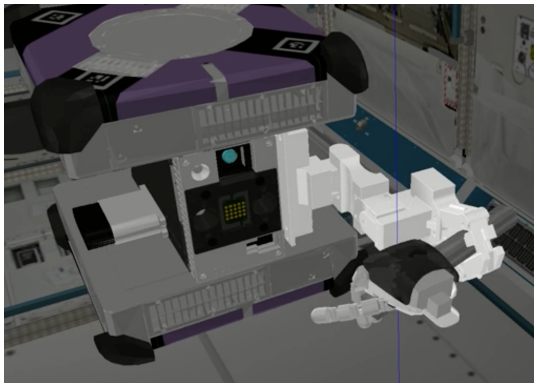
TÉCNICO LISBOA



Object Handover between Humans and Robots in Microgravity

Adriana Maria Marques Fernandes

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor(s): Professor Rodrigo Martins de Matos Ventura

Examination Committee

Chairperson: Professor José Fernando Alves da Silva
Supervisor: Professor Rodrigo Martins de Matos Ventura
Member of the Committee: Professor Alexandre José Malheiro Bernardino

February 2020

Para os meus pais

Acknowledgments

Although my name is on the front cover, the work developed would not be possible without the remarkable people that were by my side during this path.

First, I would like to express my sincere gratitude to my research supervisor, Professor Rodrigo Ventura, for, week after week, patiently clarifying my doubts, brainstorming with me and for giving me valuable, constructive and insightful suggestions that were crucial to the development of this dissertation.

Special thanks should be given to José Corujeira for the help concerning the design and review of the user study conducted. His suggestions were extremely useful and much appreciated.

I would also like to extend my thanks to Carolina Januário and Katrina Alsup for their willingness to discuss their thesis work, providing me with insights that were helpful for the work developed in this thesis.

I want to also thank to my family. There are no words that would fit into this page that would express the gratitude towards my mother. Without her, the journey that took me to this place would not be possible. She is my biggest supporter and my absolute role model. I would also like to thank my father, who, although is no longer with us, always believed in my personal and academic success and that belief continues to inspire me everyday. This thesis is for them. On the same note, I would like to show my warm appreciation for my grandparents that have a huge impact on what I've become and that are anxiously and excitedly waiting to see me graduate.

I would like to express my deep gratitude to my boyfriend Gonçalo, without whom I would not be writing this acknowledgments. His infinite patience, caring and support during the toughest moments that arised from this thesis (and from my overall academic journey) were essential to be where I am today.

Last but not least, I wanna thank my friend Mariana for the unconditional motivational support and my colleagues Raquel and Mafalda for the long study nights and the work break coffees at IST.

Resumo

A área de investigação alusiva a robots espaciais colaborativos apresenta-se atualmente como uma área com elevado impacto, uma vez que os robots desempenham missões colaborativas com mais precisão durante tarefas longas, mais rápido e sem requererem fatos espaciais, infraestruturas como suporte de vida ou de regressarem à Terra, revelando-se eficientes e economicamente viáveis. Não obstante, a colaboração humano-robot no espaço é ainda um desafio relativamente a questões relevantes como a manipulação de objetos num ambiente espacial ou a mobilidade devido à peculiar dinâmica de movimento de um robot equipado com um manipulador em microgravidade. Assim, a presente dissertação tem como objetivo a formulação de um algoritmo que permita a um robot espacial voador equipado com um manipulador entregar ou receber um objeto a/de um humano de uma forma fluente, dinâmica e bem sucedida. Adicionalmente, é proposto um estudo sistemático com utilizadores que visa explorar as preferências destes entre um comportamento de impedância rígido e não-rígido do robot durante a transferência do objecto e entender o impacto destes comportamentos no sucesso da tarefa. O algoritmo proposto é formulado com recurso a uma máquina finita de estados que inclui um controlador de impedância na fase de transferência permitindo assim uma interação dinâmica entre os dois participantes da tarefa. A implementação e validação do algoritmo é efetuada no simulador do robot voador desenvolvido pela NASA, Astrobee, e os dois comportamentos de impedância, rígido e não-rígido, são estudados com sucesso. Seguidamente, é desenvolvida uma interface de interação entre o simulador e o utilizador que engloba um modelo de uma mão controlada através do dispositivo Leap Motion. Esta interface infere uma perceção de realidade virtual do ambiente, permitindo uma interação precisa com o robot nas tarefas propostas. Por fim, foi realizado o estudo sistemático com utilizadores. Os resultados mostram que o comportamento rígido do robot apresentou maior preferência e sucesso de transferência nas tarefas propostas.

Palavras-chave: Colaboração Humano-Robot, Transferência Objectos Microgravidade, Manipulação Espacial, Controlo de Força, Controlo de Impedância, Realidade Virtual

Abstract

The collaborative space robots field is an emerging research field with high impact as robots perform collaborative missions with high precision during lengthy tasks, on an accelerated schedule and do not require spacial suits, infrastructures such as life support or to return to Earth, making them efficient and economically viable. However, human-robot collaboration in space is still a challenge concerning key issues such as manipulation of weightless objects and mobility due to the peculiar motion dynamics of the robot and the manipulator on a microgravity environment. Therefore, this dissertation aims to formulate an algorithm that enables a free-flyer robot equipped with a manipulator to perform a successful, fluent and dynamic robot-to-human and human-to-robot object handover. Additionally, a systematic user study is proposed with the goal of analysing the user's preferences between a rigid and compliant impedance robot behavior during the object handover and understanding the impact of those behaviors on the success of the task. The proposed algorithm is formulated with resource to a finite-state machine that encompasses a designed impedance controller for the transfer phase enabling a dynamic interaction between both task participants. Following, the algorithm is implemented and validated on the simulator of NASA's Astrobee free-flyer robot and the two impedance behaviors, rigid and compliant, are successfully studied. Furthermore, an user interaction interface is developed and includes an user simulated hand model controlled via a Leap Motion device. This interface generates a virtual reality perception environment that enables an accurate interaction with the robot on the proposed tasks. Lastly, the proposed systematic user study was conducted. Results showed that the rigid behavior was overall more preferable and registered higher transfer success during the tasks.

Keywords: Human-Robot Collaboration, Microgravity Object Handover, Space Manipulation, Force Control, Impedance Control, Virtual Reality

Contents

List of Tables

List of Figures

Acronyms

AC Admittance Control

AI Artificial Intelligence

API Application Programming Interface

CIMON Crew Interactive Mobile CompanioN

CM Center of Mass

CNS Central Nervous System

CPU Central Processing Unit

DDS Data Distribution Service

DOF Degree Of Freedom

DSG Deep Space Gateway

EKF Extended Kalman Filter

EVA Extra-Vehicular Activity

FAM Force Allocation Module

FSM Finite State Machine

GNC Guidance, Navigation and Control

HLP High-Level Processor

HRC Human-Robot Collaboration

HRI Human-Robot Interaction

IC Impedance Control

IMU Inertial Measurements Unit

ISECG International Space Exploration Coordination Group

ISO International Standards Organisation

ISR Institute for System and Robotics

ISS International Space Station

IVA Intra-Vehicular Activity

LEO Low Earth Orbit

LLP Low-Level Processor

MIT Massachusetts Institute of Technology

MLP Mid-Level Processor

NASA National Aeronautics and Space Administration

PHRC Physical Human-Robot Collaboration

ROS Robot Operating System

SDK Software Development Kit

SPHERES Synchronized Position Hold, Engage, Reorient, Experimental Satellites

UAV Unmanned Aerial Vehicle

URDF Unified Robot Description Format

VR Virtual Reality

Chapter 1

Introduction

1.1 Motivation

Space research is a wide growing field with high scientific and economic impact. This field encompasses distinct research branches such as earth observation, solar and space physics concerning primarily with electromagnetic effects, cosmology and research on nonliving and living materials, including humans and robots, in microgravity. This dissertation focus on the last field, more specifically, on the collaborative space robotics field.

Collaborative space robotics research is considered the development of general purpose machines that are able to operate in microgravity environments facilitating manipulation, assembling or servicing functions in collaboration with astronauts. In fact, ISECG (International Space Exploration Coordination Group) identified space robots as a critical technology not only for today's activity on the ISS (International Space Station) but also for future exploration missions to the Moon and Mars[1].

One of the reasons why space robots became progressively more relevant to the current overall ability to operate in space is their capability to function for longer periods without the need of interrupting the task, in contrast to humans due to their essential needs. Additionally, robots are able to replicate flawlessly the same task and on an accelerated schedule. As a matter of fact, occasionally robots improve task performance compared to humans. Robots also operate with extreme high precision which is crucial on a demanding and rigorous research environment such as space.

Furthermore, an outstaying advantage of space robots is the decrease of risk compared to human life. In this manner, it is safer to have space robots in operation instead of astronauts. Another advantage of having robots support humans in space is that they are impassible towards the low temperatures and thus do not need long pre-arrangements or suits before going to outer space, making robots also more affordable. As a matter of fact, astronaut Tim Peake's EVA (Extra-Vehicular Activity) suit cost €12

[1]NASA (2018). The 3^{th} Global Exploration Roadmap, Retrieved December 13, 2019, from https://www.nasa.gov/sites/default/files/atoms/files.

millions [1].

Another cost associated with humans and not with robots is the need to return to Earth. The replacement of NASA's (National Aeronautics and Space Administration) crew on board of SpaceX Dragon or Boeing CST-100 cost €56 millions per seat, in 2016[2]. Moreover, space robots do not need infrastructure (e.g. life suport) or basic supplies. In these concerns, it is interesting to refer that SpaceX Dragon Cargo Ship was contracted by NASA to carry out a minimum of 6 missions under phase 2 of SpaceX's commercial resupply services, which is worth a total of €14 billions[3]. The last launch occurred on the 5^{th} of December of 2019 with $2\ 585\ kg$ and cost €133 millions[4], which results in an average of €$51\ 450$ thousands for every kilogram lift into Low Earth Orbit (LEO). In this manner, access to space is expensive implying that robots that are lighter than humans and do not require infrastructure or basic supplies are economically attractive for a broad classes of missions.

Concerning space exploration, NASA is developing plans for the next phase of deep space exploration, specifically, Mars exploration. In this manner and given the retirement of the ISS in 2028, NASA proposal includes the brand new Deep Space Gateway (DSG) that will orbit the moon rather than Earth. The plans call for astronauts to return to the moon and then venture deeper into Mars [2]. However, the DSG encompasses a challenging environment for astronauts given the distance from Earth. As a matter of fact, in case of an emergency assistance would be weeks away, in contrast to the ISS that could be evacuated in a matter of hours. Moreover, the deep-space environment is inherently risky given the radiation which includes high-energy cosmic rays that pose as an ever-present danger to astronauts.
Given these challenges that arise from space exploration, microgravity robots are a more practical and risk-free alternative for such missions.

Furthermore, it is relevant to refer key issues concerning space robots such as manipulation and mobility. Although this issues are a basic technology in robotics, microgravity in the orbital environment requires further attention due to the peculiar motion dynamics of the robot and manipulator during IVA (Intra-Vehicular Activity) and EVA especially with human interaction. Additionally, robots autonomy and perception remains a challenge concerning space robotics.

Given the advantages of space robotics, this thesis aims to contribute to the research of the referred key issues regarding collaborative tasks, more specifically handover tasks where an object is transferred between a human and a robot.

[2]NASA (2016). Commercial Crew Transportation Program, Retrieved November 5, 2019, from https://www.nasa.gov/sites/default/files/atoms/files.

[3]A. Thompson (2019). SpaceX Launches Dragon Cargo Ship to Space Station for NASA, Retrieved December 8, 2019, from https://www.space.com/.

[4]S. Kramer and D. Mosher. Tech insider (2016). SpaceX Rocket Cargo Price by Weight, Retrieved December 8, 2019, from https://www.businessinsider.com/spacex-rocket-cargo-price-byweight-2016-6.

1.2 Objectives

The goal of the current dissertation is to formulate an algorithm that enables a free-flyer robot equipped with a manipulator to perform an object handover with a human on a successful, fluent and dynamic manner in a microgravity environment. Two tasks of interest will be explored: human-to-robot and robot-to-human handover.

Furthermore, this thesis aims to conduct a systematic user study with the goal of understanding the subjective outcome effects of a rigid and compliant impedance robot behavior during the interaction, e.g, the user's preferences between the two robot behaviors implemented. Additionally, the study intends to analyse the impact of those behaviors on the success of the task.

1.3 Approach Outline

With the aim of developing the proposed algorithm, a Finite State Machine (FSM) is formulated for both tasks. The proposed FSM encompasses three phases of the handover: approach, transfer and retraction. Several challenges in each phase will be further discussed during the algorithm formulation.

Moreover, given the objective of performing an object handover in a fluent and dynamic manner, it is crucial to develop a control-based approach that implements a dynamic relation between the free-flyer robot's manipulator and the environment during the transfer phase. In this manner, an impedance control is designed.

The algorithm is implemented and validated on the simulator of the latest robot aboard the International Space Station, NASA's Astrobee.

Additionally, two distinct robot impedance behaviors are implemented during the transfer phase: rigid or compliant. Results concerning the validation of the designed impedance controller for both behaviors are presented.

Furthermore, given the participation of a human agent during the tasks and with the aim of conducting the proposed systematic user study, an user interaction interface is implemented. The interface encompasses a simulated user hand model controlled through a hand tracking device, Leap Motion. This setup aims to deliver a virtual reality experience for the user enabling a real user interaction with the robot. The results concerning the validation of this interface interaction with the simulator are also presented.

After the implementations referred, the results regarding the proposed handover tasks are presented and discussed. Lastly, the systematic user study results are analysed.

1.4 Thesis Outline

The second chapter of this thesis, Chapter 2, includes the Background where relevant concepts concerning the work developed are explored, such as Human-Robot Collaboration and Applications, Active Compliant Control and Object Handover between Humans and Robots. Following, the State-of-the-Art regarding this dissertation main problem is studied in section 2.4.

Chapter 3 focus on the Handover Algorithm Formulation. This chapter describes the state-machine formulation that includes states and transitions regarding the robot-to-human handover and human-to-robot handover. Additionally, the impedance control approach is formulated in section 3.3.

Furthermore, Chapter 4 presents the Implementation and Results that encompasses the proposed algorithm implementation on the Astrobee's simulator and the impedance control validation (section 4.1). Followed by the human interaction implementation that includes the simulated hand model implementation, the integration of the data from the tracking device with the simulator and the interface results (section 4.2). Lastly, this chapter presents the results and discussion of the handover tasks performed (section 4.3).

Chapter 5 includes the User Study. Firstly, the aim of the study and the hypothesis are described. Additionally, the study location and the user eligibility criteria are presented and lastly, the task conditions, measures and experimental procedure are explored. The results are also presented and discussed in section 5.8.

The conclusion chapter is the Chapter 6 where a summary of the research developed and results is conducted. Moreover, the contribution of the dissertation is discussed and future work proposed.

Concerning the Appendix information, Appendix A is a relevant derivation presented on the impedance control module design and Appendix B includes the instructions delivered during the user study experiment.

Chapter 2

Background

Before presenting the current State-of-the-Art concerning the main subject focused by this dissertation, some important concepts are explored.

Given the goals of this dissertation, it is relevant to firstly understand the meaning of a Human-Robot Collaboration (HRC). Furthermore, it is pertinent to analyse some communication methods between a human and a robot, the challenges that arise from tasks shared between these agents and examples of collaborative applications.

Having into consideration the dynamic control approach proposed during the transfer phase, the field of Active Compliant Control is also explored.

Lastly and with the aim of exploring in depth the HRC task proposed by this thesis, a section is dedicated to Object Handover between Humans and Robots.

2.1 Human-Robot Collaboration

Human-Robot Collaboration is the process in which a human and a robot agent work together to achieve shared goals. More specifically, Physical Human-Robot Collaboration (PHRC) is defined when human(s), robot(s) and the environment come to contact with each other and form a tightly coupled dynamical system to accomplish a task [3]. It is important to distinguish Human-Robot Collaboration from Human-Robot Interaction (HRI) as these two terms hold different meanings. Whereas interaction entails only acting on someone or something else, collaboration is inherently *with* others [4]. Interaction determines action on someone but not necessarily profit from it and cannot be considered unavoidably a team work. Therefore, interaction is a generalized term that includes collaboration.

This research field is an accelerated growing research area through the last years. The scientific community emergent interest in Human-Robot Collaboration is presented on Figure 2.1 [5].

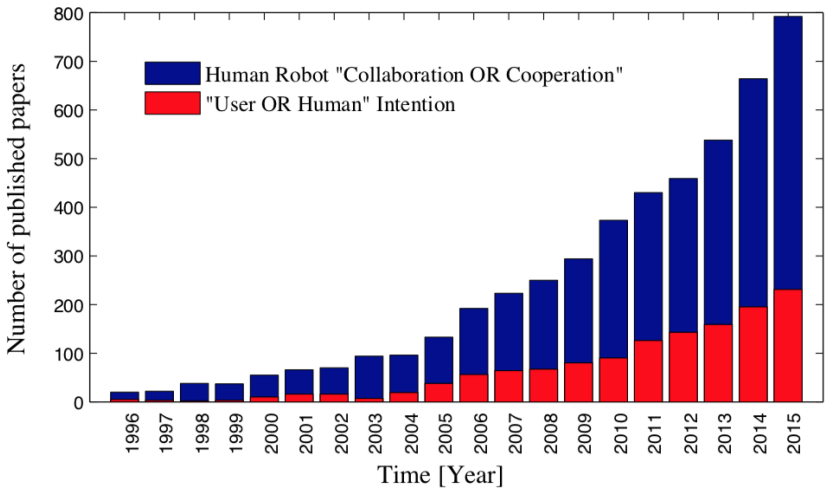


Figure 2.1: Number of publications on the research field of human–robot collaboration from 1996 to 2015 [5].

