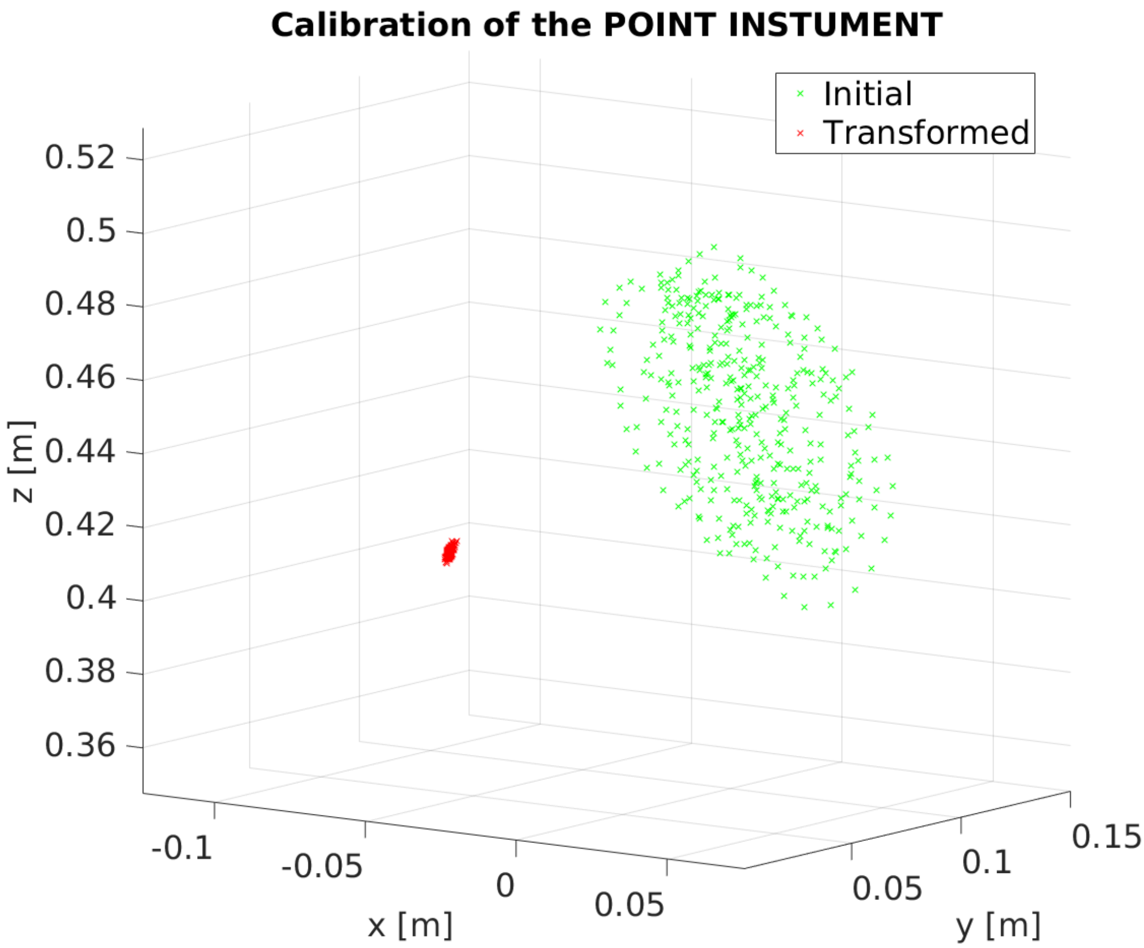


Figure 4.8: Figure shows result of T_I^P optimisation process. The recorded points of the centroid T_0^I are presented in green, while final transformation of the instrument tip T_0^P is presented in red.

.

and Tab. 4.2 as standard deviation, RMS error and min./max. error of euclidean distances of transformed points to the mean value of the dataset. The results are shown for both simulation dataset, and real measured data. The simulation dataset is noise-free, as a proof of the proposed method. Consequently, the results of the procedure are several orders of magnitude more precise than the real experiment.

Tab. 4.3 and Tab. 4.4 show calibration results for the orientation of the point-shaped virtual pen on the simulated data and real OptiTrack data, respectively. The simulated data is, again, noise-free. The tables show the results before and after the additional rotation calibration. The results show that compensation for the orientation error was achieved, and it was observed that this error remained constant, as evidenced by the negligible change in standard deviation before and after compensation.

Line-shaped virtual pen

The line-shaped virtual pen is calibrated in two phases. After the first phase of the optimization process, transformed points and lines of the dataset are shown in Fig. 4.9a. Here it

Table 4.2: Statistical analysis of euclidean distances between the optimization based transformed points and the mean value of the entire dataset.

	Measured data	Simulation data
std. deviation [mm]	0.7908	$7.5585 \cdot 10^{-5}$
RMS error [mm]	1.2718	$1.6788 \cdot 10^{-4}$
min. error [mm]	1.0276	$3.1273 \cdot 10^{-5}$
max. error [mm]	95.8326	$4.6296 \cdot 10^{-4}$

Table 4.3: Statistical analysis of the point-shaped virtual pen orientation optimization showing absolute angle error between the transformed orientation vectors and the ground-truth orientation vector collected using Blender simulation.

	before	after
mean [°]	30.6348	$0.1509 \cdot 10^{-4}$
std. deviation [°]	$0.1132 \cdot 10^{-4}$	$0.2099 \cdot 10^{-4}$
RMS error [°]	30.6348	$0.2581 \cdot 10^{-4}$
min. error [°]	30.6348	$0.1207 \cdot 10^{-5}$
max. error [°]	30.6348	$0.7268 \cdot 10^{-4}$

Table 4.4: Results of the point-shaped virtual pen orientation optimization, shown as statistical analysis of the absolute angle error between the transformed orientation vector and the mean orientation vector of the whole dataset. The dataset is collected using the Optitrack system.

	before	after
mean [°]	90.2101	1.0867
std. deviation [°]	0.8399	0.5593
RMS error [°]	90.2139	1.2215
min. error [°]	88.8771	0.2413
max. error [°]	92.4415	2.9797

Table 4.5: Statistical analysis of the angle errors between the transformed vectors that the master-slave pins span and the mean vector value of the dataset, after the first and the second phase of the optimization process.

	1. phase	2. phase
mean [°]	16.2944	0.5070
std. deviation [°]	0.7576	0.5364
RMS error [°]	16.3120	0.7380
min. error [°]	14.8107	0.0077
max. error [°]	20.4430	4.2807

is clearly shown that transformation of the points (master and slave pin) are not that far away (up to 5 mm) from the ideal line, but on the other side it is clear that direction of measured line (between master and slave pin) varies a lot from ideal line. Since the main idea of the line-instrument is to place reference line in the space which will be used for further distance measurement using point-instrument, it is extremely important to have accurate line direction measurement.

The result of the second phase of the line-instrument optimization is shown in Fig. 4.9b. As can be read in Tab. 4.5, the second phase of the optimization has significantly improved the average angle errors for measured lines w.r.t. the benchmark (average line).

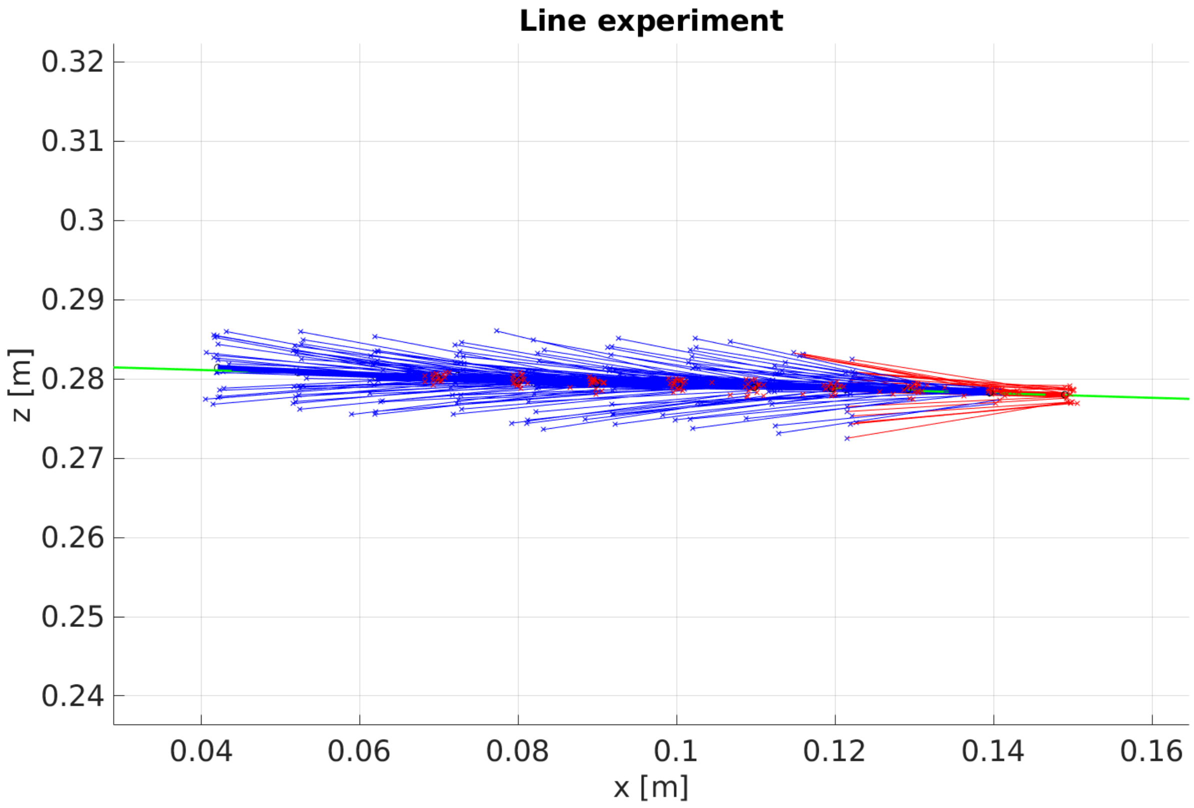
4.2.2.2 Optitrack vs. Robot measurement comparison

In order to evaluate the proposed measurement system, a similar setup for point measurement using the industrial robot *KUKA KR10* was designed. Once the robot point-tip tool is calibrated, and the robot base frame is localised in the *OptiTrack* base frame, the robot position measurements can be compared to measurements obtained manually in the *OptiTrack* system.

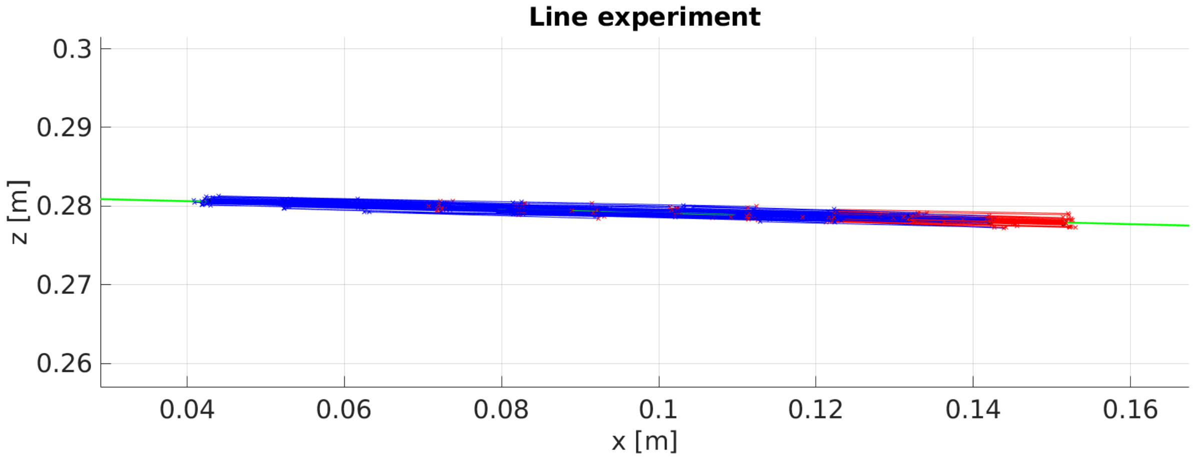
The robot is equipped with a force sensor and measurement tool similar to the point-instrument, Fig. 4.10. In order to calibrate the robot point-tip, the robot is manually guided as in the collaborative compliant control scheme. This control framework is developed in our previous work [50].

The calibration of the robot tool is conducted similarly to the calibration of the point-shaped pen. The robot flange is placed in different orientation to the calibration tool, while the measurement tool's tip was fixed in the same global point. Using the global Cartesian position of the robot flange $\mathbf{T}_{\mathrm{K}}^{\mathrm{F}}$ (from the specified kinematics of the robot and measured joint positions), the robot point-tip can be calibrated as described earlier, only using robot flange position instead of OptiTrack markers' positions. The final position of the measured point is: $\mathbf{T}_{\mathrm{K}}^{\mathrm{P}} = \mathbf{T}_{\mathrm{K}}^{\mathrm{F}} \cdot \mathbf{T}_{\mathrm{F}}^{\mathrm{P}}$.

Other than robot point-tip calibration, the robot should be localised within the *OptiTrack* frame. The transformation of the point measured in *OptiTrack* frame to robot-base frame is given by: $\mathbf{T}_{\mathrm{K}}^{\mathrm{P}} = \mathbf{T}_{\mathrm{K}}^{\mathrm{O}} \cdot \mathbf{T}_{\mathrm{O}}^{\mathrm{P}}$. The unknown transformation $\mathbf{T}_{\mathrm{K}}^{\mathrm{O}}$ is found in a three point optimization



(a) Transformation after first position of the optimization



(b) Transformation after second phase of the optimization

Figure 4.9: Showing transformed lines in the line-instrument optimization process. The red lines are measurements at the first position of the calibration tool. The green line is the average line through the whole dataset. The first figure shows dissipation after 1st phase, with average angle error of 15°. The second figure shows result of the 2nd phase, with dissipation reduced to 0.5°.

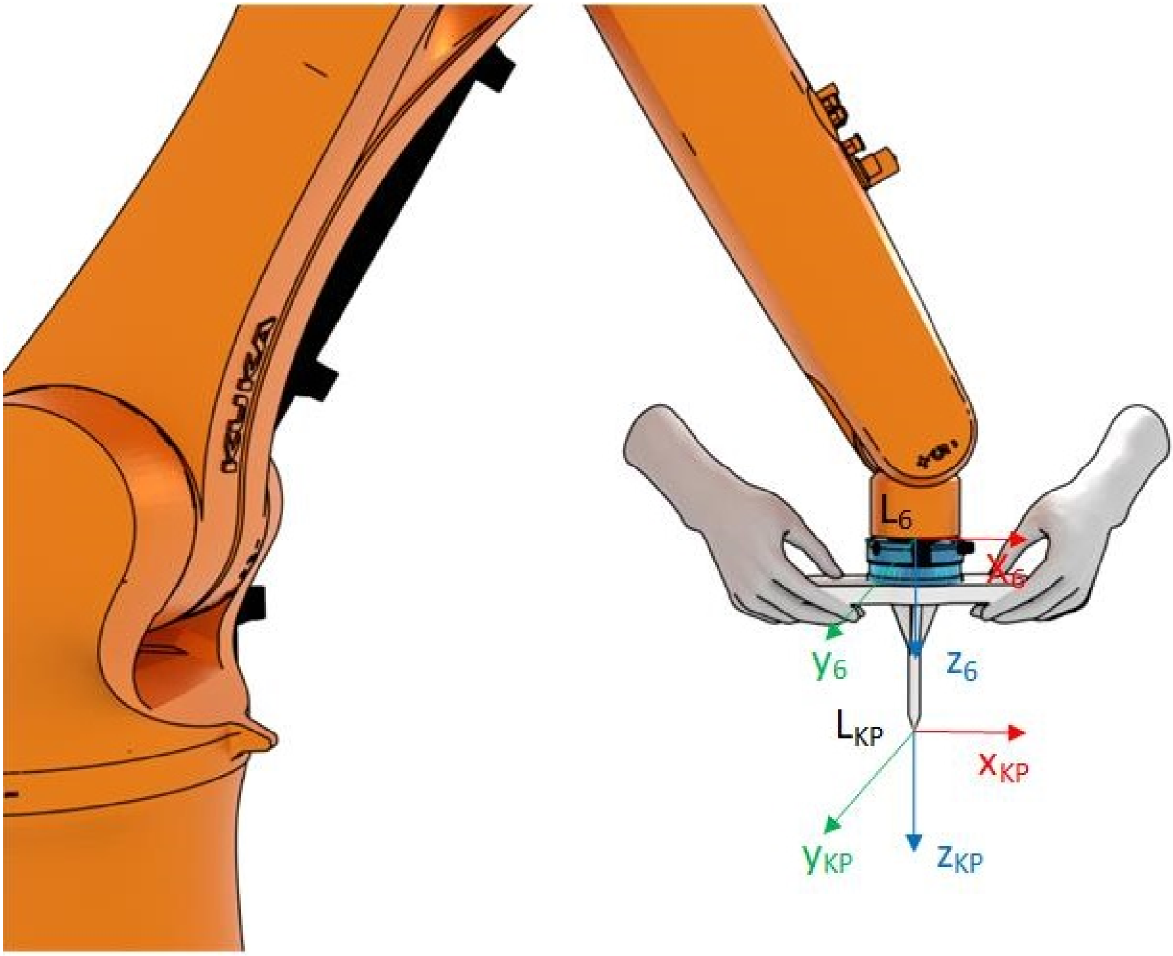


Figure 4.10: Showing KUKA KR10 robot equipped with point-tip for point measurement. The robot has collaborative behaviour due to the installed force sensor between the pin tool and robot flange, and collaborative force control system. The operator takes pin tool holder and moves robot towards point of the interest. After positioning measurement pin, robot pose is captured and used for global position measurement.

process. The three points are measured with both manual point-instrument and robot point-instrument. The transformation is optimized as before so that the transformed points align in the robot frame. The robot setup calibrated this way is used for measurement accuracy and repeatability testing.

4.2.2.3 Measurement accuracy and repeatability

In order to evaluate the measurement accuracy and repeatability, five points were marked on millimeter paper. Each point was measured ten times, with varying orientation of the measurement tool. The measurement procedure is conducted for both point-shaped pen and *KUKA* robot. The results of conducted experiments are presented as statistical analysis of the measurement error, where error is defined as euclidean norm distance of each measurement to the averaged measurement.

The Tab. 4.6 presents results of the point-instrument measurements, while Tab. 4.7 presents

Table 4.6: Statistical analysis of the conducted experiment using point-instrument. Five points were measured on millimeter paper in ten iterations. The error values are calculated as a euclidean norm distance of each measurement to the mean over measurements.

error	P_1	P_2	P_3	P_4	P_5
mean [mm]	0.2879	0.2392	0.3137	0.4467	0.5448
std. dev. [mm]	0.1841	0.1209	0.1832	0.2574	0.2988
RMS [mm]	0.3417	0.2680	0.3632	0.5155	0.6213
min. [mm]	0.0052	0.0173	0.0177	0.0236	0.0213
max. [mm]	1.2162	0.6583	1.1953	1.6967	3.6801

Table 4.7: Statistical analysis of the conducted experiment using *KUKA* robot. Five points were measured on millimeter paper in ten iterations. The error values are calculated as a euclidean norm distance of each measurement to the mean over measurements.

error	P_1	P_2	P_3	P_4	P_5
mean [mm]	0.0440	0.1112	0.0550	0.3102	0.1988
std. dev. [mm]	0.0235	0.0824	0.0296	0.1386	0.1071
RMS [mm]	0.0499	0.1382	0.0624	0.3396	0.2257
min. [mm]	0.0036	0.0194	0.0084	0.0157	0.0355
max. [mm]	0.1146	0.4169	0.1325	0.5294	0.4291

results of robot measurements. The lower standard deviation of the error values for the robot measurements imply higher repeatability when compared to point-instrument measurements. When it comes to the accuracy, point-instrument measurements were compared to the average point measurements of the robot and presented in Tab. 4.8. The results show an accuracy error up to $2.5mm$. One source of this error is in the localization error of the robot and *Optitrack* reference frames. In order to reduce the error, more points widely distributed within working area should be measured and taken into calibration procedure.

4.2.3 Deploying robot systems using Virtual pen

Deploying robots in any setting requires one to create a virtual representation of the robot's workspace. This virtual representation includes the limits of the workspace, different obstacles, and of course the targets. In this section, several procedures are described that can be followed

Table 4.8: Accuracy analysis of the proposed measurement system. The error values are calculated as a euclidean norm distance of point-instrument measurement to the mean position measurement of the *KUKA* robot.

error	P_1	P_2	P_3	P_4	P_5
mean [mm]	2.4910	2.0221	2.4134	1.7904	1.6583
std. dev. [mm]	0.0426	0.1332	0.2366	0.2416	0.2080
RMS [mm]	2.4914	2.0265	2.4249	1.8066	1.6713
min. [mm]	2.3988	1.6961	1.7833	1.2566	1.2693
max. [mm]	2.7400	2.4624	3.3999	2.7096	4.2852

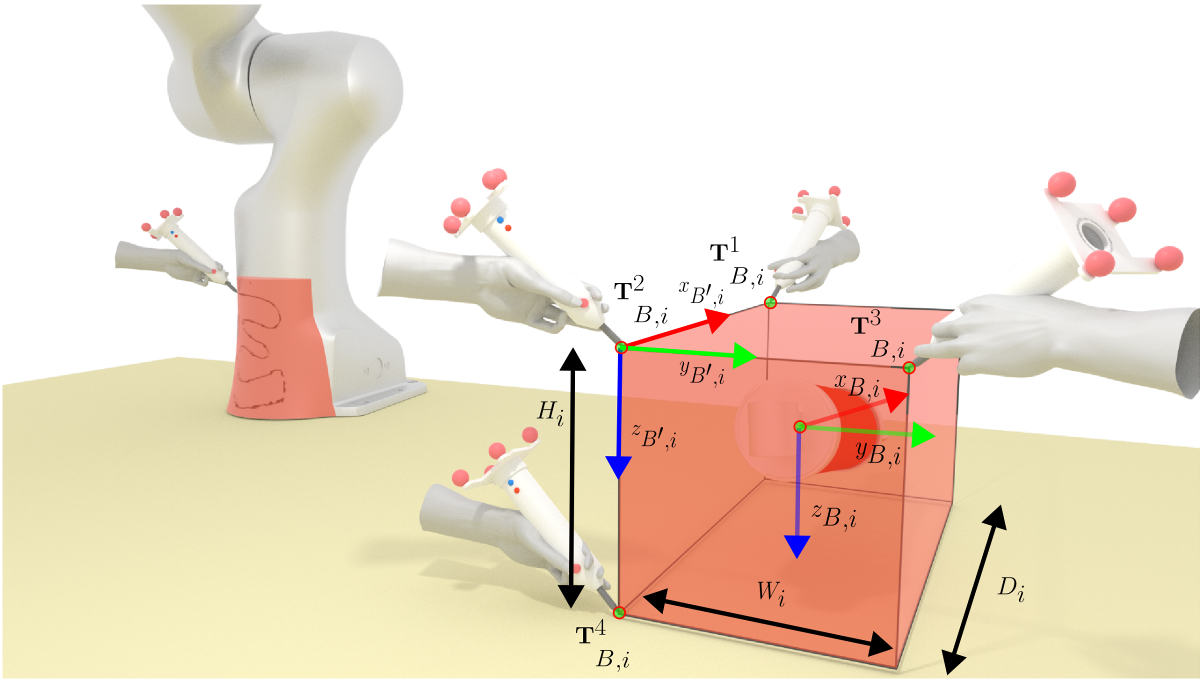


Figure 4.11: The figure illustrates two possible applications of the virtual pen. The first is the calibration of the robot's base-link to the motion capture system, achieved through the known 3D model of the robot base and pointcloud obtained by drawing along its surface with the pen. The second application involves defining the collision objects in motion planning scene.

to sketch the robots' workspace, eliminating the need to tediously measure and model their surroundings. A schematic representation of the capturing process is depicted in Fig. 4.11. The versatility and practicality of the virtual pen are highlighted, particularly in accomplishing precise calibration and enhancing safety in robot manipulation. This encompasses the annotation of trajectory waypoints for execution, marking the obstacles as bounding boxes, and achieving precise localization of objects using known 3D models.

4.2.3.1 Point(s) measurement

The utilization of the virtual pen relies on continuous OptiTrack tracking of the virtual pen's pose in 3D space. This, in conjunction with the calibration matrix $\mathbf{T}_I^P$, ensures the accurate capture of points in Cartesian 3D space, defined by the following equation:

$$\mathbf{p} = [x \quad y \quad z \quad q_x \quad q_y \quad q_z \quad q_w] \in \mathbb{R}^{1\times 7}, \tag{4.9}$$

where the virtual pen tip position is denoted as $\mathbf{p}_p = [x \quad y \quad z] \in \mathbb{R}^{1\times 3}$, while the orientation of the pen tip is expressed in quaternion form as $\mathbf{p}_q = [q_x \quad q_y \quad q_z \quad q_w] \in \mathbb{R}^{1\times 4}$. This functionality empowers the user to showcase and mark waypoints easily, moving the virtual pen freely and concurrently collecting data points.

4.2.3.2 Marking obstacles

To virtualize a priori unknown obstacles or define the boundaries of the robot's workspace, drawing on obstacles surrounding the robot is proposed. These obstacles are then virtually represented with the boxes. The point cloud collection is initiated by clicking the button on the calibration pen. Upon releasing the button, these points are averaged to calculate the centroid $\hat{\mathbf{p}}_\mathbf{i}$ of each point. To establish the boundaries of the obstacle, two approaches can be employed. The first approach involves selecting three points in a counter-clockwise direction around one side of the box, with an additional point representing the height of the box. In situations where capturing the fourth point is not feasible, as alternative the user can manually specify the desired height h in the normal direction to the plane defined by the captured points.

To ensure compatibility with the *Moveit* motion planning environment, the use of the bounding box primitive is adopted. This primitive is defined within the planning scene by specifying the coordinate frame of the box center $T^{O}_{B\ ,i}$ and the three dimensions of the box: D_i, W_i, and H_i. In the following steps, the calculation of the pose of the bounding box and its dimensions from the measured points will be outlined.

The dimensions of the box i, namely the width W_i and depth D_i, can be calculated using the following equations:

$$W_i = ||\mathbf{p}^{P3}_{B\ ,i} - \mathbf{p}^{P2}_{B\ ,i}||, \tag{4.10}$$

$$D_i = ||\mathbf{p}^{P1}_{B\ ,i} - \mathbf{p}^{P2}_{B\ ,i}||, \tag{4.11}$$

where the $\mathbf{p}^{Px}_{B\ ,i} \in \mathbb{R}^{3\times 1}$ are translation parts of measured points. If the fourth point measurement is obtainable, the height of the box H_i can be calculated using the equation:

$$H_i = ||\mathbf{p}^{P4}_{B\ ,i} - \mathbf{p}^{P2}_{B\ ,i}||. \tag{4.12}$$

If the fourth point measurement is not available, the height H_i can be defined programmatically. In the next step, a temporary coordinate frame for the bounding box is created, denoted as $L^{B'}_{0\ ,i}$, with its origin placed at point $P2$. The transformation matrix of this coordinate frame $\mathbf{T}^{B'}_{0}$ is derived from the vectors $\mathbf{x}_{B',i}$, $\mathbf{y}_{B',i}$, and $\mathbf{z}_{B',i}$, as well as the position vector of the point $\mathbf{p}^{P2}_{B\ ,i}$. First the vector $\mathbf{x}_{B',i}$ is calculated using the following equation:

$$\mathbf{x}_{B',i} = \frac{\mathbf{p}^{P1}_{B\ ,i} - \mathbf{p}^{P2}_{B\ ,i}}{||\mathbf{p}^{P1}_{B\ ,i} - \mathbf{p}^{P2}_{B\ ,i}||}. \tag{4.13}$$

Since it is unrealistic to assume that the measured points $\mathbf{p}^{P1}_{B\ ,i}$, $\mathbf{p}^{P2}_{B\ ,i}$, and $\mathbf{p}^{P3}_{B\ ,i}$ form a perfect right triangle from manual annotation, additional step has been taken to form an orthonormal coordinate frame for the box. To address this a temporary vector $\mathbf{y}^{'}_{B',i}$ is calculated using the equation:

$$\mathbf{y}^{'}_{B',i} = \frac{\mathbf{p}^{P3}_{B\ ,i} - \mathbf{p}^{P2}_{B\ ,i}}{\|\mathbf{p}^{P3}_{B\ ,i} - \mathbf{p}^{P2}_{B\ ,i}\|}. \tag{4.14}$$

The vector $\mathbf{z}_{B',i}$ can be obtained by taking the cross product of $\mathbf{x}_{B',i}$ and temporary vector $\mathbf{y}^{'}_{B',i}$. Similarly, the vector $\mathbf{y}_{B',i}$ is obtained by taking the cross product of the vector $\mathbf{z}_{B',i}$ and $\mathbf{x}_{B',i}$. The temporary coordinate frame $L^{B'}_{0\ ,i}$, formed by vectors $\mathbf{x}_{B',i}$, $\mathbf{y}_{B',i}$, and $\mathbf{z}_{B',i}$ in point $P2$ is then transformed to the center of the bounding box, resulting in the box frame $L^{B}_{0\ ,i}$:

$$\mathbf{T}^{B}_{0\ ,i} = \mathbf{T}^{B'}_{0\ ,i} \cdot \begin{bmatrix} & D_i/2 \\ \mathbf{1} & W_i/2 \\ & H_i/2 \\ 0 & 1 \end{bmatrix}. \tag{4.15}$$

Once the coordinate frame of the box and its dimensions are calculated, the bounding box can be added to the planning scene using the function *addBox*$((\mathbf{p}_i, \mathbf{q}_i), (W_i \cdot s, D_i \cdot s, H_i \cdot s))$. Here, $\mathbf{p}_i$ represents the position derived from the transformation matrix $\mathbf{T}^{B}_{0\ ,i}$, while $\mathbf{q}_i$ is orientation of the box represented in the quaternion form. The dimensions of the box, denoted as $(W_i \cdot s, D_i \cdot s, H_i \cdot s)$, are the calculated width, depth, and height of the box, respectively. Here, $s > 1$ is safety scaling factor, that is used to slightly enlarge the dimensions of the bounding box. By adding the box to the planning scene, it becomes a part of the robot's environment and can be used for collision free motion planning.

4.2.3.3 Localization of objects using known 3D models

Furthermore, the Virtual pen enables a novel approach to user-friendly object localization, allowing operators to easily locate objects by drawing shapes on their surfaces. An illustrative example of drawing a shape on the object surface is depicted in Fig. 4.12. Essentially, the Virtual pen is utilized as a portable scanning device, generating a point cloud $\mathscr{P}_c$ comprised of pen tip points $\mathbf{p} \in \mathscr{P}_c$ as described in Eq. 4.9. Subsequently, a versatile localization algorithm is presented in the following paragraph, enabling various localization and calibration functionalities.