Human-robot collaboration can be realized through actions of the individual partners aiding in fulfilling a joint intention. Accordingly to Sebanz definition in [6], a joint action can be regarded as any form of social interaction whereby two or more individuals coordinate their actions in space and time. Sebanz also propose that successful joint action depends on the ability to share representations, to predict actions, and to integrate predicted effects of own and other's actions. In this manner, to work cooperatively on a task, the team (in this case the human and the robot) need to agree on a common goal and a joint intention to reach the aim of the action. In a human-robot team it will mostly be the human who assigns a goal for the team and who therefore has an intention to reach that goal. In order to assist the human in achieving a set goal, it is the task of the robot to estimate the human intention and to act accordingly. The intention can be communicated either explicitly or implicitly.

The main ways of communicating one's intention are speech, gesture, actions, haptic signals and physiological signals [3]. Speech provides explicit information about intentions through words but can also provide information about emotions implicitly. Explicit body gestures are another communication method that rely on pointing gestures, primitive signs or sign language gestures, while implicit body gestures are called manipulative gestures. Some research developed concerning speech communication method can be found on [7] and [8] and related to gestures recognition technique on [9] and [10]. Another action that can convey intention is proactive task execution, which is used by one partner to detect the exact intention of another partner through provoking an action. When partners are connected haptically either directly or through an object that they both manipulate, intentions can be communicated in a haptical form where force is detected to anticipate the objective of the human and/or to control the cooperation

effort. For example two team partners carrying a large object together may know the other team partner motion intentions by the applied forces or torques on the object [11]. Due to the simplicity of this last communication mechanism it is one of the most implemented in human-robot collaborative tasks [12, 13].

In terms of the robot mechanical design in HRC, applications usually determine the requirement of a low robot mass compared to the payload in order to enable mobility and minimize the injury risk. Furthermore, the International Standards Organisation (ISO) has defined standards or requirements that should be met by companies using work cells with HRC. The standard ISO 10218 defines several requirements like safety-rated monitored stop, hand guiding, speed and separation monitoring, and power and force limiting [14].

When humans interact physically with robots, the robot control faces critical challenges in achieving performances while ensuring stability. Indeed, the stability of the human–robot system depends on the coupled dynamics. Interactions can alter both performance and stability of the two systems when coupled; hence, even if the two systems are stable in isolation, the coupled system may be unstable and not satisfy performance objectives. For example, a human may physically grab the robot to adapt the nominal trajectory to variability in the task: this interaction should always be stable regardless of the forces exerted by the human, gripping force, hand stiffness while grabbing and steering the robot, etc. Most controllers for such interacting systems regulate forces or dynamics behavior of the robot at the interaction port (the location of the interaction, e.g., the end-effector). This control laws will be further explained in section 2.2.

Furthermore, other features are crucial to optimize the collaboration such as legibility. Legibility is the property of generating legible motions that can be easily understood by the human partners. Other important feature is anticipation, i.e., the property of predicting the human intention, the goal of the collaboration, and optimizing the robot control policy to take into account the human action, reducing the human effort or improving some shared performance criteria [15].

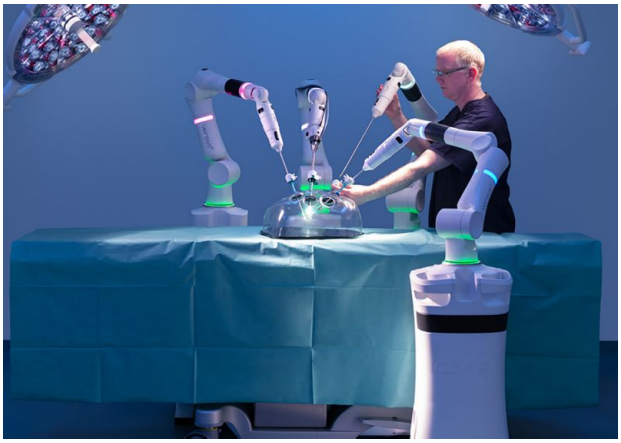
2.1.1 Applications of Human-Robot Collaboration

Robots that until now were mainly on manufacturing environments due to their precision, straight, low price, high speed, durability and robustness are now entering more complex and unpredictable everyday environments with human presence.

One of these environments include healthcare facilities where robots are being developed to, for example, guide the blind [16], support people with walking disabilities, assist in elderly care and work on therapy of autistic children [17]. These robots are all designed to aid diseased or fragile humans and, consequently, must be especially well formulated and scrutinized. Concerning healthcare environments,

robots are also being used to assist medical staff during treatments, diagnosis and surgeries (Figure 2.2). Ethical issues have to be considered particularly in the field of healthcare and medicine. Therefore, in these areas few collaborative-robotic products are commercially available, though they are a major field of research.

(a) NSK robotic guide dog for the visual impaired[1].



(b) Versius robot surgeon[2].

Figure 2.2: Examples of human and robot collaboration applications in healthcare environments.

A seminal field for HRC is construction, where robots can relieve human of carrying heavy loads and ease repeating construction tasks. First steps toward joint construction have been taken by the mobile robot helper that can share loads and handle objects in collaboration with a human.

Robots assisting services that concern home environments are also being developed and some are commercially available. They provide necessary information for daily life, look after the house while the owners are absent and communicate with them [18].

Another field where robots support humans and take on dangerous tasks is urban search and rescue [19]. Robots are designed to collect data, move into buildings with the risk of collapsing and rescue victims, thus avoiding life risk situations to the rescuing team.

Lastly, robots are being developed to collaborate with humans in outer space. One example of these free-flyer robots is CIMON (Crew Interactive Mobile CompanioN) developed by Airbus [20]. CIMON is designed to support astronauts in performing routine work, for example by displaying procedures or – thanks to its neural AI (Artificial Intelligence) network and its ability to learn. With CIMON, crew members can do more than just work through a schematic view of prescribed checklists and procedures; they can also engage with their assistant. In this way, CIMON makes work easier for the astronauts when car-

[1] Mada At Portal (2019). The Guide Dog Robot: Future Mobility Aid for the Blind and Visually Impaired. Retrieved July 3, 2019, from https://madaportal.org/blogs/.

[2] Andy Shipley. Edinburgh News (2019). Meet the robot surgeon hospital bosses want to operate on Edinburgh patients. Retrieved October 10, 2019, from https://www.edinburghnews.scotsman.com/health/.

rying out every day routine tasks, helps to increase efficiency, facilitates mission success and improves security, as it can also serve as an early warning system for technical problems.

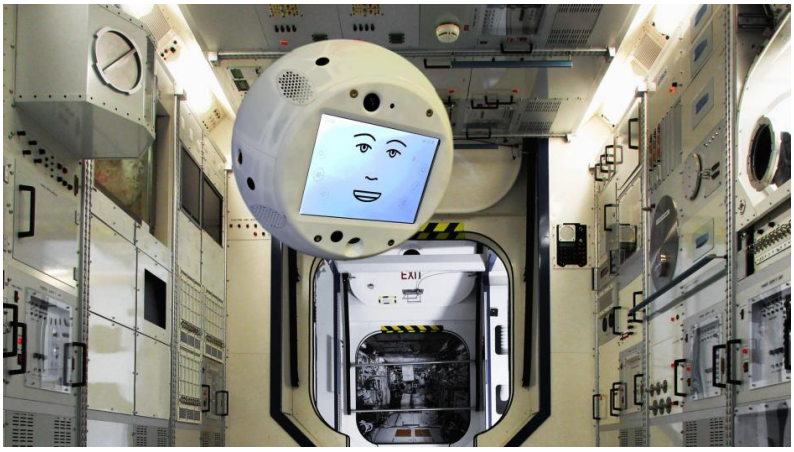


Figure 2.3: Airbus free-flying robot, CIMON [20].

Another free-flyer robot is NASA Ames's robot, Astrobee [21]. This robot is a cube-shaped autonomous free-flying robot with 30.5 cm on each side and a mass of approximately 10 kg designed for IVA and it is fully operable inside the ISS since August of 2019. This robot is an upgrade of the SPHERES (Synchronized Position Hold, Engage, Reorient, Experimental Satellites) developed by NASA and MIT (Massachusetts Institute of Technology). Its general mission is to support astronauts in a variety of missions such as surveying, monitoring, sensing and scientific tasks. Thus, Astrobee's capabilities include markerless vision-based localization, autonomous docking and charging, perching on handrails to conserve energy and carrying modular payloads. Furthermore, the Astrobee is designed to interact with people and it is used for human-robot interaction research due to the additionally relevant features such as a touch screen, speaker and microphone, flash lights and a laser pointer. This components can be used to primarily help crew understand the robot's state and intentions.

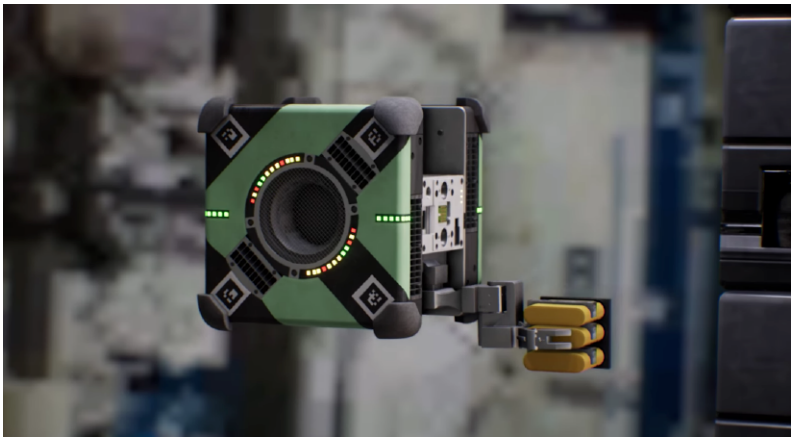


Figure 2.4: Astrobee, developed by NASA, carrying an object [21].

Additionally, Space CoBot developed by ISR (Institute for System and Robotics) [22], is a free-flyer robot designed to operate in microgravity environments due to its hexrotor based propulsion system. It encompasses cameras for navigation, a space for extension modules, a pair of docking ports, a screen for telepresence and a robotic arm. In this manner, the robot was developed to participate in collaborative tasks with humans such as telepresence and cooperative mobile manipulation.

Figure 2.5: Free-flyer robot Space CoBot developed by ISR [22].

2.2 Active Compliant Control

An usual robotic approach to deal with external perturbations and uncertainty is based on robust controllers that preserve stability, in contrast to humans that adapt the interaction endpoint force and viscoelasticity to compensate for external forces, minimising error and effort while ensuring a constant stable attitude. In fact, there is evidence that humans learn and adapt internal dynamic models of their arm/hand and of the environment during interaction and these models are used for predicting muscle activation during planning movements [23]. Franklin in [24] presented evidence that the human Central Nervous System (CNS) regulates the arm compliance by contracting the muscles. More specifically, it was shown that the arm stiffness increased along the direction of the instability. Moroever, Krutky in [25] also demonstrated that the stiffness value was increased along the directions of the destabilizing environments. These findings suggest that at a fixed posture, external perturbations can induce task-appropriate changes in the arm mechanics that are tuned according to the disturbance. This leads into the conclusion that the CNS does not simply contract all the muscles but acts on the endpoint to selectively adapt it to the environment.

Accordingly, an important aspect in setting up a successful, fluent and dynamic human-robot collaboration framework is also to achieve a similar robot adaptability to the task conditions, environmental

constraints and perturbations instead of the usual position control, in which the robot is treated essentially as an isolated system. Thus, the future generation of robots will have to cope with contact tasks under uncertainty in a dynamic manner. To achieve this, the field of compliant control is widely studied.

Compliant control can be defined as the control technology to produce compliant motion [26] and it can be considered part of the continuous feedback force control domain. This method allows to virtually manipulate the compliance properties and respective dynamic behavior of a controlled system. Moreover, compliant motion has been defined as a manipulation task which specifies the contact force between a robot's manipulator and the environment [27] and it occurs when the manipulator position is constrained by the task geometry. Therefore, compliant motion may be produced either by a passive mechanical compliance built in to the system that offers some performance advantages or by an active compliance implemented in the control servo loop. This last method offers the advantage of programmability and it is the one explored in this section.

It is important to state that stiffness and compliance describe a static relation, while impedance and admittance refers to the dynamic relation between the deviation in force $E_F(s) = F_R(s) - F(s)$ and displacement $E_X(s) = X_R(s) - X(s)$. Using Laplace notation, impedance $I(s)$ and admittance $A(s)$ can be computed as:

$$I(s) = \frac{E_F(s)}{E_X(s)} = A^{-1}(s) \tag{2.1}$$

where $X_R(s)$, $X(s)$, $F_R(s)$ and $F(s)$ represent the reference and measured position and force, respectively. The order of an impedance or admittance then refers to the highest exponent of s.

Although numerous specialized variants and combinations of active compliant control approaches can be found in literature, there are four main distinct branches on this control field: parallel force/position control, hybrid force/position control, admittance control and impedance control [28]. Figure 2.6 represents each one of this methods and the selection process given the system properties and the goal of the task.

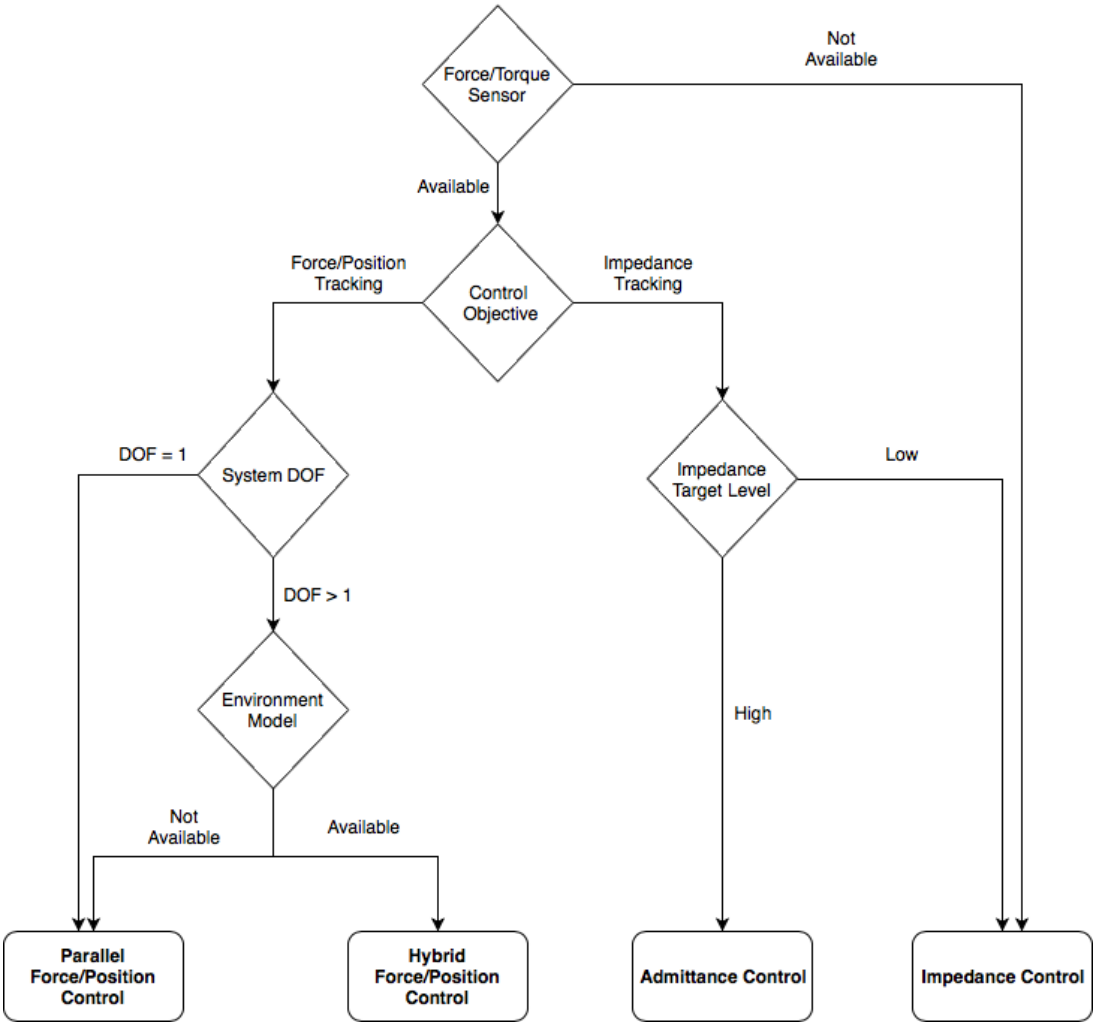


Figure 2.6: Active compliant control branches.

Analysing the previous figure it is possible to conclude that only one of the four main active compliant control approaches does not require force/torque sensor. Therefore, if a system does not have access to force data the implementation is limited to impedance control. Regarding the control goal, while Hybrid and Parallel Force/Position control aim at tracking a target force or position, impedance and admittance control regulate their relation, but not explicitly their individual trajectories. The Hybrid Force/Position control approach requires the system to have multiple degrees of freedom (DOF) and a detailed environment or a task model to enable the division of the working space in position and force-controlled sub-spaces according to the environmental constraints. If these conditions are fulfilled, it can be beneficial to be applied as it allows to exploit the model knowledge and it provides faster dynamics that reduced design complexity compared to the more robust parallel force/position control approach [28].

Lastly, it is important to state that *force* holds for *force* and *torque* and the term *position* stands for *posi-*

tion and *orientation*.

The next sections describe more in-depth these control concepts while exploring their robotic applications and relevant work developed.

2.2.1 Hybrid Force/Position Control

The objective of Hybrid Force/Position control is the tracking of the reference motion trajectory in unconstrained directions and the control of the contact forces arising in constrained directions.

This control method is described, using Laplace notation, by:

$$U(s) = S\ P(s)\ (X_R(s) - X(s)) + (I - S)\ W(s)\ (F_R(s) - F(s)) \tag{2.2}$$

where $U(s)$ represents the control and $U(s) = U_x(s) + U_f(s)$ with the indices x standing for position and f for force. The position and force control law is represented by $P(s)$ and $W(s)$, respectively, and their structure and parameters can be selected accordingly to the system characteristics and implementation objectives. In literature, they are usually selected following a PID or PD approach [29, 30]. Furthermore, I is the identity matrix with $n \times n$ size. Additionally, the compliance selection matrix is represented by (2.3).

$$S = diag(S_j) = \begin{bmatrix} S_1 & & & 0 \\ & S_2 & & \\ & & ... & \\ 0 & & & S_n \end{bmatrix} \tag{2.3}$$

with $j = 1...n$ and n being the number of degrees of freedom dividing the workspace into complementary orthogonal sub-spaces, which are either motion or force-controlled. While motion constrained directions must be force-controlled, $S_j = 0$, free directions need to be motion-controlled, thus $S_j = 1$.

The sub-spaces and their corresponding constraint types are derived from an accurate environment model, which is always required for hybrid force/position control [31]. In fact, this control approach was developed very closely together with the Task Frame modeling and specification approach where a contact state is modeled by means of a so-called Task Frame or Compliance Frame. A Task Frame is an orthogonal reference frame, placed at an appropriate location in the contact, such that the available degrees of freedom of the objects in contact can be modeled as either pure motion freedoms or pure contact forces along one or more of the frame's axes [32, 33]. Therefore, the directions of the Task Frame are divided into motion-controlled directions and force-controlled directions as mentioned. Force-

controlled directions are transformed into motion-controlled directions, by means of a feedback action resulting in a twist setpoint with dimensional translational and angular velocity for the Task Frame that then can be transformed to be given as a input for the robot controller [34].

The major advantage of this method is the execution efficiency. Moreover, developing an appropriate Task Frame is usually not complex for most simple contacts. This method also has the advantage of the independent design and implementation of position and force control law and it is effective for high stiffness environments. Furthermore, Raibert in [35] shows that this approach achieves stable, accurate control of force and position trajectories for a variety of test conditions.

High complexity implementation for non-planar surfaces can be seen as an disadvantage. Other disadvantages may include the requirement for force measurements.

2.2.2 Parallel Force/Position Control

In the same way as Hybrid Force/Position control, the Parallel Force/Position control aims to track the reference motion trajectory in the unconstrained directions and to control the contact forces arising in the constrained directions. The difference from Hybrid Position/Force control lays on the fact that the force and motion controller outputs are superimposed and can hence act on the same directions on the Task-Frame. So, force and motion are not decoupled. The integration of the Task Frame with the Parallel Force/Position control scheme is on the same low complexity as the integration with the Hybrid control approach.

This method control law can be described by:

$$U(s) = P(s)\ (X_R(s) - X(s)) + \ W(s)\ (F_R(s) - F(s)) \tag{2.4}$$

This approach has the advantage of a decoupled force and position trajectory tracking, robustness and safety in the presence of environment model uncertainties and planning errors. Additional, it is useful on industrial robots as Chiaverini have shown in [36] and [37] where a successful experimental industrial robot research was conducted using PID-type parallel controller. As in Hybrid Force/Position control a force measurement is required. Moreover, this control method presents slower dynamics compared to the Hybrid approach.

2.2.3 Impedance Control

Impedance Control (IC) is also called force-based impedance control, impedance control without force feedback or equilibrium point control and aims to implement a target relation between force and motion

but does not necessarily track their individual trajectories.

The pioneering work in this control field is presented by Hogan in [38] and [39] where it is explored the impedance and admittance control framework, as a dual representation.

Most controllers for interacting systems regulate forces or dynamics behavior of the robot at the interaction port/end-effector. A distinction between impedance control and those conventional approaches to manipulator control is that the controller attempts to implement a dynamic relation between manipulator variables such as end-point position and force rather than just control these variables alone, modelling the system as a spring-mass system, with desired stiffness and damping, where the mechanical impedance of the manipulator is regulated to a target model. Thus, the goal of impedance control is to make the manipulator behave according to a desired dynamic model. To achieve this goal, the impedance control system creates a virtual new impedance for the manipulator, which is being able to interact with the environment as if new mechanical elements had been included in the real manipulator.

Moreover, both force control and motion control are achieved simultaneously and indirectly by controlling a dynamic relationship between the two. This contrasts with the Hybrid and Parallel Force/Position control which uses two separate feedback loops for position control and force control. Additionally, this approach can achieve motion control of the manipulator in free space, i.e., in the absence of any contact.

Furthermore, IC can be implemented explicitly or implicitly depending on the availability of force data. In the first case, there has to be access to force measurements and a compromise for a well performing but stable force controller has to be implemented. This control law can be described by:

$$U(s) = W(s)\ [Imp(s)\ (X(s) - X_R(s)) + F_R(s)] - F(s) \tag{2.5}$$

with

$$Imp(s) = \sum_{k=0}^{m} a_k\ s^k \tag{2.6}$$

and

$$U(s) = U_f(s) \tag{2.7}$$

This method depends on a mechanical impedance $Imp(s)$ which is reflected by the linear impedance control law as polynomial of mth order with the parameters a_k as is illustrated in (2.6) [28].

It is important to notice that $X_R(s)$ is also called rest instead of reference or desired position.

In the implicit case, no force data is required. Therefore, only the position $X(s)$ is measured and used for the inner loop reference force $U_f(s)$ as shown in Equation (2.8).

$$U_f(s) = Imp(s)\ (X(s) - X_R(s)) + F_R(s) \tag{2.8}$$

The impedance parameters choice, a_k, can be considered a trade-off between the allowed contact force and deviation from the reference motion trajectory. For a high stiffness relation between the system and the environment, the system endpoint tends to reach the desired steady-state rest position at the expense of penetration into the environment. For a low one, the endpoint rather adopts to environmental constraints [40]. The commonly used m values are 0, 1 and 2. Theoretically, the general approach allows for $m > 2$ but most of the literature does not mention the practical implementation of a such high order and for $m < 2$ physically meaning can be attributed to the parameters as going to be described next.

With $m = 0$ it is considered impedance control of zeroth order or stiffness control [41]. In such case, the impedance can be represented by:

$$m = 0 \Leftrightarrow Imp(s) = a_0 = k_D \tag{2.9}$$

with k_D being the target stiffness. In the case of implicit control, it is formally equal to a P position controller. It is relevant to state that stiffness control is less computationally demanding, complex and sensitive to model uncertainties than high-order and dynamic model-based impedance control but also possibly reduces stability and accuracy in the explicit case.

The case where $m = 1$ is usually called damping control where the velocity error, $\dot{X}_R(s) - \dot{X}(s)$, is fed back and multiplied by the target damping coefficient k_B [42]. This can be described by:

$$m = 1 \Leftrightarrow Imp(s) = a_1\ s = k_B\ s \tag{2.10}$$

However, pure damping control can barely be found in the literature. Generally, a stiffness term is added which then results in impedance control of the first order. Therefore, in the first order implicit case, impedance control is formally equal to a PD position controller [40, 41, 43], resulting in a desired dynamic relationship between the contact force and the motion of the robot.

$$m = 1 \Leftrightarrow Imp(s) = a_0 + a_1\ s = k_D + k_B\ s \tag{2.11}$$

Lastly, implicit second order impedance control is formally equal to PID velocity control [44]. Equation (2.12) describes this impedance formulation.

$$m = 2 \Leftrightarrow Imp(s) = a_0 + a_1\ s + a_2\ s^2 = k_D + k_B\ s + k_M\ s^2 \tag{2.12}$$

where k_M is the at the target mass or inertia.

Figure 2.7 represents schematically an impedance control behavioral scheme using a mass-damper-spring simplified system.

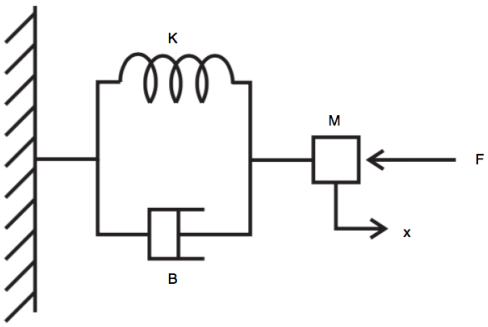


Figure 2.7: Mass-damper-spring system schematically representing an impedance controller behavior, where M, B and K are respectively mass, damping constant and spring constant.

Lastly, as disadvantages of this control method, it is possible to point out the absence of an exact tracking of the position and the force and the requirement for force data in the explicit case.

2.2.4 Admittance Control

Admittance Control (AC) is also called position-based impedance control or impedance control with force feedback and it is an approach dual to impedance control. In IC the controller is an impedance and the manipulator is an admittance and in AC the controller is an admittance and the manipulator is an impedance [45].

As in impedance control, admittance control can also be implemented explicitly or implicitly. For implicit admittance control, the outer force feedback loop is excluded. Therefore, depending on the selection of

the inner loop position control law, implicit admittance control is equivalent to implicit impedance control. Hence, implicit admittance control is commonly not considered. Furthermore, explicit admittance control can be described by:

$$U_x(s) = Adm(s)\ (F(s) - F_R(s)) + X_R(s) \tag{2.13}$$

with

$$Adm(s) = (\ \sum_{k=0}^{m}\ a_k\ s^k\ )^{-1} \tag{2.14}$$

where $Adm(s)$ is a mechanical admittance that relates the measured external force $F(s)$ to the reference motion $U_x(s)$ for the inner position control loop.

Impedance and admittance control distinctly differ in their capabilities to render certain target impedance levels and in the key characteristics of their inner control loop. While for IC a high target impedance level results in high control gains, which possibly cause instability, AC is prone to instability in case of a low target impedance level.

Regarding the m values, as in IC, most of the literature does not mention the practical implementation of $m > 2$. Accordingly, for $m = 0$, called the admittance control of zeroth order is the counterpart to stiffness control. The control parameter selection can be related to the system stability [45]. Stiffness and compliance control can also be referred to by a generalized spring or by artificial or active compliance.

Moreover, zeroth order AC is a result of an admittance described by:

$$m = 0 \Leftrightarrow Adm(s) = (a_0)^{-1} = (k_D)^{-1} \tag{2.15}$$

With $m = 1$ the control law becomes admittance control of first order or accommodation control [46] as seen in Equation (2.16).

$$m = 1 \Leftrightarrow Adm(s) = (a_0 + a_1\ s)^{-1} = (k_D + k_B\ s)^{-1} \tag{2.16}$$

It is an approach dual to damping control and synonymously named active accommodation or generalized damping [32]. Accommodation control is especially relevant for slow motion applications as the

inertia characteristics of the closed-loop system are negligible in this case.

Lastly, for $m = 2$ or admittance control of second order, it is possible to manipulate the mass or inertia characteristics of the system. This is shown in (2.17).

$$m = 2 \Leftrightarrow Adm(s) = (a_0 + a_1\ s + a_2\ s^2)^{-1} = (k_D + k_B\ s + k_M\ s^2)^{-1} \tag{2.17}$$

2.3 Object Handover between Humans and Robots

An object handover occurs anytime two parties exchange an object without the use of an external tool. Object handover is a basic, but essential capability for robots to interact with humans in several applications.

The agents involved in a handover are the giver and the receiver. The giver is defined as the agent presenting the object. The receiver is defined as the agent who is accepting the object. While both participate in the exchange, each have a different goal: the giver wants to safely release the object while the receiver wants to readily acquire the control of the object and establish a stable grasp [47].

It appears deceptively simple, as humans perform object handover almost flawlessly and in a very fast way. These fast and seamless events occur in a wide range of environmental conditions and for objects ranging in size and mass. Furthermore, object handovers between humans involve kinodynamic (kinematic + dynamic) negotiation, gesture and motion prediction, coordinated hand-eye motions, and tactile sensing on the part of both participants - all within a matter of seconds.

The success of humans, however, belies the complexity of an object handover as collaborative physical interaction between two agents with limited communication. In fact, object handover between human and robots is an instance of the more general problem of cooperative manipulation [48] involving two asymmetric agents with limited communication. To become useful partners in human-robot teams, robots will need to approach a certain level of capability to be able to perform fluent and dynamic object handovers as both giver and receiver.

Generally, an handover consist of a three states: an approach phase where one or both parties move(s) to the handover location, a transfer phase where the object is physically transferred from the giver to the receiver and, after transferring the entire object load, often the receiver retracts or otherwise moves the object to signal the giver that the handover is complete, ending the retraction phase and consequently the handover task [49].

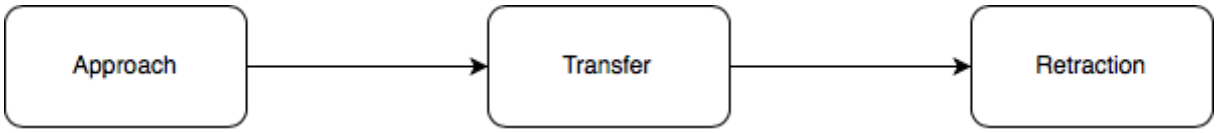


Figure 2.8: Object handover schematic phases.

Although all handovers may be described by the previously states phases, the work developed within this dissertation can be considered to span two different tasks: robot-to-human handovers and human-to-robot handovers. At first glance, it may appear that there is very little distinction between these two domains since both have robot and human agents participating in a single type of activity. However, from the perspective of the robot, these tasks are very different: In the case of a robot giver passing an object to a human, this includes having the robot provide clear and understandable cues to indicate that it wants to hand over the object, fluently and dynamically negotiate where and when the handover is to occur, recognize when the human user has grabbed the object and appropriately release it before safely retracting. For the inverse situation where a robot receives an object from a human, the robot must interpret the human nonverbal or verbal cues that the object is placed on the its gripper, fluently and dynamically negotiate where and when the handover is to occur, obtain and maintain a stable grasp of the object and retract with it. Both of this tasks encompass very different technical challenges.

Therefore, the capability of robots to carrying out handovers autonomously requires perception, planning and decision inference capabilities for each of the handover phases. Answers to the technical challenge arising from both tasks of object handover determine the handover configurations, i.e., the configurations in which the object transfer occurs. Most of the studies on human–robot handovers present in literature propose several configurations and methods on how provide the robot with the capabilities mentioned. Thus, the literature on this field will be explored in the Section 2.4 concerning the State-of-the-Art on this dissertation main problem.

2.4 State-of-the-art

Firstly, it is important to highlight that, to the best of the authors knowledge, *no* record on the research field regarding object handover between humans and robots in microgravity was found in the literature. Consequently, the state-of-the-art research was conducted concerning human-robot object handover in a terrestrial environment and relevant microgravity work is also acknowledged.

It is interesting to state that the field of object handover between humans and robots has generated curiosity on the research community since the late 90's from both physical and social-cognitive perspectives. The pioneer work done on this topic goes back to 1992 [50]. Nowadays, with the growth and impact in the collaborative robots, human and robot object handover is a vast and emerging field. Thus, a considerable large number of publications concerning this topic can be found in literature in which

several aspects of the previously described handover phases are explored.

In order to achieve a clear and organized review of the studied literature, the State-of-the-Art presented is divided in each handover phase.

2.4.1 Approach Phase

Concerning the approach phase, certain findings regarding proxemics or social communication have strong implications on how handover interactions should be designed. Therefore, Takayama and Pantofaru in [51] explored how personal space varies when approaching and being approached by a robot based on the human's experience with robots and where the robot looks during the approach. Such factors should be considered when designing handover behaviors in order to respect the human's personal space.

Furthermore, Aleotti, Micelli, and Caselli in [52] stated that, in a robot-to-human handover task, robots should take into consideration how the human will grasp the object and thus robots should approach the user with the most appropriate and easiest to grasp part of the object, usually an handle.

Another study conducted by Sisbot and Alami in [53], designed a safe but socially acceptable robot motion in handover tasks. The author achieved this proposed handover characteristics by considering human kinematics, vision field, posture, and preferences, as well as the legibility of the robot's actions. In a handover task performed with this framework, the system chooses robot paths such that the human can easily grasp the object being handed over, while the robot both maintains a safe distance and moves in a manner that is visible to the human. Additionally, the action is made legible by directing the robot's gaze toward the object being handed over at the appropriate time.

In a different study by Cakmak et al. (2011) [54], the author explored the use of the robot's postures and gestures in handovers and found that certain postures can better communicate a robot's intent. The researchers conducted an online survey asking participants to classify different postures of a robot holding an object. Results showed that postures with an extended arm were most frequently classified as handing over.

In [55] Koay studied which angle should the robot approach a person in the robot-to-human handover task. In experiment conducted, a robot was programmed to approach the person receiving the object from different angles. The conclusion reported from this study was that most participants preferred the robot to approach them and transfer the object from the front declaring that it provided the most visibility of the robot's motion.

2.4.2 Transfer Phase

A crucial aspect of an human-robot handover task is the object exchange or the transfer phase. Several aspects of this phase have been proposed in literature.

For example, concerning the object and the robot's gripper pose in which the transfer occurs, Edsinger and Kemp in [56] found that, during a human-to-robot handover, humans will pose an object in the robot's stationary hand regardless of the robot's hand pose, demonstrating that humans adapt to the robot's hand pose in the case of human-to-robot task, thereby simplifying robotic grasping and offering opportunities to simplify the manipulation task. Additionally, the authors also implemented an state decision tree, that included the handover three phases described in the previous section. An additional useful finding by Chan et al. (1996) [57] was that the giver has a post-unloadind phase towards the end of the handover, where the giver applies a positive grip force even though their load force is approximately zero. This, according to the authors, implies that the giver takes responsibility for the safety of the object.

Another relevant characteristic of the transfer phase, is the communication of intent done by signaling. This includes having the robot provide clear and understandable cues to indicate that it is ready to hand over an object, in the case task of a robot-to-human handover, or that it wants to accept the object on the other way around. Dragan showed how this signaling can be done by motion [58] or head orientation [59] in robot-to-human tasks. Hendrich and Bistry in [60] implemented signalling with sounds and speech to indicate that the robot was ready to the handover. Moreover, Strabala in [61] studied the importance of robots responding to humans using human-like gestures and signals. However, in addition to human-like signals, the authors claim that special signals can be used when the human and robot share the meaning of these signals in a common ground.