Localization algorithm

The object needs to be a priori known, which makes this approach good for localizing specific objects in the robot workspace that are accurately modeled or robot parts (i.e. known 3D model of the robot base, links, etc). The acquisition agnostic algorithm is presented in Fig. 4.13, which means the the algorithm can be utilized with point clouds obtained not only with

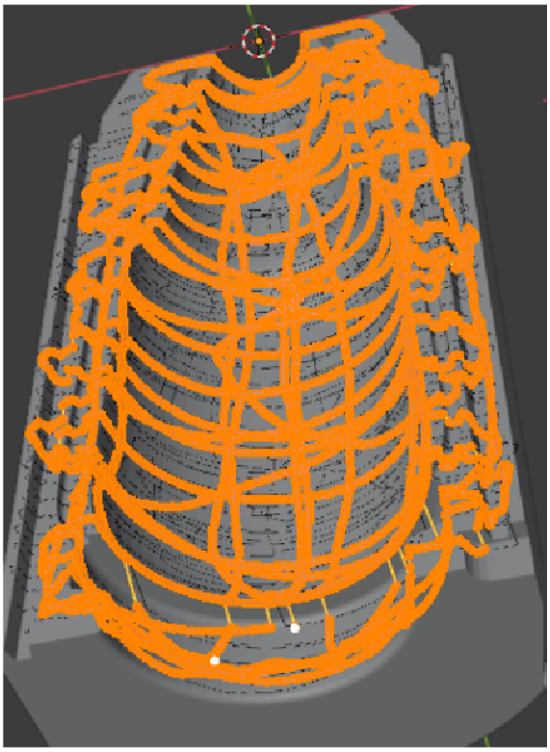


Figure 4.12: Drawing curves on the surface of the model.

the virtual pen, but also with other sources like RGB-D cameras, lasers and lidars etc.

Before the matching process, the object scan undergoes several preprocessing steps, including statistical outlier removal, voxelization, and uniform downsampling, culminating in the calculation of surface normals. Concurrently, a voxelized CAD model of a known object is converted into a point cloud. In the subsequent step, global registration is employed, involving feature calculation based on Fast Point Feature Histograms (FPFH) [51] or the 3D Harris approach to establish correspondences between the point clouds of the model and the scan. To determine the initial transformation between the two clouds, the Sample Consensus Prerejective (SCP) method is iteratively applied until a predefined fit threshold is achieved [52]. Finally, the transformation is further refined using the Iterative Closest Point (ICP) algorithm [53], resulting in a transformation matrix that transforms the scanned point cloud to the known model.

4.3 PbD paradigm: User experience survey

In the preceding sections, two approaches to the learning phase of the Programming by Demonstration (PbD) paradigm [54] are outlined. The kinesthetic approach, discussed in Sec. 4.1, involves the operator guiding the robot's flange, while a novel human-robot interface based on tool motion tracking and the virtual pen device is presented in Sec. 4.2. Despite the widespread popularity of the kinesthetic approach, this section argues that its usage is limited, particularly in tasks that demand highly skilled demonstration. Essentially, this section conducts a user ex-

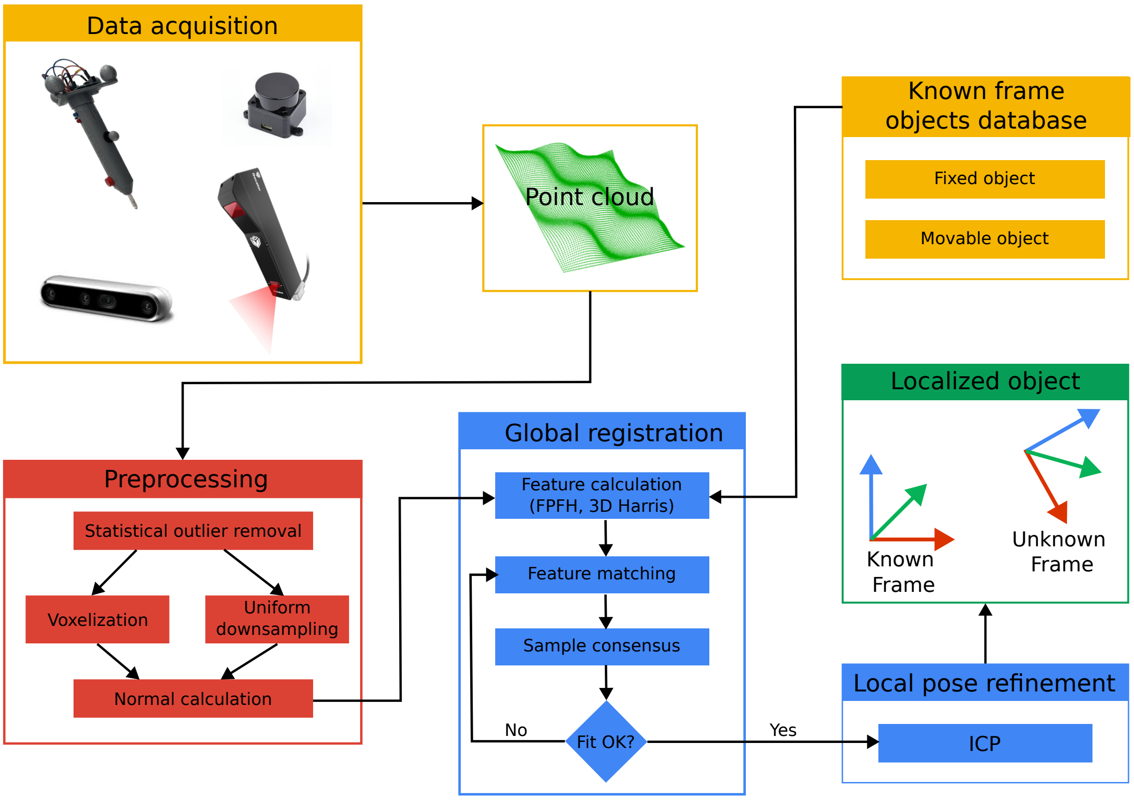


Figure 4.13: The figure illustrates acquisition agnostic algorithm used for different localization and calibration procedures, built using the point-cloud measurements, preprocessing method and algorithms for globally and locally optimal registration.

perience survey on the PbD paradigm, comparing the two approaches. The main hypothesis of the survey is that the Virtual pen is a user-intuitive and friendly approach that enables the system to capture the pure essence of the demonstration, while contrary to that, the kinesthetic approach pushes the operators outside their comfort zone, resulting in highly biased demonstrations. The hypothesis is examined through a drawing task that requires skill possessed by everyone.

4.3.1 Experiment Methodology

A user study was conducted to assess the ergonomics and user-friendliness of the virtual pen in PbD tasks. To ensure a broad survey that encompasses a skill possessed by everyone, participants were tasked with demonstrating the drawing of a simple shape on a whiteboard. Demonstration was performed both with the virtual pen and by guiding the flange of the collaborative manipulator *Franka Emika Panda* in the usual collaborative manner. Our primary objective was to validate our claims regarding the potential of the new Human-Robot-Interface (HRI) to enhance the adoption of robot manipulators, even among unskilled labor.

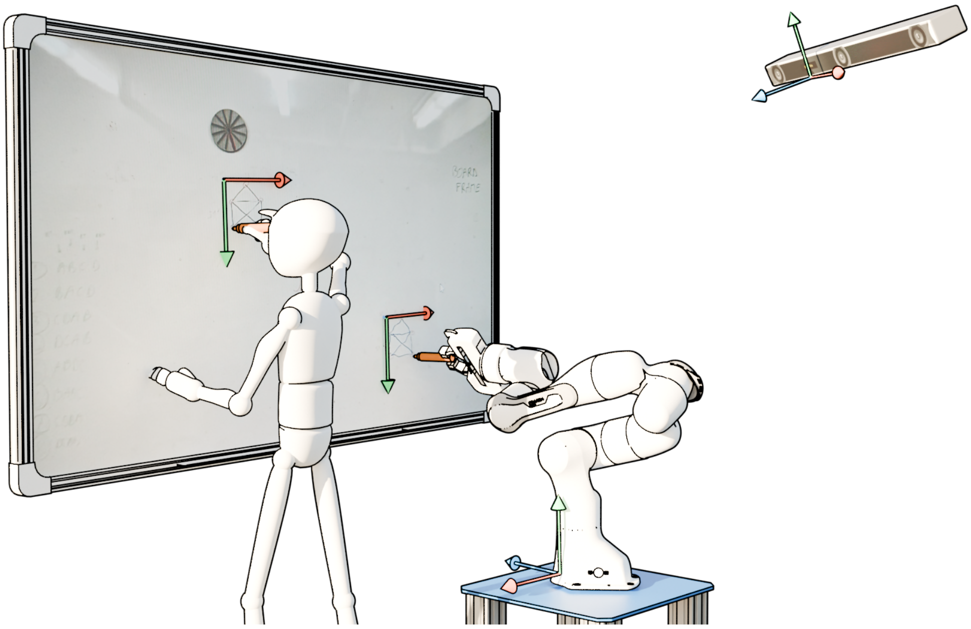


Figure 4.14: The experimental setup for the user-experience study comprised a motion capture system, specially developed virtual markers, and the collaborative robot *Franka Emika Panda*. Participants were tasked with drawing simple images using both the virtual marker and the robot. Subsequently, user-experience, task load, and task evaluation were assessed through a questionnaire and task analysis.

4.3.1.1 Hardware setup

In the experimental setup, depicted in Fig. 4.14, the virtual pen tool was used alongside the *Optitrack* mocap system, a collaborative manipulator *Franka Emika Panda*, a force/torque sensor, and a whiteboard. Recognizing the significance of the contact force in the demonstration, the force-torque sensor was strategically positioned beneath the whiteboard. This setup allowed concurrent tracking of the contact force along with tracing the virtual marker. Two marker versions were used: one that left a trace on the whiteboard, and the other one that did not. Additionally, a marker holder was designed and mounted on the robot flange. The holder was designed for tight positioning of the whiteboard marker while allowing for easy marker replacement. Both marker versions had the same physical dimensions, differing only in their tracing capability.

Given that the Optitrack motion capture system tracks the global pose $\mathbf{T}_W^I$ of the centroid of the fiducials placed on top of the virtual pen in world frame, precise calibration of the pen tip becomes crucial. As detailed in Sec. 4.2.1.1, the calibration process involves deriving the relative transformation between the virtual pen tip and the Optitrack marker, represented as $\mathbf{T}_I^P$. Building upon that, the calculation of the global pose of the virtual pen tip $\mathbf{T}_W^P$ is obtained as:

$$\mathbf{T}_W^P = \mathbf{T}_W^I \cdot \mathbf{T}_I^P. \tag{4.16}$$

The calibration results in two homogeneous transformation matrices: $\mathbf{T}_{O_k}^{P_t}$ and $\mathbf{T}_{O_k}^{P_n}$, for virtual pen that leaves and does not leave trace respectively. A similar calibration procedure was followed for the robot flange to tool transformation $\mathbf{T}_F^P$, where the difference lies in using the flange position derived from the robot's Forward Kinematics instead of the mocap system measurement $\mathbf{T}_W^I$.

4.3.1.2 Drawing frames

Drawing frames are established to standardize user drawings created with both the virtual marker and the robot, thus facilitating uniform post-analysis. These frames are selected and marked on the whiteboard, and a 3D printed template is aligned within the frame to mark five waypoints for the drawing. This procedure ensures consistency and comparability across all participant drawings. The virtual marker drawing frame is labeled as L_{F_k}, while the robot drawing frame is denoted as L_{F_r}.

The drawing frames were identified using the calibrated tool tip, encompassing both the virtual pen tip $\mathbf{T}_W^P$ and the robot tip $\mathbf{T}_B^P$. To elucidate the methodology, let's denote the pose of the tool tip, whether it's the pen or robot tip, as $\mathbf{T}_{O_x}^P$, its translational component as $\mathbf{p}_{O_x}^P$, and the drawing frame as L_{F_x}. The drawing frame L_{F_x} was localized using three points to form a freely chosen right triangle, marked in the following sequence: 1) $\mathbf{T}_{F_x}^{P_1}$ on the $+x$ axis; 2) $\mathbf{T}_{F_x}^{P_2}$ at the origin; and 3) $\mathbf{T}_{F_x}^{P_3}$ on the $+y$ axis. Following this, coordinate frame unit vectors are derived using:

$$\mathbf{x}_{O_x}^{F_x} = \frac{\mathbf{p}_{O_x}^{P_1} - \mathbf{p}_{O_x}^{P_2}}{||\mathbf{p}_{O_x}^{P_1} - \mathbf{p}_{O_x}^{P_2}||}, \tag{4.17}$$

$$\mathbf{y}_{O_x}^{F_x} = \frac{\mathbf{p}_{O_x}^{P_3} - \mathbf{p}_{O_x}^{P_2}}{||\mathbf{p}_{O_x}^{P_3} - \mathbf{p}_{O_x}^{P_2}||}. \tag{4.18}$$

Having calculated unit vectors $\mathbf{x}_{O_x}^{F_x} \in \mathbb{R}^{3\times1}$ and $\mathbf{y}_{O_x}^{F_x} \in \mathbb{R}^{3\times1}$, remaining one is calculated as: $\mathbf{z}_{O_x}^{F_x} = \mathbf{x}_{O_x}^{F_x} \times \mathbf{y}_{O_x}^{F_x}$. Following that, the matrix of homogeneous transformation of the frame $T_{O_x}^{F_x}$ is derived as follows:

$$T_{O_x}^{F_x} = \begin{bmatrix} \mathbf{x}_{O_x}^{F_x} & \mathbf{y}_{O_x}^{F_x} & \mathbf{z}_{O_x}^{F_x} & \mathbf{p}_{O_x}^{P_2} \\ & \mathbf{0} & & 1 \end{bmatrix} \tag{4.19}$$

where $\mathbf{p}_{O_x}^{P_2} \in \mathbb{R}^{3\times1}$ is translational component of the measured point $\mathbf{T}_{0_x}^{P_2}$.

Once the drawing frame is localized, the drawing experiment can commence. Considering the tool tip pose is tracked in the global frame of the mocap system or the robot base link, generally expressed as $\mathbf{T}_{O_x,i}^P$. To transform the obtained points $\mathbf{T}_{O_x,i}^P$ into the drawing frame, the following transformation is applied:

$$\mathbf{T}_{F_x,i} = \mathbf{T}_{O_x}^{F_x\,-1} \cdot \mathbf{T}_{O_x,i}^P. \tag{4.20}$$

4.3.1.3 Task and Population samples

To show difference between different ways of demonstrating robot to execute certain task, human subjects were divided in 8 different groups to mitigate influence of knowing tasks beforehand. To each study participant video was shown to demonstrate how to hold pen, and how to guide robot manipulator (RM). Each user had to complete two different tasks. One task was demonstrating with a pen on whiteboard. Another task was using robot manipulator to demonstrate same pattern on a whiteboard. For each task user used pen that leaves trace and pen that leaves no trace.

There was 24 study participants. Participants ranged in age from 22 - 28. Gender information is omitted because it doesn't affect study outcome. Participants were divided into 8 groups, and each group had three participants that executed tasks in different ordering (ABCD, ABDC, BACD, BADC, CDAB, DCAB, CDBA, DCBA). Study limitation arises from the small diversity of the study population. It is not known to which extent age and gender affect subjective workload felt. In this case, small age diversity in the study population contributes to the ability to easily compare workload imposed by different PbD techniques, rather than exploring age effect on the felt workload.

4.3.2 Experimental results

Each study participant had to write with the pen that leaves trace on blackboard (A) and that leaves no trace on the blackboard (B). Each participant also had to force-guide RM that leaves trace (C), and RM that leaves no trace (D). Therefore, every participant executed 4 different tasks in certain ordering (e.g. ABCD), and after each task, participant was prompted to fill out NASA raw Task Load Index (rTLX) [55], [56] and the system usability scale (SUS) [57] test. The NASA raw Task Load Index (rTLX) is measurement developed by NASA to assess operator's workload when instructing or operating a machine. A NASA raw task load index [58] survey was employed, where each participant had to estimate each of the following workload categories: physical demand (PD), mental demand (MD), frustration (F), performance (P), temporal demand (TD) and effort (E). In addition, the usability of the system was examined through the system usability scale (SUS) test. In this work, main focus was kept on task evaluation, as explained in Subsec. 4.3.2.1.

4.3.2.1 Task evaluation

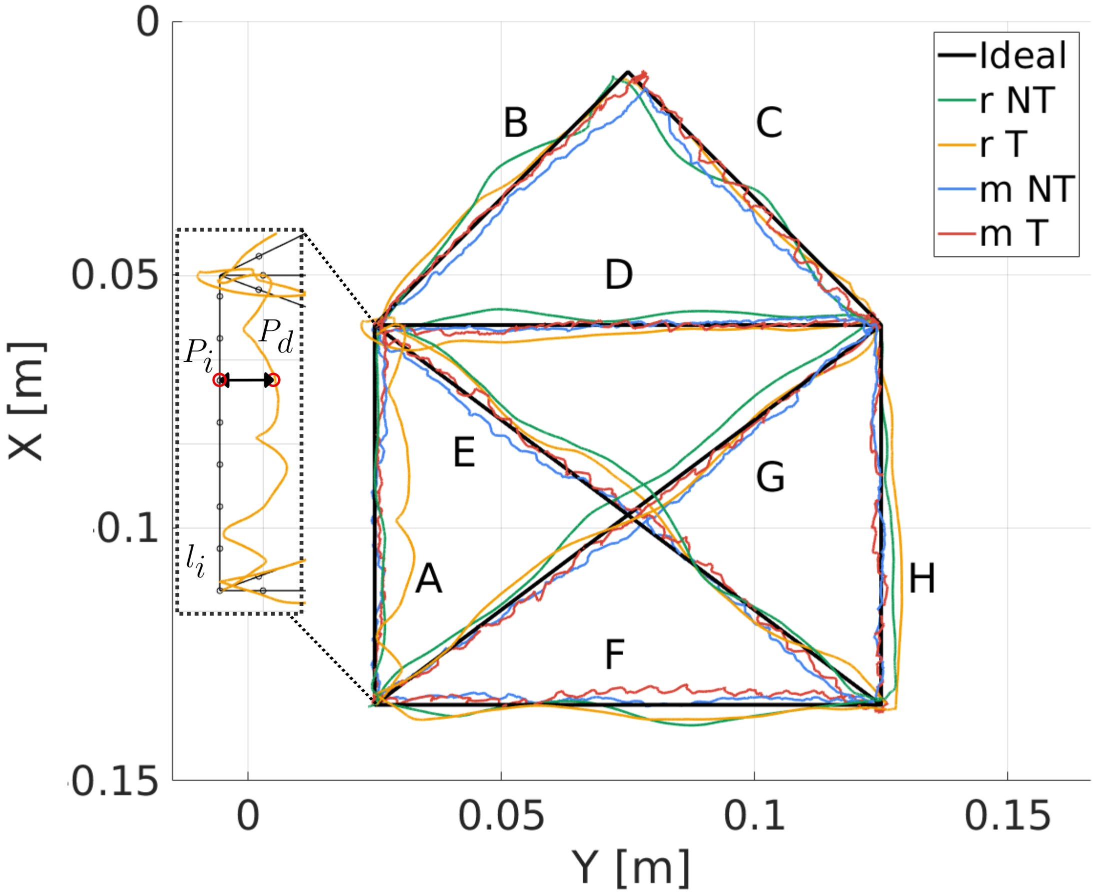
Given that the task involved connecting marked points with straight lines, the aim was to evaluate the deviation of demonstrated trajectories from the ideal lines connecting these marked waypoints. Initially, all recodings were transformed in it's drawing frame, using the Eq. 4.20 and obtained frame transformations L_{Fk} and L_{Fr}. Then all displayed trajectories were segmented into eight sections. Given the variations in demonstration speeds, leading to differing numbers of points in the demonstrated trajectories, a prerequisite for conducting any analysis on these points was the resampling of demonstrated trajectories. This approach ensures an equal number of trajectory points across participants and drawing cases.

In the resampling process, the ideal segment line l_i was uniformly divided into a consistent number of points per segment. Each point P_i on l_i was then paired with a corresponding point P_d on the demonstrated trajectory, positioned orthogonally to l_i and P_i. The orthogonal projection of P_d to l_i represents the distance error from the demonstrated to the ideal trajectory. This process was conducted for each participant and drawing case, including pairing contact forces exerted between the marker and the whiteboard. Fig. 4.15 illustrates the prepared trajectories for a single participant, showing ideal and demonstrated trajectories (robot and marker), distance errors, and contact forces on the whiteboard.

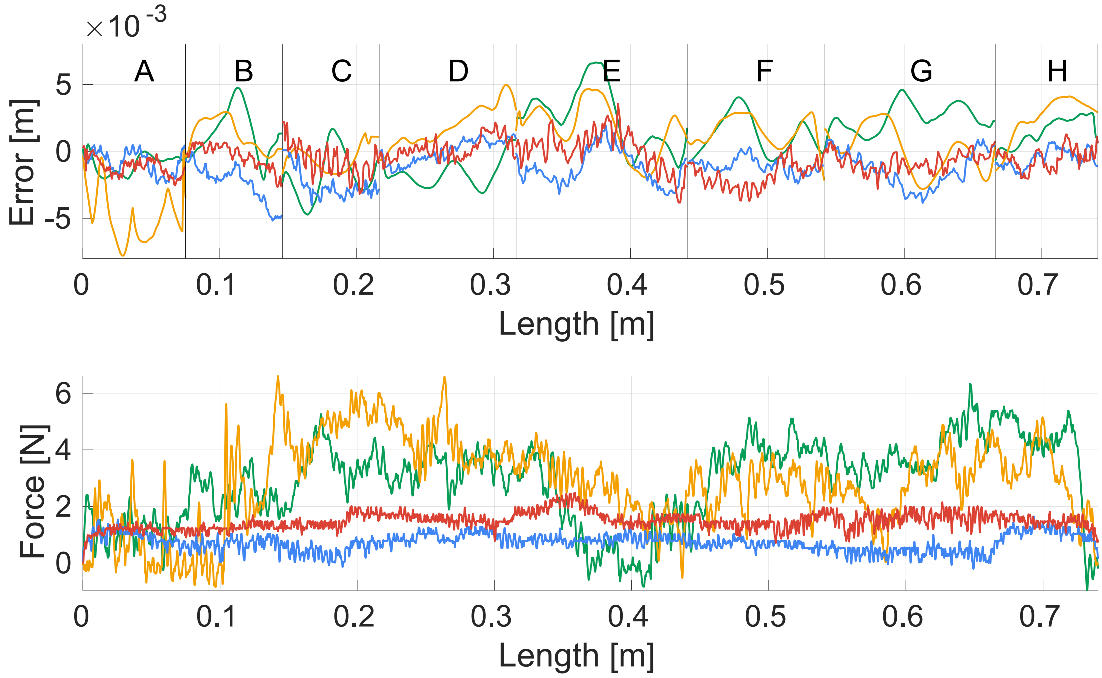
In order to overlap the trajectories of all participants for each test case, an envelope encompassing all trajectories per segments was derived, as illustrated in Fig. 4.16. This depiction highlights that the robot's demonstration spans a broader area compared to the ideal waypoints. Additionally, the mean trajectory was computed, obtained from the distance errors from the demonstrated to the ideal trajectory. Alongside the mean trajectory, the standard deviation is presented, representing the dispersion around the mean trajectory.

To substantiate that trajectories demonstrated with the virtual pen are more compact, closely aligned with the ideal lines, and exhibit a smaller dissipation rate, additional metrics was employed. The histogram depicted in Fig. 4.17 illustrates the groups of distances of the trajectory points from the ideal line, with the corresponding count of such points. Additionally, the *Epsilon* area in blue is highlighted, representing a narrow 3 mm wide zone around the ideal lines, where the majority of trajectory points are anticipated to fall. It is evident that for virtual pen demonstrations, a greater number of trajectory points lie within the *Epsilon* area.

Returning to the contact force F_z during the demonstration, as depicted in Fig. 4.15b, two observations stand out. Firstly, the force amplitude is higher in the robot demonstration, and secondly, the force signal is significantly more variable in such cases. In contrast, the contact force signal in virtual marker demonstrations has a lower amplitude but is more consistent. The variation in the force signal is particularly interesting in our experiment. To investigate further, Fast Fourier Transformation (FFT) was conducted on the force signal F_z for each demonstration. In Fig. 4.18, the results of the FFT are presented in the form of a histogram. The y-axis represents the count of frequencies found in the force signal, with amplitudes higher than a set threshold. It is evident that force signals from robot demonstrations encompass a broader range of frequencies, observed in both markers that leave a trace and those that do not. This can be attributed to the challenges associated with guiding the robot flange, often resulting in the loss and re-establishment of contact between the marker and the whiteboard.



(a) The recordings of a single participant with the cobot (no-trace in green and trace in yellow) and the virtual marker (no trace in blue and trace in red). The detail demonstrates the sampling of demonstration, with marked point P_i on ideal line l_i and point P_d on demonstrated trajectory



(b) The upper graph shows the distance error between the demonstrated trajectories and the ideal trajectory for each segment (A-H), while the lower graph shows contact force for each demonstrated trajectory along the drawing path.

Figure 4.15: Task evaluation results.

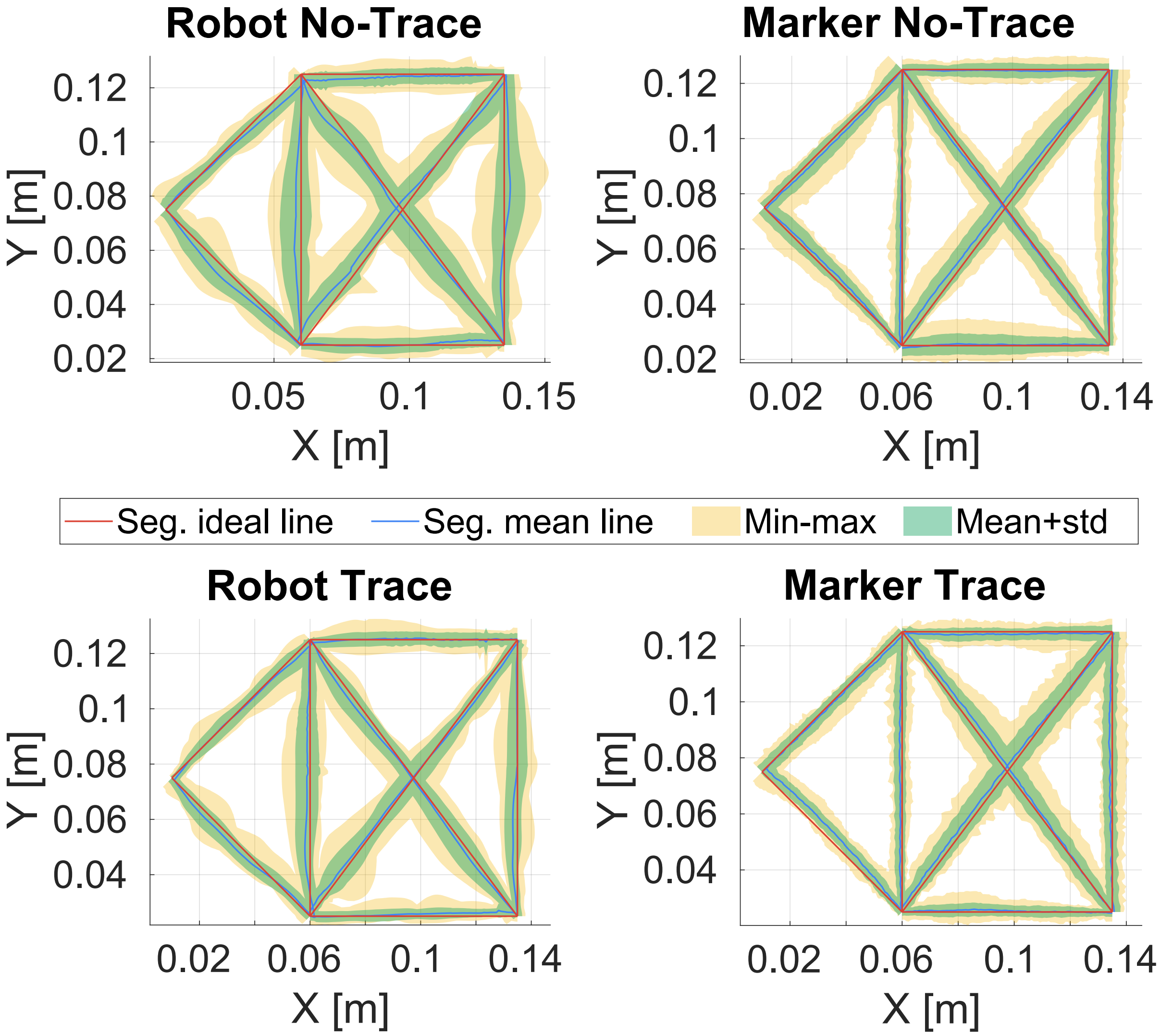


Figure 4.16: The illustration showcases demonstrated trajectories, with the ideal trajectory highlighted in red. The blue line depicts the mean value, derived from distances between the points of trajectories and their respective orthogonal projections onto the ideal segment line. Yellow indicates the range between minimum and maximum trajectory values, while the green area portrays the standard deviation from the mean trajectory.

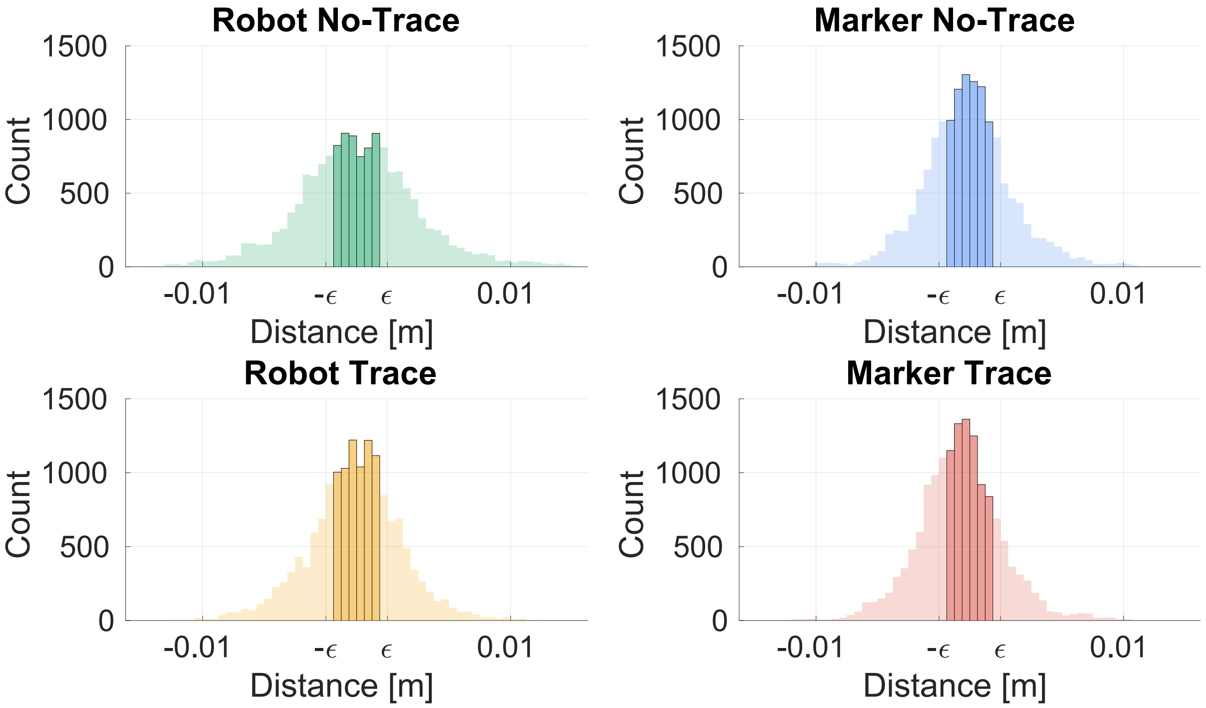


Figure 4.17: Distribution of distances of trajectory points from the ideal segment's lines, accompanied by the corresponding count of such points. The narrow ε area, indicating where the majority of well-demonstrated trajectory points are expected, is highlighted.