Several techniques have been proposed to decide when a robot hand or gripper should release an object for a human to receive or to detect that an object has been placed in the robot's end-effector. Dragan in [58] studied the robot's decision to release the object through an Hidden Markov Model that estimates the state of the handover. Nagata, Oosaki, Kakikura, and Tsukune et. al (1998) [62] presented a grasping system based on force and torque feedback that senses when the humans has a stable grasp on the object, after which the robot can release the object. Hendrich and Bistry in [60] supported a force threshold to release the object. Moreover, their experiments indicate that force thresholds must be matched to the object weight to achieve interaction that feels natural to the users. This approach is not feasible in microgravity as the objects are weighless is such an environment. Furthermore, to ensure object safety, Bohren in [63] developed an displacement-based method. In this study, the robot was programmed to serve beverages to people and when handing over a drink, to avoid accidentally dropping it, the robot was programmed to release it when someone pulls hard enough on the bottle such that the robot's arm displaced more than one centimeter vertically. Additionally, Edsinger and Kemp in [56] monitored the velocity of the robot's end-effector in order to detect when an object has been placed in it or is being pulled out of its gripper.

Certain handovers can be planned in advance, however the robot needs to be responsive and dynamic during the interaction, particularly during the transfer of the object. Thus, controller-based approaches for robot handovers exist, and generally attempt to reach a desired goal position [64], often by following a desired trajectory [65]. Edsinger and Kemp [56] used a time-based handover controller and showed that in certain situations a time-based handover controller can results in a high rates of dropping the object. Nagata in [62] implemented a grasping algorithm controller for transferring objects from and to humans. The study platform consisted of a four-fingered gripper with individual 6-axis force-torque sensors at each fingertip. Using force-torque data from each fingertip, the grasping algorithm continually evaluates the grasp stability by monitoring finger slippage. The author algorithm took into account manoeuvrability, joint angle range, and contact surface geometry. During grasping, if the stability constraint of a finger is violated, the algorithm reposition the finger to maintain a stable grasp. Upon sensing contact at one of the fingers, the robot softly closes its fingers, as the person adjusts the object into a comfortable position and orientation in the robot's hand. To take the object from the robot, the person is required to pull on the object such that it becomes unstable in the robot's hand. The robot then let the object go when stability conditions are no longer satisfied. Although this algorithm allows the robot to continually maintain grasp stability, this controller has the disadvantage of the requirement of individual 6-axis force torque sensors. On Kupcsik's study [66] a Cartesian impedance control was used. The author supported that for static handover tasks using Cartesian compliant control, compliance parameters are less important for success and high stiffness is always preferred and highly rated. Furthermore and continuing their previous work, Chan and Kumagai in [67] presented an implementation of an human-inspired handover controller on a robot with compliant under-actuated hands, using joint position error measurement for estimating the amount of applied grip force. Through the experiment, the authors concluded that is possible to achieve safe, smooth, and intuitive robot-human handovers despite the lack of accurate grip force control of their robot.

Although Hermsdörfer et al. (2000) [68] research was not conducted with the goal of specifically studying the transfer phase of the handover, it is important to be reviewed due to the fact that the author explored the effects of microgravity on human's grip force control when manipulating objects. Hermsdörfer concluded that humans are able to adjust their grip force to account for dynamic load forces under microgravity, even though the pattern of load force changes were completely different from those under normal gravity. Moreover, the coupling between grip force and load force was also preserved during transitions between different gravity levels. In this manner, it is possible to concluded that his study showed that human's grip force control remains consistent even with massive environmental condition changes. This research is important to ensure the safety of the object from a human perspective in the transfer phase on microgravity robot-human handovers.

2.4.3 Retraction Phase

Concerning the last handover phase, there is not a considerable quantity of information in literature. According to Strabala in [61], no signal needs to be done to demonstrate that the transfer phase is over. After transferring the entire object load, often the receiver will retract or otherwise move the object, indirectly signalling the giver that the handover is complete. The giver will then also retract and thus ending this phase and, consequently, the handover task.

Chapter 3

Handover Algorithm Formulation

An handover structure resonates with the interaction theory in Clark [69] by representing the handover as a sequence of phases using a common ground between the giver and the receiver, i.e., the information that the participants in the handover believe they share. Moreover, the human and the robot participating in the task must synchronize with the transitions between these phases.

The formulation of a successful, fluent and dynamical handover requires the integration of multiple states in each handover phase that together allow a robot to perform the task. The development process of such states often involves the use of a Finite State Machine (FSM). A FSM can change from one state to another in response to some external inputs and/or to condition satisfaction and this change is called a transition. A FSM is defined by a list of states and conditions for each transition. State machines are vastly used in robotic due to its various properties, such as the ability to make the program execution transparent and reproducible [70]. This is particularly important for robot experiments which need to be designed in a such a way that consistent results are obtained. Additionally, the use of a state machine design in the proposed tasks allows to easily change transitional conditions or states within each handover phase.

Before analysing the algorithm proposed, it is important to refer the requirements and assumptions concerning the robot performing the tasks:

- It is assumed that the robot base is a free-flying system designed to actuate in a microgravity environment;
- It must be equipped with a manipulator with an operational gripper. The gripper will also be further referenced as the robot's end-effector.

Regarding the requirements on the object manipulated, it must be rigid and its physical properties known.

Figure 3.1 displays the state-machine based algorithm formulated. This formulation encompasses all relevant states and transitions concerning an object handover. Furthermore, the next two sections explore

a predetermined sequence of states and transitions that describes the two tasks of interest: robot-to-human handover or human-to-robot handover. Additionally, it is relevant to note that the state *close gripper* differs from the state *grasp object* in the sense that the gripper may be closed but not holding the object.

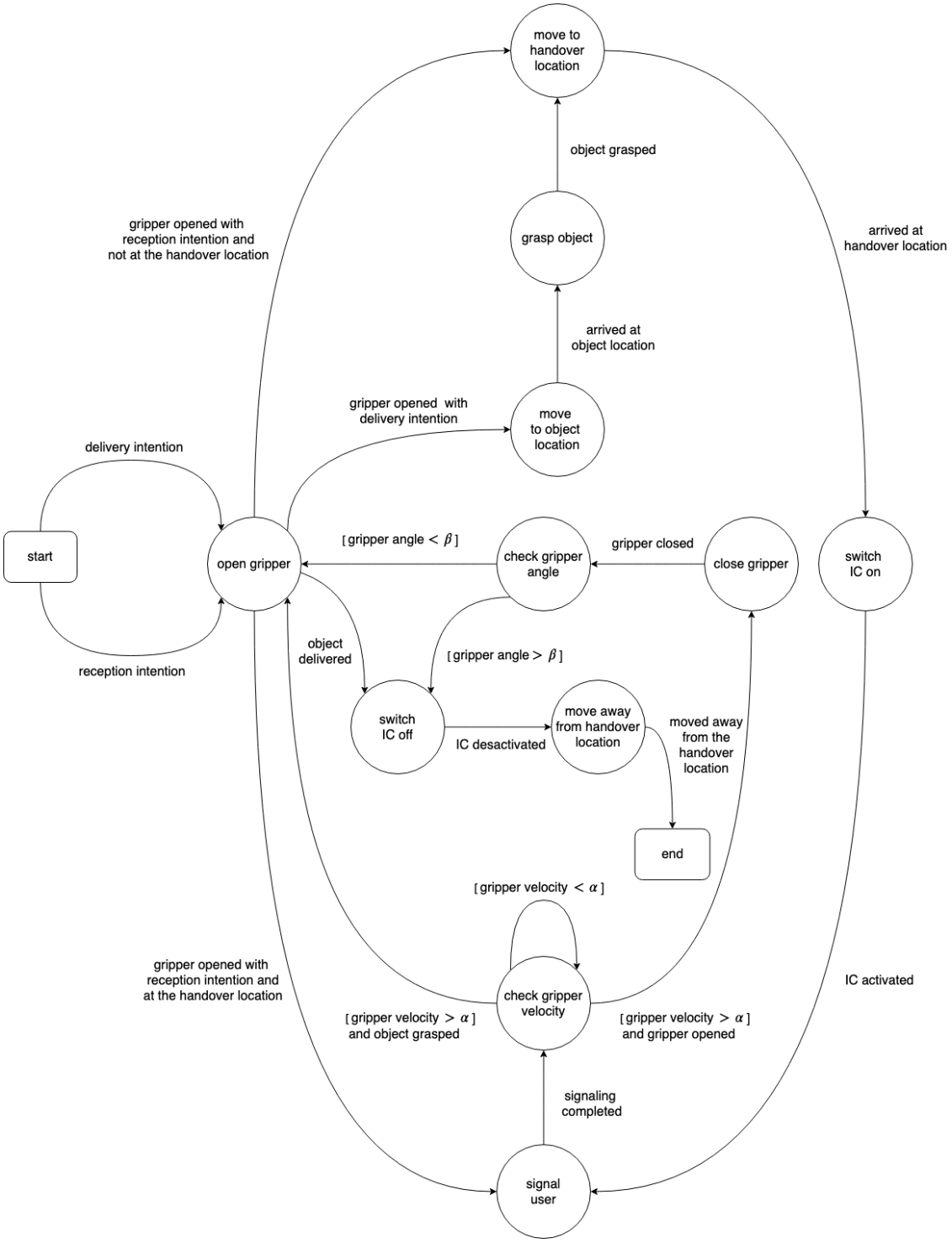


Figure 3.1: State-machine based algorithm concerning all the possible states and transitions of an handover between a robot and a human.

3.1 Robot-to-Human Handover

A robot-to-human object handover task aims to achieve an object transfer from a robot to a human where the robot acts as the giver and the human as the receiver.

Assuming that the receiver is ready to participate in the handover, the robot is required to move to the handover location with the object, to provide a clear and understandable cues to indicate that it wants to hand over the object, recognize when the receiver has grabbed the object and, lastly, to appropriately release it, before retracting away from the handover location.

Having the handover requirements into consideration, a sequence of states and transitions of the FSM proposed were selected as Figure 3.2 displays in blue.

As mentioned in Chapter 2, the handover task comprehends three distinct phases: approach, transfer, retraction. Thus, in order to encounter with the task requirements, each phase hold several states of the proposed FSM based algorithm.

3.1.1 Approach

Initially and assuming that the intention to perform a handover has already been established, the first state involves the opening of the gripper, followed by the movement to the object location which is assumed to be known by the robot. Upon arrival, the object must be grasped. It is important to refer that no specific grasping algorithm was designed given that the object grasping field is considered out of the current research goals scope. However, as referred in Section 2.4.1, it should be taken into consideration how the human will grasp the object and thus the robot should approach the human with the most appropriate and easiest to grasp part of the object, usually an handle. In the case of a cuboid or cylindrical object, the robot should grasp one end of the object, leaving space for the person comfortably grasp the other end. Additionally, the chosen object shape and size must encounter the robot's gripper capabilities.

The next states involves the robot's movement into the handover pre-assigned location. As this thesis does not focus on trajectory planning and rather focus on the dynamical relation during the handover interaction, the robot's mobility can be achieved resorting to any position controller such as a simple PID. Moreover, the robot should approach the user from the front as this angle provides him/her the most visibility of the robot's motion [55]. Furthermore, the robot's arm should be extended as this indicates the intention of approaching the user to hand over the object [54].

3.1.2 Transfer

Once arriving to the handover location, the next phase encompassed in the handover sequence is the transfer. In this phase the object is physically transferred from the giver (the robot) to the receiver (the user).

Most handovers can be planned in advance, however an important aspect in setting up a successful, fluent and dynamic object transfer is to achieve robot adaptability to the task conditions, environmental constraints and perturbations, i.e., the user interaction, instead of the usual position control in which the robot is treated essentially as an isolated system. As the robot needs to present a dynamic response between the environment acting on its structure and its motion, a control-based approach that implements this dynamic relation between the manipulator and the environment must be formulated. Having this requirements into consideration and analysing Figure 2.6 regarding several compliant control approaches, an impedance controller was chosen (section 2.2.3), more specifically an implicit impedance control as this method controls both force and motion achieving simultaneously and indirectly a dynamic relationship between the two, instead of using two separate feedback loops for position and force control as done in hybrid and parallel position/force approach. This control method can also achieve motion control of the manipulator in free space, i.e., in the absence of any contact. Additionally, this choice relies on the fact that the robot may not be equipped with a force/torque sensor, leaving implicit force control the obvious choice within compliant control. Furthermore, the goal of the impedance control approach is to make the manipulator behave according to a desired dynamic impedance model. Due to the extensive control formulation, this module is further analysed in section 3.3 of the current chapter.
The robot is ready to deliver the object when the impedance control is activated.

The impedance control activation is followed by the state regarding the user signaling in which the robot should communicate to the user that it is ready to deliver the object. As referred in the Section 2.4.2, the handover signaling can be done using several approaches such as motion, head-orientation, posture, speech, sounds or gestures. Furthermore, in [61] the author claims that special signals can be used if the human and the robot share the meaning of that signals in a common ground. Therefore, the signaling strategy can be adapted to the robot's technical capabilities and functionalities.

Another relevant stage of the transfer phase is the robot's decision concerning the appropriate instance of releasing the object. Several methods to achieve this knowledge have been explored in the state-of-the-art section regarding terrestrial handover. Such methods include the commonly used force threshold that matched the object weight [60]. This approach is unfeasible in microgravity as objects are weightless. Another methods were studied based hidden Markov models that estimated the state of the handover [58] or approaches based on force/torque sensing [62]. This last approaches requires the end-effector force/torque data and thus can not be implemented in robots without a gripper sensor. Therefore, following the work developed by Edsinger and Kemp in [56], the robot's end-effector velocity was monitored in order to detect when an object was being pulled out of the robot's gripper. This ap-

proach allows to manually tune the velocity threshold, α, accordingly to the robotic behavior proposed. In this manner, if the end-effector velocity is higher then the defined threshold and the robot is grasping the object, the user's receiving intention is detected and the robot will open the gripper. Otherwise, it will continue checking the gripper velocity holding on to the object.
It is considered a success handover when the object load is delivered to the receiver without being supported anywhere else and it is always in contact with at lest one of the participating agents.

3.1.3 Retraction

The retraction phase is the last phase of the handover sequence. If the object has been delivered, the robot must switch off the formulated impedance control as no more dynamical interaction will be made. As referred in the approach phase, a simple position controller is again activated to allow the robot's movement to another location other then the handover one.

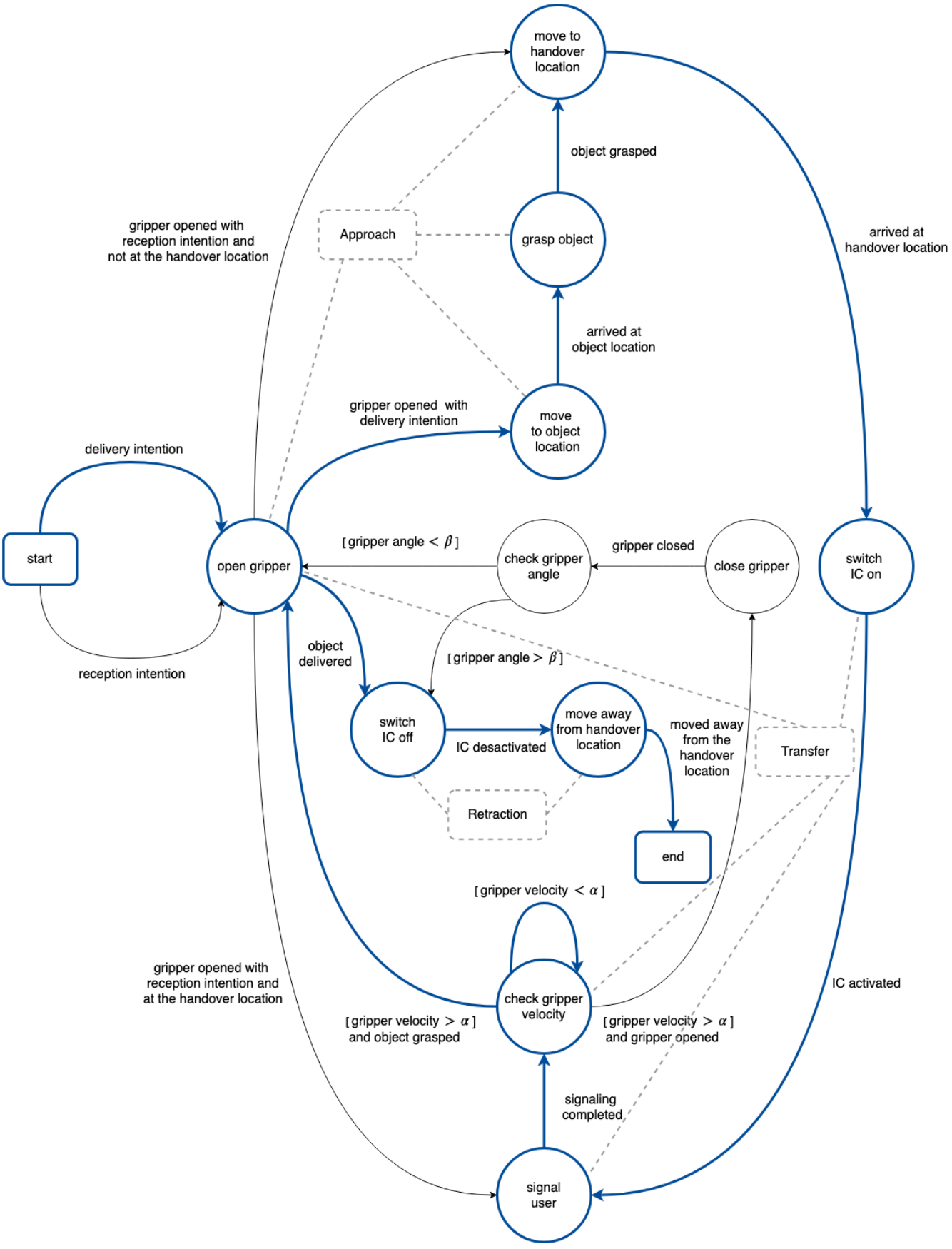


Figure 3.2: State-machine based algorithm sequence, in blue, regarding a robot-to-human handover. The three phases on the handover are also presented.