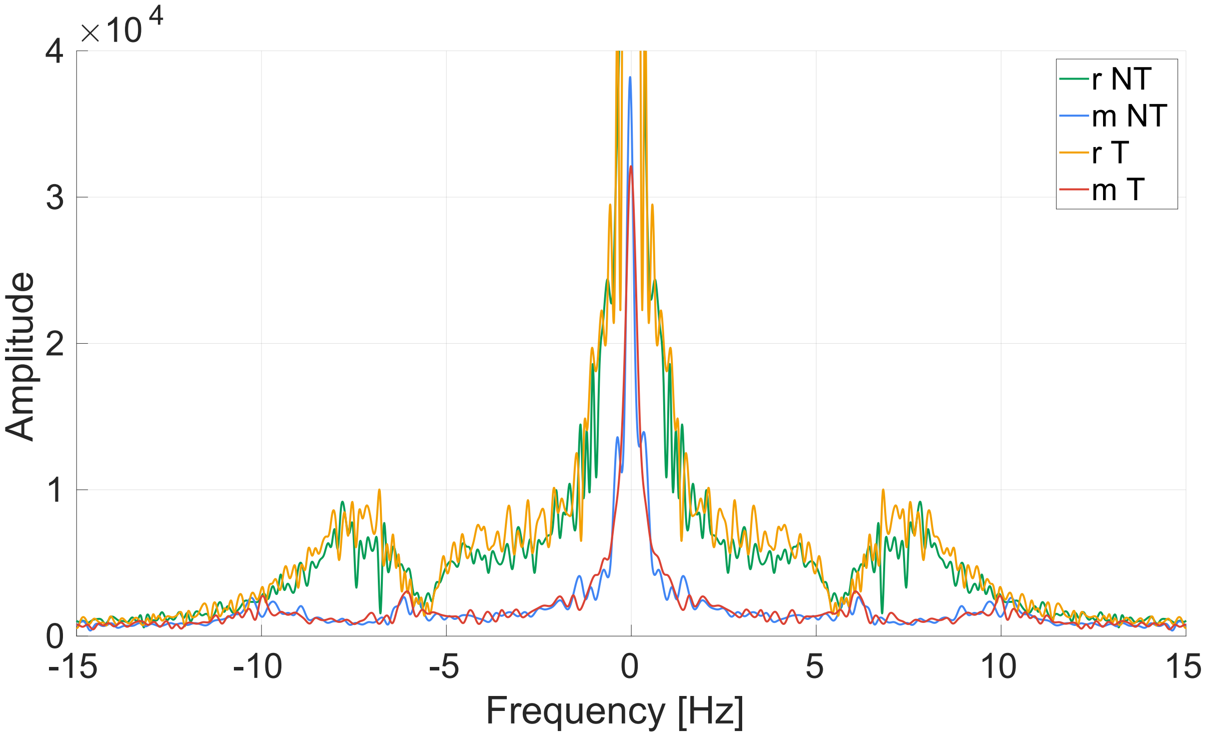


Figure 4.18: Fast-Fourier-Transformation of force signals. Demonstrations where marker does not leave trace are shown in green (robot) and blue (marker), while yellow (robot) and red (marker) denote demonstrations with the marker that does leave trace.

4.3.3 Discussion

In the conducted user-experience survey, it is apparent that demonstrating with the virtual pen induces significantly less operator workload compared to directly guiding the cobot. Similarly, participants found the virtual pen system more useful in PbD applications compared to the cobot. Regarding the task evaluation, it is demonstrated that trajectories shown with the virtual pen were closer to the ideal task, exhibiting lower error rates and less variation than those demonstrated with the cobot. Furthermore, demonstrations with the cobot exerted higher contact force with greater variations compared to virtual marker demonstrations. All these findings suggest that directly guiding the cobot to demonstrate a specific task is highly biased by the operator being outside their familiar environment and comfort zone. This situation may lead to demonstrations that fail to capture the true essence of the demonstrated motion

To contextualize these results in a terms of industry, the outcomes of two demonstration approaches in the same industrial use case of a robotic deep-micro-hole drilling of moulds used in the glass manufacturing industry were compared. The first approach, as presented by Ochoa et al. [59], involves operators guiding the robot's end-effector during the demonstration. In contrast, our recent works [60] focus on capturing how the operator performs the task based on virtual pen approach, analyzing both the forces exerted and position profiles of the drilling tool. Both of our studies demonstrate significant performance differences compared to Ochoa et al. [59], underscoring the importance of keeping operators within their comfort zone and familiar environments for the precise capture of expert skills.

4.4 Demonstration replication

After the demonstration phase of PbD, the replication phase aims to translate the demonstrated motion into executable robot actions. This phase involves analyzing and interpreting the recorded data to extract relevant features and patterns that represent the desired task. These features serve as the basis for generating robot trajectories or behaviours that replicate the demonstrated actions. Various techniques can be employed in the replication phase, depending on factors such as the complexity of the task and the capabilities of the hardware. A straightforward approach involves directly replaying the recorded trajectories on the robot. However, this approach may be limited in cases where adjustments to the demonstrated motion, such as initial and target conditions, are needed. For more complex tasks, a more sophisticated approach is required to generate appropriate robot motions. This may involve the use of advanced algorithms to replicate the underlying task dynamics. These approaches will be further discussed in the following subsections, using the demonstrations obtained through the user-experience study presented in Sec. 4.3.

4.4.1 Time-Based Trajectory Planning

Replicating demonstrated skills often encounters challenges in achieving velocities and accelerations as found in the demonstration. These issues may stem from task constraints or hardware limitations. Consequently, the recorded demonstration often undergoes time-based trajectory planning methods, wherein the replication trajectory is constructed based on specific velocity or acceleration preferences or limits. An example of such a requirement is the joint limitations of the robot manipulator. By understanding the robot's constraints, the trajectory derived from the demonstrated motion can be optimized to ensure joint velocities and accelerations that are feasible for execution on the robot. This optimization process helps tailor the replicated trajectory to suit the capabilities and limitations of the robot, ensuring smoother and more accurate execution of the task. On the other hand, the goal of trajectory planning can be defined in task space. For instance, it may involve optimizing the linear motion of the robot's tool or ensuring a constant processing speed of the tool as it works on the workpiece.

While various approaches and algorithms can be employed in such scenarios, this work utilizes the Time-Optimal Path Parameterization Based on Reachability Analysis (TOPP-RA) algorithm [61]. An example of demonstration replication using a time-based trajectory planning approach is depicted in Fig. 4.19. In this case, the planning was conducted in task space with a velocity limit. The replicated trajectory traverses all the demonstrated points while adhering to the specified velocity constraints. Real-case scenario where this approach is employed is shown in Sec. 6.1.

4.4.2 Dynamic Motion Primitives

When replicating complete demonstrated motion, including the demonstrated positions, velocities, and accelerations, the previously mentioned approach encounters limitations. Contrary to that, Dynamic Motion Primitives (DMP) algorithm [30, 62, 63, 64] excels at reproducing dynamic and intricate motions, often inspired by human demonstrations [65]. Fundamentally, the DMP algorithm operates as a nonlinear dynamical system utilizing standard linear differential equations coupled with a trainable nonlinear forcing term. In our case, the form of the system can be represented as:

$$\begin{aligned} \tau\dot{v} &= K_{\mathscr{D}}(d_t - d) - D_{\mathscr{D}}v - K_{\mathscr{D}}(d_t - d_0)s + K_{\mathscr{D}}f(s) \\ \tau\dot{d} &= v, \end{aligned} \tag{4.21}$$

where d and v are drilling depth and feed rate of the system, d_0 and d_t are the initial and target depth, while $K_{\mathscr{D}}$, $D_{\mathscr{D}}$, τ represent spring constant, damping term and temporal scaling factor, respectively, representing DMP trajectory parameters. The initial drilling depth d_0 is generally set to zero since the drilling process typically starts at the piloted hole. The nonlinear function

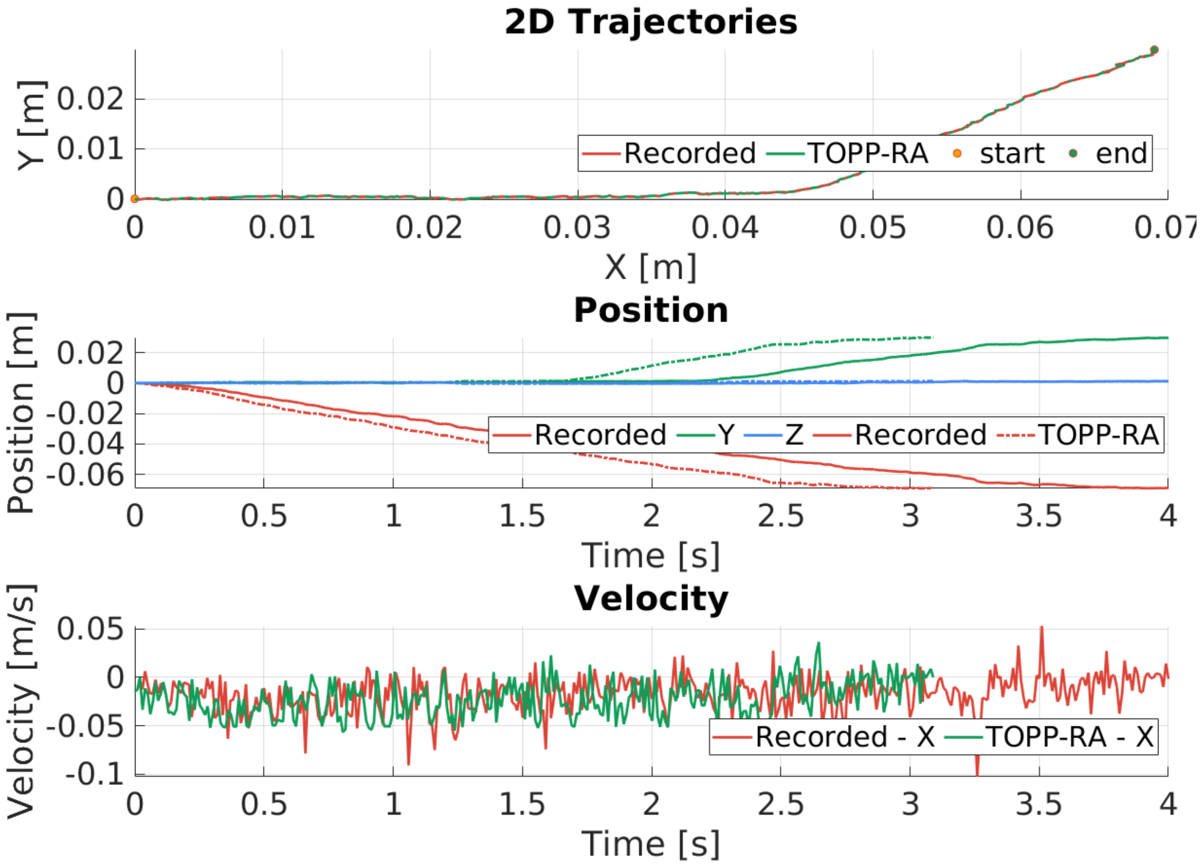


Figure 4.19: The figures illustrate time-based trajectory replication conducted using TOPP-RA algorithm.

of forcing term is defined as:

$$f(s) = \frac{\Sigma_i^{N_b} \psi_i(s)\Theta_i s}{\Sigma_i^{N} \psi_i(s)}, \tag{4.22}$$

where $\psi_i(s) = exp^{-h_i(s-c_i)^2}$ represent Gaussian basis function with center c_i and width h_i, and adjustable parameters Θ_i, while N_b represents number of basis function. Through modeling and training this forcing term on demonstrated datasets, the dynamical system becomes adept at reconstructing and mimicking the desired motion. The functions mentioned above do not directly depend on time. Instead, they depend on a phase variable s regulated by a canonical system:

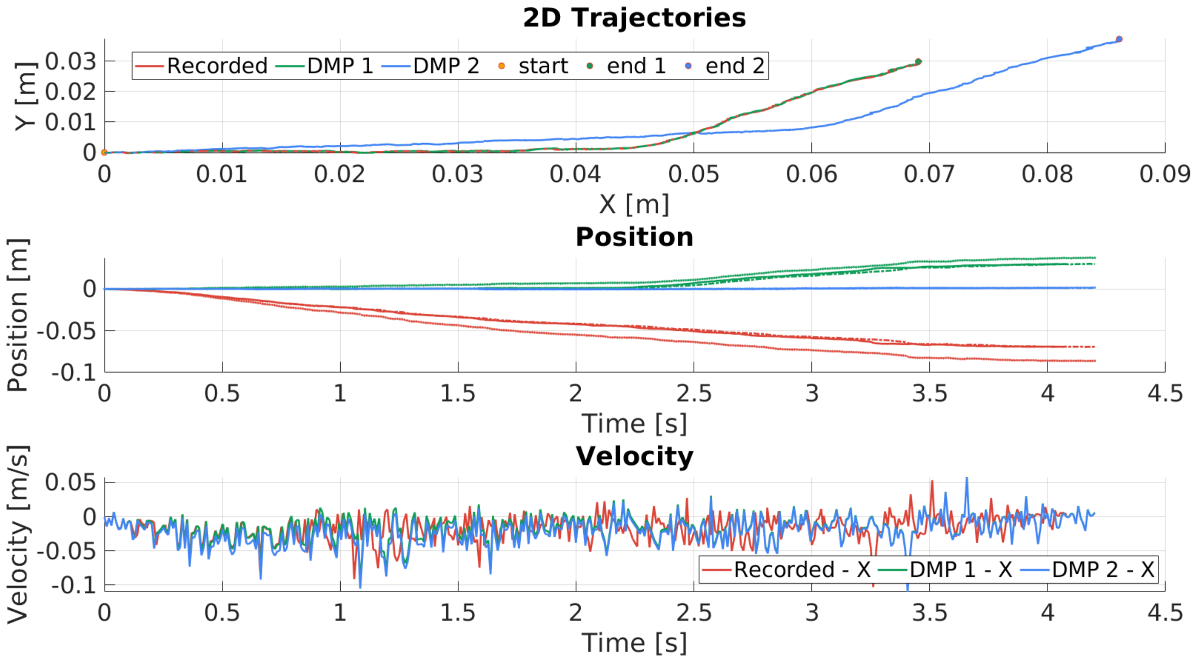
$$\tau\dot{s} = -\alpha s, \tag{4.23}$$

which monotonically changes from 1 to 0 during the motion, with α being time constant.

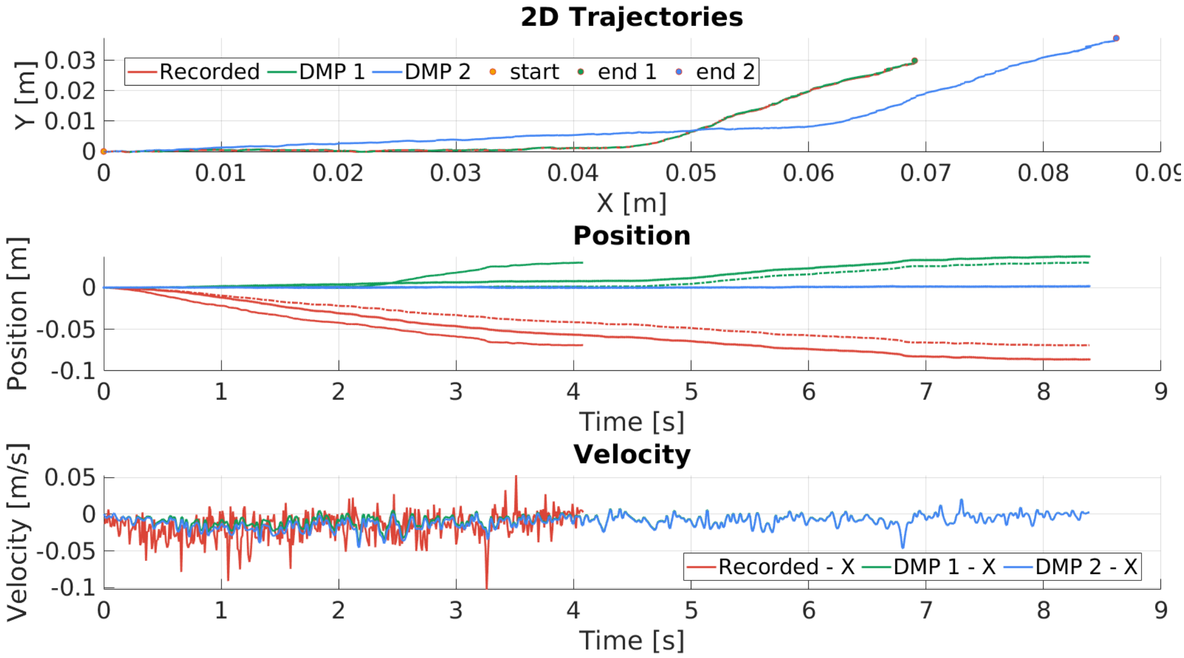
To encode the demonstrated motion, $d(t)$, into the DMP, the parameters Θ_i are optimized. This optimization is designed to ensure that the non-linear forcing term guides the system to follow the demonstrated trajectory. In the initial phase, the derivatives of $v(t)$ and $\dot{v}(t)$ for the demonstrated motion $d(t)$ are computed at discrete time stamps for $t = 0, \ldots, T$. Simultaneously, the value of $s(t)$ is determined using Eq 4.23. Subsequently, the target forcing term, $f_{target}(s)$, is calculated using Eq. 4.21. The final set of parameters, Θ_i, are then derived by minimizing the criterion $J = \sum_i (f_{target} - f(s))^2$ through a linear regression approach.

DMP trajectory

An example of demonstration replication using a DMP approach is depicted in Fig. 4.20. Given that one of the key aspects of the DMP algorithm involves modifying the initial and target conditions, two modifications were tested. Firstly, the target position was slightly adjusted. It is evident that the trajectory reaches the target position while maintaining a trajectory shape similar to the demonstrated one. Additionally, a time variation was introduced by increasing the execution time to reach the target position. In terms of position, the system reaches the target position following the same path between the start and target points, albeit at a lower velocity. Real-case scenario where DMP approach is employed is shown in Sec. 6.2.



(a) Same duration.



(b) Different duration.

Figure 4.20: Demonstration replication using a DMP approach.

CHAPTER 5

System Safety

Since the early stages of industrial robotization, large and powerful industrial manipulators have increasingly been integrated into various branches of industry, primarily handling repetitive and labor-intensive tasks in large production lines. However, due to their lack of sensing capabilities and non-flexible pre-programming approach, such robotic systems are unable to understand their environment and adapt their behaviour accordingly. Consequently, industrial manipulators are typically enclosed within safety fences, restricting their movement only when human operators are at a safe distance. While this approach ensures safety, it significantly reduces production efficiency, especially in industries where certain operations are still performed manually by human operators. In contrast, collaborative manipulators are torque-controlled and capable of sensing contact with the environment by measuring joint torques. This enables a high degree of human-robot collaboration [8, 9, 10], leading to new safety approaches that goes beyond traditional safety fences. However, off-the-shelf collaborative robots often lag behind industrial manipulators in terms of range, payload capacity, and precision. This chapter introduces a modern computer vision system that enables the use of industrial manipulators in a collaborative manner while retaining all the safety features of collaborative manipulation. The system is based on an improved collision detection method using the GPU-Voxels library [36, 37], which extends the ability of collaborative systems to share workspace between the operator and the robot. The system's safety is designed in accordance with the requirements of ISO technical specifications for collaborative robots.

Safety requirements for collaborative robots

When it comes to the safety issue of industrial manipulation applications, risk assessment is required. In the conventional industrial application of manipulators it is generally standardized by several norms [66, 67]. Since collaborative manipulators go beyond the limits of the well known standards, International Standardization Organization (ISO) published the technical specification ISO /TS 15066 [68] in 2016. The specification is still not a standard, but it is a first step towards standardization of safety issues for collaborative robots. It is important to note that ISO/TS 15066 does not replace the requirements for risk assessment by ISO EN 10218, but provides a development platform for future editions of the standards. It contains a general hazard analysis and risk assessment for collaborative human-robot interaction. The specification defines four collaborative operations:

- **Safety-rated monitored stop** is a mode of operation that stops the robot without shutting off power when the operator is in the collaborative workspace. As soon as the human leaves the robot's workspace, the robot continues its movement. An example of such an operation is a manual loading station.
- **Speed and distance monitoring** is the next level of collaborative interaction where the robot dynamically reduces speeds and accelerations depending on the distance between human and robot.
- **Manual guidance**, an operation where an operator and robot are in direct contact and where the robot is controlled directly by the input of an operator. Typically used for robot-assisted operations and learning through demonstration tasks, where the human operator guides the robot directly through a manual movement. This operation can be derived using force/torque sensors on the robot tool or haptic devices.
- **Power and force limiting** is a final stage of the collaborative operation that allows the robot to fully share the workspace with humans. This stage covers situations where contact between robot and human is already established, reducing the risk of unintended or unpredictable behaviour. Safety is ensured by limiting the contact forces and pressures. The force and pressure limits are given as biomechanical limits.

The following sections will address the safety issue of the proposed framework, especially in context of requirements of the technical specification ISO/TS 15066.

5.1 Collision detection algorithm

One of the primary concerns regarding system safety during close human-robot interaction is tracking the motions of both the human and the robot and predicting potential hazardous collisions. In this work, a collision detection algorithm based on an advanced computer vision system is proposed. The system is sensor-agnostic, meaning that various sensors ranging from RGB-D cameras to 3D lidars can be utilized.

Table 5.1: Showing time duration of each stage of the camera registration algorithm.

Action	without cutoff, t [s]	with cutoff, t [s]
Cutoff	0.0	0.345
Filtration	0.013	0.007
Normal computing	0.017	0.008
Feature computing	0.103	0.057
Robot detection	34.650	23.376
Fine tuning	89.661	41.853
Total	**124.444**	**65.646**

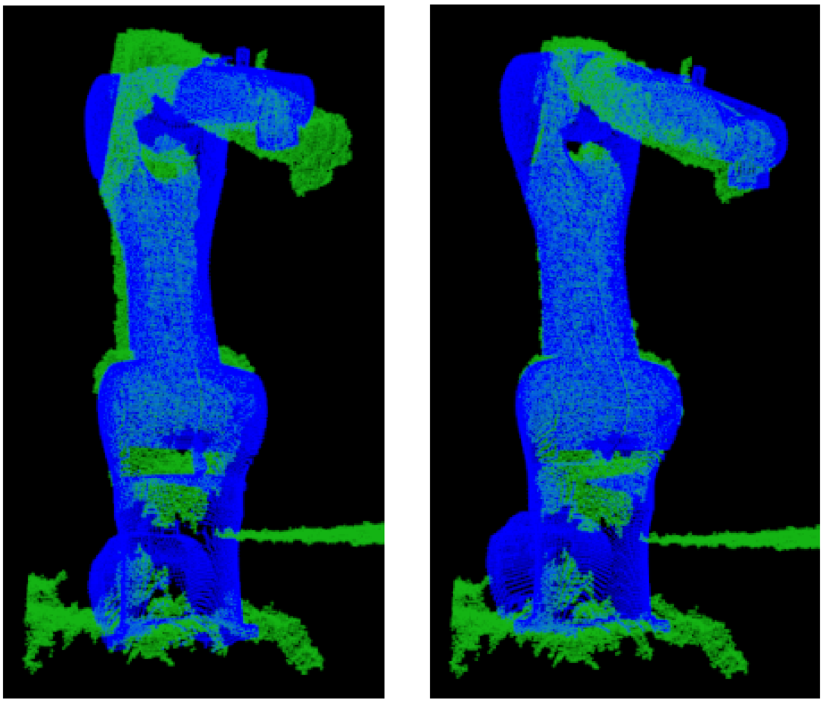
5.1.1 Sensors position calibration

In order to ensure proper coverage, multiple camera are typically placed around the work area to ensure that the people are always detected. Having this in mind, the view from multiple cameras needs to be calibrated and located within the robot's working area. Once the system is properly calibrated, it is possible to determine the exact position of each sensor, the robot and the environment. To address this issue, sensors position calibration method is proposed in this work. The sensor position calibration procedure is tailored in two steps:

1. **Data acquisition**: To obtain the calibration dataset, the robot is positioned at a chosen pose with a joint state vector $\mathbf{q}_0$. Robot is captured with each sensor k, resulting in the point cloud $\mathscr{P}_k$. Simultaneously, the known 3D model of the robot, along with a joint state vector $\mathbf{q}_0$, is used to synthesize the ground-truth point cloud data $\mathscr{P}_0$.
2. **Sensor calibration**: The scope of the calibration procedure is to register the robot's point cloud data $\mathscr{P}_k$ obtained by each sensor k and compare it with the ground truth point cloud $\mathscr{P}_0$. The procedure results in a transformation matrix $T_{S,k}$ with respect to the base frame of the robot. For this process, the localization procedure presented in Sec. 4.2.3.3 is utilized. The result of the sensor calibration procedure is shown in Fig. 5.1, while the duration of the process is shown in Tab. 5.1. Since the calibration procedure is performed only when the position of the sensor changes, although the calibration procedure takes more than 2 minutes, it works adequately fast.

5.1.2 Collision detection based on Dynamic Swept Volumes

When it comes to collision detection, the concept of "swept volumes", introduced in [36], and the GPU-voxel-based collision detection algorithm [37] are utilized in this work. The presented algorithm [37] reserves the entire space (voxels) that is swept by the body of the robot executing a given trajectory. The space reservation is performed offline before the robot starts moving. If the voxels detected by sensors collide with the occupied swept volume, a robot stop alarm is triggered, even if the obstacle is at a safe distance from the robot (i.e., the intrusion occurs along



(a) After SCP.
(b) After ICP.

Figure 5.1: The sensor data transformation after two phases of the optimization process. The blue point cloud represents ground truth data synthesized using the actual robot pose during informative motion execution, while the green point cloud represents the transformed sensor read point cloud. Fig. 5.1a shows the sensor data transformation according to the global registration, which is used for rough registration of sensor data to ground truth data. The next phase is local registration using the Iterative Closest Point (ICP) algorithm shown in Fig. 5.1b.

the trajectory the robot has yet to reach). The reserved space is released while the robot visits certain parts of the trajectory. In contrast, this thesis presents the online/dynamic allocation of swept volume depending on the velocity of the end-effector. This means that the algorithm allocates less swept volume when the end-effector moves slowly, while it allocates more space when the end-effector moves fast. With this approach, potentially dangerous behaviour can still be detected, while the operator can better utilize the robot's workspace.

Dynamic swept volumes generation

Since the timely performance of collision detection depends mainly on the algorithm execution time, it is crucial to ensure that it is performed in real time. For this reason, most of the collision detection is executed on Graphics Processing Unit (GPU). The algorithm is parallelized so that different processes such as sensor data filtering, dynamic allocation of swept volume and collision detection are executed in parallel on different processor cores. In addition to GPU multi-threading, Central Processing Unit (CPU) multi-threading is also used to acquire sensor point cloud data. The execution time of the collision detection algorithm is mainly limited by the GPU performance, but kept between 30 and 50 msec. The most important hardware property of the GPU is the number of *Streaming Multiprocessors (SM)*. The number SM is a physical property of the hardware and determines how many threads can actually be executed in parallel. For real-time execution, at least one SM is required for each sensor used.

The algorithm of dynamic sweep volume allocation is described in Alg. 1. To assign swept volume voxels, it is essential to know a priori the complete trajectory that the robot is executing. In each cycle of swept volume assignment, the algorithm locates the current position of the end-effector on the trajectory and determines a segment of the trajectory where the Euclidean distance between the segment start and the end is equal to the given distance S_v.

```
$S_v$ - given size of the swept volume
$\mathbf{p}$ - current end-effector (EE) position
$i$ - point index on the trajectory corresponding to the current position EE
$n = 1$ - check of each $n^{th}$ trajectory point
$d = 0$ - current width of the swept volume
while $d < S_v$ do
    $i = i + n$ - next point on the trajectory
    $\mathbf{p}_i$ - calculate EE position of $i^{th}$ trajectory point (direct kinematics $T_b$)
    Append $i^{th}$ point to swept volume
    $d = d(\mathbf{p}, \mathbf{p}_i)$ - update width dimension of swept volume
    if end of trajectory then
        break
    end if
end while
```

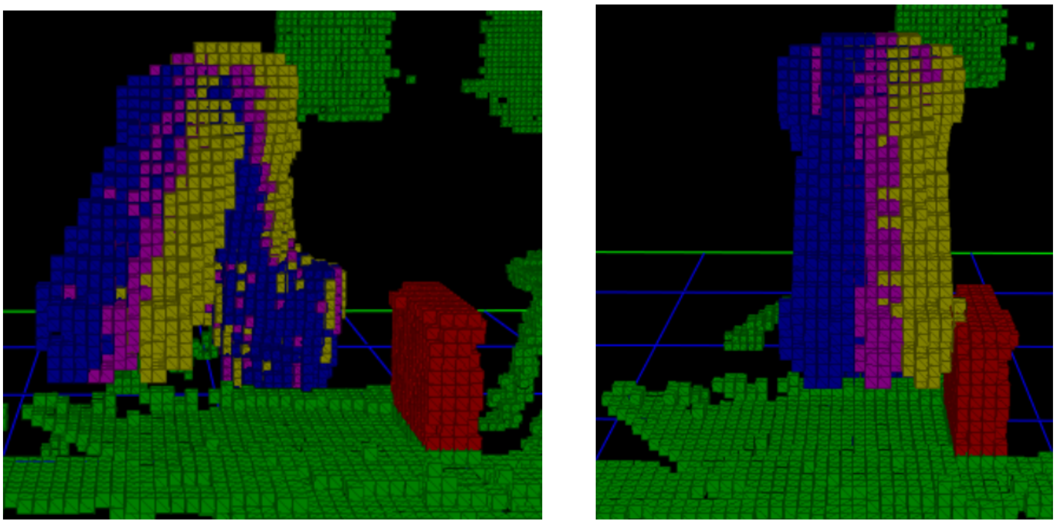
Algorithm 1: Dynamic swept volume generation

Swept volume scaling

As described above, the swept volume is scaled depending on the velocity of the robot end-effector. The faster the robot moves, the larger is the volume of a potential collision. In order to scale the swept volume, it is necessary to determine the speed of the end-effector and determine volume size depending on speed and stopping time of the robot. The velocity of the end effector $\mathbf{v} \in \mathbb{R}^{3\times1}$ is determined from the joint velocities $\mathbf{q} \in \mathbb{R}^{n\times1}$ and the reduced Jacobian matrix $\mathbf{J} \in \mathbb{R}^{3\times n}$ (Eq. 3.9). The reduced form of the standard Jacobian matrix is used, where only the translation part of the end-effector pose vector $\mathbf{p}$ is required. It is important to note that the scaling of the distance of the swept volume is performed before the dynamically swept volume itself is observed.

5.1.3 Experimental results

The scope of the experimental evaluation is to represent and evaluate the dynamic swept volume allocation and volume scaling as a function of the speed of the end effector. For this reason, experiment with a static obstacle at a fixed position was designed, while the robot performs the same linear motion from one side to the other. 15 experiment iterations were performed. In each iteration the end-effector speed was increased by 20%. The safety distance was set to



(a) Beginning of the trajectory.

(b) Collision detected moment.