3.2 Human-to-Robot Handover

The most commonly implemented handover tasks between humans and a robots rely on the robot as a giver and the user as the receiver. Nonetheless, several tasks also require the other way around: human-to-robot object handover.

In this handover task, the robot is the receiver and the human performs the giver role. Although both tasks may resemble fairly similar due to the participation of a human and a robot on the object transfer task, from a robot's perspective encounters different challenges: the robot has to signal the user of its intention on receiving the object, interpret the human nonverbal or verbal cues that the object is placed on the its end-effector, fluently and dynamically negotiate where and when the handover is to occur, obtain and maintain a stable grasp the object and retract with it.

To overcome this challenges, a sequence of transitions and states of the proposed FSM describes a human-to-robot handover and it is presented in Figure 3.3, in blue.

As in the robot-to-human handover, the proposed handover encompasses three phases: the approach, the transfer and the retraction phase, in which several states were developed to encounter the task requirements. Although some states are shared between the two handover tasks, others are exclusive to the human-to-robot handover.

Lastly, the same requirements as in the previous section are made regarding the robot's mobility, manipulator and object.

3.2.1 Approach

The approach phase is initiated without the object and it is assumed that the robot already acknowledge the intention of receiving an object. As in the previous task, the robot's arm should be extended in the approach phase [54], as this indicates the intention of moving to the handover location to receive an object. Furthermore, the opened gripper during the approach stage emphasis that intention, not only for the user participant in the task, but for others present in the same space. The movement to the pre-assigned handover location relays on a PID position controller

Furthermore, the transition to the first state of the transfer phase depends on the arrival to the handover location.

3.2.2 Transfer

The transfer phase in a human-to-robot handover also relays on the robot's dynamical behavior towards the external perturbations, i.e., the user interaction. Therefore, the formulated controller must implement

a relationship between the external generalized forces acting on the robot's structure and its motion. On that premise, the same impedance control approach was used for this task. As already mentioned, further details on the control formulation are available in Section 3.3. Upon activating the impedance control, the robot must signal the user with the aim of communicating that it is ready to receive the object. Several approaches to accomplish this signaling were already discussed in Section 2.4.2 and on the previous section concerning the robot-to-human handover.

Furthermore, the robot must detect that the object has been placed in its end-effector. As the object weight can not be taken into consideration for the detection in a microgravity environment and force/torque sensing data may not be available, the work developed by Edsinger and Kemp in [56] was again taken into consideration. The main difference between the previous task is the value of velocity threshold, α, as this must remain dependent of the robotic behavior implemented and must be once again tuned given the different interaction. Lastly, if the gripper's velocity is higher then the defined threshold, α, the robot must close the gripper. Otherwise, the robot must keep checking the gripper velocity.

Upon closing its gripper, the robot must verify the object reception success. This can be done by checking the resulting grasp aperture, which gives a strong indication of whether or not the object was successfully grasped. The grasp aperture or gripper angle, is typically defined as the angle between the gripper main links. If it is positive and above a threshold, β, then the gripper is assumed to be wrapped around an object. In order to define β, the grasp aperture of the closed end-effector without holding any object, β_0, must be determined. The value of β is then defined given that β must be higher then β_0, otherwise the gripper is closed without holding to an object.
If the referred condition is true, then the retraction phase initiates, otherwise the robot must re-open the gripper and signal the user regarding the availability to receive the object once again, showing the acknowledgement of the a failed transfer. The described states and transitions generate a failure reception module.

On the same note, it is important to refer that a failure delivery module was not considered for the robot-to-human handover task given that when the robot opens the gripper and the object is not being grasped by the user, it is highly probable that the object thrived away from the robot's gripper due to external perturbation that leaded to the exceed of the velocity threshold. In this manner, a more complex failure delivery module would be required such as visual recognition of the object or the user.

3.2.3 Retraction

Finally, in the retraction phase the impedance control must be switched off and, implicitly, a mobility controller switch on, allowing the robot to move away from the handover location with the object.

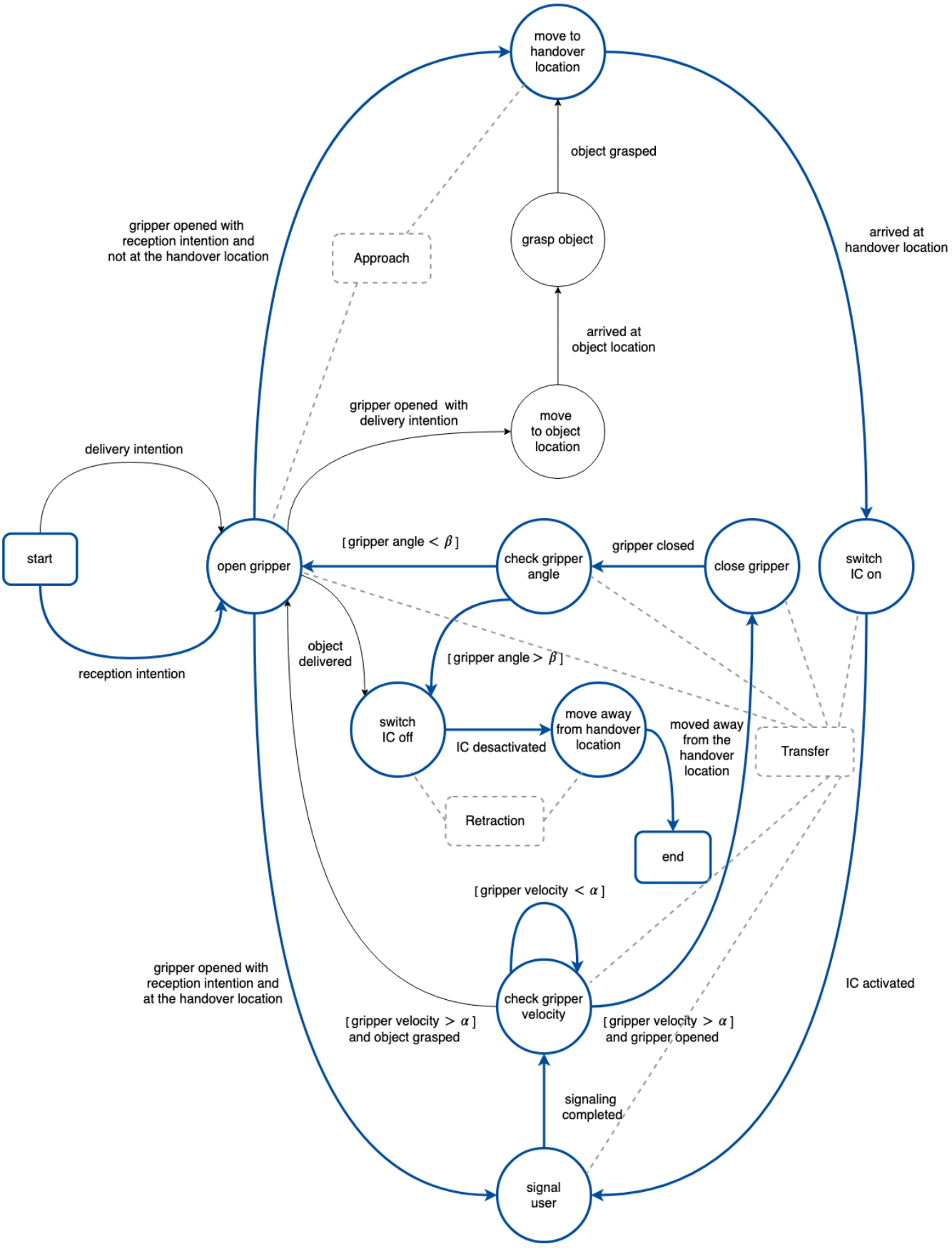


Figure 3.3: State-machine based algorithm sequence, in blue, regarding a human-to-robot handover. The three phases on the handover are also presented.

3.3 Impedance Control

A fluent and dynamic human-robot handover may be achieved due to the robot's adaptability to the task conditions, environmental constraints and perturbations instead of simply controlling its position, in which the robot is seen as an isolated system. Therefore, most controllers for interacting systems regulate dynamics behavior of the robot at the interaction port/end-effector. As a result, active compliant control was selected as the controlling approach for the transfer phase of the formulated FSM-based handover algorithm, more specifically impedance control (section 2.2.3).

Impedance control is not as commonly implemented solution in aerial manipulation robotics applications as in robotic manipulations tasks with a fixed base. Nevertheless, this section aims to design an impedance controller that generates a dynamical relationship between a free-flyer robot manipulator and external forces acting on it during the transfer phase of the handover.

Impedance control aims at controlling the dynamic behavior of the robot manipulator when contacting the environment, not by controlling the exact interaction forces but the properties of the contact such as the stiffness and damping of the interaction. To achieve this dynamic relationship, the impedance control system creates a virtual new impedance for the manipulator, which is being able to interact with the environment as if new mechanical elements had been included in the real manipulator. Additionally, it does not requires force/torque sensing data when implemented on a implicit manner. This advantage allows formulated algorithm to be implemented in a wider robot range.

As stated in previous sections, it is assumed that the robot is an unmanned aerial vehicle equipped with a robotic arm and a gripper as end-effector. Although, UAV systems with a manipulator provide aerial robots with the capability of performing dexterous manipulation tasks, the presence of the mounted robot arm arise challenges since its dynamics depends on the actual configuration of the whole state of the system. In this manner, the kinematic and dynamic model, using an Euler-Lagrange formulation, of an 6-DOF holonomic free-flying robot equipped with n-DOF arm are described. This formulation is adapted to this dissertation goals from the research developed by Lippiello and Ruggiero in [71]. Additionally, an impedance control law is proposed given a desired dynamic impedance relationship between the external forces acting on the robot's end-effector and the closed-loop system behavior.

3.3.1 Kinematic Model

The derivation present in the current section considers that the system is composed by a microgravity free-flying robot equipped with a robotic manipulator. It is relevant to refer that the derivation is independent of the robot's base configuration as the actuation transformation is further reference on Chapter 4. Furthermore, it also does not depend on the arm placement choice as, depending on the particular vehicle configuration, the best mounting of the manipulator that allows a stable behavior will be considered. Additionally, the manipulator consists of n rigid links connected by joints q_l, with $l = 1, 2, 3, ..., n$. At the

end of the last link, the one not connected to the vehicle, is a gripper, e.g., end-effector.

In this manner, a schematic representation of free-flying robot equipped with actuators and a manipulator is presented in Figure 3.4 as well as the reference frames considered.

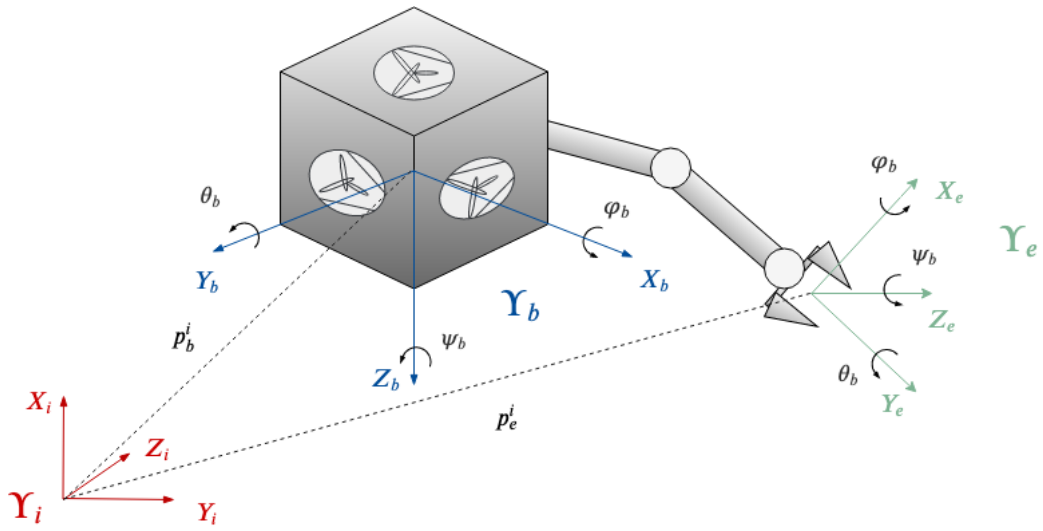


Figure 3.4: Schematic representation of a generalized free-flyer robot equipped with a manipulator and respective reference frames.

The inertial frame is denoted by Υ_i, the body-fixed reference frame placed at the spacecraft center of mass by Υ_b and the end-effector coordinates attached to the interaction point of the manipulator by Υ_e. Furthermore, the absolute position of Υ_b with respect to Υ_i is described as $p_b^i = [x_b \ y_b \ z_b]^T$ and the system attitude is expressed in roll-pitch-yaw Euler angles being denoted by $\phi_b^i = [\varphi_b \ \theta_b \ \psi_b]^T$. Additionally, the absolute transitional velocity of Υ_b is represented by $\dot{p}_b^i$ and $\dot{p}_b^b$, with respect to Υ_i and to Υ_b, respectively. Regarding the absolute rotational velocity, ω_b^i refers to the absolute rotational velocity of the vehicle and ω_b^b denotes the absolute rotational velocity with respect to Υ_b. If the rotation matrix of frame Υ_b with respect to frame Υ_i is defined by R_b^i, the spacecraft linear velocity representation in Υ_b coordinates is transformed to its representation in Υ_i coordinates from:

$$\dot{p}_b^i = R_b^i \, \dot{p}_b^b \tag{3.1}$$

with

$$R_b^i = \begin{bmatrix} cos(\theta_b)\ cos(\psi_b) & cos(\varphi_b)\ sin(\theta_b)\ cos(\psi_b) - cos(\varphi_b)\ sin(\psi_b) & cos(\varphi_b)\ sin(\theta_b)\ cos(\psi_b) + sin(\varphi_b)\ sin(\psi_b) \\ cos(\theta_b)\ sin(\psi_b) & cos(\varphi_b)\ sin(\theta_b)\ sin(\psi_b) + cos(\varphi_b)\ cos(\psi_b) & cos(\varphi_b)\ sin(\theta_b)\ sin(\psi_b) - sin(\varphi_b)\ cos(\psi_b) \\ -sin(\theta_b) & sin(\varphi_b)\ cos(\theta_b) & cos(\varphi_b)\ cos(\theta_b) \end{bmatrix} \tag{3.2}$$

Moreover, if the transformation matrix between the time derivative of ϕ_b^i and ω_b^i is defined by N_b^i, the transformation of the UAV absolute rotational velocity is obtained as:

$$\omega_b^i = N_b^i \, \dot{\phi}_b^i \tag{3.3}$$

with

$$N_b^i = \begin{bmatrix} cos(\theta_b) \, cos(\psi_b) & sin(\psi_b) & 0 \\ -cos(\theta_b) \, sin(\psi_b) & cos(\psi_b) & 0 \\ sin(\theta_b) & 0 & 1 \end{bmatrix} \tag{3.4}$$

From (3.1) and (3.3) holds:

$$\omega_b^b = (R_b^i)^T \, \omega_b^i = (R_b^i)^T \, N_b^i \, \dot{\phi}_b^i = Q_b^i \, \dot{\phi}_b^i \tag{3.5}$$

with $Q_b^i = (R_b^i)^T \; N_b^i$ being the mapping of the time derivative of ϕ_b^i into the body absolute rotational velocity with respect to Υ_b. The transformation equations (3.1)-(3.5) are valid as long as the matrices R_b^i, N_b^i, and Q_b^i are non-singular.

Furthermore, direct kinematics of the spacecraft are defined by the following transformation matrix:

$$K_b^i(p_b^i, \phi_b^i) = \begin{bmatrix} R_b^i & p_b^i \\ 0_{1\times3} & 1 \end{bmatrix} \tag{3.6}$$

where, $0_{1\times3}$ is a (1×3) vector composed only by zeros.

Direct kinematics of the manipulator with respect to Υ_b are expressed as:

$$K_e^b(q) = \begin{bmatrix} R_e^b(q) & p_e^b(q) \\ 0_{1\times3} & 1 \end{bmatrix} \tag{3.7}$$

with q describing the $(n \times 1)$ vector of the robot manipulator joints variables, R_e^b the rotation matrix between Υ_e and Υ_b and $p_e^b = [x_e^b \; y_e^b \; z_e^b]^T$ the position of the end-effector with respect to Υ_b.

Combining (3.6) and (3.7):

$$K_e^i(\xi) = K_b^i \, K_e^b \tag{3.8}$$

where $\xi = [{p_b^i}^T \; {\phi_b^i}^T \; q_1 \; ... \; q_n]^T$ is the $((6+n) \times 1)$ generalized vector of the system joints variables.