Figure 5.2: Showing dynamically swept volume scaling at the beginning of the trajectory (Figure 5.2a), and in the moment of collision. Green voxels represent scanned environment of the system, red voxels are highlighted as obstacles, while blue, pink and yellow represents different swept volume scaling depending on the robot's speed. It is clear that yellow voxels represents the fastest movement of three given.

$S_d = 25cm$. The robot is expected to stop no closer than $25cm$ to the obstacle, no matter how fast it moves.

The dynamic assignment of the swept volume at different end-effector speeds is shown in Fig. 5.2. The Fig. 5.2a shows the swept volumes at the beginning of the trajectory, while Fig. 5.2b shows the robot position and swept volumes at the time a potential collision is detected. The results are analyzed statistically and give a mean stop distance to the object of about $25cm$, with a standard deviation of less than $2cm$. Considering that the voxelization process is set to a voxel size of $2cm \times 2cm \times 2cm$, one can conclude that the deviation is below one voxel. Smaller voxel size would result in significantly increased computational time, potentially leading to hazardous situation.

The initial joint speeds are given as a joint speed vector $\dot{\mathbf{q}}_\mathbf{0}$:

$$\dot{\mathbf{q}}_\mathbf{0} = [0.38, 0.37, 0.47, 0.67, 0.54, 0.86]^T, \tag{5.1}$$

while the collision detection results are displayed in Tab. 5.2. The table displays the collision detection results, indicating the distance from the stopped robot position to the obstacle for various speed scaling scenarios. The algorithm demonstrates good performance, consistently halting the robot safely across all tested velocity scenarios. The distance error values and their standard deviation remain within the size of a single voxel.

Table 5.2: Showing the results of the experiment carried out in which the robot was moving towards the obstacle at different speeds. The safety distance to the obstacle was set to $S_d = 25cm$. The results show that the avarage distance to the obstacle is close to $25cm$, while the standard deviation is less than $2cm$. Considering that the resolution of voxelization is set to $2cm \times 2cm \times 2cm$, it can be concluded that the deviation is bellow a voxel size.

Max. joint velocities	Distance to obstacle [cm]
$\dot{\mathbf{q}}_0 \cdot 1.0$	24.5328
$\dot{\mathbf{q}}_0 \cdot 1.2$	22.9231
$\dot{\mathbf{q}}_0 \cdot 1.4$	28.1432
$\dot{\mathbf{q}}_0 \cdot 1.6$	25.0431
$\dot{\mathbf{q}}_0 \cdot 1.8$	23.8214
$\dot{\mathbf{q}}_0 \cdot 2.0$	27.2822
$\dot{\mathbf{q}}_0 \cdot 2.2$	25.8112
$\dot{\mathbf{q}}_0 \cdot 2.5$	25.1354
$\dot{\mathbf{q}}_0 \cdot 3.0$	26.2139
Mean	**25.4340**
Standard deviation	**1.5443**

5.2 Framework compliance to ISO/TS 15066

This section discuss the proposed framework compliance to ISO technical specification 15066 [38] for collaborative robots. Since the proposed framework is tailored as add-on system to standard off-the-shelf available industrial manipulators, special care must be paid to the safety.

Safety-rated monitored stop

The one of two capabilities in term of safety of the poposed GPU-Voxel based algorithm (Sec. 5.1) is online collision detection. The goal of the algorithm is to increase as much as possible usage of collaborative workspace while keeping high level of operator safety. The algorithm distinguishes human and robot motions that lead to collision, from the movements that does not lead to collision. This way, the operator and the robot can perform tasks simultaneously, sharing collaborative workspace extensively.

The minimal distance of dynamical swept volume, proposed in Sec. 5.1.2, is defined as :

$$S_v = \mathbf{v}_R \cdot (T_R + T_B) + S_d, \tag{5.2}$$

where v_R is robot's end-effector velocity, T_R collision detection system reaction time, T_B robot stopping time, and S_d minimal separation distance between the human and the robot. This ensures collision detection system will have enough time to react and stop the robot at a safe distance from human.

In contrast to the initial version of the swept volume collision detection introduced in [36], the algorithm proposed in this thesis enhances system effectiveness by allocating less space

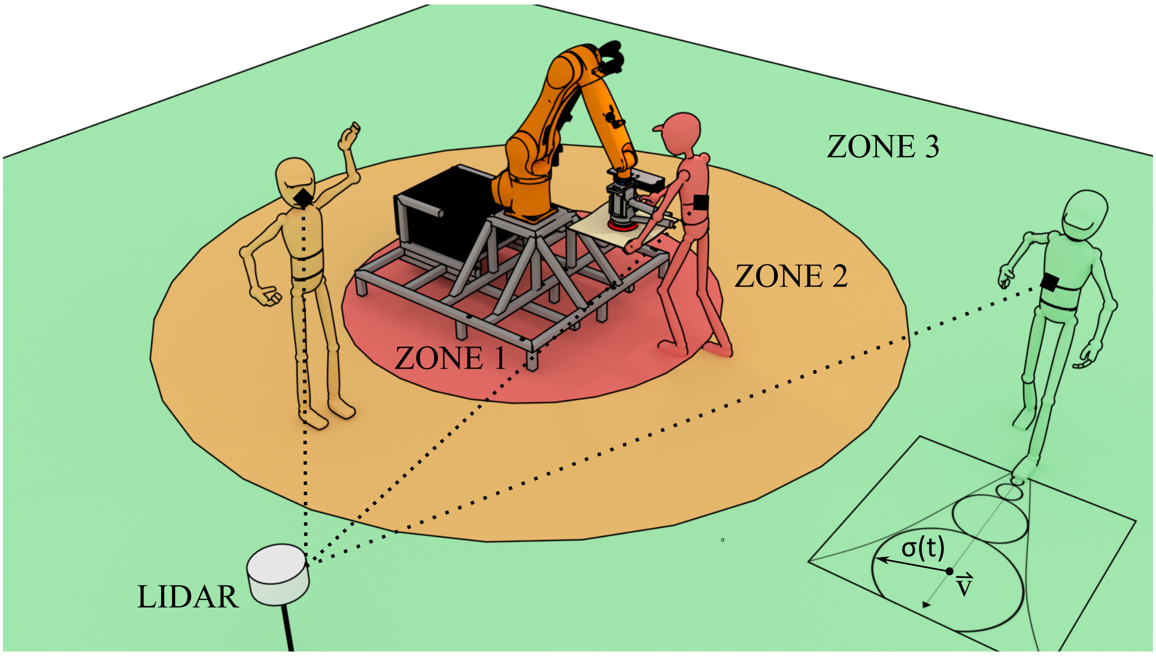


Figure 5.3: Illustrating the concept of safety zones, where the space in front of the robot is divided into several zones, each with a different maximum robot speed set for the presence of a human in the zone.

while maintaining a high level of safety. This approach is critical in industrial scenarios where human operators and robots need to handle objects simultaneously. For example, in grinding large pieces where the robot trajectory spans across the entire workpiece, once the robot moves away from one side, there is no reason why a human operator cannot perform additional tasks on the area already treated by the robot. At the same time the robot continues its task on the other side. To further improve the effectiveness of the proposed system and reduce the allocation distance even further, incorporating the human operator's approaching speed into Eq. 5.2 is suggested. Nevertheless, this approach significantly enhances the collaboration level between human operators and robots, with a steadfast commitment to prioritizing operator safety.

Speed and separation monitoring

The second capability of the proposed collision detection system is the concept of safety zones monitoring, as shown in Fig. 5.3, introduced in [50]. This concept aligns with the requirements of the technical specification, which demand a reduction in the robot's speed depending on the distance between the robot and the operator. The space in front of the robot can be divided into several zones, with the closest zone resulting in a greater reduction in the robot's maximum speed. The primary goal is to maintain a sufficient distance between the robot and the operator. This mode is primarily used for collaborative operations that are occasional, where the human and robot are not in collaborative cooperation most of the time. For example, a human operator may supervise a group of industrial manipulators, bring materials, and check important stages of the process.

Hand guiding

Collaborative robotics has opened numerous new opportunities for manipulator utilization, particularly for tasks where operators can directly and intuitively control the robot. An example of such operation is hand guiding, also known as kinesthetic teaching, where the operator has direct control over the robot's movements, not through programming, but by physically guiding the robot's tool. In the collaborative system proposed in this thesis, the manipulator is equipped with a 6-DOF force/torque sensor and a specially designed and tuned impedance controller, as detailed in Sec. 4.1, which enables kineshtetic teaching using the industrial manipulator.

Power and force limiting

As the final stage of collaborative mode, as outlined in ISO/TS 15066, power and force limiting are presented with the goal of restricting the forces, torques, and pressures that a robot can exert on the operator and the environment. On the one hand, collaborative manipulators are joint torque-controlled, enabling them to sense contact with the environment and limit contact forces. Established contact forces are measured as joint torque disturbances. With this approach, the robot can sense contact with the environment throughout the entire kinematic chain.

On the other hand, industrial manipulators are joint position controlled and cannot sense contact with the environment. To address this limitation, the robot is equipped with a 6-DOF force/torque sensor mounted between the robot's flange and tool, enabling it to sense contact acting on the tool. However, it is not possible to recognize any other contact acting on the kinematics chain up to the tool. In order to provide full safety and recognize potential harmful contact between the whole kinematics chain and the environment, it is possible to equip the robot with pneumatic coating [8], as shown in Fig. 5.4. The proposed safety system consists of air pockets placed on each link of the robot. When contact occurs, the pressure disturbance of the single air pocket is measured and triggers an emergency state. By setting desired pressure thresholds, it is possible to achieve biomechanical limits that are safe for collaborative operation.

---

[8] `https://www.airskin.io/product/airskin-for-kuka`

Figure 5.4: *AIRSKIN*® safety system, consisting of air pockets that are put on each link of the robot. When in contact, pressure disturbance within a single pocket triggers an emergency state.

CHAPTER 6

Framework Applications to Industrial Scenarios

In the preceding chapters, three key components of the collaborative framework for industrial manipulators have been introduced: compliance control, programming-by-demonstration, and system safety. This chapter addresses the practical applications of this framework in industrial scenarios. Firstly, the application of delicate grinding of composite materials used in aircraft manufacturing is presented. Here, compliant Cartesian motion is employed alongside a kinesthetic teaching approach to capture the grinding motion from experienced human operators. Another application is deep-micro-hole drilling in the glass mould container production industry. Despite widespread robotization in mould production, the process of deep-micro-hole drilling is primarily performed by human operators due to its delicate nature. Recognizing the limitations of the kinesthetic approach, especially when guiding the robot pushes the human operator outside their comfort zone, this scenario shifts focus towards the utilization of a motion capture system and a tool tracking algorithm to accurately capture the demonstration while keeping the operator in their comfort zone. Lastly, the proposed compliance control system was tested in the domain of manipulating deformable objects, specifically in a plastering application.

Robot hardware setup

The robotic setup used in the experimental validation of the proposed framework is built upon the industrial manipulator *KUKA KR10* along with the *KRC C4 Compact* control unit. The control unit offers the Robot Sensor Interface (RSI), serving as an interface for communication between the robot and its sensor systems. Additionally, it provides an Ethernet interface utilized for integration with the Robotic Operating System (ROS). The ROS integration is crucial as it

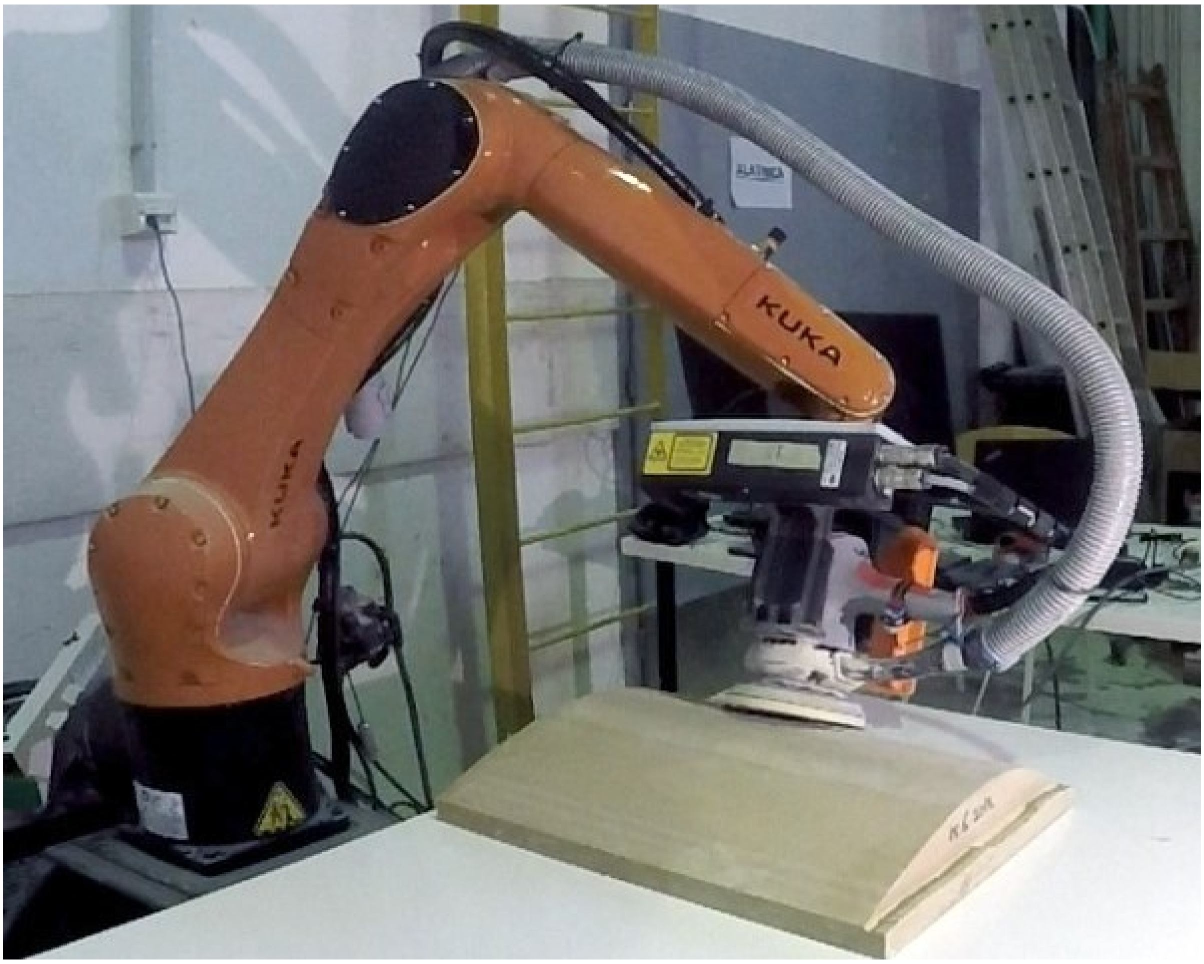


Figure 6.1: Showing compliance control setup involving industrial manipulator KUKA KR10 performing delicate sanding of complex surfaces in an industrial scenario.

enables easy and efficient integration of all framework components. To enable compliant behaviour, the robot is equipped with a force/torque sensor positioned between the robot's flange and the tool. Two different force/torque sensors were used: the *OnRobot* force/torque sensor and the industrial *ATI Gamma* and *Delta* sensors. Furthermore, for efficient and high-precision localization of workpieces, the robot is equipped with a *Gocator 2150* 2D laser scanner.

6.1 Delicate grinding of composite materials used in aircraft manufacturing industry

In the first industrial scenario, the framework was used for delicate grinding of composite materials used in the aircraft manufacturing industry, as depicted in Fig 6.1. The scenario was built within the ENDORSE project (Effective Robotic GriNDing of Surface Areas through HORSE framework) [9]. Although the grinding process is currently partially robotized using a standard

---

[9] `https://larics.fer.hr/larics/research/endorse?`

pre-programmed industrial manipulator with a single DOF active force-control flange [10], a significant portion of it is still manually handled by human workers. Given the health risks associated with grinding, the primary goal was to minimize the time human operators spend in such hazardous environments. Planning robot operations for sanding complex 3D surfaces is typically challenging, relying on qualitative assessments by experienced workers that are difficult to model or code. In this section, a robotic system is introduced that utilizes the skill and expertise of operators through a collaborative human-robot interaction framework. The system not only significantly reduces human labor, but also provides flexibility to intuitively programming the process for objects with complex surfaces.

6.1.1 Robotic Grinding based on Industrial Manipulator

Considering the weight of the grinding tool, typical grinding forces, and required reach due to large workpieces, the utilization of the industrial manipulator is crucial. Ensuring high-quality grinding necessitates the system to ensure that the end-effector grinding tool perfectly conforms to the product's surface throughout the trajectory. Therefore, the system must be capable of tracking the grinding trajectory and controlling contact forces and torques in all six degrees of freedom simultaneously.

6.1.1.1 Compliant industrial manipulator

In order to control the industrial manipulator in a compliant way, compliant control algorithm presented in Ch. 3 was utilized. Algorithm combines three fundamental algorithms: impedance control, admittance control and force control. The force contribution of all three parts is then merged and processed from the Cartesian space up to joint state using the forward dynamics simulation (Fig. 3.3). The mass and inertia, as well as the compliance parameters are freely chosen to match the compliant behaviour respective to the environment compliance. The different environment stiffness demands modification of end-effector compliance, as it was discussed in Sec. 3.3.

Kinesthetic teaching

The grinding process often necessitates trajectories that prove challenging to generate within the CAD environment, particularly in highly curved and narrow surfaces. However, an experienced human operator perform such grinding tasks effortlessly, maneuvering the grinding tool freely over the workpiece. To replicate this grinding motion of the operator to the robotic system, the kinesthetic teaching approach, as presented in Sec. 4.1, was utilized. In this Programming-by-Demonstration approach, an experienced operator guides the robot while

---

[10] https://www.youtube.com/watch?v=gDel8wzub0M

holding the end-effector, i.e. grinding tool placed on the robot. Here, it is important to ensure operator intuitive Cartesian compliance in tool frame, irrespective of the tool's orientation.

6.1.1.2 Force/torque measurements

Due to the high-frequency noise generated by the orbital sanding tool, reaching amplitudes of up to 50N, a spectral analysis of the force and torque signals has been conducted. Firstly, dataset was recorded, by measuring forces and torques across all 6-DOF of the sensor, with the grinding tool set to different speeds. Following to that, spectral analysis of measured signals was performed, as shown (in red) in Fig. 6.2. To reduce the noise level, a customized 3^{rd} order linear filter was designed with set cuttoff frequency at $90Hz$, resulting in an approximate threefold reduction in noise. Additionally, the impact of the sanding tool, weighing approximately 3kg ($\sim$ 30N), on force measurements cannot be disregarded. Filtered force signals are shown in blue in Fig. 6.2.

Additionally, the impact of the sanding tool, weighing approximately 3kg ($\sim$ 30N), on force measurements cannot be disregarded. Sanding complex curved 3D surfaces necessitates movement of the end-effector approach axis in various directions, causing a varied distribution of the tool's weight across the 6 degrees of measurement in the force sensor. To estimate this distribution, a calibration dataset was gathered, measuring forces and torques with different tool orientations when the tool was not in contact with the environment. The sensor offset was modeled using the following equation:

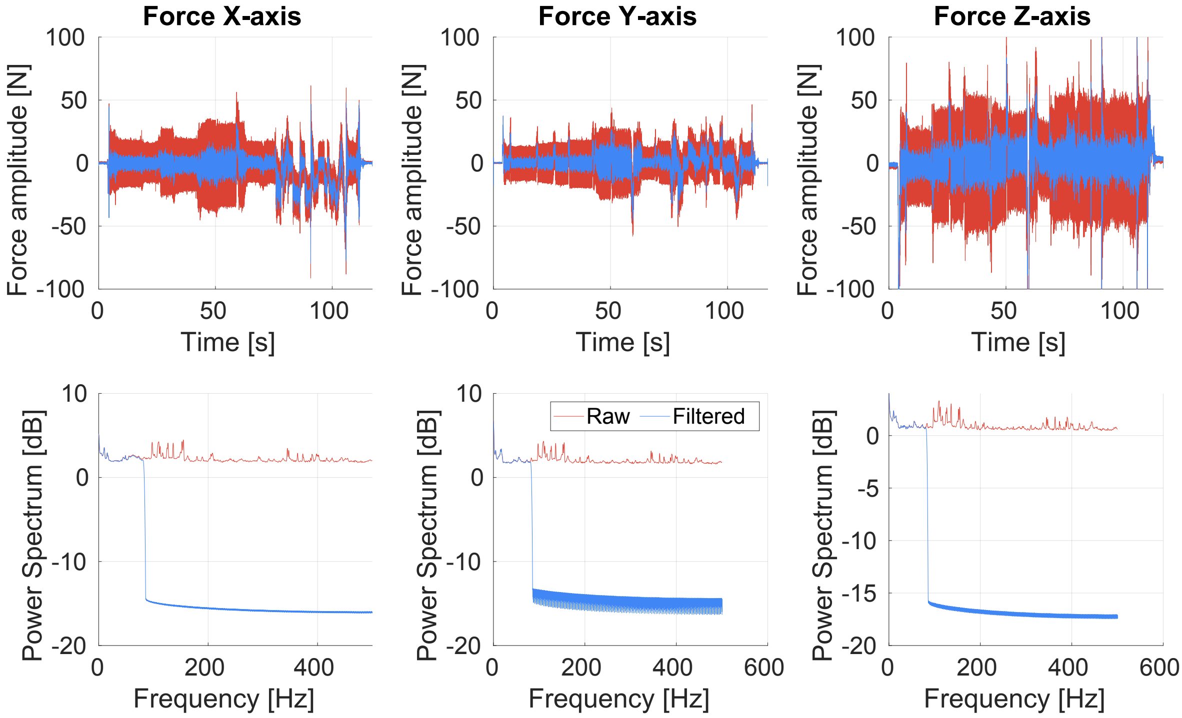
$$F_0(\alpha,\beta) = \Phi_0 + \Phi_1 \cdot \sin(\Phi_2 \cdot \alpha + \Phi_3) \cdot \cos(\Phi_4 \cdot \beta + \Phi_5), \tag{6.1}$$

where α and β represent the roll and pitch angles of the tool, respectively. The tool weight distribution in the force sensor reading follows a sinusoidal form. The optimization parameters Φ_{0-5} encompass force amplitude, frequency, phase shift for both angles, and static amplitude offset. The optimization process is conducted within the MATLAB environment, utilizing the least mean squares (LMS) method.

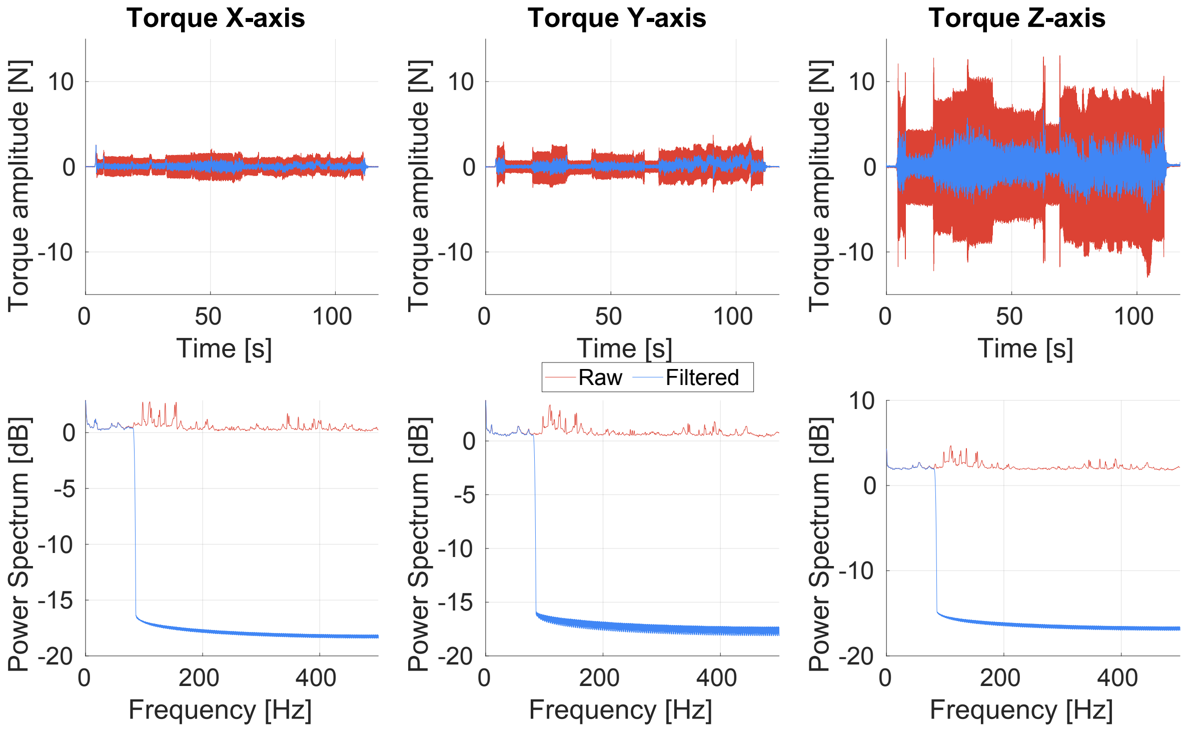
The results of force signal filtration and calibration are illustrated in Fig. 6.3, specifically highlighting the pitch axis, where force correction has the most pronounced impact. Force estimation can reach up to 20 N in certain poses. Without force correction, trajectory tracking error would be as high as $5mm$ given the standard stiffness of the impedance controller in this axis ($c = 4000\frac{N}{m}$), posing a potential risk of significant damage to the treated object.

6.1.1.3 Scan matching, localization and trajectory generation

The utilization of computer-aided design (CAD) data for free-form surfaces [69] is integral in generating tool trajectories for various applications, encompassing cutting, drilling, paint-



(a) Forces



(b) Torques

Figure 6.2: The figures depict the raw force/torque signals (shown in red), which are highly saturated with high-frequency noise produced by the grinding tool. After conducting spectral analysis, a 3^{rd} order polynomial low-pass filter was designed to reduce the noise level. The filtered signals are shown in blue.

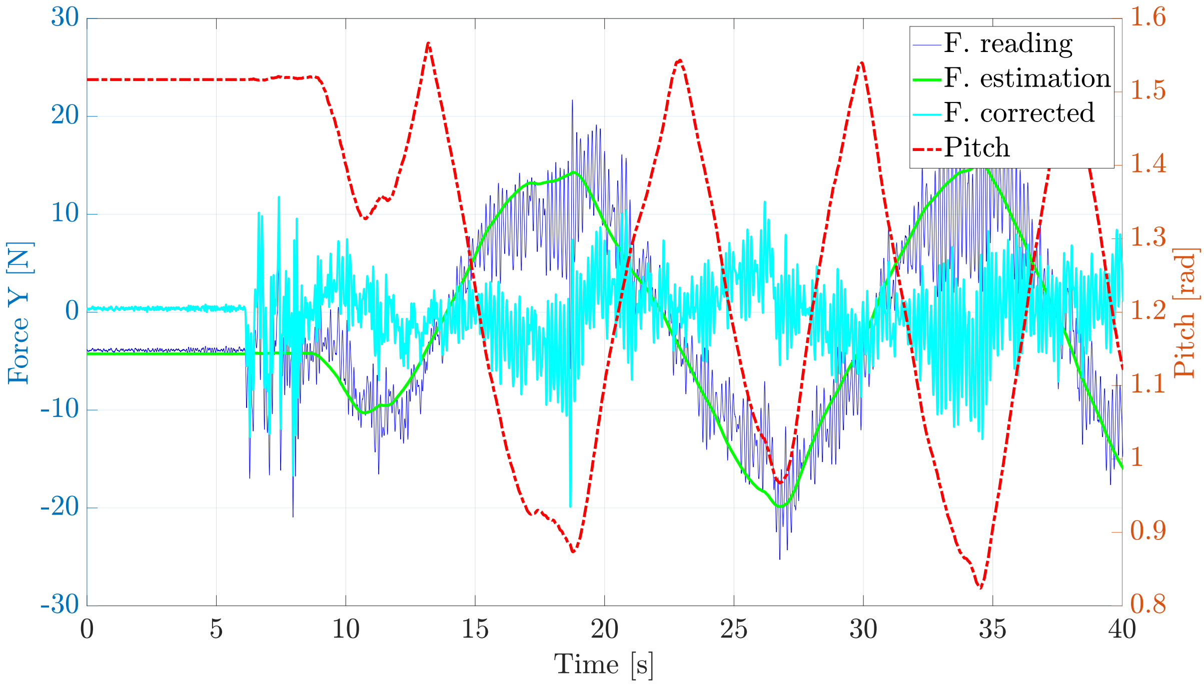


Figure 6.3: Showing force calibration including force reading, force offset estimation and final corrected value in pitch axis where the most significant effect of the force calibration is present. If the impedance stiffness in given axis is assumed to be $c = 4000\frac{N}{m}$ it will result up to $5mm$ error in trajectory tracking without force offset estimation and correction.

ing, and grinding. Planning algorithms rely on 3D surface data (including faces and normals), tool specifications (such as radius and shape), and surface material properties for initial trajectory generation. However, due to the inherent complexity of free-form surfaces, additional modifications are frequently necessary to achieve satisfactory results. Among these applications, grinding presents notable challenges for robots, prompting researchers to delve into the analysis of parameters influencing the final outcome. These parameters encompass raster path generation [70], tool rotation speed, sandpaper type [71], applied force [72], and more. The impact of these factors is particularly pronounced on surfaces with significant curvature, often necessitating human operators to perform sanding on such parts.

Ensuring precise positioning of each workpiece relative to the robot frame is usually unfeasible, especially when dealing with elastic workpieces and mounts that may not maintain a consistent position and orientation of workpiece. To overcome this issue, a laser scanner, mounted on the robot to scan each workpiece individually, was utilized, as previously described, in order to enable precise localization for each workpiece. In order to perform accurate trajectory planning on the designated surface, a reference virtual 3D model of the object is required. To localize the workpiece, localization algorithm presented in Sec. 4.2.3.3 was utilized. The localization procedure aligns the reference model with the scanned model, resulting in localization accuracy of $1.47(\bar{x}) \pm 1.74\text{mm}(\sigma)$.

The final step involves planning the trajectory of the robot's end-effector in the robot base

Product	1DOF	6DOF	PbD
A	60%	75%	80%
B	34%	50%	50%
C	0%	60%	98%

Table 6.1: 3 product use case: 1DOF-current state of production, 6DOF - proposed impedance control improvement, PbD - programming by demonstration

frame, considering the transform chain derived from scan matching and localization. The planning considers the user-selected area of the product, dividing it into predefined segments known from the product model and transformed upon localization. Each segment is defined by its location and a normal to the surface. Here, a simple lawnmower path implemented across the surface is presented, ensuring each segment is visited only once. Additional details about scan matching, localization, and trajectory generation can be found in [50].

6.1.2 Experimental validation

Benchmarking the results of the framework against the current production line was facilitated by working closely with the end user, operators, and inspectors. This clearly showed that the amount of surface treated by the robot and the variety of products can be increased. This case study involved 3 products from the end user's production line labeled as A, B and C, with their respective improvements shown in Tab. 6.1. Five instances of each product have been treated in 4 phases of sanding and paint coating. All 15 instances were subjected to standard quality assessment control usually conducted in the company, with only a single instance returned for additional sanding. Each of the four stages requires different amount of force applied to the surface ranging from 10N to 40N. Additionally, every stage achieves different surface roughness, with the final stage reaching N9 ISO grade surface roughness number. It is interesting to note that the proposed system allowed the company to increase the amount of robot treated surface in product C from zero to 98% of the surface.

Due to copyright restrictions, demonstration of the framework's capabilities on real products is not possible. Instead, the capabilities are demonstrated on a mockup version [11] with similar curvature as product C. One of the problems of applying standard single-visit lawnmower path planning is that it results in visible lines at the final sanded product shown in Fig. 6.4 a). Following the surface normal results with uneven sanding since the tool radius exceeds the local curvature of the surface. At the same time, the experienced worker uses the same tool, performs similar movements and successfully smooths the product. However, these workers are certainly not educated enough to participate in trajectory planning and programming, through the proposed PbD framework worker's experience has been converted to the trajectory planning.

---

[11] `https://youtu.be/EAezNw0kz10`

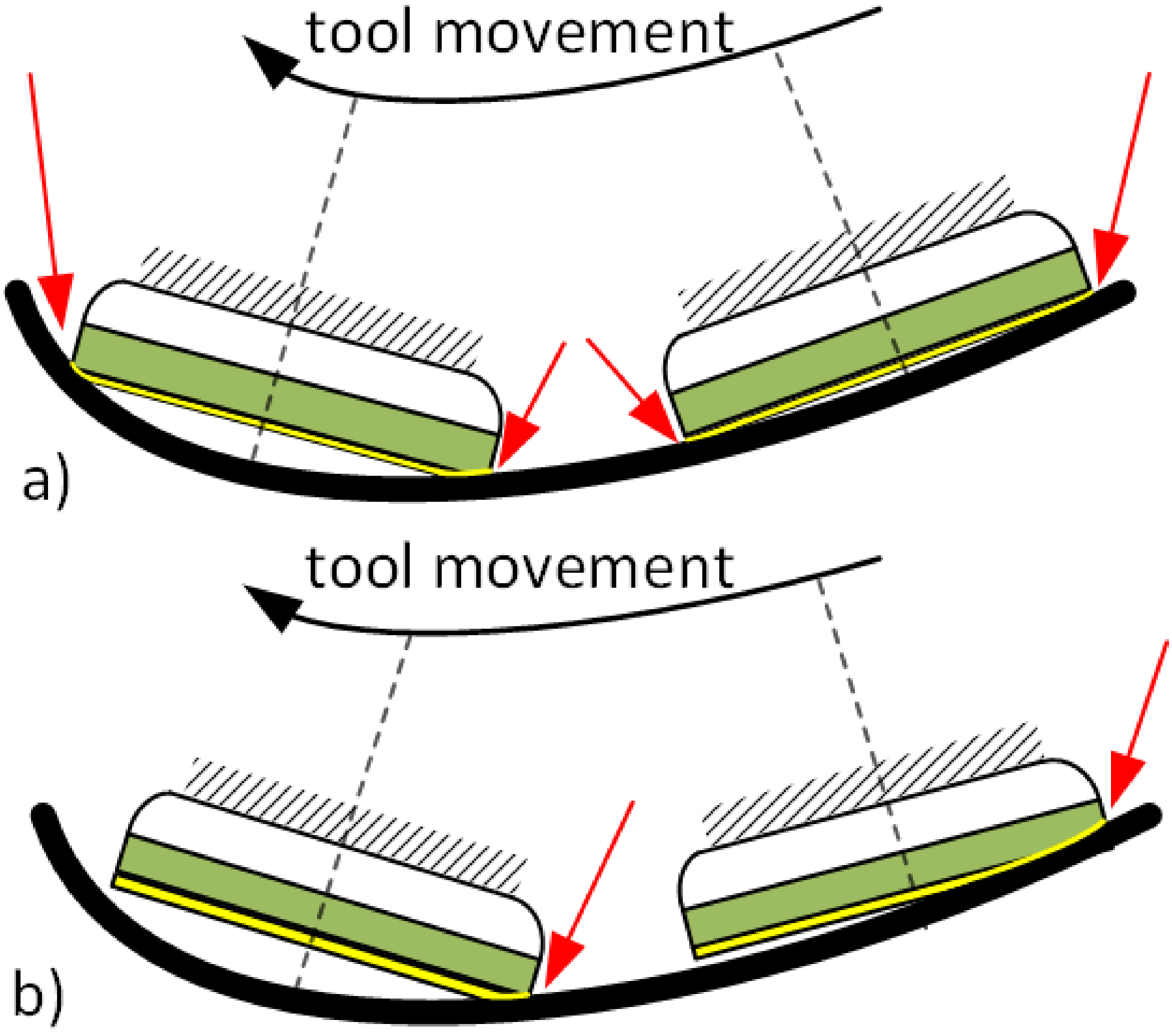


Figure 6.4: The tool trajectory movement: a) generated tool movement follows the surface normals; b) sanding trajectory provided by PbD. The presented tool elements are sanding paper (yellow) and buffering pad (green).

A single segment of PbD is shown in Fig. 6.4b) for clarity. It shows the experienced worker tilts the trajectory's normal using one side of the tool, resulting in a more uniform sanding.

6.2 Deep-micro-hole drilling in glass mould container production industry

The process of manufacturing a complete range of mould parts for glass container production is a highly robotized and automated process. Even in this context, specific stages of the production line necessitate the skilled intervention of human workers to perform delicate manipulation of the product. This makes the production line dependent on the skills and craftsmanship of but

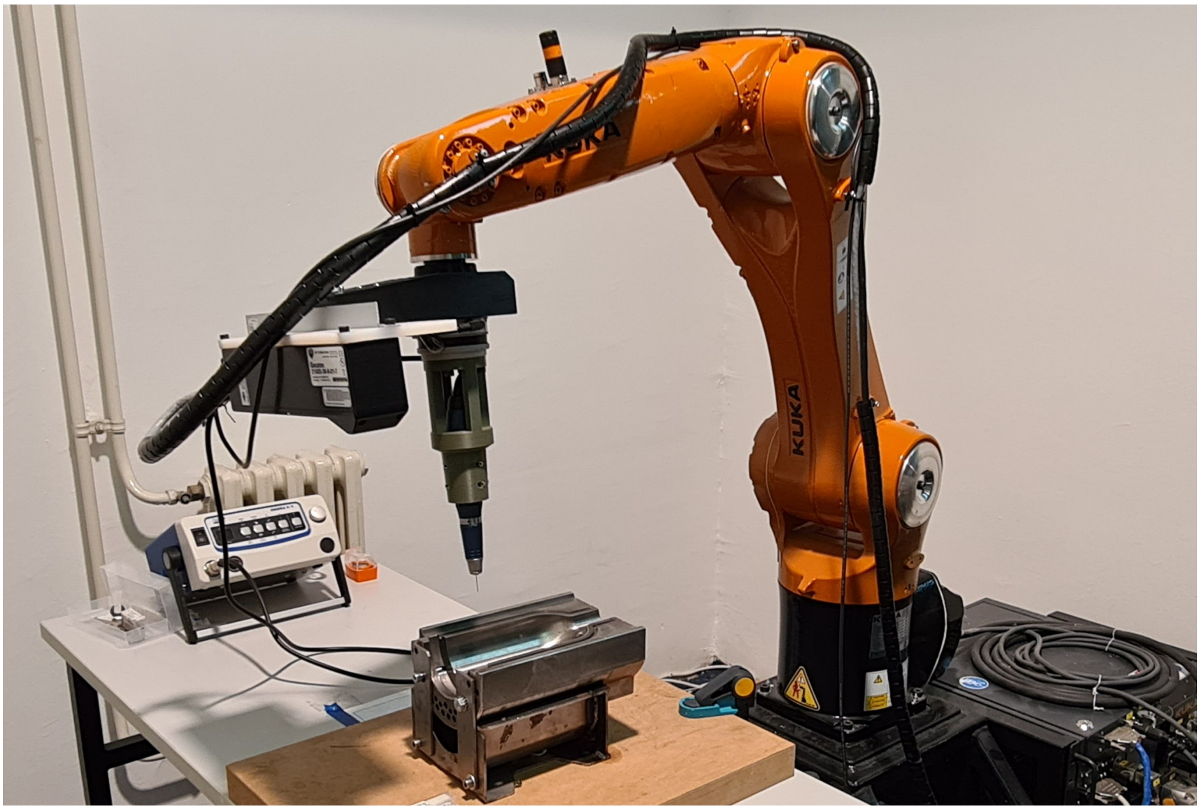


Figure 6.5: Experimental setup for deep-micro-hole robotic drilling using an industrial manipulator Kuka KR10.

a handful of skilled workers. In previous Sec. 6.1, kinesthetic teaching approach (Sec. 4.1) was utilized in similar scenario to record the expert skill. However, as presented in Sec. 4.3, kinesthetic teaching can push the expert beyond their comfort zone, resulting with significant performance impairment. Here it is no different, especially with a delicate task like deep-micro-hole drilling. In addition, the narrow mould cavity presents challenges in guiding the robot accurately. To overcome these issues, focus in this section is shift towards developing a framework that records human worker skills (Sec. 4.2.3) in their familiar environment.

6.2.1 Drilling problem

To produce glass bottles, molten glass is poured into the mould where pressure forms the desired shape of the bottle. To guarantee a perfect fit with the surface of the mould, micro-holes are drilled towards the vacuum chambers to extract the air trapped between the molten glass and the mould. Drilling micro-holes is vital for surfaces with engravings and other intricate surface features. A schematic view of a typical drilling scenario is shown in Fig. 6.6.

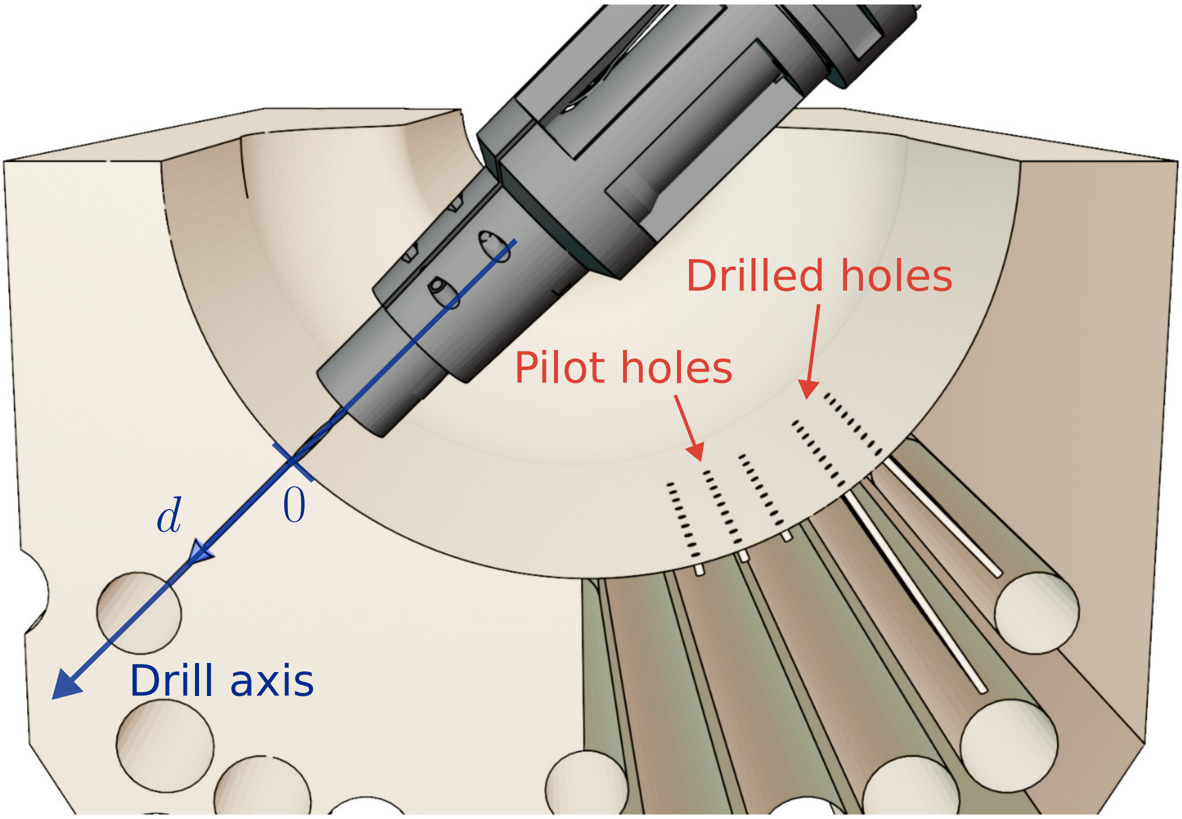


Figure 6.6: Visualizing the deep-micro-hole drilling process within a mould utilized in glass container manufacturing. Initially, a rigid drill is employed for piloting, followed by the placement of a high-speed-steel drill in the pilot hole to proceed with the deep-micro-drilling along the drilling axis.

6.2.1.1 Drilling technology

Drilling tasks are widely common operations in industry, starting from metal industry, over many other processing industries like PCB [73] or wood production, up to different medical applications [74, 75]. Depending on the drilling bit size, the drilling task can be categorized as micro-hole drilling, starting from 30 micrometers up to 3 millimeters, and the macro-drilling with hole diameter above 3 millimeters [76]. Although there are different micro-hole drilling technologies available, mechanical micro-hole drilling is often preferred over other technologies due to the shorter processing time, better hole roundness and overall surface integrity [77]. Various analyses [73, 78] showcase the difficulties in such processes. Many of challenges in mechanical micro-hole drilling stem from the large ratio of length (L) and diameter (d) of the micro drill (up to 50) and the small diameter of the micro drill itself (0.3 mm to 0.7 mm in the glass mould industry). Large ratio of L/d classifies the drilling in the glass mould industry in the category of deep-micro-hole drilling. With deep micro-holes and small drill diameters, the removal of separated particles during the drilling process becomes difficult, which leads to an increase in friction and diminishes heat dissipation. This causes higher tool wear which with low micro drill rigidity leads to frequent tool breaks and ultimately lower productivity [79]. One of the traditional approaches to these challenges in machine drilling is periodic retraction of the micro drill from the borehole (so-called peck drilling technique) [80]. With this technique the drilling time is significantly increased, due to the fact the feed pitch is reduced while the number of retractions is increased. In addition, the aforementioned challenges become even more emphasized due to the mould material inhomogeneity and local hard inclusions that are present due to the casting process.

On the other hand, an experienced human worker can adapt and overcome these challenges. However, mastering the delicate micro-drilling technique usually takes several months of training. The lack of a highly trained workforce is a serious challenge in sustaining an efficient production line. This has sparked research interest in creating a system capable of accurately capturing, extracting and repeating the precise micro-drilling skill demonstrated by the experienced and trained workers. With a keen sense of touch, by guiding the drill with fingertips, an operator can usually complete the entire drilling process using a single high-speed-steel (HSS) drill bit. The approach taken in this work resembles traditional machining, where this problem is resolved by utilizing a specially designed rigid drill bit for piloting, followed by the use of a HSS flexible drill to continue the process. The piloting stage removes less than a millimeter of material, making it straightforward to implement and is thus not further discussed in this work.

6.2.1.2 Deep-micro-hole drilling kinematics

Observed in a world frame, controlled by either a human worker, a robot, or a CNC machine, the drilling task configuration is defined with the position $\mathbf{P}_W^d \in \mathbb{R}^{3\times1}$ and the orientation of the drill frame L_d. The orientation is defined by matching the z-axis of L_d with the drill direction vector $\mathbf{d}_{dir} \in \mathbb{R}^{3\times1}$, as shown in Fig. 6.6. A $\mathbf{w}^d(t) \in \mathbb{R}^{6\times1}$ represents the configuration vector of the drill consisting of the position $\mathbf{P}^d \in \mathbb{R}^{3\times1}$ and the unity orientation vector $\hat{\mathbf{z}}^d$, all defined in the drill task space.

For practical reasons, the one degree of freedom drilling task $\mathscr{T}_n$ is defined in the mould frame as $\{\mathbf{P}_{M,n}^0, \mathbf{d}_{drill,n}, d_t, \mathscr{D}_n\} \in \mathscr{T}_n$, wherein $\mathbf{P}_{M,n}^0 \in \mathbb{R}^{3\times1}$ is the micro-hole's position on the mould surface, along with the drilling direction vector $\mathbf{d}_M^{drill,n} \in \mathbb{R}^{3\times1}$. The target depth for the micro-hole is denoted by $d_t \in \mathbb{R}$, while the drilling trajectory needed to reach the desired depth is $d(t) \in \mathscr{D}_n$. Consequently, the drill's position during the drilling of the micro-hole indexed as n can be represented as:

$$\mathbf{P}_M^d(t) = \mathbf{P}_{M,n}^0 + d(t) \cdot \mathbf{d}_M^{drill}{}_{,n}, \tag{6.2}$$

where $d(t) \in \mathbb{R}$ represents the drill tip's offset from the initial point $\mathbf{P}_{M,n}^0$ along the drilling axis $\mathbf{d}_M^{drill,n}{}_{,n}$ into the borehole. The mould remains fixed in the world frame throughout drilling, while its position T_W^M is determined through a localization procedure. Observed in L_W, drill end-efector configuration vector $\mathbf{w}_W^d(t)$ transforms to:

$$\mathbf{w}_W^d(t) = \begin{bmatrix} \mathbf{P}_W^d(t) \\ \mathbf{d}_{dir} \end{bmatrix} = \mathbf{J}_d(\mathbf{q}_H) \cdot \mathbf{q}_H(t) + \mathbf{w}_W^0, \tag{6.3}$$

with $\mathbf{w}_W^0 \in \mathbb{R}^{6\times1}$ denoting the initial configuration vector of the drill, while $\mathbf{J}_d(t) \in \mathbb{R}^{6\times\mathrm{DOF}}$ denotes the Jacobian matrix, and $\mathbf{q}_H(t) \in \mathbb{R}^{\mathrm{DOF}\times1}$ represents the state-space vector of the manipulator.

6.2.2 Expert skill

The goal of developing an autonomous drilling system was to gain a thorough understanding of how human workers perform the drilling task. This encompassed studying the motion of the drill bit within the mould frame and analyzing the drilling forces exerted on the mould. To track drilling trajectories during expert demonstration, the virtual pen device introduced in Sec. 4.2 was extended. Furthermore, the localization algorithm previously presented in Sec. 4.2.3.3 was employed, to analyze tool motion relative to the mould.

6.2.2.1 Skill recording

Similar to the demonstration setup detailed in the user-experience survey in Sec. 4.3, the process of recording drilling skill involved capturing tool motions and forces exerted on the mould. To accommodate the varying shapes, sizes, and mass of the moulds, a portable prototype for on-site recording was developed, as illustrated in Fig. 6.7. The use of linear bearings facilitated smooth motion while reducing load and sensor noise. Between the static and movable part of the holder *Optoforce* force/torque sensor was placed. Due to the linear bearings integrated into the holder, force measurement is restricted to the z axis. Nonetheless, this limitation is inconsequential when drilling the central part of the mould cavity.

Before recording the skill, pilot holes were drilled in the mould to establish consistent starting points for both the human operator and the robotic system. Using the a prior known 3D models of the moulds to create a mesh of N_h drilling tasks, denoted as $\mathcal{T}_n$ for $n = 1 \ldots N_h$, the robot was programmed to drill the pilot holes. The pilot holes were drilled to optimize hole quantity, not focusing on precisely representing the typical hole placement in the mould. The mould preparation facilitated capturing the skill of deep-micro-hole drilling and simplified the identification of holes in the recordings. Preparing for on-site recording involved several steps, including setting up the motion capture system, attaching virtual markers to the operator's tool, and performing calibration procedures. The entire preparation process took less than 20 minutes, while switching moulds required only a few minutes in between.

It is important to note that different moulds vary in terms of material type, casting method,

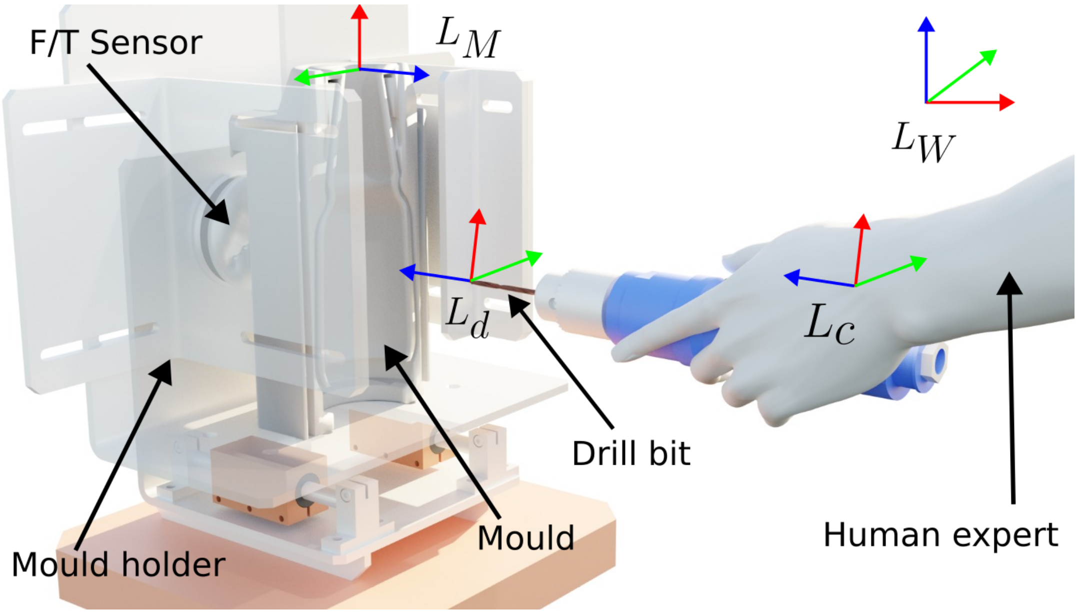


Figure 6.7: The mould holder's mechanical setup accommodates moulds of various sizes and weights. It includes linear bearings that counterbalance the mould's weight, ensuring smooth movement during drilling. After localizing the mould's frame L_M within the world frame L_W, and calibrating the tool's frame L_d in carrier frame L_c, the drill was continuously tracked to record the expert's motion.

and shape. Consequently, different drilling strategies are employed in DMH drilling. To capture a comprehensive range of drilling strategies, the expert demonstration recording encompassed five different moulds, each representing a specific material type commonly used in mould production. Two drilling bit sizes, 0.5 mm and 0.6 mm, were recorded during the experiments.

On-site scan matching and localisation

To capture the worker motion in a most convenient way, the optimal mould scanning procedure needed to be established, taking into account the complexity, accuracy and speed of obtaining a scan. For this reason, virtual pen device presented in Sec. 4.2.3 was utilized to enable easy scanning simply by drawing a shape on the surface of the mould. Later, to localize the mould in the motion recording coordinate frame, localization procedure presented in Sec. 4.2.3.3 was employed. Here, a voxelized CAD model of a given mould, converted into a point cloud, is used (*mould model* M_m). Mould model is then matched against a scan of the real mould (*mould scan* M_s), and the pose of the mould is extracted. Contrary to the onsite scanning, in the robot drilling setup, scanning is performed using a line scanner mounted on the robot (see Fig. 6.5).

6.2.2.2 Processing recordings and extracting human drilling specifics

Performing on-site localization allowed us to determine precisely at which piloted hole the drilling demonstration occurred. Once the drill sequence is matched with the piloted hole n,

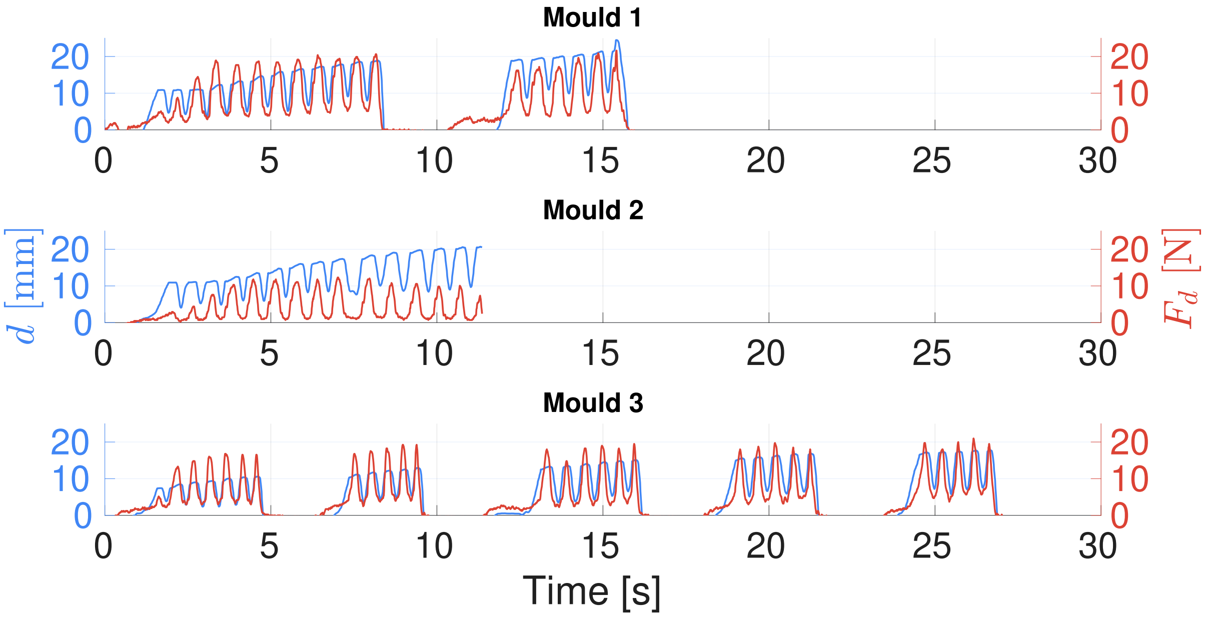


Figure 6.8: Specific drilling trajectories for several moulds drilled. The left graphs axis shows the drilling depth d, i.e. the dept that the bit penetrates the material, while the right graphs axis shows the force F_d acting on the mould. It can be seen that different moulds require different drilling strategies.

taking into account the position $P_{0,n}$ and drill direction $\mathbf{d}_{drill,n}$, the single-DOF drilling trajectory $d(t) \in \mathscr{D}_n$ is obtained using the Eq. 6.2, where $\mathbf{P}^0_{M,n}$ represents the pilot hole position, while $\mathbf{P}^d_M(t)$ represents the drill tip position in the mould frame, obtained as the translational part of the homogeneous transformation matrix $\mathbf{T}^d_M$:

$$\mathbf{T}^d_M = \mathbf{T}^{M^{-1}}_W \cdot \mathbf{T}^c_W \cdot \mathbf{T}^d_c. \tag{6.4}$$

Here, $\mathbf{T}^M_W$ denotes the mould frame pose obtained via localization, $\mathbf{T}^c_W$ is the carrier frame transformation acquired through the *Optitrack* motion capture system, and $\mathbf{T}^d_c$ refers to the tool calibration matrix. Therefore, the primary focus of capturing the drilling skill was set on the drilling depth variable $d \in \mathscr{D}$, at the same time using the drilling force profile to identify the drill bit breaking point.

Selected drilling trajectories $\mathscr{D}_n$ recorded for three distinct moulds are shown in Fig. 6.8. Notably, these trajectories differ in duration, the number of peck cycles, and how they are grouped. Upon analysing the recorded trajectories and after gathering operator feedback, the conclusion was that harder materials require a lower penetration rate (depth of cut in a peck cycle), leading to a higher number of peck cycles. To manage an increased number of peck cycles, the operator organizes them into groups, typically consisting of up to 5 cycles. The pause between these groups is used to cool down the drill bit. In terms of control and skill replication, this segment is both the fastest and the most challenging in the trajectory. The drilling force in this skill ranges from 10 N to 20 N, depending on the specific material and drilling scenario.

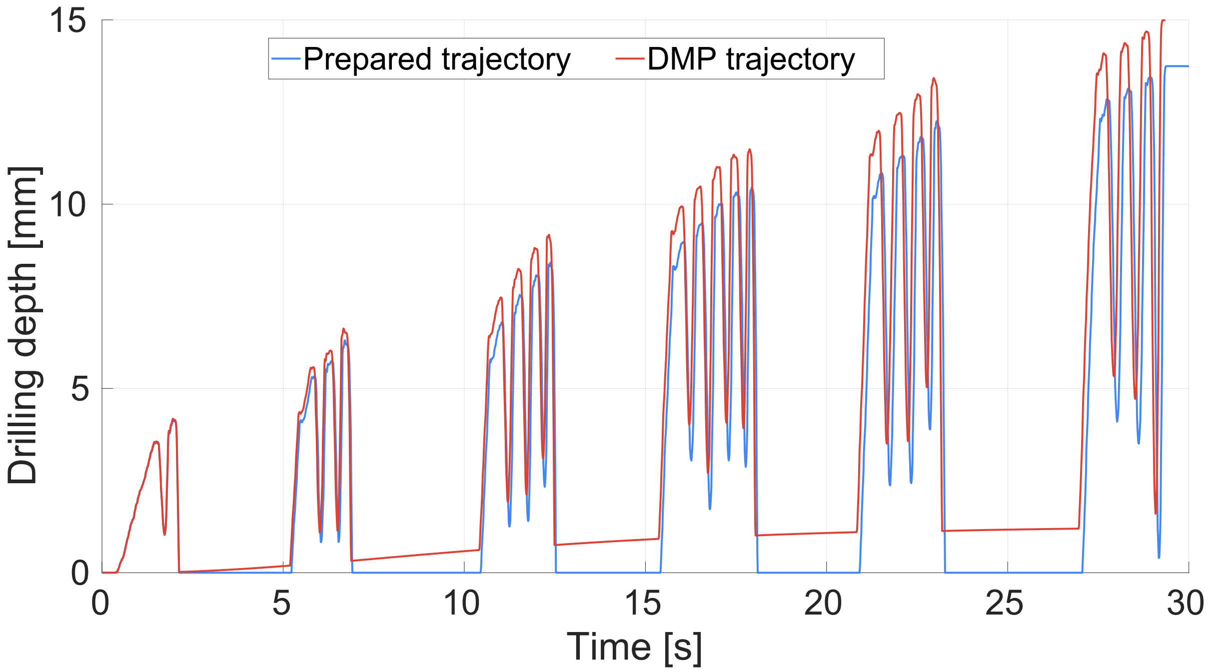


Figure 6.9: To replicate expert skills to robot, recorded demonstration were transformed into a 1-DOF drilling trajectory, and filtered it to establish initial conditions. The prepared trajectory, represented by the blue line, served as input for the DMP algorithm. The resulting drilling motion for the target depth of 15 *mm*, generated by the DMP is shown by the red trajectory.

6.2.2.3 Replicating expert skill

The analysis of the on-site demonstration revealed remarkable consistency in drilling motion performed by the operator. However, significant discrepancies in terms of drilling parameters (i.e. drilling speed, feed rate, etc.) were found, compared to standard CNC drilling . For instance, the operator's feed rate reaches up to 300 mm/min during on-site recordings, whereas the typical range in CNC drilling is $60-80$ mm/min. While the operator achieves a high level of efficiency and precision through tactile feedback, their motion, particularly in the drilling direction (depth d), is challenging to replicate using conventional trajectory planning algorithms. Moreover, the robotic system must be capable of adjusting drilling motions dynamically, including duration and drilling depth. To replicate the drilling motion, addressing the aforementioned challenges in terms of feed rates and drilling trajectories, the *Dynamic Motion Primitives (DMP)* algorithm presented in Sec. 4.4.2 was utilized.

Utilizing the DMP algorithm allowed for the adjustment of duration and drilling targets. For simplicity, drilling trajectory targets were set in depth increments of 5 mm. The generated drilling trajectories are depicted in Fig. 6.9, with the employed parameters being $K_{\mathscr{D}} = 10.000$, $D_{\mathscr{D}} = 400$, and $N_b = 4$. It is noteworthy that only the drilling depth variable was utilized for replication on the robot, while the drilling force was controlled using the impedance-based drilling controller, as later presented.