Moreover, the end-effector absolute position with respect to the inertial frame is defined as $x = [p_e^i \; \phi_e^i]$ where $p_e^i = [x_e \; y_e \; z_e]^T$ and the manipulator's attitude is also expressed in roll-pitch-yaw Euler angles being denoted by $\phi_e^i = [\varphi_e \; \theta_e \; \psi_e]^T$ with respect to Υ_i. The vector of absolute generalized velocity of the manipulator's end-effector can consequently be expressed as $\dot{x} = [\dot{p}_e^i \; \dot{\phi}_e^i]$. Furthermore, the same consideration as Equation (3.3) holds for the end-effector with:

$$\omega_e^i = N_e^i \, \dot{\phi}_e^i \tag{3.9}$$

The transformation between $\dot{x}$ and the time derivative of the system generalized joints variables can be written as:

$$\dot{x} = J \, \dot{\xi} \tag{3.10}$$

This equation describes the relationship between the generalized joint vector velocities, $\dot{\xi}$, and the end-effector operational space velocities, $\dot{x}$, where J is the so called Jacobian $(6 \times (6+n))$ matrix of the system. This Jacobian can be presented as a composition of the spacecraft contribution, J_s, and the manipulator component, J_m, as presented in Equation (3.11).

$$J = \Big[J_s \;\Big|\; J_m \Big] = \left[\begin{array}{cc|c} I_{3\times3} & -R_b^i \, [p_e^b]_\times \, N_b^i & R_b^i \, J_t^{l_i} \\ 0_{3\times3} & N_b^i \, N_e^{i-1} & R_b^i \, J_r^{l_i} \, N_e^{i-1} \end{array} \right] \tag{3.11}$$

where, $I_{3\times3}$ is an (3×3) identity matrix, $0_{3\times3}$ is a (3×3) matrix composed only by zeros, $J_t^{l_i}$ and $J_t^{l_i}$ denotes the Jacobians that describe transitional and rotational contributions of the joints velocities up to the current link i. Additionally, $[\;\;]_\times$ symbolizes the skew-symmetric matrix. This matrix is a square matrix whose transpose equals its negative and, in the present case, is defined by:

$$[p_e^b]_\times = \left[\begin{array}{ccc} 0 & -z_e^b & y_e^b \\ z_e^b & 0 & -x_e^b \\ -y_e^b & x_e^b & 0 \end{array} \right] \tag{3.12}$$

Moreover, time deriving equation (3.10) yields:

$$\ddot{x} = J \, \ddot{\xi} + \dot{J} \, \dot{\xi} \tag{3.13}$$

Furthermore, the linear and angular velocity of the center of mass of the link i with respect to Υ_b, $\dot{p}^b_{bl_i}$ and $\omega^b_{bl_i}$, respectively, can be expressed by:

$$\dot{p}^b_{bl_i} = J^{l_i}_t \ \dot{q} \tag{3.14a}$$

$$\omega^b_{bl_i} = J^{l_i}_r \ \dot{q} \tag{3.14b}$$

Lastly, the position of the center of mass of the link i with respect to Υ_i is described as:

$$p^i_{l_i} = p^i_b + R^i_b \ p^b_{bl_i} \tag{3.15}$$

3.3.2 Dynamic Model

The controller designed for the proposed system present in Figure 3.4 should consider the dynamics of the free-flyer as the manipulator aerial base and of the manipulator, simultaneously. Accordingly, using a *Euler-Lagrange* formulation and kinematics relations derived in previous section, the dynamics of the system in the generalized joint space, ξ, is obtained.

According to the *Euler-Lagrange* formulation, a system mechanical structure can be described by the sum of its total kinetic energy, T, and potential energy, U. In a microgravity environment, the potential energy is considered to be identically zero as $g \approx 0$. Therefore, the total energy, L, of the proposed system is:

$$L = T \tag{3.16}$$

The Lagrange equations can be defined as:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\xi}_i} - \frac{\partial L}{\partial \xi_i} = u_i \tag{3.17}$$

with i describing the i-th generalized coordinate of ξ and assuming values of $i = 1, ..., ((6+n))$. The i-th generalized force is represented as u_i.

The total kinetic energy of the system being studied is composed by the energy contributions concerning the motion of the spacecraft, T_b and the energy associated with motion of each link of the manipulator,

T_{l_i}, as express in Equation (3.18).

$$T = T_b + \sum_{i=1}^{n} T_{l_i} \tag{3.18}$$

The aerial vehicle kinetic energy contribution is expressed by:

$$T_b = \frac{1}{2}\, m_b\, \dot{p}_b^{i\,T}\, \dot{p}_b^i + \frac{1}{2}\; \omega_b^{i\,T}\; R_b^{i\,T}\; H_b\; R_b^i\; \omega_b^i \tag{3.19}$$

with H_b and m_b representing the inertia and mass matrix of the free-flyer, respectively.

Knowing (3.5), the previous equation can be re-written as:

$$T_b = \frac{1}{2}\, m_b\, \dot{p}_b^{i\,T}\, \dot{p}_b^i + \frac{1}{2}\; \dot{\phi}_b^{i\,T}\; Q_b^{i\,T}\; H_b\; Q_b^i\; \dot{\phi}_b^i \tag{3.20}$$

Having into account that the $R_{l_i}^b$ is the rotation matrix between the center-of-mass of the i-th manipulator link and Υ_b, m_{l_i} is the mass and H_{l_i} is the inertia matrix of the i-th link, the kinematic energy of the manipulator is then described as:

$$T_{l_i} = \frac{1}{2}\, m_b\, \dot{p}_{l_i}^{i\;T}\, \dot{p}_{l_i}^i + \frac{1}{2}\; \omega_{l_i}^{i\;T}\; R_b^i\; R_{l_i}^b\; H_{l_i}\; R_b^{l_i}\; R_b^{i\,T}\; \omega_{l_i}^i \tag{3.21}$$

Knowing (3.5), (3.14a), (3.14b), (3.20) and (3.21) the total kinetic energy is:

$$L = T = \frac{1}{2}\, \dot{\xi}^T\; B\; \dot{\xi} \tag{3.22}$$

with B being an $((6+n)\times(6+n))$ symmetric and positive inertia matrix, whose block elements are:

- $B_{11} = I_{3\times 3}\; (m_b + \sum_{i=1}^{n} m_{l_i})$
- $B_{12} = {B_{21}}^T = -\sum_{i=1}^{n} (m_{l_i}\; [R_b^i\; p_{bl_i}^b]_\times\; N_b^i)$
- $B_{13} = {B_{31}}^T = \sum_{i=1}^{n} (m_{l_i}\; R_b^i\; J_t^{l_i})$
- $B_{22} = {Q_b^i}^T\; H_b\; Q + \sum_{i=1}^{n} (m_{l_i}\; {N_b}^T\; {[R_b^i\; p_{bl_i}^b]_\times}^T\; [R_b^i\; p_{bl_i}^b]_\times\; N_b + {Q_b^i}^T\; R_{li}^b\; H_{li}\; R_b^{li}\; Q_b^i$
- $B_{23} = {B_{32}}^T = \sum_{i=1}^{n} ({Q_b^i}^T\; R_{li}^b\; H_{li}\; R_b^{li}\; J_r^{l_i} - m_{l_i}\; N_b\; {[R_b^i\; p_{bl_i}^b]_\times}^T\; R_b^i\; J_t^{l_i})$
- $B_{33} = \sum_{i=1}^{n} (m_{l_i}\; J_t^{l_i}\; J_r^{l_i} + {J_r^{l_i}}^T\; R_{li}^b\; H_{li}\; R_b^{li}\; J_r^{l_i})$

Lastly, computing the Lagrange equation (3.17), the dynamics of the system in the generalized joint space are given by:

$$B(\xi)\,\ddot{\xi} + C(\xi,\dot{\xi})\,\dot{\xi} = u + u_{ext} \tag{3.23}$$

where u describes the generalized input forces vector $((6+n)\times 1)$ and u_{ext} represents the external generalized forces vector at a joint level, $((6+n)\times 1)$. Furthermore, C is an $((6+n)\times(6+n))$ matrix that encompasses the Coriolis and centrifugal terms and whose generic element is c_{ij}:

$$c_{ij} = \sum_{k=1}^{(6+n)} \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial \xi_k} + \frac{\partial b_{ik}}{\partial \xi_j} + \frac{\partial b_{jk}}{\partial \xi_i}\right)\dot{\xi}_k \tag{3.24}$$

with b_{ij} representing the generic element of $B(\xi)$, $i,j = 1,...,(6+n)$.

3.3.3 Control Law

Given that the kinematics and dynamics model of the system were already described, this section uses those formulations to design a control law that agrees with an impedance dynamical model of the end-effector motion and external forces.

Let $\ddot{x}_d$, $\dot{x}_d$ and x_d be the end-effector desired *rest* acceleration, velocity and position, respectively, and the actual position error as $\tilde{x} = x_d - x$. Moreover, during the transfer phase on the the handover tasks formulated it is assumed that $\ddot{x}_d = 0$ and $\dot{x}_d = 0$. Given these considerations, a suitable law control can be designed:

$$u = J^T\,(-k_B\,\dot{x} + k_D\,\tilde{x}) \tag{3.25}$$

With k_D and k_B representing the $((6+n)\times(6+n))$ symmetric and positive definite matrices of the chosen stiffness and damping, respectively. It is important to refer that this matrices can be tuned to the desired system's behavior.

Finally, substituting (3.25) into (3.23) and considering (3.10) and (3.13), the joint space dynamics can be expressed in terms of the manipulator's end-effector configuration, x, in the inertial Cartesian coordinates representing an impedance dynamic model as presented in Equation (3.26). This derivation can further be analysed in the Appendix A.

$$B_x\,\ddot{x} + (C_x + k_B)\,\dot{x} - k_D\,\tilde{x} = f_{ext} \tag{3.26}$$

with f_{ext} representing the vector $((6+n)\times 1)$ of the external generalized forces at the Cartesian coordinate level and B_x and C_x describing the inertia and Coriolis matrices with respect to the x variable:

$$B_x = J(\xi)^{-T}\ B(\xi)\ J(\xi)^{-1} \tag{3.27a}$$

$$C_x = J(\xi)^{-T}\ (C(\xi,\dot{\xi}) - B(\xi)\ J(\xi)^{-1}\ \dot{J}(\xi))\ J(\xi)^{-1} \tag{3.27b}$$

Summarizing, a control law was designed for the transfer phase of the formulated FSM-based handover algorithm given the kinematics and dynamics of a microgravity free-flying robot equipped with a manipulator, obtaining an impedance dynamic model that describes the mapping between the end-effector motion and external generalized forces.

Chapter 4

Implementation and Results

The previous chapter described the handover algorithm formulation while the current chapter concerns about its implementation and validation on a simulator. Furthermore, an implementation of a human interaction interface with the system is also described and validated.

4.1 Formulated Algorithm Implementation

4.1.1 Astrobee Robot Software

Given the current collaboration between NASA and ISR and the availability of an open-source Astrobee software platform designed to conduct research, this free-flyer robot simulator was used as an implementation platform to showcase and verify the proposed handover algorithm.

The Astrobee open-source flight software drives the robot's capabilities, running on three interconnected processors and interfacing with seven micro-controllers, six cameras, other peripherals such as flashlights, touch screen and laser pointer. Lastly, also encompasses two propulsion modules that allows an holonomic motion. These modules are composed by an air container which is pressurized by an impeller. Each propulsion module has six nozzles with a fixed direction and symmetric aligned. With the thrust out of the total twelve nozzles, force (F) and torque (M) can be simultaneously applied in X,Y and Z direction. This is possible due to two flappers implemented in each nozzle that increase or decrease the air flow out of it. In this manner, the robot can fully actuated in both force and torque on a microgravity enviroment.

The Robot Operating System (ROS) handles internal software communication while the Data Distribution Service (DDS) communicates externally (Version 1). The Astrobee Robot Software also uses a Gazebo 7.0 based simulator.

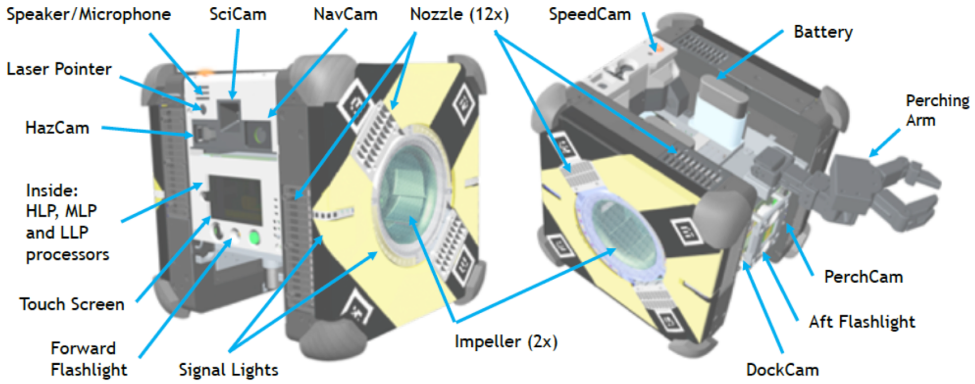


Figure 4.1: Astrobee's main components [21].

The software architecture is composed by a set of modular, distributed, and loosely-coupled components. This is implemented in practice by 46 ROS nodelets. Dependencies between nodelets are kept to a minimum by strongly-defining their responsibilities and interface messages, services and actions. Moreover, the nodelets are grouped into modules running across three main computers (Figure 4.2):

- The Low-Level Processor (LLP) - runs the Guidance, Navigation and Control (GNC) module. The GNC module consists of an Extended Kalman Filter (EKF), a PID controller (CTL) and a Force Allocation Module (FAM). The EFK estimates the Astrobee's base position and attitude out of the cameras data and the Inertial Measurements Unit (IMU). The controller gives the system input, force (F) and torque (M) out of the estimated position and attitude and desired state. In the FAM, the forces and torques are converted to air flow out of the nozzle and, consequently, converted to nozzle's flappers opening angle. Moreover, this processor controls the peripherals such as laser and flashlights. The LLP runs in Ubuntu 16.04.
- Mid-Level Processor (MLP) - runs most of the software developed by the Astrobee team, including the compute-intensive machine vision algorithms concerning the localization module and encompasses the mobility module. The MLP also runs in Ubuntu 16.04.
- The High-Level Processor (HLP) - primarily used for guest scientist software, but also hosts less critical functions such as streaming HD video compression and touch screen interaction. The HLP runs in Android (Nougat 7.1).

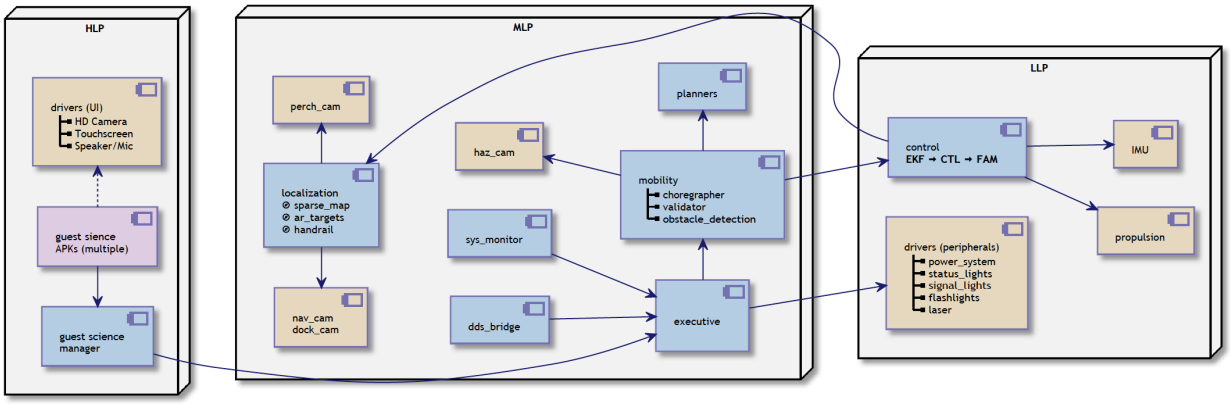


Figure 4.2: Astrobee's software three processors and its main components. The blue colored components on this scheme represent logical groupings composed of multiple ROS nodelets. The arrows indicate dependencies, not flow of information. Beige color represent drivers and purple external software [21].

The work developed concerning the handover algorithm implementation was focused on the Middle Level and Low Level Processor. These processor's components rely solely on the ROS communication framework to exchange information. ROS messages included in ROS topics are used for data distribution, ROS services for simple instantaneous requests and ROS actions for complex longer operations.

4.1.2 Handover ROS Node Arquitecture

Furthermore, the implementation of the handover algorithm requires the integration of multiple functionalities in each phase that together allow a robot to perform the task.

In this manner, a ROS node (*handover_node*) was developed. This node connects to the several components of the LLP and MLP of the Astrobee Software and Simulator and includes multiple modules that are used on the algorithm implementation. A module is considered to be a group of ROS topics, services and/or actions that aim to realize a common goal. Moreover, it is important to refer that several algorithm states on different tasks share the same module implementation however relevant differences will further be highlighted. Figure 4.3 displays a diagram of the architecture of the ROS node and modules developed.

The proposed algorithm was implement using Python 3.0, Ubuntu 16.04 LTS and ROS Kinetic.

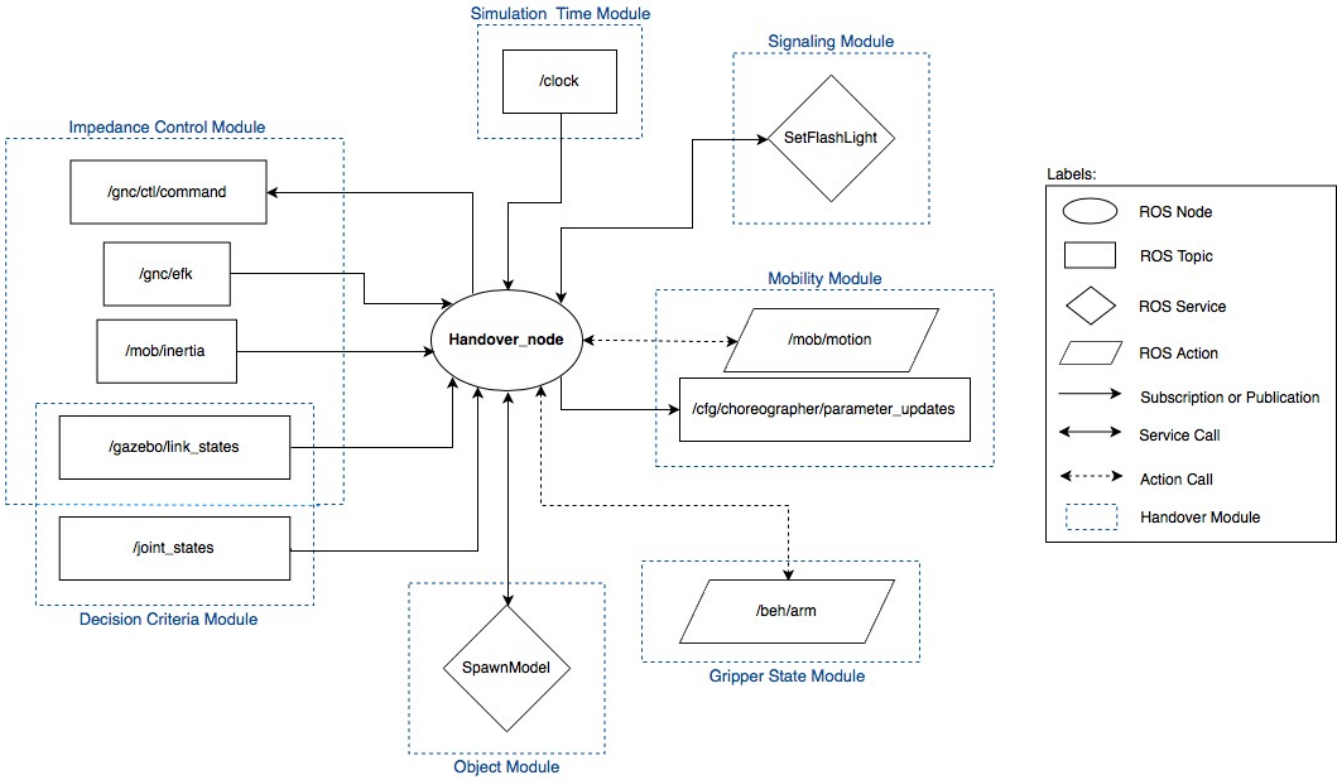


Figure 4.3: Block diagram of the architecture of the ROS node developed to implement the handover algorithm formulated on the Astrobee robot.