6.2.2.4 Drill Controller

To execute the robotic drilling, a single-degree-of-freedom drill controller is employed, as detailed in Sec. 3.2.4. This controller operates with d_{ref} as the reference (representing drilling depth) commanded from the DMP trajectory generation algorithm, while d serves as the control value. The controller integrates an impedance filter F_{imp} and force feedback F_{sens} to maintain the desired motion (d_{ref}) using a PD controller. Through the impedance filter and sensor feedback, the system promptly responds to external forces F_{sens}, ensuring the desired motion d_{ref} is maintained. Following the computation of the PD control value, forward dynamics motion is simulated utilizing the virtual mass coefficient M. This motion is then translated into the robot joint space through motion error calculation and the Jacobian matrix.

6.2.3 Robot Low-level Control

In the experimental setup detailed in Ch. 6, the *KUKA KR10* robot is utilized. Alongside the robot, the control unit *KR C4 Compact* is employed to control the robot, featuring the *Robot Sensor Interface (RSI)* used for integration with the *Robotic Operating System (ROS)*. The RSI offers two motion interfaces: *Absolute* and *Relative* joint motions. In *Absolute* mode, the RSI brings the joints to the specified target position within each duty cycle and stops the motion. On the other hand, *Relative* mode allows the RSI to move the joints by a defined increment without stopping, requiring appropriate control for each joint. While *Absolute* mode is generally recommended, fast drilling movements can result in oscillatory robot motion and significant displacement of the tool tip from the intended linear trajectory.

6.2.3.1 Simplified kinematics model

Since the drilling motion arises from the rotations of all joints, initial analysis focuses on understanding how the dynamics of individual joints impact the overall dynamic behaviour of the drill. As mentioned earlier in Eq. 3.17, the drill controller provides joint references $\mathbf{q}_{ref}$ to ensure proper axial motion, for small drilling motion d. Yet, non-axial motion between the joint configurations of two control cycles $\mathbf{q}_{ref}(k)$ and $\mathbf{q}_{ref}(k+1)$ is observed. In this section, the cause of this effect is addressed, along with strategies to mitigate it.

Without loss of generality, firstly a simplified forward kinematics model is derived to understand how joint dynamics contribute to tool motion in a representative drilling robot configuration. In this configuration the drill bit aligns with the robot's base negative Z axis, as illustrated in Fig. 6.10, and is primarily driven by joints A2, A3 and A5. Using existing joint controllers on the robot, commanding drilling trajectory along the Z axis generates significant deviation of the drill bit in the X axis, as illustrated with green path in Fig. 6.10. The effect is exaggerated for visualization purposes, but it is important to note that even the offset of the diameter of drill

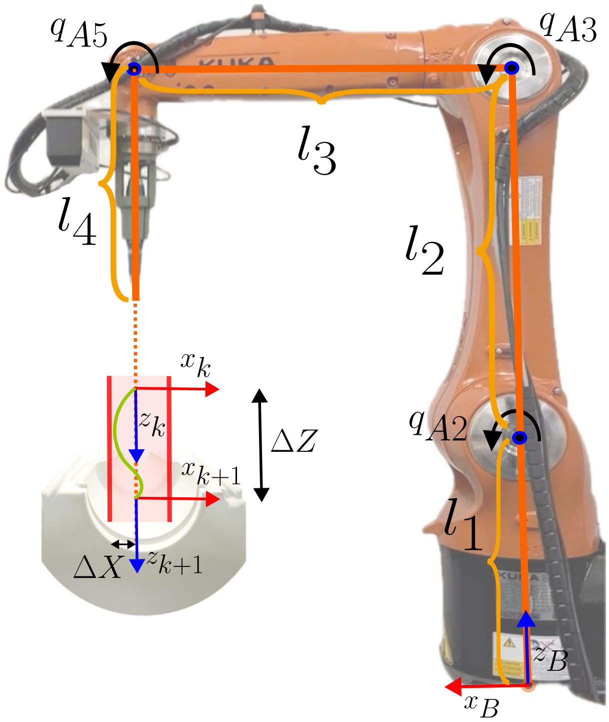


Figure 6.10: Simplified kinematics model of the robot employed to analyze the impact of dynamic responses of joints $A2$, $A3$, and $A5$ on the non-axial motion of the drilling tool in the drilling robot configuration. The effect of non-axial motion is exaggeratedly depicted in green, highlighting significant tip offset during fast motion changes. In drilling scenarios, this non-axial motion, represented by ΔX, is critical as it can cause the bit to break.

bit (≈ 0.5 *mm*) usually causes drill bit to break. To track the drill position with respect to the base frame and calculate this offset, the forward kinematics model is derived as:

$$\begin{bmatrix} w_x \\ w_z \end{bmatrix} = \begin{bmatrix} l_2 sin(q_2) + l_3 cos(q_2+q_3) - l_4 sin(q_2+q_3+q_5) \\ l_1 + l_2 cos(q_2) - l_3 sin(q_2+q_3) - l_4 cos(q_2+q_3+q_5) \end{bmatrix}, \tag{6.5}$$

where $l_{1,\dots,4}$ represent the lengths of the links. From Eq. 6.5, Jacobian matrix $J_{\mathbf{q_0}} = \frac{\delta \mathbf{w}}{\delta \mathbf{q}}$ can be derived as:

$$\begin{bmatrix} v_x \\ v_z \end{bmatrix} = \begin{bmatrix} l_2 - l_4 & -l_4 & -l_4 \\ -l_3 & -l_3 & 0 \end{bmatrix}_{\mathbf{q_0}} \cdot \begin{bmatrix} \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_5 \end{bmatrix}, \tag{6.6}$$

describing the mapping of joint velocities to velocities of the drill tip for given robot's pose $\mathbf{q_0} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$.

6.2.3.2 Joint Dynamics and Drill Motion

To inspect how the dynamics of the joints affect the dynamics of the drill bit, firstly open-loop system identification for joints A2, A3 and A5 is conducted. The open-loop behaviour of each

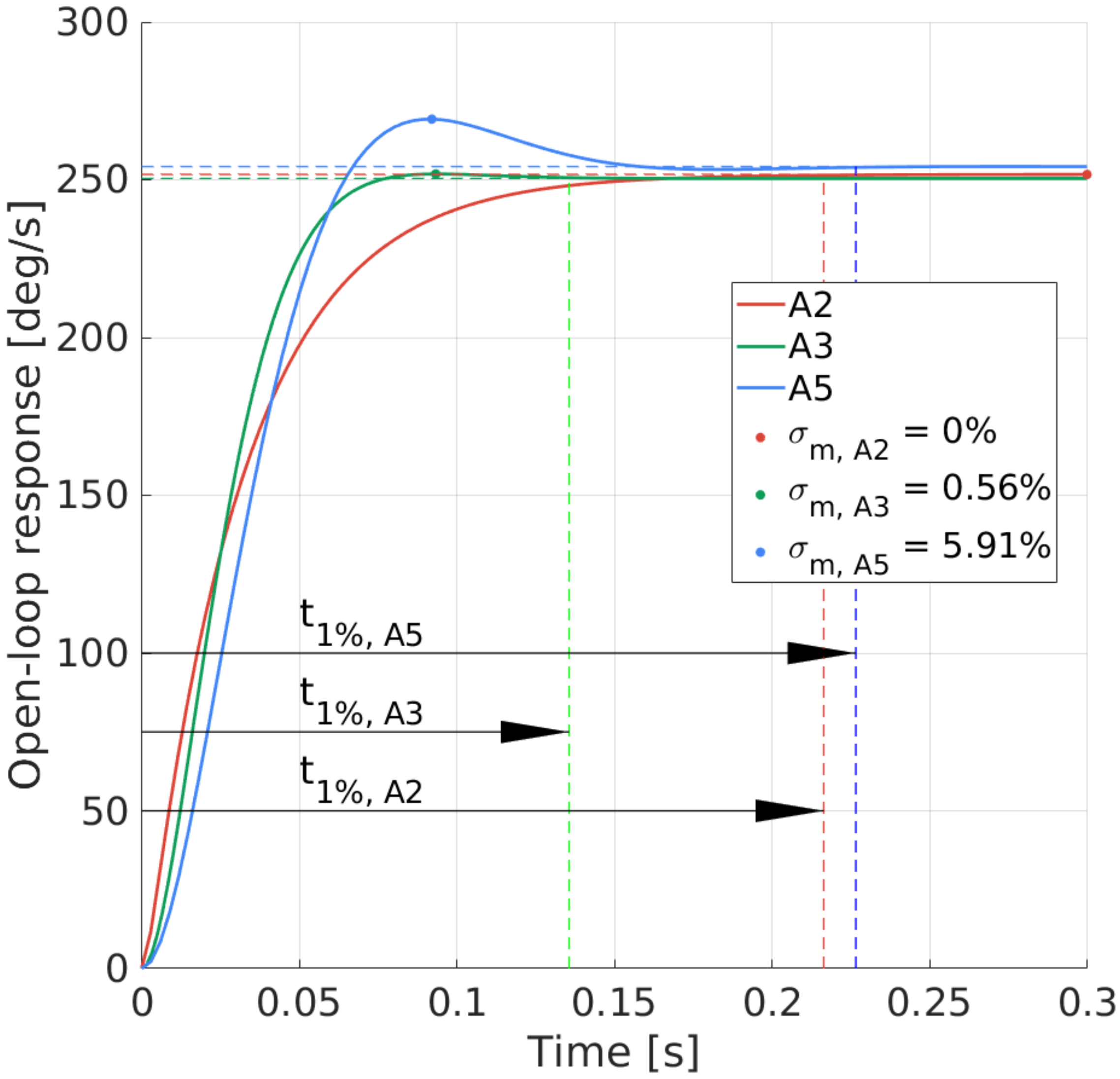


Figure 6.11: Open-loop step response for joints $A2$, $A3$ and $A5$.

joint is modeled as a third-order system:

$$G_o(s) = G_{An}(s) \cdot \frac{1}{s} = \frac{K_{An}}{\frac{1}{w_{n,An}^2}s^2 + \frac{2\zeta_{An}}{w_{n,An}}s + 1} \cdot \frac{1}{s}. \tag{6.7}$$

To identify system parameters K_{An}, $w_{n,An}$ and ζ_{An}, for each joint $An \in \{A2, A3, A5\}$, the dynamic response of each joint was recorded using a chirp signal. Comparing the recorded responses (Fig. 6.11) to model output and applying Nelder-Mead Simplex optimization method [81], the model parameters for each joint were obtained. Identified parameters are shown in Tab. 6.2, while step responses are shown in Fig. 6.11.

After conducting open-loop system identification, the next step is to analyze how joint dynamics influence drill motion. To model the drill motion as a function of joint dynamics, the derived Jacobian matrix in Eq. 6.6 is used to describe the behaviour of the tool tip in the robot's initial position based on the small drill reference d in the Laplace domain:

$$\begin{aligned}\Delta X(s) &= (l_2 - l_4)\cdot\Delta q_{A2}(s) - l_4\cdot\Delta q_{A3}(s) - l_4\cdot\Delta q_{A5}(s)\\ \Delta Z(s) &= -l_3\cdot\Delta q_{A2}(s) - l_3\cdot\Delta q_{A3}(s),\end{aligned}$$

where $\Delta X(s)$ and $\Delta Z(s)$ represent the dynamic response of the tool as the offset from initial position in x and z axes, respectively, while the response of each joint is modelled as:

$$\Delta q_{An}(s) = G_{cl,An}(s)\cdot q_{An,ref}(s), \tag{6.8}$$

with $G_{cl,An}(s)$ being the transfer function of the closed loop for joint An. Now, the expression for $\Delta X(s)$ can be rewrited as:

$$\begin{aligned}\Delta X(s) =&(l_2 - l_4)\cdot G_{cl,A2}(s)\cdot q_{A2,ref}(s)\\ &-l_4\cdot G_{cl,A3}(s)\cdot q_{A3,ref}(s)\\ &-l_4\cdot G_{cl,A5}(s)\cdot q_{A5,ref}(s).\end{aligned}\tag{6.9}$$

Recalling that $q_{An,ref}$ is calculated using Eq. 3.17, it is known that the following holds:

$$0 = (l_2 - l_4)\cdot q_{A2,ref} - l_4\cdot q_{A3,ref} - l_4\cdot q_{A5,ref}. \tag{6.10}$$

From Eqs. 6.9 and 6.10, two solutions for $G_{cl,An}$ can be identified, ensuring that $\Delta X(s) = 0$, where the solution $G_{cl,An}(s) = 0$ can be discarded. The second solution that ensures $\Delta X(s) = 0$ is:

$$G_{cl,A2}(s) = G_{cl,A3}(s) = G_{cl,A5}(s) = G_{cl,A}(s), \tag{6.11}$$

since then the Eq. 6.9 can be factored into:

$$\Delta X(s) = G_{cl,A}(s)\cdot\left[(l_2 - l_4)\cdot q_{A2,ref}(s) - l_4\cdot q_{A3,ref}(s) - l_4\cdot q_{A5,ref}(s)\right]. \tag{6.12}$$

From Eqs. 6.10 and 6.12, it is clear that the proposed drill controller should ensure minimal offset from the drilling axis if the dynamics of the closed control loops of robot joints are the same. One cannot ensure zero offset in the X axis since the aforementioned analysis holds only for infinitesimally small offsets, since as soon as the robot moves, the Jacobian matrix changes. To numerically validate this result, a model transfer function for the closed control loops of robot joints was designed to simulate the dynamic response of the tool during drilling. A uniformed joint dynamics response was simulated, where all six joint dynamics were modeled with the following parameters: $t_{1\%} = 120\ ms$ and $\sigma_m = 5\%$. The dynamic response of joints to a chirp reference signal is mapped into dynamic response of the tool by feeding the resulting joint values to the forward kinematics model of the whole robot, as shown in Fig. 6.11a. The response for this idealistic case is shown in the top plot of Fig. 6.11b, showcasing that, if the dynamic response of all joints is the same, there is virtually no offset from the drilling axis in the X direction.

However, the idealistic tool motion generated by the same closed loop behaviour of all joints is not attainable in a realistic scenario. To investigate the effect the difference in responses of joints have on the non-axial motion of the tool, three additional simulation experiments were performed, results of which are shown at the bottom plot of Fig. 6.11b. In each of these simulations, the model closed loop transfer function of one of the dominant joints in the drilling configuration (A2, A3, and A5) was altered by extending it's settling time by 1 *ms*. During the experiment the offset from the drilling axis was evaluated.

From Fig. 6.11b, it can be observed that slowing down any of the joints introduces the undesired offset in the X axis, confirming that the motion of the tool is best aligned with the drilling axis if the dynamics of the joints are the same. Interestingly, delaying joint A2 has significantly smaller effect on the amount of ΔX, which can be explained by observing the Jacobian matrix in Eq. 6.6. Since $|l_2 - l_4| << |l_4|$, it is evident that the motion of joint $A2$ has a relatively minor impact on tip motion in the X axis compared to joints $A3$ and $A5$.

From this analysis, it can be concluded that the non-axial motion of the drill bit is minimized if the dynamics of joints are similar, and that the motion of the joint A2 has comparatively small effect on this undesired behaviour. With these conclusions in mind, the focus is shift to design of the joint controllers.

6.2.3.3 Joint Controllers

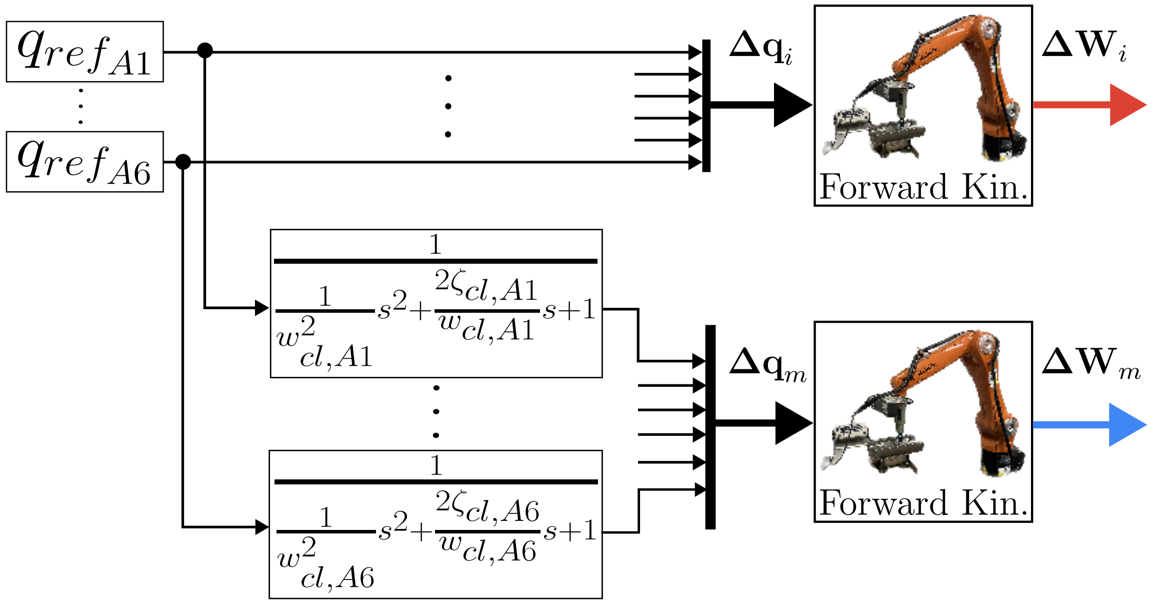
To optimize response times and improve joint synchronization, a cascade joint control system was implemented on top of the Relative mode in the RSI control program. Fig. 6.12 illustrates the cascade control schematic for a single joint, in which the closed loop transfer function can be written in the following form:

$$G_{cl,An}(s) = \frac{G_{PD,An}(s) \cdot G_{An}(s)}{1 + G_{PD,An}(s) \cdot G_{An}(s)}, \tag{6.13}$$

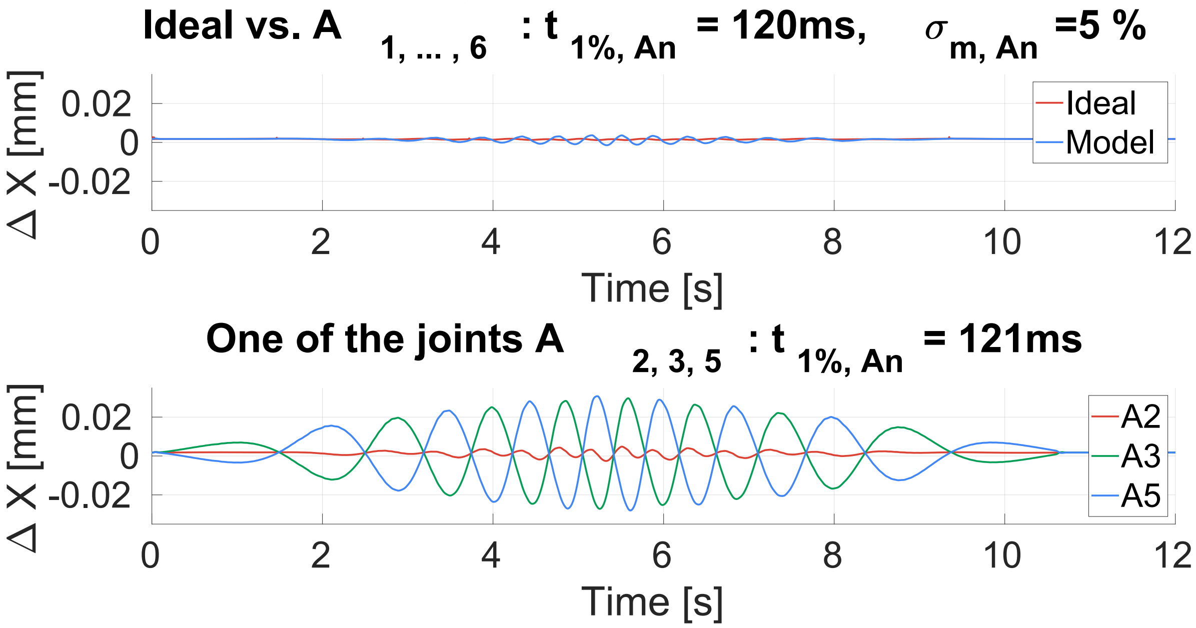
where $G_{An}(s)$ is open-loop model for joint An, and $G_{PD,An}(s)$ is the controller transfer function.

The control system is based on the discrete PD-Controller provided within the RSI environment and has the following transfer function:

$$G_{PD}(z) = \frac{(T + T_d)K_p z - K_p T_d}{Tz}, \tag{6.14}$$



(a) Simulation schematics.



(b) Simulation results.

Figure 6.11: To assess the impact of distinct joint responses on tool tip non-axial motion to the drilling axis, a simulation environment (Fig. 6.11a) was used. The tool tip's global pose $\mathbf{W}$ was compared in two scenarios: ideal joint response $\mathbf{W}_i$ and modeled joint response $\mathbf{W}_m$ for several joint dynamics configuration as shown in Fig. 6.11b.

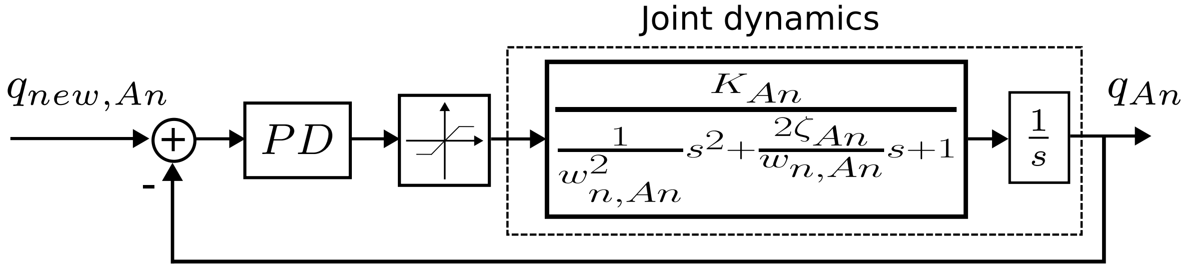


Figure 6.12: Schematic view of KUKA RSI control in Relative mode for a single joint. The joint angle reference $q_{new,An}$ is commanded via Ethernet communication from the computer running the Robot Operating System (ROS). The actual position of the joint and the PD-controller are provided within the RSI interface. The transfer function of the PD-controller is derived from the recursive form of the RSI PD-Controller.

Joint	K	ω_n	ζ	K_p	T_d
A2	251.7050	136.5076	68.2538	0.122	0.003
A3	250.4301	64.6105	32.3052	0.1482	0.0146
A5	254.1584	46.2180	23.1090	0.1451	0.0177

Table 6.2: Parameters of RSI Relative mode, both identified parameters of the open-loop and controller parameters for the closed-loop system.

with K_p being proportional gain, T_d derivative time constant, and $T = 4ms$ the control cycle time. To design the controller for each of the joints, firstly the joint model transfer functions are designed.

When designing the model transfer function for each joint, the aim was to make each joint as fast as possible to achieve the tool velocity required to replicate feed rates of the human worker drilling. The closed-loop controllers for joints $A3$ and $A5$ were able to match a second-order model with settling time $t_{1\%} = 120\ ms$ and overshoot $\sigma_m = 5\%$. However, joint $A2$ was unable to reach similar dynamics due to physical limitations of the robot and needed a slightly slower response in the model function. Having in mind that the dynamics of the joints should be similar to minimize non-axial movement, the nominal action should be to increase the settling time of model functions for joints A3 and A5. Recalling that the difference in dynamics of joint A2 has a significantly smaller effect on the non-axial motion of the tool compared to A3 and A5, a compromise was made to sacrifice a small amount of non-axial motion for the overall faster response of the tool. As a result, the model function for joint A2 was designed to be different from A3 and A5, with a settling time of $t_{1\%} = 180\ ms$ and overshoot $\sigma_m = 5\%$ in joint $A2$.

Utilizing the pole-placement method, the controller parameters K_p and T_d were determined based on the identified open-loop behaviour and the desired model transfer functions for each joint. All parameters are shown in Tab. 6.2.

6.2.3.4 Low-level control: Experimental validation

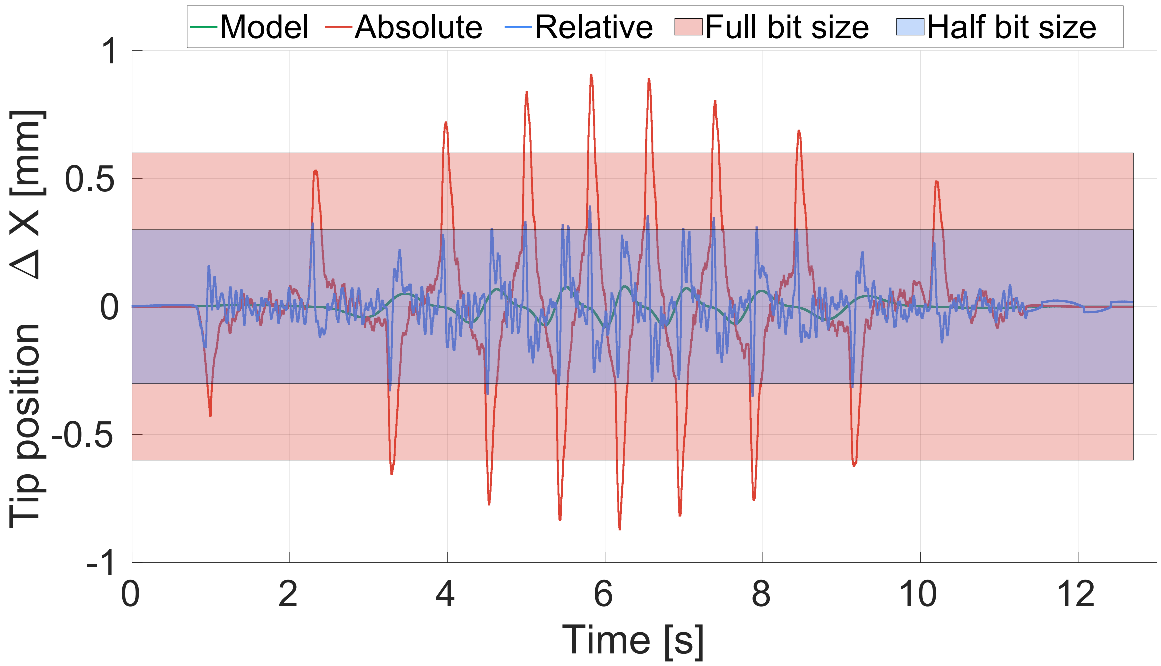
To evaluate the improvement of the proposed robot low-level control system, again, a chirp signal was while drilling along the $-Z$ axis of the robot base was used to compare the system's response in both *RSI Absolute* and *RSI Relative* modes. The chirp signal parameters were chosen to encompass a tip feed rate ranging from the minimum to the maximum typically observed in drilling scenarios. As shown in Fig. 6.13, when the robot is controlled in *RSI Absolute* mode, a significant displacement of the tip in the X axis is observed. This motion exceeds the drill bit diameter, making it almost impossible to drill as the bit tends to break due to radial displacement. Deploying the proposed control approach in *RSI Relative* mode reduces the tip's deviation in the X axis to half the drill bit diameter. Additionally, it is worth noting that no motion is observed in the Y axis due to the robot's kinematics in the selected joint configuration.

6.2.4 Experimental validation

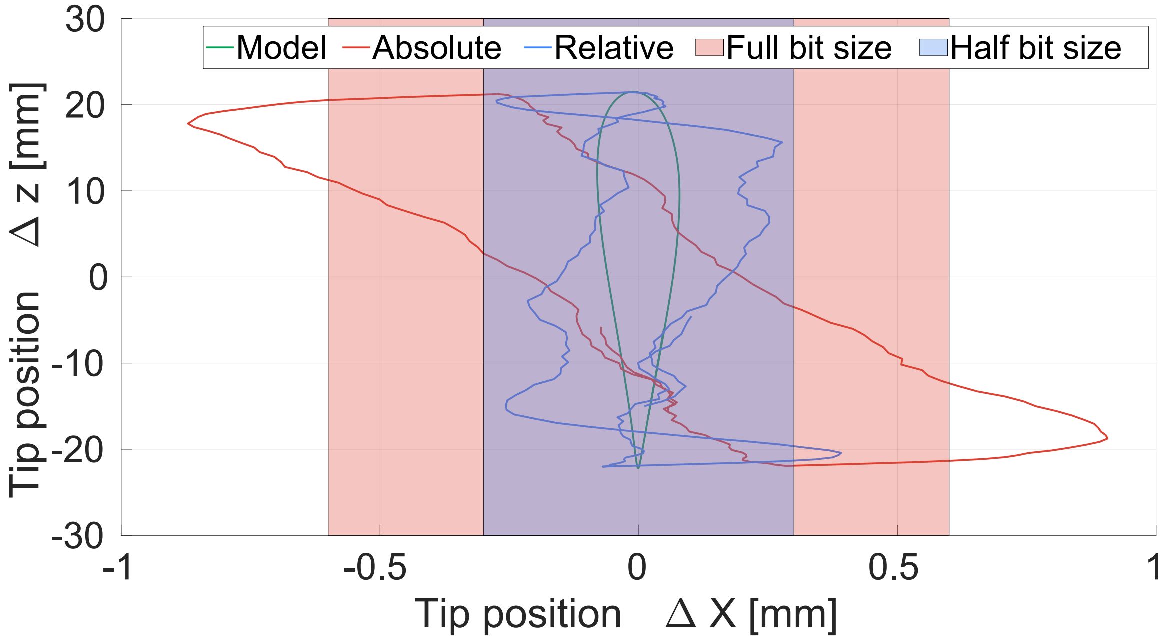
In the experimental validation of the proposed methods and algorithms, the first objective was to reproduce the recorded human drilling skill on the robotic system while maintaining a similar duration and depth. For simplicity, initiated the experiments with depths of 10 mm and 15 mm, adjusting them according to the particular mould type. Each mould type that was included in the human skill recording was subsequently drilled by the robot using the DMP trajectory tailored for that specific mould type. To evaluate the effectiveness of the drilling process, the number of holes that could be drilled with a single drill bit was quantified. This metric served as an indicator of the overall performance and efficiency of the robotic drilling system.

The results of the experiments, conducted on five different mould types, are summarized in Tab. 6.3. Furthermore, the performance of the human operator is compared to the robotic system for the same mould for which the DMP trajectory was presented in Fig. 6.9. In the visual representation in Fig. 6.14, piloted holes are distinguished (in red), along with holes drilled by the operator (in green), and those completed by the robot (in blue). The results clearly demonstrate that the robotic system consistently surpasses the human operator in terms of hole quantity for each mould type. In most instances, the robotic system achieves at least double the number of holes compared to the operator, with some cases exceeding seven times the operator's hole count.

The programmed impedance behaviour of the system is apparent in the difference between the commanded and the achieved depths, shown in Fig. 6.15. It should be noted that the system may not achieve the desired goal depth due to the spring-like behaviour of the impedance filter. However, since the DMP trajectory goal can be easily adjusted, the desired depth can be iteratively attained. The drilling feed rates range up to 300 mm/min, while the service feed rates typically reach up to 3.000 mm/min. To prevent drill breakage and the drill tip getting stuck



(a) Tip position responses in time.



(b) Tip position responses in XZ plane.

Figure 6.13: Evaluating joint responses' impact on tool tip motion. Ideally, motion should be purely in the Z-axis, but in RSI Absolute mode, substantial motion in X-axis of drilling is observed (red line), exceeding the drill diameter. This motion is significantly reduced, roughly half the drill bit's diameter, using closed loop RSI relative mode (blue line).

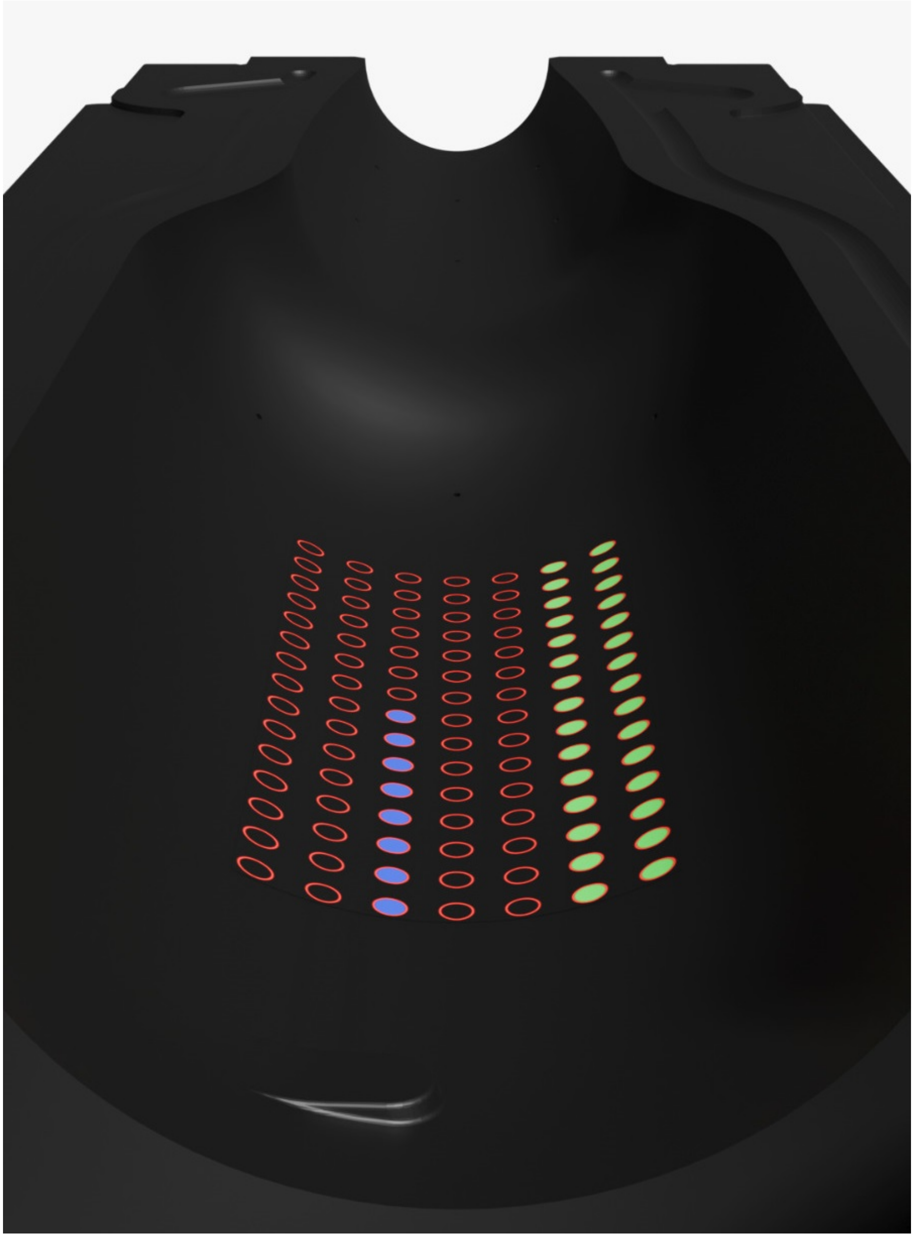


Figure 6.14: Illustrating robotic system performance: Red circled points are pilot holes ($1mm$ deep) optimized for experiment consistency and post-analysis hole localization. Blue filled dots indicate operator-drilled holes, and green filled dots represent holes completed by the robot.

	Operator performance	Robot performance
Mould 1	up to 25	more than 120
Mould 2	8	60
Mould 3	8	20
Mould 4	4	19
Mould 5	7	10

Table 6.3: Performance comparison between the human operator and robotic system, presented as the number of drilled holes per mould type.

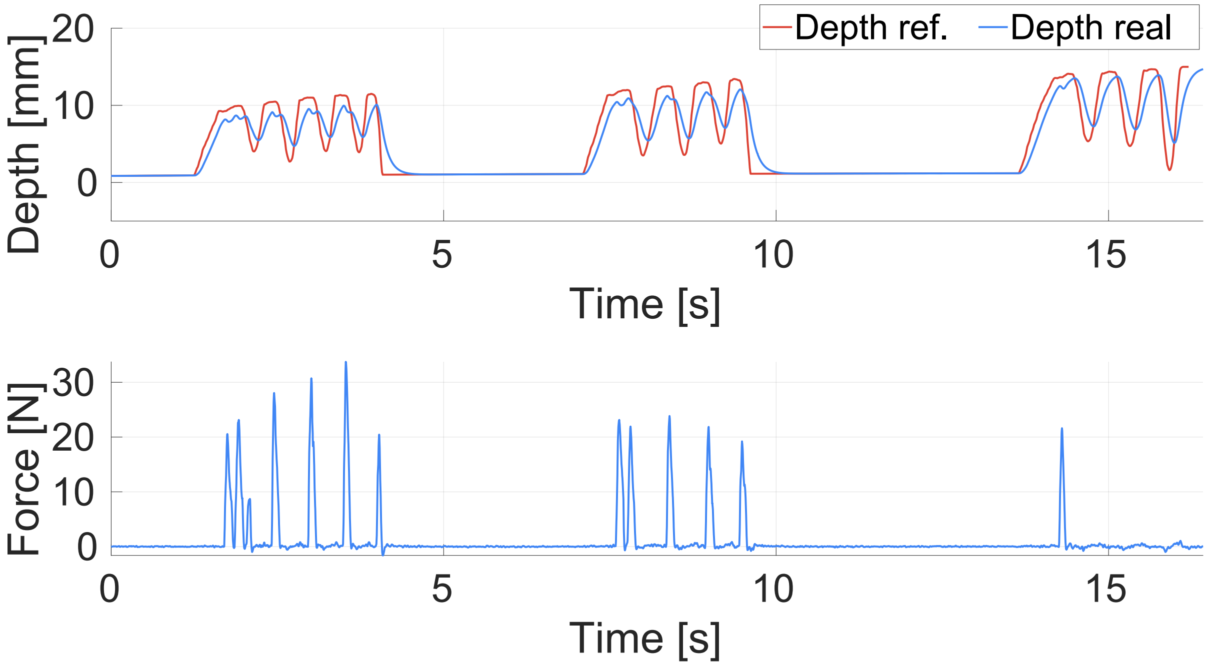


Figure 6.15: Drilling of a 15 *mm* hole with the robot using the DMP obtained from human motion recording. The drilling feed rates typically reach up to 300 mm/min, while the service feed rates typically reach up to 3.000 mm/min. The drilling forces reach up to $30N$.

in the hole, the drilling process is carefully monitored, and if the drilling force exceeds a given threshold, the process can be canceled.

In the second stage of the experiment, the goal was to drill moulds with similar material properties to those used in the initial recording. The objective was to determine if a single DMP trajectory could be effectively used across different mould shapes and sizes. This enabled to assess the adaptability and efficiency of the proposed system to moulds that were not previously drilled by the operator. The results demonstrate that the robotic system is able to achieve a comparable number of drilled holes to that of the original mould, indicating the adaptability and effectiveness of the transfer learning approach in the robotic drilling process.

6.3 Plastering

This section aims to extend the capabilities of the proposed collaborative framework to another application, in particular, to apply a coat of putty using a standard knife tool, as depicted in Fig. 6.16. Based on the experience of human operators, the plastering process is controlled both with the contact force and the angle between the knife tip and the treated object. By varying these two variables, the robotic system can perform different plastering strategies, such as applying a thicker layer, or removing excess plaster. However, the plastering knife is a flexible tool (deformable object), undergoing physical deformation under application of force, which significantly complicates the problem of coupled force and angle control. The domain of deformable

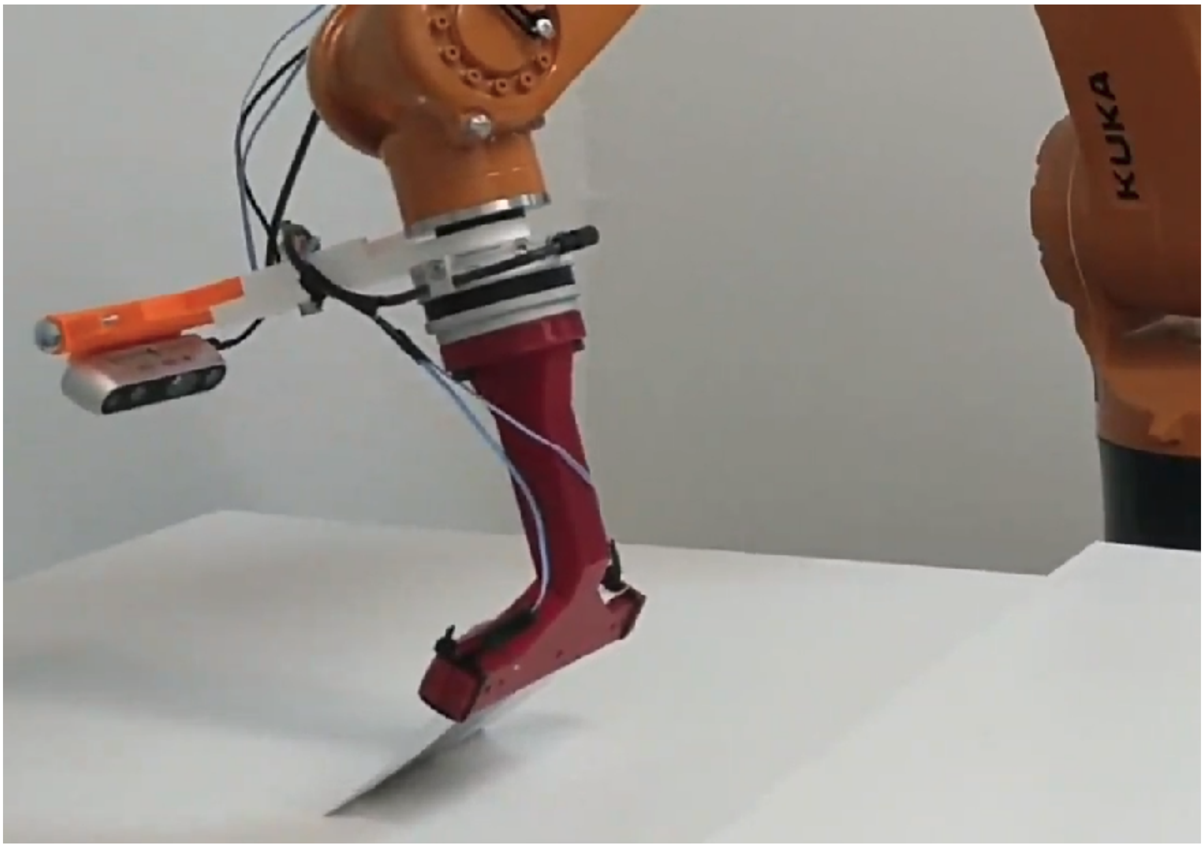


Figure 6.16: Flexible plastering knife tool is mounted on the robot end effector, along with the Intel RealSense RGB-D camera and a torque sensor. Plastering is an example of an industrial task that involves deformable object manipulation.

object manipulation is becoming ever more interesting with the development of sensing and computational capabilities. As opposed to rigid body manipulation, the position control of deformable objects is not such a straight-forward task since their shape is affected by physical interactions. Recent research results related to robotic plastering [82] show a mechanical setup that ensures the tool remains in a predefined contact with the wall. This system, deploying artificial neural network to process walls and plan the procedure, shows promising results, but is focused on large, flat wall surfaces.

6.3.1 Shape modelling and control

As mentioned earlier, two variables are important in plastering: applied force and the angle between the knife tip and the treated object. Due to the deformable nature of the plastering knife, special care must be taken into account when estimating the shape of the plastering knife. In this work *KUKA KR 10* robot was equipped with one such off-the-shelf available knife tool in a setup shown in Fig. 6.16. The mount of the tool attaches to the flexible knife and an *Intel RealSense* RGB-D camera, mounted so that it captures the complete plastering knife within the camera frame. A torque sensor is deployed to measure contact forces with the treated surface.

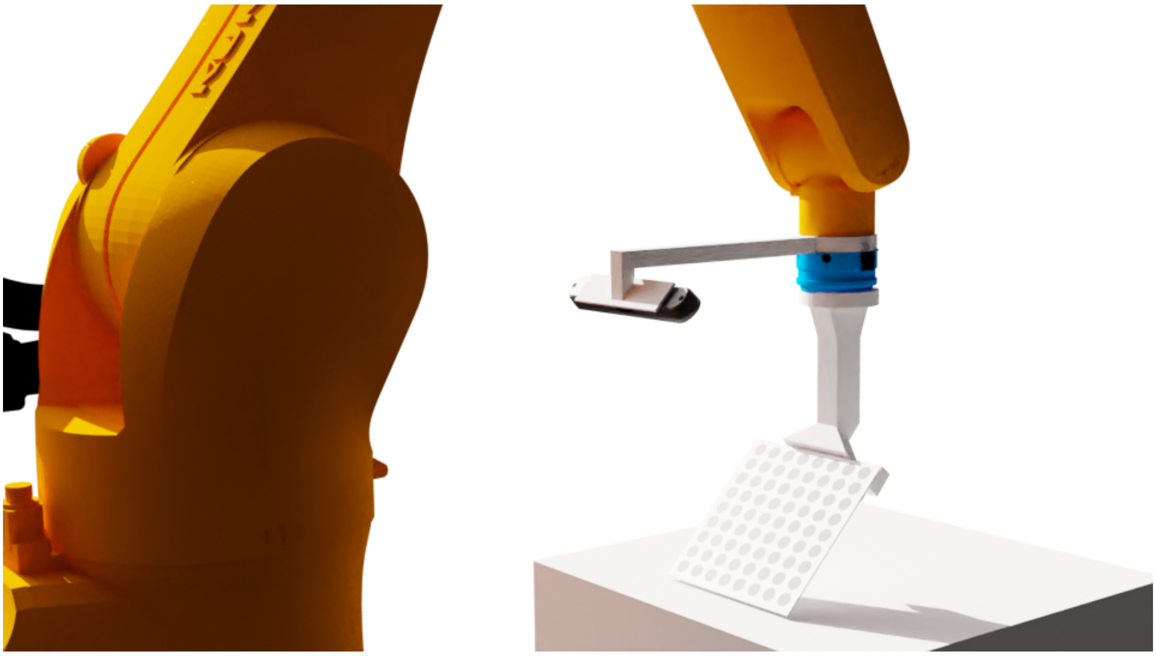


Figure 6.17: Flexible knife tool is mounted on the robot end effector along with the sensory aparatus consisting of an Intel RealSense RGB-D camera and a torque sensor.

Inspired by the results from visual based tactile sensing development [83], the plastering tool was enriched with visual cues in the form of regular grid of dots, as showed in Fig. 6.17, for easier deformation modeling through image analysis.

To ensure complete surface coverage, high level control layers generate the desired tool trajectory and the desired force. These control variables, the applied contact force and the shape of the knife tool, are coupled due to the flexibility of the plastering knife. Namely, the shape of the tool is a result of the contact force acting between the surface and the knife. The shape of the knife is described with the relative position of the tip with respect to the robot flange, and the inclination angle of the knife tip with respect to the treated surface, i.e. Δz, Δx and $\Delta \phi$ as shown in Fig. 6.18. The trajectory thus defines desired knife tip waypoints $\mathbf{T}_{d}{}^{K}_{B}$ that describe the poses and inclination angles of the knife, and the desired force, applied to the surface along the local surface normal vector $\mathbf{n}_B$.

The control system for robotic plastering is decoupled into two subsystems: the knife inclination control, and the impedance based controller presented in Ch. 3. In this section, the focus is on the knife inclination angle control, based on robot flange rotation to provide the desired inclination of the knife regardless of the contact force.

Knife inclination control

Planned trajectories rely on the known shape of the treated surface, and an optimal shape of the knife. To make sure that the desired knife tip pose is tracked, the following control strategy is proposed. Starting from knife shape shown in Fig. 6.18, the knife tip angle $\Delta \phi$ is considered

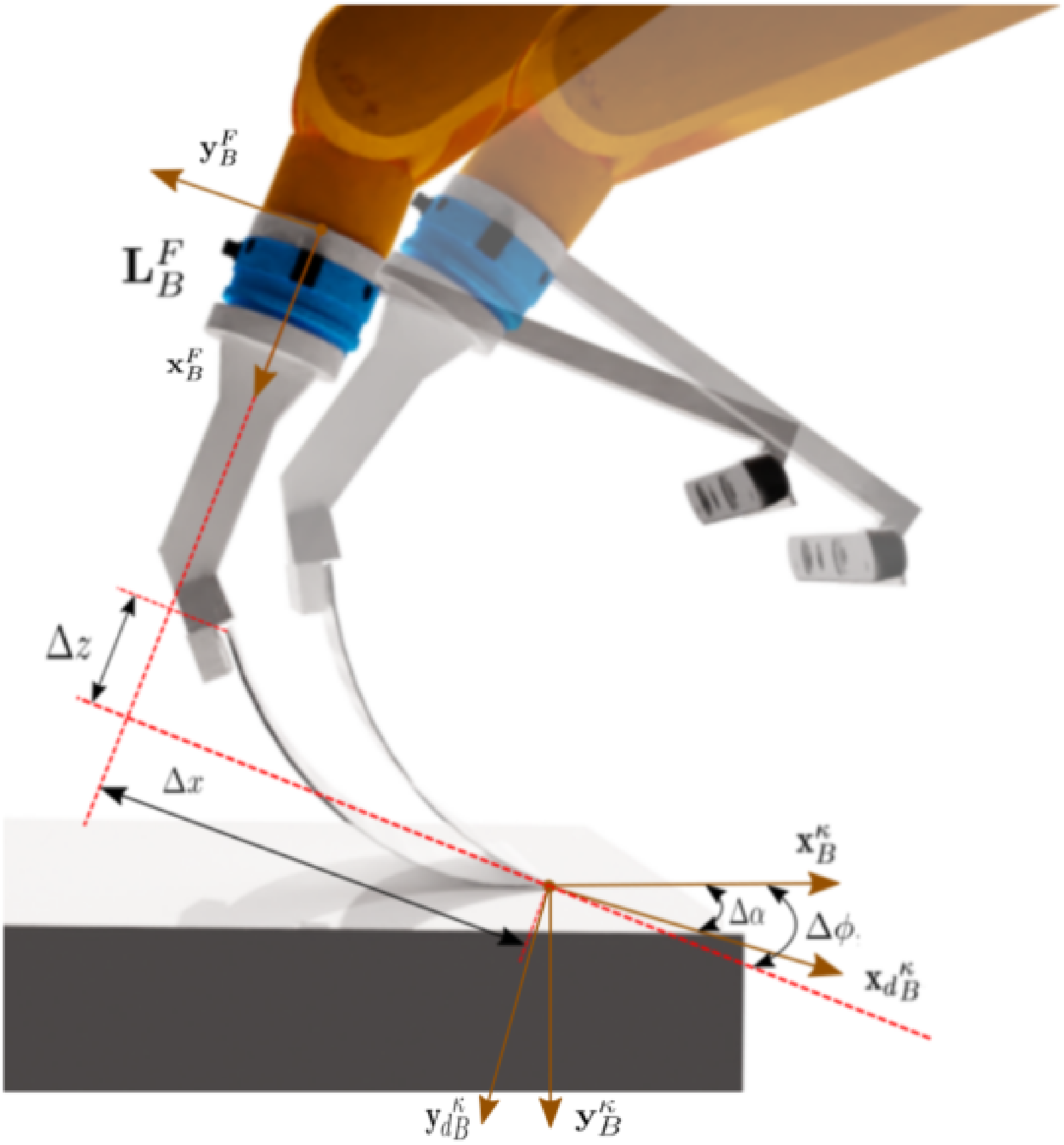


Figure 6.18: Knife shape is estimated with deflection variables ΔX, ΔZ, $\Delta\phi$. The estimated and desired approach vectors of the knife tip are $\mathbf{x}_B^\kappa$ and $\mathbf{x}_{dB}^\kappa$, respectively. The controller compensates the angle error $\Delta\alpha$ by rotating the robot flange.

w.r.t. the fixed part of the knife tool. The current and desired knife tangents are denoted as $\mathbf{z}_{B}^{\kappa}$, and $\mathbf{z}_{d}{}_{B}^{\kappa}$, respectively. The control strategy is to rotate the robot flange $\mathbf{L}_{B}^{F}$ to align $\mathbf{z}_{B}^{\kappa}$ with desired $\mathbf{z}_{d}{}_{B}^{\kappa}$, while keeping the knife tip $\mathbf{p}_{B}^{\kappa}$ at the same position. In other words, the robot flange is to be rotated around the estimated knife tip position by an angle $\Delta\alpha$, calculated using the small angle approximation of the vector product between the desired and the actual orientation:

$$\Delta\alpha \sim \frac{\|\mathbf{z}_{d}{}_{B}^{\kappa} \times \mathbf{z}_{B}^{\kappa}\|}{\|\mathbf{z}_{d}{}_{B}^{\kappa}\| \cdot \|\mathbf{z}_{B}^{\kappa}\|}. \tag{6.15}$$

The calculated angle error and the axis $\mathbf{x}_{B}^{\kappa}$ are used to generate a rotation matrix $\Delta\mathbf{T} = \mathbf{Rot}(\alpha, \mathbf{x}_{B}^{\kappa})$. The new flange pose is corrected using the following nonlinear equation:

$$\mathbf{T}_{B,new}^{F}(t_{k+1}) = \mathbf{T}_{B}^{\kappa}(t_k) \cdot \Delta\mathbf{T}(\alpha) \cdot {\mathbf{T}_{F}^{\kappa}}^{-1}, \tag{6.16}$$

where $\mathbf{T}_{B}^{\kappa}$ and $\mathbf{T}_{F}^{\kappa}$ denote the desired and actual transformation the knife w.r.t. Δx and Δz.

From the control perspective, angle error $\Delta\alpha$ is a nonlinear function. Here, the discussion is limited to the linearized point around the desired shape of the knife, for surface irregularities in a single degree of freedom. For this class of problems, $\Delta\alpha$ takes the following open loop behaviour:

$$\Delta\alpha(s) = \mathbf{J}_{B}^{\kappa}(\mathbf{q}, \Delta z, \Delta x, \Delta\phi) \cdot g_{\kappa}(s) \cdot G_{CC}(s), \tag{6.17}$$

where $\mathbf{J}_{B}^{\kappa}(\mathbf{q}, \Delta z, \Delta x, \Delta\phi)$ denotes the Jacobian linearization of the robot pose including plastering knife shape, at joint angles $\mathbf{q}$. The dynamic elasticity of the knife is described with $g_{\kappa}(s)$, and $G_{CC}(s)$ describes the closed loop compliance control of the flange. The last part of the equation encompasses the dynamic behaviour of the robot arm control function, tuned as a PT2 system for a desired behaviour.

In effect, controlling $\Delta\alpha$ with a steady set zero reference value was proposed. This ensures the knife tip follows the desired tangent approach vector $\mathbf{x}_{d_B}^{\kappa}$. The system identification for the open loop transfer function Eq. 6.17 was performed. Second order approximation of the system dynamics yielded the following transfer function:

$$G_{\Delta\alpha}(s) = \frac{1.63}{0.0043s^2 + 0.1036s + 1}. \tag{6.18}$$

Knowing the linearized transfer function of the system Eq. 6.17, PI controller was designed using standard pole placement technique, resulting with the closed loop behaviour of 7% overshoot and $455[ms]$ settling time.

6.3.2 Knife shape estimate

The deformation of the plastering tool, essential for a successful task execution, is accounted for by the robot controller through a closed control loop. The three deformation features, describing the net effect of the relative knife pose and the exerted force, are obtained from a deep CNN based black-box model. The back-bone of the CNN model is the MobileNet V2 architecture, chosen for its efficiency and suitability for online inference in closed loop control. The history of CNN development is closely related to visual scene analysis, resulting in models trained for object detection and semantic segmentation. As such, it is extensively trained for classification on different benchmark datasets. Although critical to control the knife angle, knife shape estimation goes beyond of scope of this work. Here the black-box model presented in [25] was used. However, the estimation includes following steps:

- **Transfer Learning** is employed for knife angle estimation, which involved removing the classification layer of a pretrained network and adding four feed-forward (FF) layers. Here the deeper layers utilize rectified linear activation, while the output layer employs linear activation to estimate continuous deformation values with three output neurons. A training procedure is then performed on a custom dataset adjusts weights solely for the last five layers, retaining the rest of the network.
- **Dataset Generation and Input pre-procesing**: The training dataset is generated from experiments where external force deforms the tool, recorded using an *Intel RealSense* D435 RGB-D camera. Ground truth for three deformation features is extracted from point cloud recordings. Initially, a 3D point cloud is transformed into a 2D reading, representing a vertical slice of the data within a predefined patch. After filtering points based on a known knife length and identifying the knife starting point using discontinuities in the z axis, a third-order polynomial curve is fitted to the extracted knife profile. This polynomial is then used to derive the knife tip position feature, while its derivative is used to derive the orientational features at the knife tip relative to the tool mount. For pre-processing the *Intel RealSense* camera images, the process involves cropping to the area of interest, adjusting brightness and contrast, and scaling to the MobileNet's input size. Additionally, normalizing the label set of the predicted variables to the $[0,1]$ range is performed to ensure equal contribution to MSE.
- **Deployment**: Using the *MobileNet V2* architecture with an *NVidia GeForce GTX 1060* GPU in the *TensorFlow* environment achieves a measurement rate of $\sim 30-50$Hz. Estimates from the *Intel RealSense D435* camera are available at ~ 30Hz. However, the outer control loop of the industrial manipulator operates at 100Hz, necessitating up-sampling via a white-noise linear velocity Kalman filter. Despite small estimate errors during plastering task execution, the inner control loop at 250Hz is sensitive. To mitigate slight shaking motion, the Kalman estimate undergoes filtering with a simple low-pass PT1

filter.

- **Sensing results**: The transfer learning-trained network is tested on separate test data for three knife deformation features. Mean absolute errors are less than 3% of the x and z axis variable ranges, with sub-millimeter errors ($0.61mm$ and $0.93mm$). The mean absolute angular error is 1.88°, less than 4% of the variable range.

6.3.3 Experimental results

Experiments are conducted with the plastering tool on experimental surfaces, evaluating the precision of knife inclination control during the plastering procedure. The robot is provided with a position trajectory for the knife tip, as well as the desired knife inclination with respect to the treated surface. The position references are effectively filtered through the compliance controller, since the force control is turned off for the sake of knife tip pose control validation.

The control loop is closed through the neural network provided estimations of flexible tool position and inclination while in contact with the manipulated surface. The controlled variable was the knife inclination, identified as crucial in the plastering process. The other two estimated deformation features were not directly controlled, but were used in robot position reference adaptation, accounting for the tool deformation in the robot position control. The task motion was referenced along the x axis, with z axis reference only provided for realizing contact with the surface. A realistic reference of 45° for the knife inclination was commanded in the experiments.

The experiments [11] were conducted on a piece-wise-flat surface, mimicking a step change unmodeled irregularity of the manipulated surface. The measurements of the experiments in the open loop in Fig. 6.19 clearly show this irregularity at $t \approx 12s$, corresponding to the more prominent knife deformation upon coming into contact with a protruding surface profile. This step disturbance can be observed in the lower right corner of the Fig. 6.16. In addition, Fig. 6.20 shows how a tuned PI controller reaches the inclination reference. Then, at $t \approx 13.5s$, the robot reached the surface irregularity. The measured inclination error was again accounted for by the robot position control. The results in Fig. 6.20 show one randomly chosen representative experiment, out of 5 successful consecutive repetitions with the same reference.

Increased signal-to-noise ratio can be observed in the measured angle signal in the closed loop experiments. Since the system is in contact with the surface even in the case of open loop experiments, the noise should not be attributed to the neural network prediction imprecision, but to the fact that the measurement rate is lower than the robot control rate. Even though upsampled from 30Hz to 100Hz, the higher rate only provided scheduled measurements to the real time robot control. The signal filtering in the upsampling process is however not completely accurate, not taking into account the control signals. Since the amplitude of the measurement

---

[11] `https://youtu.be/RxIiJySSC0k?feature=shared`

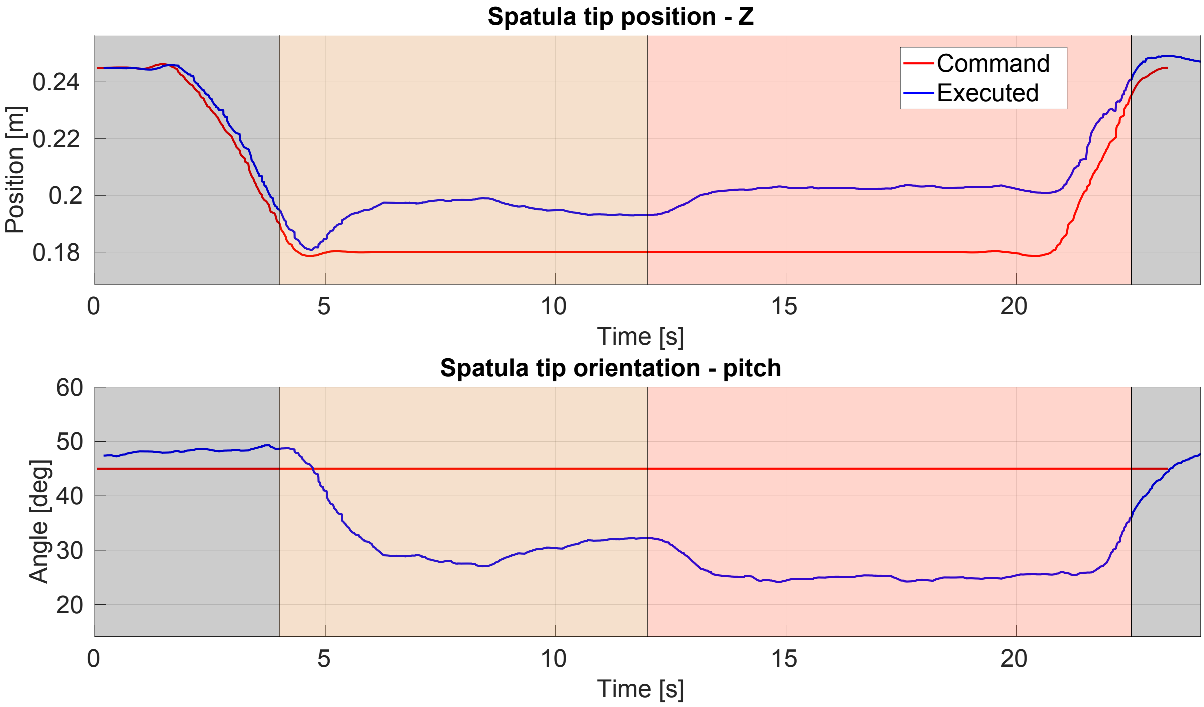


Figure 6.19: Showing global position of knife tip in robot base frame during the open-loop motion.

noise is low enough not to deteriorate the reference following, this issue was not addressed within this work, but it remains a part of the future work, since this measurement signal also results in a "shaky" behaviour of the robot during task execution.