Simulation Time Module

Initially, the developed node access the Gazebo simulation time through the subscription of the */clock* topic.

Object Module

The implemented node spawns an Unified Robot Description Format (URDF) object file into a simulation pre-defined position, i.e., closer to the robot or closer to the human on the robot-to-human handover and on the human-to-robot handover, respectively. The object simulation on Gazebo is done through a service call request to the *gazebo* ROS node. This object was implemented as a rigid box with side lengths of 0.021 mm, 0.195 mm and 0.017 mm and mass of 0.1 kg.

Gripper State Module

The object is later grasped by the robot or by the user, depending on the goal of interaction. To execute the grasping motion or to open the gripper to receive/give the object, a *SimpleActionClient* is created given the available arm action library (*/beh/arm*) in which a gripper goal state (close or open) is sent to the action server.

Mobility Module

Regarding the movement to/from the handover location, another *SimpleActionClient* is created concerning the mobility action library */mob/motion*. This action server connects with the Astrobee's baseline mobility capacity to maneuver the robot, using a PID controller, given a goal position and orientation. Additionally, a *face foward* boolean was dynamical reconfigured to false resorting to the *cfg* module. This enabled the robot's movement in any direction without the need of any particular orientation during the trajectory.

Signaling Module

Concerning the states encompassed on the transfer phase of the handover tasks, the implementation the signaling is done through the call of the *SetFlashLight* service of the *gazebo* node and by setting the brightness to *200* followed by setting it to *0*.

Decision Criteria Module

On the tasks formulated, several decision criteria transitions arose. On the one hand, in the robot-to-human and in the robot-to-human handover, the end-effector velocity concerning the decision of opening/closing the gripper is acquire through the subscription of the */gazebo/link_states* topic. As different handover directions and robot's behaviors generate distinct motions on the robot, the threshold was tuned to $\alpha = 0.059m/s$ and $\alpha = 0.08m/s$ for the robot-to-human handover with a rigid and a compliant behavior, respectively. While for the human-to-robot handover, the threshold was defined as $\alpha = 0.021m/s$ and $\alpha = 0.037m/s$ for the rigid and compliant behavior, respectively.

On the other hand, in the human-to-robot handover task an additional decision criteria was designed concerning the failure reception module and thus the gripper angle value required is obtained from the */joint_states* topic. Furthermore, the Astrobee's gripper angle threshold was defined as $\beta = 0.71rad$.

Impedance Control Module

As already stated, a PID controller was used as the baseline mobility platform for the mobility module described previously. However, during the transfer phase, the pre-defined CTL controller must be replaced by the impedance controller formulated.

The EKF estimates the Astrobee's base position and attitude out of the cameras data and the IMU. Therefore, the output data of the EKF must be the input data of IC implemented. In this manner, the node developed subscribes to the */gnc/efk* topic and the IC module is called every time the EKF state

is published. Furthermore, the IC module also makes use of the subscription of the */gazebo/link_states* topic to obtain the end-effector's state values required. Furthermore, the controller publishes the Forces (F) and Torques (M) calculated on the */gnc/ctl/command* topic with the same frequency as the FAM subscribes to that the same topic, 67.5 Hz.

Moreover, other measurements required such as mass and inertia values are obtain by subscribing to the available */mob/inertia* topic.

The impedance control formulation for a microgravity free-flyer robot equipped with a manipulator was described on the previous Chapter (section 3.3). However, the Astrobee's simulator presented considerably small tolerances for the joints state goals and thus not allowing the movement of the arm links for small controlled angles. Therefore, the joint variables are assumed to be fixed for this implementation. In this manner:

$$\dot{\xi} = [\dot{p}_b^{i\,T} \ \dot{\phi}_b^{i\,T} \ 0 \ ... \ 0]^T \tag{4.1}$$

Consequently, the input vector of the manipulator actuation torques is irrelevant and the input vector of forces, f, is composed by the free-flyer base actuator force vector, F (3×1), and torque vector, M (3×1):

$$f = \begin{bmatrix} F^T \\ M^T \end{bmatrix} \tag{4.2}$$

Furthermore, considering the configuration of the Astrobee, the vector u has the following expression:

$$u = \begin{bmatrix} R_b^i \\ Q_b^{i\,T} \end{bmatrix} f \tag{4.3}$$

With this assumption, the end-effector desired position and orientation is reached through the movement of the Astrobee's base. This desired position and orientation was defined as the handover position and orientation.
Furthermore, given the desired dynamic impedance model of Equation (3.26) and the fixed manipulator assumption, the end-effector intuitively behaves as if it were attached to the environment, e.g., to the desired position, by linear spring-damper systems in the 3 linear degrees of freedom and also by torsional spring-dampers to control rotational orientation in the other 3 DOF. This intuitive behavior is represented in the simplified 1 DOF scheme of Figure 4.4.

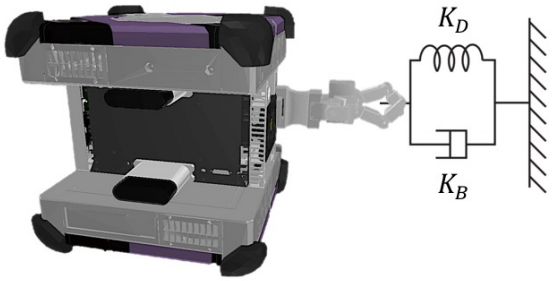


Figure 4.4: Impedance control concept on the Astrobee where an imaginary spring-damper is attached to the end-effector of a fixed manipulator. This figure is a simplification of the concept as there should be independent springs and dampers in each degree of freedom - including rotational degrees of freedom.

Moreover, two types of impedance behaviors were implemented: rigid and compliant. This behaviors can be defined by tuning the values of the matrices in Equation (3.25) accordingly to each behavior and to avoid oscillations. A high value of the stiffness matrix will generate a rigid behavior while a low value is associated with a compliant behavior.

The tuned values for the matrices are presented in Table 4.1 with $m_{3\times3}$ being a (3×3) matrix representing the mass of the system and its inertial vales being presented by $H_{3\times3}$.

		Rigid behavior	Compliant behavior
Linear Component	k_D	$7m_{3\times3}$	$0.7m_{3\times3}$
	k_B	$15I_{3\times3}$	$5I_{3\times3}$
Rotational component	k_D	$70H_{3\times3}$	$7H_{3\times3}$
	k_B	$5I_{3\times3}$	$I_{3\times3}$

Table 4.1: Matrices values in Equation (3.25) tuned accordingly to the desired Astrobee behavior: rigid or compliant.

Lastly, the next section presents the validation results concerning this control implementation.

4.1.3 Impedance Control Validation

In order to obtain the formulated impedance controller validation during interaction, several generalized external forces, f_{ext}, were applied to the robot's end-effector, for both behavior study cases: rigid and compliant.

Furthermore, with the aim of validating the dynamic impedance model proposed in Section 3.3 (Equation (3.26)), the expected values of the end-effector position and orientation error, $\tilde{x}'$, were calculated from

Equation (3.26) given the end-effector simulated acceleration, $\ddot{x}$, the end-effector simulated velocity, $\dot{x}$, the B_x, C_x, K_D and K_B matrices and the f_{ext}, as following:

$$\tilde{x}' = [B_x\ \ddot{x} + (C_x + k_B)\ \dot{x}\ - f_{ext}]\ /\ k_D \tag{4.4}$$

The calculated end-effector position and orientation error, $\tilde{x}'$, as well as the actual simulated end-effector position and orientation error, $\tilde{x}$, are presented for each axis in Figures (4.5)-(4.10) for the rigid behavior case and in Figures (4.11)-(4.16) for the compliant behavior case. Additionally, the external generalized forces generated for each simulation are represented.

For all tests performed, the end-effector initial state and the end-effector desired state were defined as $x_{initial} = [0, -0.07, 0, 0, 0, 0]$ and $x_d = x_{initial}$, respectively.

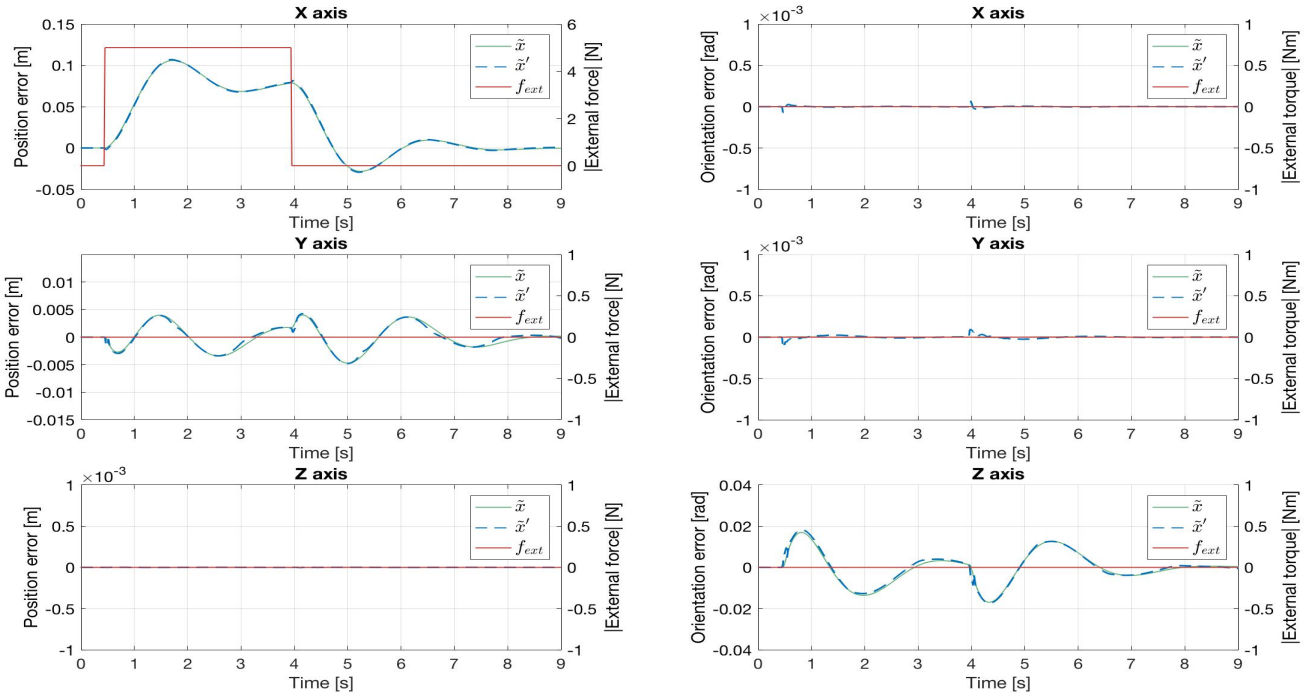


Figure 4.5: Actual and calculated (from impedance model) end-effector position and orientation error, with a rigid behavior. A $5N$ force was applied on the X axis.

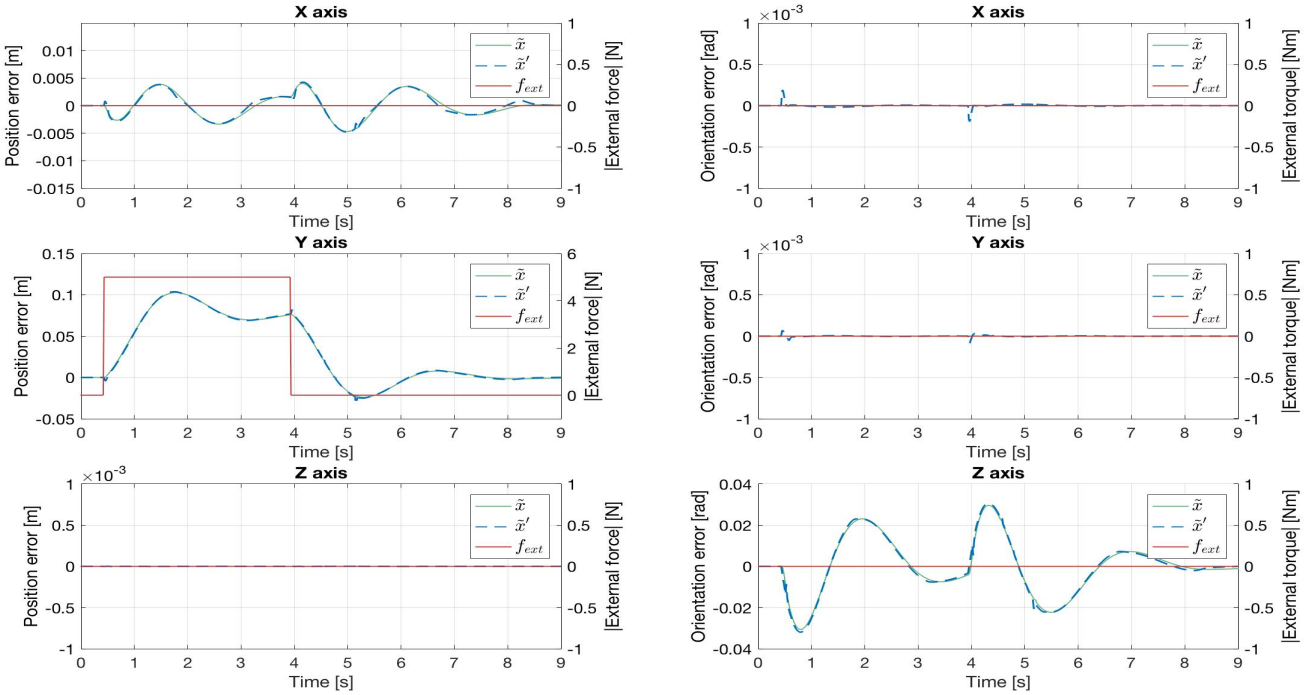


Figure 4.6: Actual and calculated (from impedance model) end-effector position and orientation error, with a rigid behavior. A $5N$ force was applied on the Y axis.

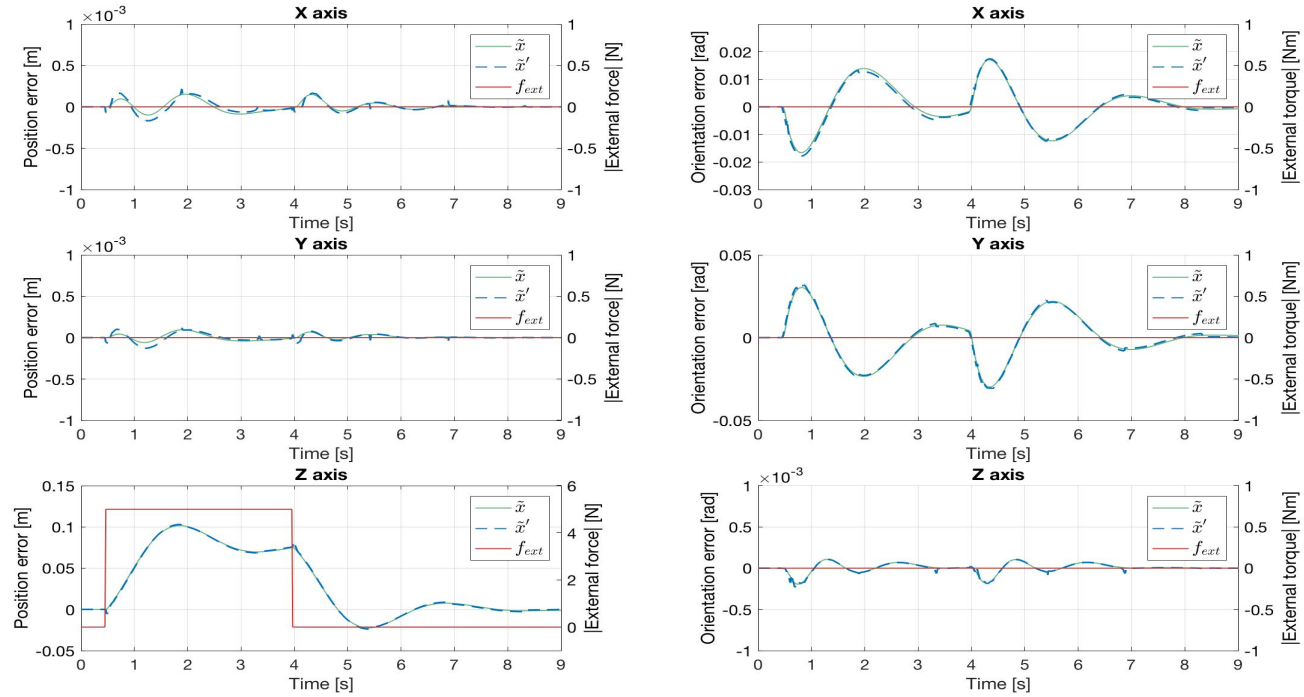


Figure 4.7: Actual and calculated (from impedance model) end-effector position and orientation error, with a rigid behavior. A $5N$ force was applied on the Z axis.