It is worth noting that the z axis deformation is not directly controlled, and the inclination error compensation results in a larger tracking error of this deformation feature. This would be resolved by closing the control loop for the knife tip position as well.

6.4 Discussion

In conclusion, this chapter showcased the experimental validation of the proposed collaborative framework developed for industrial manipulators in three distinct scenarios. Firstly, framework was utilized in the delicate grinding of complex-shaped surfaces used in aircraft manufacturing industry. Although being partially robotized with standard industrial manipulator equipped with active flange that acts only in the contact direction, the grinding process still heavily relies on experienced operators. Employing the proposed compliant control enabled the follow the grinding in Cartesian space while simultaneously exhibiting soft and complaint behaviour in the contact axis. When it comes down to the grinding process itself, two approaches were explored. The first involved grinding trajectory generation based on CAD data, while the second enabled experienced human operators to perform delicate sanding using a programming-by-demonstration approach. A significant increase in the treated area ratio of the proposed 6-DOF compliant system compared to the old 1-DOF system with an active flange was demonstrated.

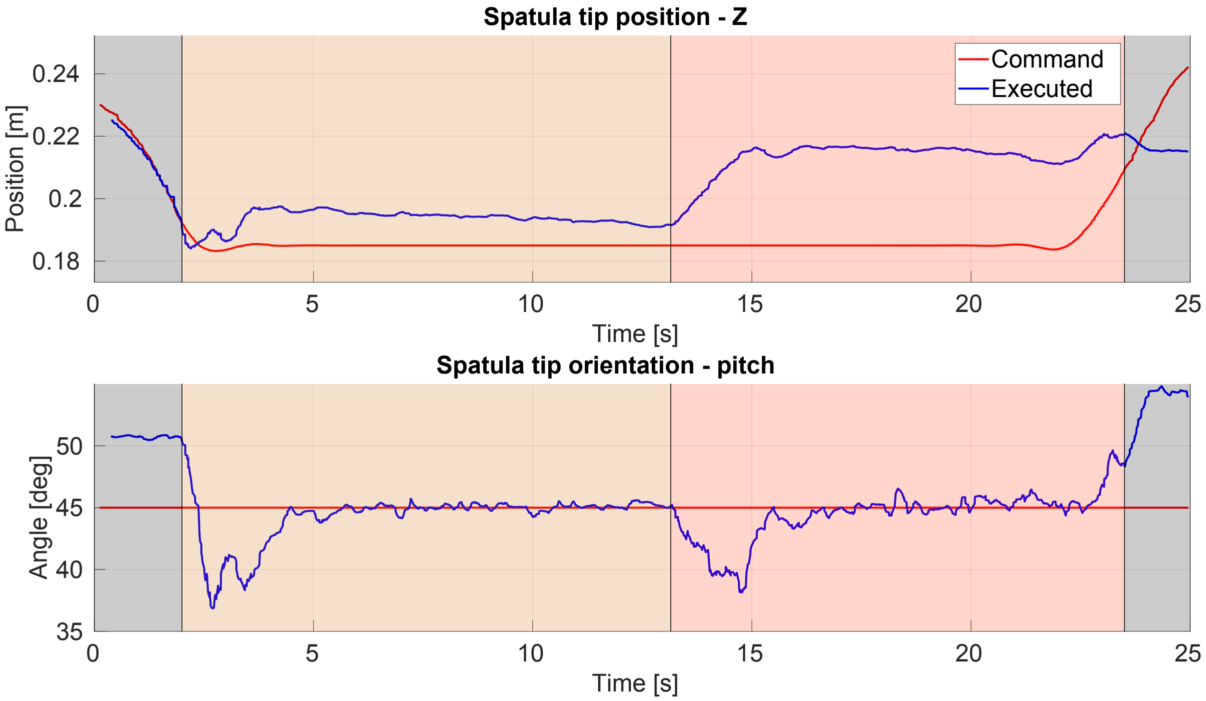


Figure 6.20: Showing global position of knife tip in robot base frame during the close-loop motion.

Notably, the proposed system increased the amount of robot-treated surface for some products from zero to 98% of the total surface area.

Secondly, the scenario of deep-micro-hole drilling in cast iron moulds for glass container manufacturing was presented. This task poses a challenge for robotic processing and is an area with limited research focus. Autonomous robotic drilling was attempted by Ochoa et al. [59] by having operators kinesthetically guide the robot's end-effector during demonstration. However, this method struggles to authentically capture the operator's skill, often taking them out of their comfort zone and regular work environment. In order to keep the operator in their comfort zone and capture the pure essence of delicate demonstration, a novel concept of demonstration recording based on tool tracking was utilized. The results have significantly outperformed Ochoa et al. in terms of the number of holes being drilled with a single bit, emphasizing the importance of keeping operators in their comfort zone within familiar environments to capture expert skill accurately.

Lastly, the framework was utilized in industrial application focused on manipulating deformable objects, i.e. robotic plastering. A deep learning method was utilized to estimate the shape of the plastering tool. Control of plastering system was decoupled in controlling the knife angle between the tool tip and treated object, and controlling the contact force through proposed compliant control of industrial manipulator. While deep learning has found roles in higher control levels in robotic applications, primarily as decision modules in state machines, the system demonstrated promising results for deploying deep learning-based inference within the lower-level robot control. Validation involved testing a reduced scope experimental setup, where a neural network model controls a single deformation feature of a flexible robot tool. Experimental results affirm that this control architecture ensures reference tracking, even in the presence of external disturbances

In conclusion, this chapter has effectively demonstrated the efficacy of the developed framework for the collaborative utilization of industrial manipulators across various industrial scenarios. The advantages of industrial manipulators, including precision, repeatability, reach, and payload capacity, were successfully leveraged and integrated in a collaborative context. The first component of the presented framework, compliant control, played a crucial role in all three applications involving compliant interactions between the robot and the environment. The second component focused on Programming-by-Demonstration, employing two different approaches for grinding and deep-micro-hole drilling. Both approaches demonstrated effectiveness in replicating complex demonstrations. Throughout the framework development, special attention was paid to safety issues, ensuring compliance with all requirements of the norm for collaborative robotics.

CHAPTER 7

Conclusion

Having in mind rapid development of industry, and more prominent shortage of human labour, there is growing demand for the robots to enter small-scale industries. This necessitates the deployment of flexible robotic systems that exhibit compliance, facilitate safe human-robot interaction, and are easily programmable even by untrained personnel. Consequently, the adoption of collaborative robotic systems becomes imperative. Considering these factors alongside industrial requirements, particularly regarding precision, repeatability, payload reach, a collaborative system has been developed atop standard off-the-shelf available industrial manipulator. These manipulators, in their original form, lack the collaborative features crucial for modern industrial applications. The primary contribution of this thesis lies in the development and experimental validation of a safe collaborative framework for compliant industrial manipulators. To better elucidate this contribution, it can be decoupled as follows.

A compliant control system for an industrial manipulator based on real time forward dynamics computation for both soft and rigid body contact for industrial applications involving deep micro drilling, plastering and polishing

As the foundational component of the proposed framework, the compliance control algorithm tailored for industrial manipulators was presented. This algorithm integrates fundamental principles such as impedance, admittance, and force control. Given that all compliant behaviours are executed in Cartesian space, particular emphasis was placed on mapping Cartesian motion into robot joint motion. This mapping is achieved through forward dynamics simulation, which maps forces acting on the robot's end-effector into joint motion. Furthermore, a simplified approach that circumvents the need for simulating forward kinematics was devised by leveraging the Jacobian matrix. This enables additional use cases, such as position-constrained motions like drilling along a single axis. In the stability analysis, a detailed examination of how different system parameters influence system behaviour was conducted. Contact interaction was modeled using a three mass-damping-spring model, and stability analysis was conducted using root locus method. Additionally, experimental validation was performed to confirm the derived model and stability analysis. Our experiments demonstrated that the stiffness and response time of the robot must align with those of the environment for stable contact. Increasing the stiffness or response time of the robot relative to the environment enhances the stability of the contact interaction.

Given the framework's strong focus on industrial applications, a critical aspect of this thesis was to validate the developed system through experimental testing in real-world scenarios across various industrial applications. Initially, the developed framework was implemented in the delicate grinding of composite materials used in the aircraft manufacturing industry. While this process already featured partial robotization using a standard industrial manipulator equipped with an active flange, the bulk of the work was still performed by human operators. Due to the hazardous nature of grinding, especially the generation of fine dust, the primary goal was to minimize the time human operators spent in such environments. Compared to the previous pre-programmed robotic systems that were compliant only in one axis, an active compliance control was employed across all six degrees of freedom in the robot system. This led to an increase in the quantity of surfaces treated by the robot. Additionally, a programming-by-demonstration approach was introduced, where human operators demonstrated how to treat special parts involving motions that were difficult to describe and program manually. With this final component, an effectiveness increase of over 90% in some cases was observed.

In the second case study, the framework was applied to the deep-micro-hole drilling of moulds used in glass container manufacturing. Unlike the previous case study, which relied on a kinesthetic approach to PbD, such an approach was deemed insufficient here. This was particularly true due to the highly demanding nature of the process, which involved subtle yet characteristic movements and low contact forces. Keeping the operator in their comfort zone was found to be instrumental to ensure proper recording of skill demonstrations. In this application, a novel PbD concept utilizing tool tracking was introduced. The results validated the effectiveness of this approach, as the robotic system outperformed the human operator in terms of the number of holes drilled. Finally, in the third case study, the task of autonomous plastering was addressed. Here, the developed compliance control was leveraged into manipulating a deformable object, i.e. plastering knife. This application serves as another illustration of the versatility and effectiveness of the developed framework in addressing various industrial tasks.

Safe collaborative human-robot interface for industrial manipulators which enables programming-by-demonstration

Given that industrial manipulators are large and powerful machines, ensuring human safety is imperative. This concern becomes even more critical in collaborative systems, which entail close cooperation between humans and robots. Through the integration of compliant behaviour into industrial manipulators and the development of a camera-based collision detection algorithm, a safe human-robot interaction has been enabled. Furthermore, the safety and compliance of the proposed system with ISO technical specification for collaborative manipulators have been demonstrated. The demonstrated results indicate that the system is both safe and fully compliant with these requirements.

Bibliography

[1] Patel, S., Sobh, T., "Task based synthesis of serial manipulators", Journal of Advanced Research, Vol. 6, No. 3, 2015, pp. 479-492, editors and International Board Member collection, available at: https://www.sciencedirect.com/science/article/pii/S2090123214001532

[2] Merlet, J.-P., "Parallel manipulators: state of the art and perspectives", Advanced Robotics, Vol. 8, No. 6, 1993, pp. 589-596, available at: https://doi.org/10.1163/156855394X00275

[3] Patel, Y., George, P. *et al.*, "Parallel manipulators applications—a survey", Modern Mechanical Engineering, Vol. 2, No. 03, 2012, pp. 57.

[4] Das, M. T., Canan Dülger, L., "Mathematical modelling, simulation and experimental verification of a scara robot", Simulation Modelling Practice and Theory, Vol. 13, No. 3, 2005, pp. 257-271, available at: https://www.sciencedirect.com/science/article/pii/S1569190X04001200

[5] Suri, S., Jain, A., Verma, N., Prasertpoj, N., "Scara industrial automation robot", in 2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC), 2018, pp. 173-177.

[6] Mohan, A., Wara, U. U., Shaikh, M. T. A., Rahman, R. M., Zaidi, Z. A., Shaikh, M. T. A., "Telesurgery and robotics: an improved and efficient era", Cureus, Vol. 13, No. 3, 2021.

[7] Feng, Z., Hu, G., Sun, Y., Soon, J., "An overview of collaborative robotic manipulation in multi-robot systems", Annual Reviews in Control, Vol. 49, 2020, pp. 113-127, available at: https://www.sciencedirect.com/science/article/pii/S1367578820300043

[8] Calinon, S., Robot Programming by Demonstration, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2009.

[9] Sheng, W., Thobbi, A., Gu, Y., “An integrated framework for human–robot collaborative manipulation”, IEEE Transactions on Cybernetics, Vol. 45, No. 10, Oct 2015, pp. 2030-2041.

[10] Maric, B., Polic, M., Tabak, T., Orsag, M., “Unsupervised optimization approach to in situ calibration of collaborative human-robot interaction tools”, in 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2020, pp. 255-262.

[11] Vanderborght, B., Albu-Schaeffer, A., Bicchi, A., Burdet, E., Caldwell, D., Carloni, R., Catalano, M., Eiberger, O., Friedl, W., Ganesh, G., Garabini, M., Grebenstein, M., Grioli, G., Haddadin, S., Hoppner, H., Jafari, A., Laffranchi, M., Lefeber, D., Petit, F., Stramigioli, S., Tsagarakis, N., Van Damme, M., Van Ham, R., Visser, L., Wolf, S., “Variable impedance actuators: A review”, Robotics and Autonomous Systems, Vol. 61, No. 12, 2013, pp. 1601-1614, available at: https://www.sciencedirect.com/science/article/pii/S0921889013001188

[12] Wang, W., Loh, R. N., Gu, E. Y., “Passive compliance versus active compliance in robot-based automated assembly systems”, Industrial Robot: An International Journal, Vol. 25, No. 1, 1998, pp. 48–57.

[13] De Schutter, J., “A study of active compliant motion control methods for rigid manipulators based on a generic scheme”, in Proceedings. 1987 IEEE International Conference on Robotics and Automation, Vol. 4, March 1987, pp. 1060-1065.

[14] Hogan, N., “Impedance control: An approach to manipulation: Part II—implementation”, Journal of Dynamic Systems, Measurement, and Control, Vol. 107, No. 1, 1985, pp. 8, available at: https://doi.org/10.1115%2F1.3140713

[15] Hogan, N., “Impedance control: An approach to manipulation: Part III—applications”, Journal of Dynamic Systems, Measurement, and Control, Vol. 107, No. 1, 1985, pp. 17, available at: https://doi.org/10.1115%2F1.3140701

[16] Ott, C., Mukherjee, R., Nakamura, Y., “Unified impedance and admittance control”, in Proceedings - IEEE International Conference on Robotics and Automation, 05 2010, pp. 554-561.

[17] Anderson, R. J., Spong, M. W., “Hybrid impedance control of robotic manipulators”, IEEE Journal on Robotics and Automation, Vol. 4, No. 5, Oct 1988, pp. 549-556.

[18] Volpe, R., Khosla, P., "A theoretical and experimental investigation of explicit force control strategies for manipulators", IEEE Transactions on Automatic Control, Vol. 38, No. 11, Nov 1993, pp. 1634-1650.

[19] Volpe, R., Khosla, P., "An experimental evaluation and comparison of explicit force control strategies for robotic manipulators", in 1992 American Control Conference, June 1992, pp. 758-765.

[20] Scherzinger, S., Roennau, A., Dillmann, R., "Forward dynamics compliance control (fdcc): A new approach to cartesian compliance for robotic manipulators", in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2017, pp. 4568-4575.

[21] Sanchez, J., Corrales, J.-A., Bouzgarrou, B.-C., Mezouar, Y., "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey", The International Journal of Robotics Research, Vol. 37, No. 7, 2018, pp. 688–716.

[22] Polic, M., Car, M., Petric, F., Orsag, M., "Compliant plant exploration for agricultural procedures with a collaborative robot", IEEE Robotics and Automation Letters, 2021, pp. 1-1.

[23] Navarro-Alarcon, D., Yip, H. M., Wang, Z., Liu, Y.-H., Zhong, F., Zhang, T., Li, P., "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model", IEEE Transactions on Robotics, Vol. 32, No. 2, 2016, pp. 429–441.

[24] Scalera, L., Seriani, S., Gasparetto, A., Gallina, P., "Watercolour robotic painting: a novel automatic system for artistic rendering", Journal of Intelligent & Robotic Systems, Vol. 95, No. 3, 2019, pp. 871–886.

[25] Polic, M., Maric, B., Orsag, M., "Soft robotics approach to autonomous plastering", in 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), 2021, pp. 482-487.

[26] Zhang, J., Wang, Y., Xiong, R., "Industrial robot programming by demonstration", in 2016 International Conference on Advanced Robotics and Mechatronics (ICARM), 2016, pp. 300-305.

[27] Tykal, M., Montebelli, A., Kyrki, V., "Incrementally assisted kinesthetic teaching for programming by demonstration", in 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2016, pp. 205-212.

[28] Zhang, T., McCarthy, Z., Jow, O., Lee, D., Chen, X., Goldberg, K., Abbeel, P., "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation", in 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 5628-5635.

[29] Scherzinger, S., Roennau, A., Dillmann, R., "Contact skill imitation learning for robot-independent assembly programming", in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nov 2019, pp. 4309-4316.

[30] Pastor, P., Kalakrishnan, M., Chitta, S., Theodorou, E., Schaal, S., "Skill learning and task outcome prediction for manipulation", in 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 3828-3834.

[31] Wang, K., Fan, Y., Sakuma, I., "Robot programming from a single demonstration for high precision industrial insertion", Sensors, Vol. 23, No. 5, 2023, available at: https://www.mdpi.com/1424-8220/23/5/2514

[32] Ajaykumar, G., Huang, C.-M., "Multimodal robot programming by demonstration: A preliminary exploration", 2023.

[33] Steinmetz, F., Nitsch, V., Stulp, F., "Intuitive task-level programming by demonstration through semantic skill recognition", IEEE Robotics and Automation Letters, Vol. 4, No. 4, 2019, pp. 3742-3749.

[34] Hermann, A., Klemm, S., Xue, Z., Roennau, A., Dillmann, R., "Gpu-based real-time collision detection for motion execution in mobile manipulation planning", in 2013 16th International Conference on Advanced Robotics (ICAR), Nov 2013, pp. 1-7.

[35] Pan, J., Chitta, S., Manocha, D., "Fcl: A general purpose library for collision and proximity queries", in 2012 IEEE International Conference on Robotics and Automation, May 2012, pp. 3859-3866.

[36] Hermann, A., Drews, F., Bauer, J., Klemm, S., Roennau, A., Dillmann, R., "Unified gpu voxel collision detection for mobile manipulation planning", in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 4154-4160.

[37] Hermann, A., Klemm, S., Xue, Z., Roennau, A., Dillmann, R., "Gpu-based real-time collision detection for motion execution in mobile manipulation planning", in 2013 16th International Conference on Advanced Robotics (ICAR), 2013, pp. 1-7.

[38] Rosenstrauch, M. J., Krüger, J., "Safe human-robot-collaboration-introduction and experiment using iso/ts 15066", in 2017 3rd International conference on control, automation and robotics (ICCAR). IEEE, 2017, pp. 740–744.

[39] Ferraguti, F., Bertuletti, M., Landi, C. T., Bonfè, M., Fantuzzi, C., Secchi, C., “A control barrier function approach for maximizing performance while fulfilling to iso/ts 15066 regulations”, IEEE Robotics and Automation Letters, Vol. 5, No. 4, 2020, pp. 5921-5928.

[40] Song, P., Yu, Y., Zhang, X., “Impedance control of robots: An overview”, in 2017 2nd International Conference on Cybernetics, Robotics and Control (CRC), July 2017, pp. 51-55.

[41] Valency, T., Zacksenhouse, M., “Accuracy/robustness dilemma in impedance control”, Journal of Dynamic Systems Measurement and Control-transactions of The Asme - J DYN SYST MEAS CONTR, Vol. 125, 09 2003.

[42] Zeng, G., Hemami, A., “An overview of robot force control”, Robotica, Vol. 15, 09 1997, pp. 473-482.

[43] Seul Jung, Hsia, T. C., Bonitz, R. G., “Force tracking impedance control of robot manipulators under unknown environment”, IEEE Transactions on Control Systems Technology, Vol. 12, No. 3, May 2004, pp. 474-483.

[44] Jazar, R. N., Theory of applied robotics. Springer, 2010.

[45] Felis, M. L., “Rbdl: an efficient rigid-body dynamics library using recursive algorithms”, Autonomous Robots, Vol. 41, No. 2, Feb 2017, pp. 495–511, available at: https://doi.org/10.1007/s10514-016-9574-0

[46] Smits, R., Aertbelien, E., Bruyninckx, H., Shakhimardanov, A., “Kinematics and dynamics library (kdl)”, 2017.

[47] Eppinger, S., Seering, W., “On dynamic models of robot force control”, in Proceedings. 1986 IEEE International Conference on Robotics and Automation, Vol. 3. IEEE, 1986, pp. 29–34.

[48] Eppinger, S., Seering, W., “Introduction to dynamic models for robot force control”, IEEE Control Systems Magazine, Vol. 7, No. 2, 1987, pp. 48–52.

[49] Dang, H., Allen, P. K., “Robot learning of everyday object manipulations via human demonstration”, in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010, pp. 1284–1289.

[50] Maric, B., Mutka, A., Orsag, M., “Collaborative human-robot framework for delicate sanding of complex shape surfaces”, IEEE Robotics and Automation Letters, Vol. 5, No. 2, 2020, pp. 2848-2855.

[51] Rusu, R. B., Blodow, N., Beetz, M., “Fast point feature histograms (fpfh) for 3d registration”, in 2009 IEEE international conference on robotics and automation. IEEE, 2009, pp. 3212–3217.

[52] Buch, A. G., Kraft, D., Kamarainen, J.-K., Petersen, H. G., Krüger, N., “Pose estimation using local structure-specific shape and appearance context”, in 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013, pp. 2080–2087.

[53] Arun, K. S., Huang, T. S., Blostein, S. D., “Least-squares fitting of two 3-d point sets”, IEEE Transactions on pattern analysis and machine intelligence, No. 5, 1987, pp. 698–700.

[54] Billard, A., Calinon, S., Dillmann, R., Schaal, S., Robot Programming by Demonstration. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1371–1394, available at: https://doi.org/10.1007/978-3-540-30301-5_60

[55] G., H. S., “Nasa task load index (tlx)”, in NASA Ames Research Center Moffett Field, CA United States), January 1986, available at: https://ntrs.nasa.gov/citations/20000021488

[56] Hart, S. G., “Nasa-Task Load Index (NASA-TLX); 20 Years Later”, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 50, No. 9, 2006, pp. 904–908, available at: https://doi.org/10.1177/154193120605000909

[57] Brooke, J., “Sus: A quick and dirty usability scale”, Usability Eval. Ind., Vol. 189, 11 1995.

[58] Bustamante, E. A., Spain, R. D., “Measurement Invariance of the Nasa TLX”, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 52, No. 19, 2008, pp. 1522–1526, available at: https://doi.org/10.1177/154193120805201946

[59] Ochoa, H., Cortesão, R., “Impedance control architecture for robotic-assisted micro-drilling tasks”, Journal of Manufacturing Processes, Vol. 67, 2021, pp. 356–363.

[60] Maric, B., Petric, F., Stuhne, D., Ranogajec, V., Orsag, M., “Replicating human skill for robotic deep-micro-hole drilling”, in 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), 2022, pp. 2238-2244.

[61] Pham, H., Pham, Q., “A new approach to time-optimal path parameterization based on reachability analysis”, IEEE Transactions on Robotics, Vol. 34, No. 3, June 2018, pp. 645-659.

[62] Pastor, P., Hoffmann, H., Asfour, T., Schaal, S., "Learning and generalization of motor skills by learning from demonstration", in 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 763-768.

[63] Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S., "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors", Neural Computation, Vol. 25, No. 2, 02 2013, pp. 328-373, available at: https://doi.org/10.1162/NECO_a_00393

[64] Pastor, P., Kalakrishnan, M., Meier, F., Stulp, F., Buchli, J., Theodorou, E., Schaal, S., "From dynamic movement primitives to associative skill memories", Robotics and Autonomous Systems, Vol. 61, No. 4, 2013, pp. 351–361.

[65] Hera, P. L., Morales, D. O., Mendoza-Trejo, O., "A study case of dynamic motion primitives as a motion planning method to automate the work of forestry cranes", Computers and Electronics in Agriculture, Vol. 183, 2021, pp. 106037, available at: https://www.sciencedirect.com/science/article/pii/S0168169921000557

[66] "Safety of machinery — General principles for design — Risk assessment and risk reduction", International Organization for Standardization, Geneva, CH, Standard, November 2010.

[67] "Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots", International Organization for Standardization, Geneva, CH, Standard, July 2011.

[68] "Robots and robotic devices — Collaborative robots", International Organization for Standardization, Geneva, CH, Technical Specification, February 2016.

[69] Kharidege, A., Ting, D., Yajun, Z., "A practical approach for automated polishing system of free-form surface path generation based on industrial arm robot", The International Journal of Advanced Manufacturing Technology, Vol. 93, July 2017, pp. 3921-3934.

[70] Tam, H.-Y., Lui, O., Mok, A., "Robotic polishing of free-form surfaces using scanning paths", Journal of Materials Processing Technology, Vol. 95, No. 1-3, 10 1999, pp. 191–200.

[71] Fernandez, A., Dieste, J., Javierre, C., Santolaria, J., "Surface roughness evolution model for finishing using an abrasive tool on a robot", International Journal of Advanced Robotic Systems, Vol. 12, 09 2015, pp. 1.

[72] Ding, Y., Min, X., Fu, W., Liang, Z., “Research and application on force control of industrial robot polishing concave curved surfaces”, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Vol. 233, No. 6, 2019, pp. 1674–1686.

[73] Zheng, L., Wang, C., Fu, L., Yang, L., Qu, Y., Song, Y., “Wear mechanisms of micro-drills during dry high speed drilling of pcb”, Journal of Materials Processing Technology, Vol. 212, No. 10, 2012, pp. 1989–1997.

[74] Brett, P., Taylor, R., Proops, D., Coulson, C., Reid, A., Griffiths, M., “A surgical robot for cochleostomy”, in 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, pp. 1229-1232.

[75] Wang, W., Shi, Y., Yang, N., Yuan, X., “Experimental analysis of drilling process in cortical bone”, Medical engineering & physics, Vol. 36, No. 2, 2014, pp. 261–266.

[76] Butler-Smith, P., Axinte, D., Daine, M., Kennedy, A., Harper, L., Bucourt, J., Ragueneau, R., “A study of an improved cutting mechanism of composite materials using novel design of diamond micro-core drills”, International Journal of Machine Tools and Manufacture, Vol. 88, 2015, pp. 175–183.

[77] Kim, D. W., Lee, Y. S., Park, M. S., Chu, C. N., “Tool life improvement by peck drilling and thrust force monitoring during deep-micro-hole drilling of steel”, International Journal of Machine Tools and Manufacture, Vol. 49, No. 3, 2009, pp. 246-255, available at: https://www.sciencedirect.com/science/article/pii/S0890695508002204

[78] Li, X., Meadows, A., Babitsky, V., Parkin, R., “Experimental analysis on autoresonant control of ultrasonically assisted drilling”, Mechatronics, Vol. 29, 2015, pp. 57–66.

[79] Kudla, L. A., “Fracture phenomena of microdrills in static and dynamic conditions”, Engineering Fracture Mechanics, Vol. 78, No. 1, 2011, pp. 1-12, available at: https://www.sciencedirect.com/science/article/pii/S0013794410004170

[80] Hasan, M., Zhao, J., Jiang, Z., “A review of modern advancements in micro drilling techniques”, Journal of Manufacturing Processes, Vol. 29, 2017, pp. 343-375, available at: https://www.sciencedirect.com/science/article/pii/S152661251730244X

[81] Lagarias, J. C., Reeds, J. A., Wright, M. H., Wright, P. E., “Convergence properties of the nelder–mead simplex method in low dimensions”, SIAM Journal on optimization, Vol. 9, No. 1, 1998, pp. 112–147.

[82] Li, X., Jiang, X., "Development of a robot system for applying putty on plastered walls", in 2018 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2018, pp. 1417–1422.

[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, 2019, pp. 3671-3678.

Biography

Bruno Marić is a Ph.D. student and 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 from the same institution in 2017, receiving his MSc degree. During his education, he received a Rector's Award for the work entitled "Design and development of a small-scale multi-rotor vehicle with moving mass based control".

His research interests are robotics, collaborative industrial manipulation, and robotics in medicine. During his PhD studies, he actively contributed to various research projects such as ENDORSE (Effective Robotic Grinding of Surface Areas through HORSE framework), SPECULARIA (Structured Ecological Cultivation with Autonomous Robots in Agriculture), and HMIGOS (Gesture-based Human Machine Interface for Guided Orthopedic Surgery). Presently, he is engaged in an industrial-focused research project aimed at developing an autonomous system for the delicate deep-micro-hole drilling and polishing of moulds used in the glass container manufacturing industry. So far, he has authorized and co-authorized 7 research papers.

List of publications

Journal papers

1. Collaborative Human-Robot Framework for Delicate Sanding of Complex Shape Surfaces. Marić, B.; Mutka, A.; and Orsag, M. IEEE Robotics and Automation Letters, 5(2): 2848-2855. April 2020.

Conference papers

1. A novel concept of attitude control for large multirotor-UAVs based on moving mass control. Haus, T.; Prkut, N.; Borovina, K.; Marić, B.; Orsag, M.; and Bogdan, S. In Control and Automation (MED), 2016 24th Mediterranean Conference on, pages 832–839, 2016.

IEEE

2. Unsupervised optimization approach to in situ calibration of collaborative human-robot interaction tools. Marić, B.; Polić, M.; Tabak, T.; and Orsag, M. In 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pages 255-262, 2020.
3. Vision based collision detection for a safe collaborative industrial manipulator. Marić, B.; Jurican, F.; Orsag, M.; and Kovačić, Z. In 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR), pages 334-337, 2021.
4. Structured Ecological Cultivation with Autonomous Robots in Indoor Agriculture. Polić, M.; Ivanović, A.; Marić, B.; Arbanas, B.; Tabak, J.; and Orsag, M. In 2021 16th International Conference on Telecommunications (ConTEL), pages 189-195, 2021.
5. Soft robotics approach to autonomous plastering. Polić, M.; Marić, B.; and Orsag, M. In 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), pages 482-487, 2021.
6. Replicating human skill for robotic deep-micro-hole drilling. Marić, B.; Petric, F.; Stuhne, D.; Ranogajec, V.; and Orsag, M. In 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), pages 2238-2244, 2022.

Životopis

Bruno Marić je doktorand i istraživač u Laboratoriju za Robotiku i Inteligentne Sustave Upravljanja u sklopu Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu (UNIZG-FER). Diplomirao je 2017. godine na istom fakultetu, stekavši titulu magistra inženjera. Tijekom studija, dobio je rektorovu nagradu za rad pod naslovom "Dizajn i razvoj male multirotorske letjelice s upravljanjem zasnovanim na pomičnim masama".

Njegove interesne sfere istraživanja su robotika, kolaborativna industrijska manipulacija i robotika u medicini. Tijekom doktorskog studija, aktivno je sudjelovao na različitim istraživačkim projektima poput ENDORSE(Effective Robotic Grinding of Surface Areas through HORSE framework), SPECULARIA (Structured Ecological Cultivation with Autonomous Robots in Agriculture) te HMIGOS (Gesture-based Human Machine Interface for Guided Orthopedic Surgery). Trenutno se bavi industrijskim istraživačkim projektom čiji je cilj razvoj autonomnog sustava za delikatna duboka mikrobušenja i poliranja kalupa, korištenih u proizvodnji staklene ambalaže. Dosad je autor ili koautor 7 znanstvenih radova.