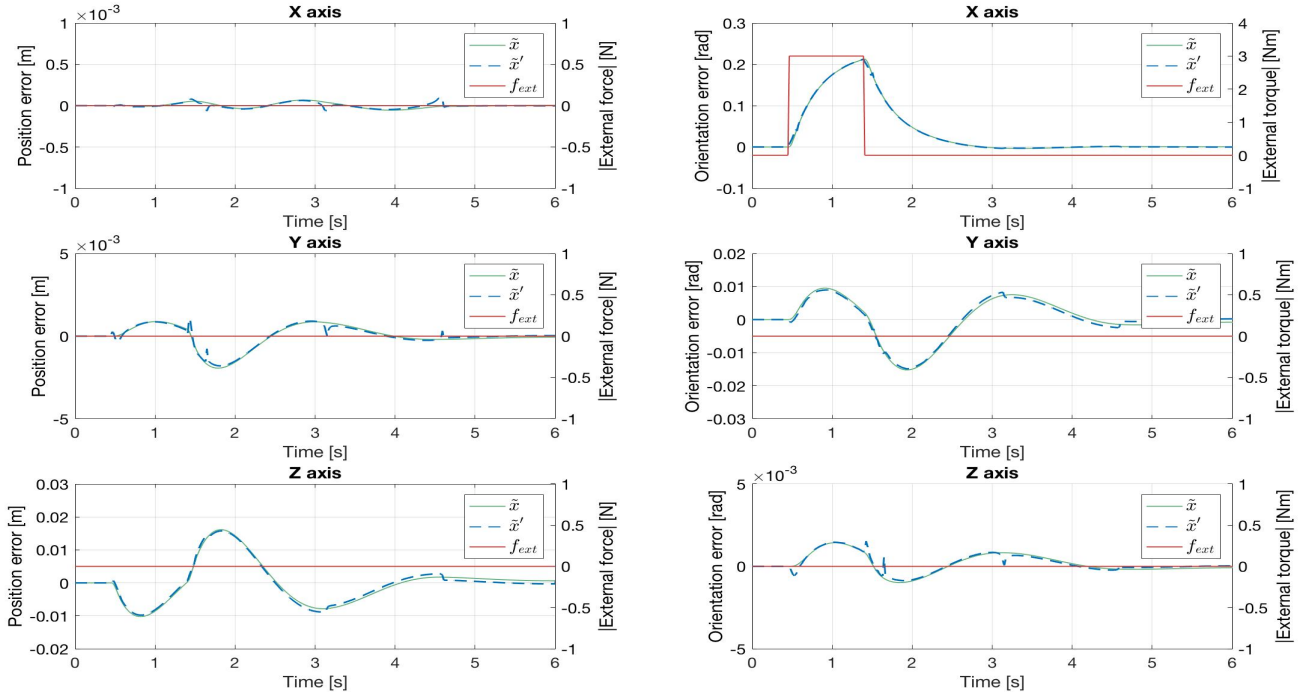


Figure 4.8: Actual and calculated (from impedance model) end-effector position and orientation error, with a rigid behavior. A $3Nm$ torque was applied on the X axis.

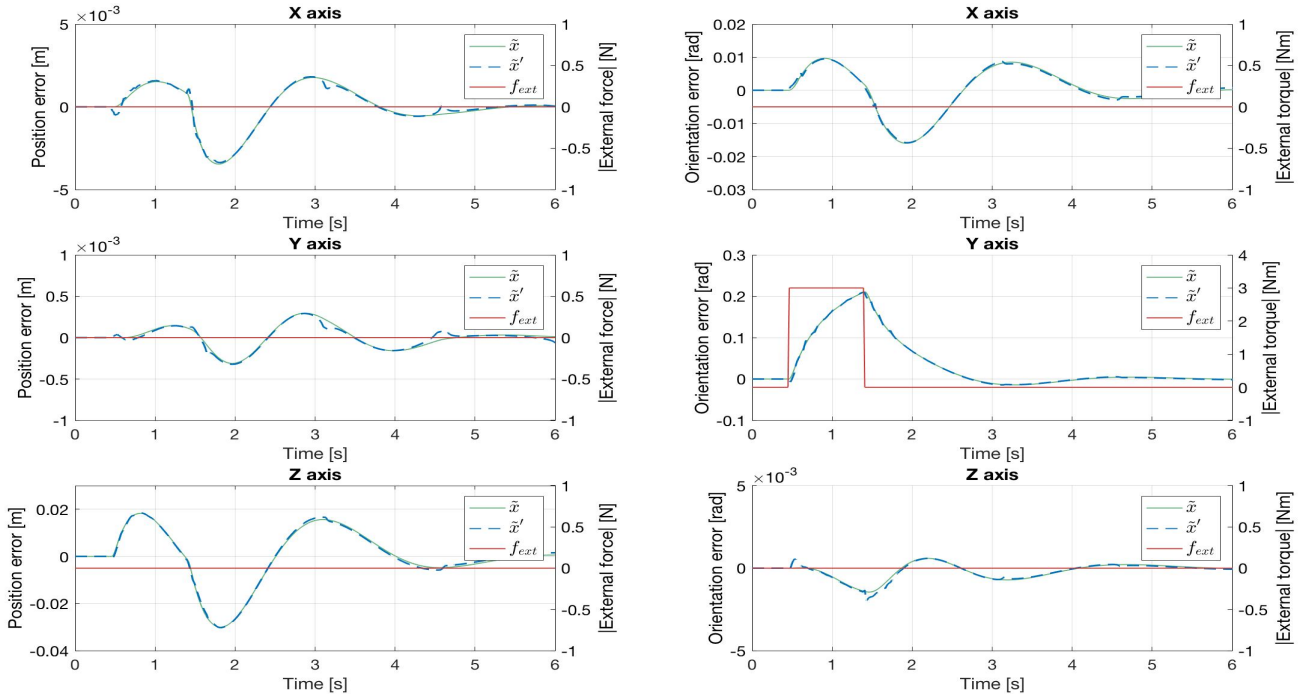


Figure 4.9: Actual and calculated (from impedance model) end-effector position and orientation error, with a rigid behavior. A $3Nm$ torque was applied on the Y axis.

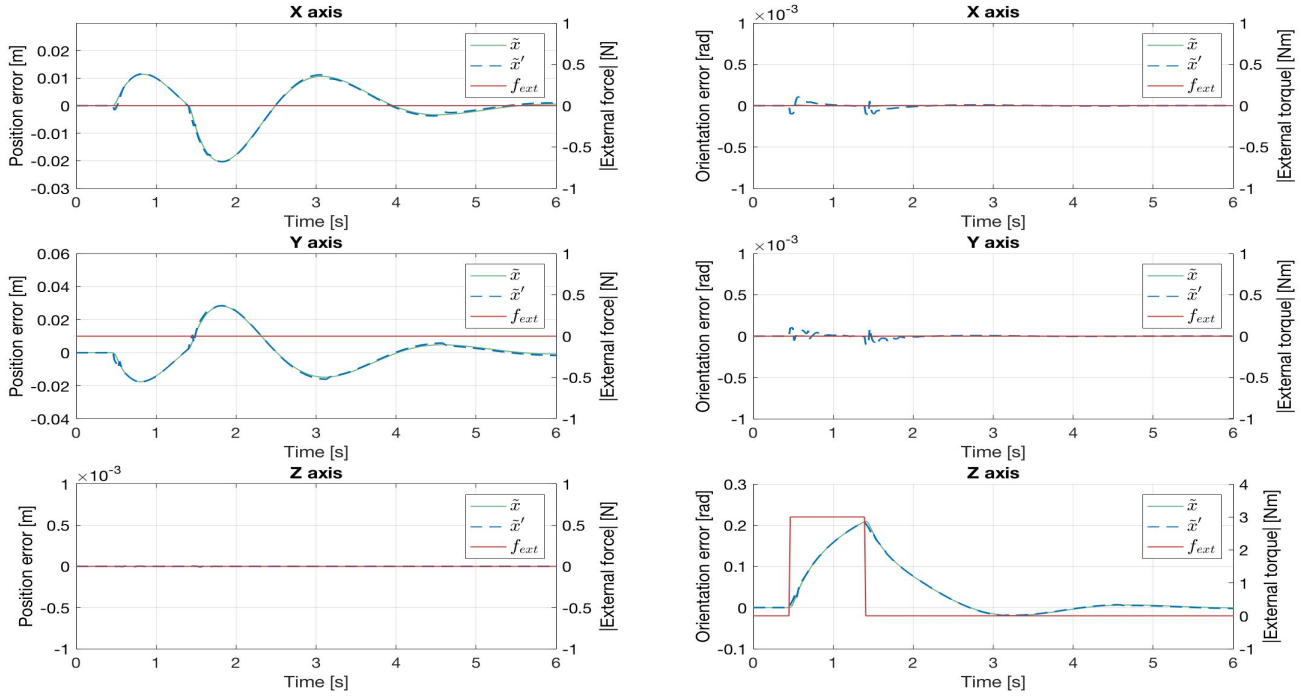


Figure 4.10: Actual and calculated (from impedance model) end-effector position and orientation error, with a rigid behavior. A $3Nm$ torque was applied on the Z axis.

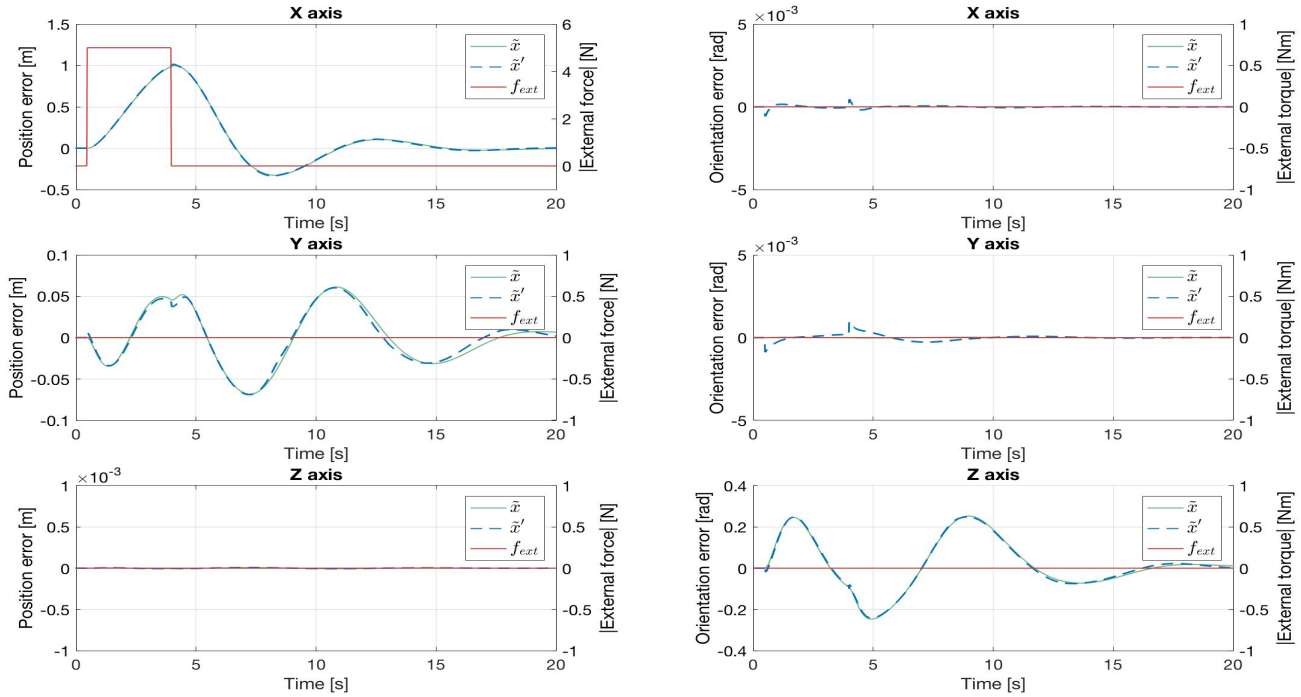


Figure 4.11: Actual and calculated (from impedance model) end-effector position and orientation error, with a compliant behavior. A $5N$ force was applied on the X axis.

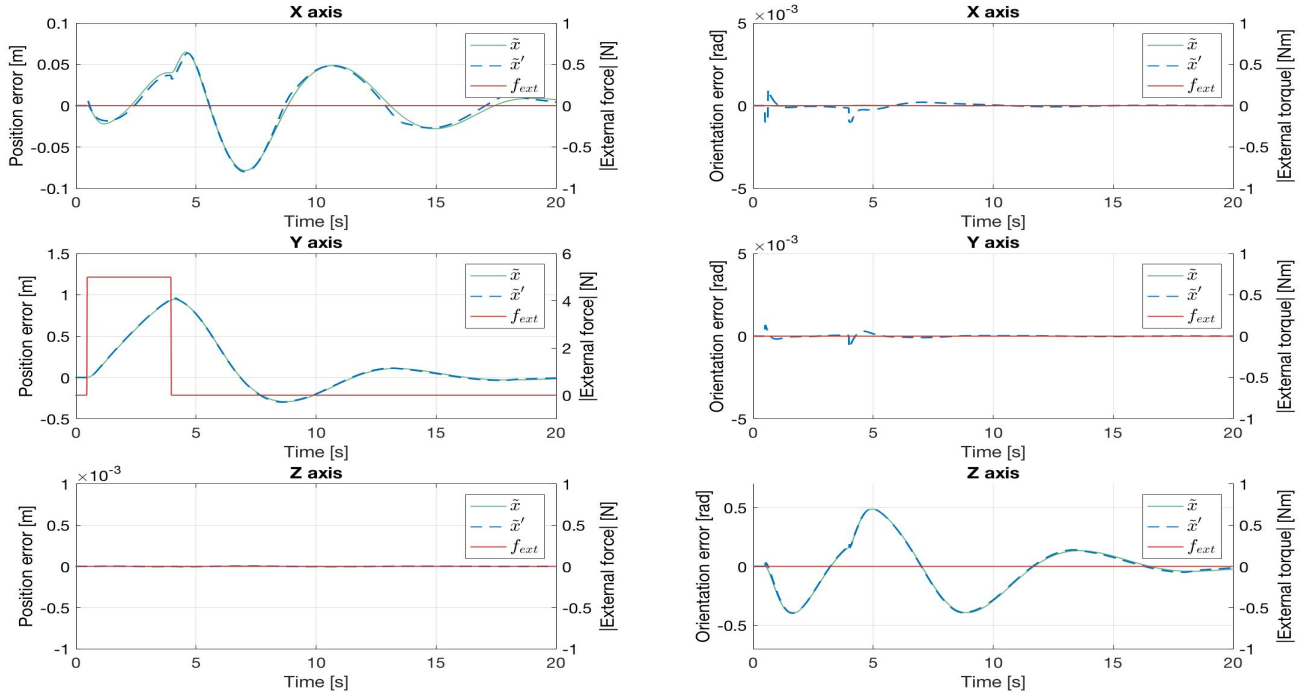


Figure 4.12: Actual and calculated (from impedance model) end-effector position and orientation error, with a compliant behavior. A $5N$ force was applied on the Y axis.

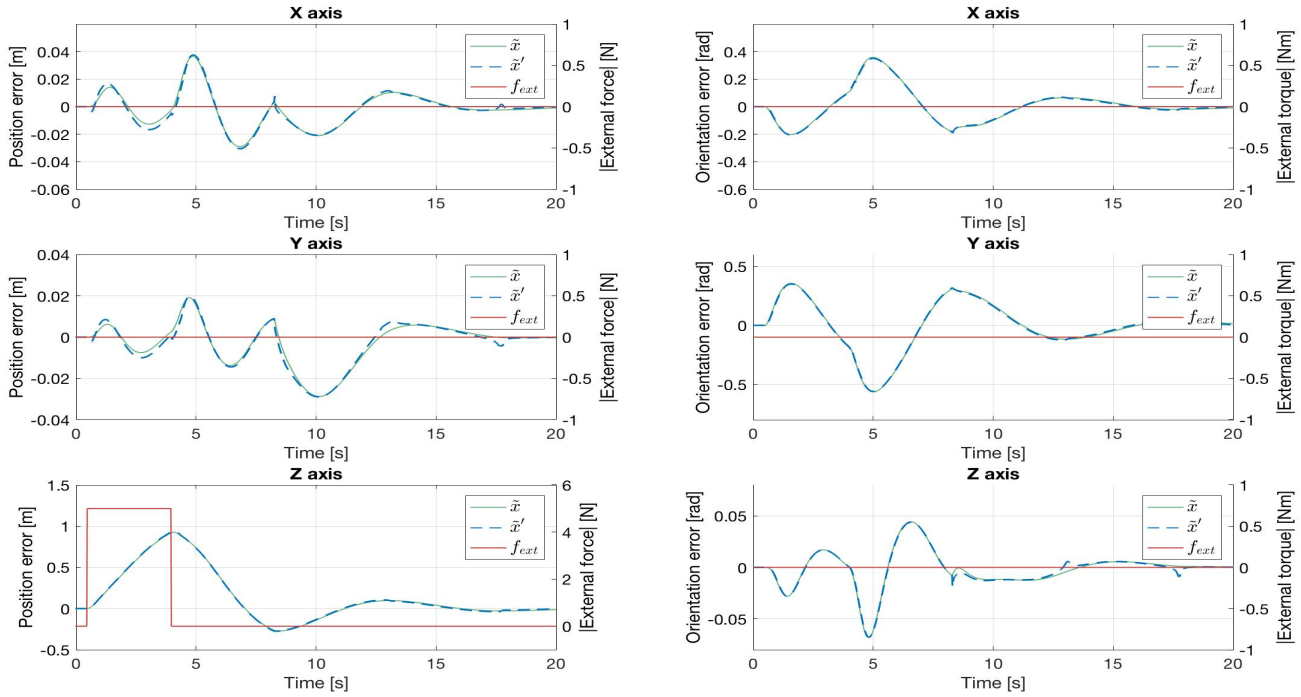


Figure 4.13: Actual and calculated (from impedance model) end-effector position and orientation error, with a compliant behavior. A $5N$ force was applied on the Z axis.

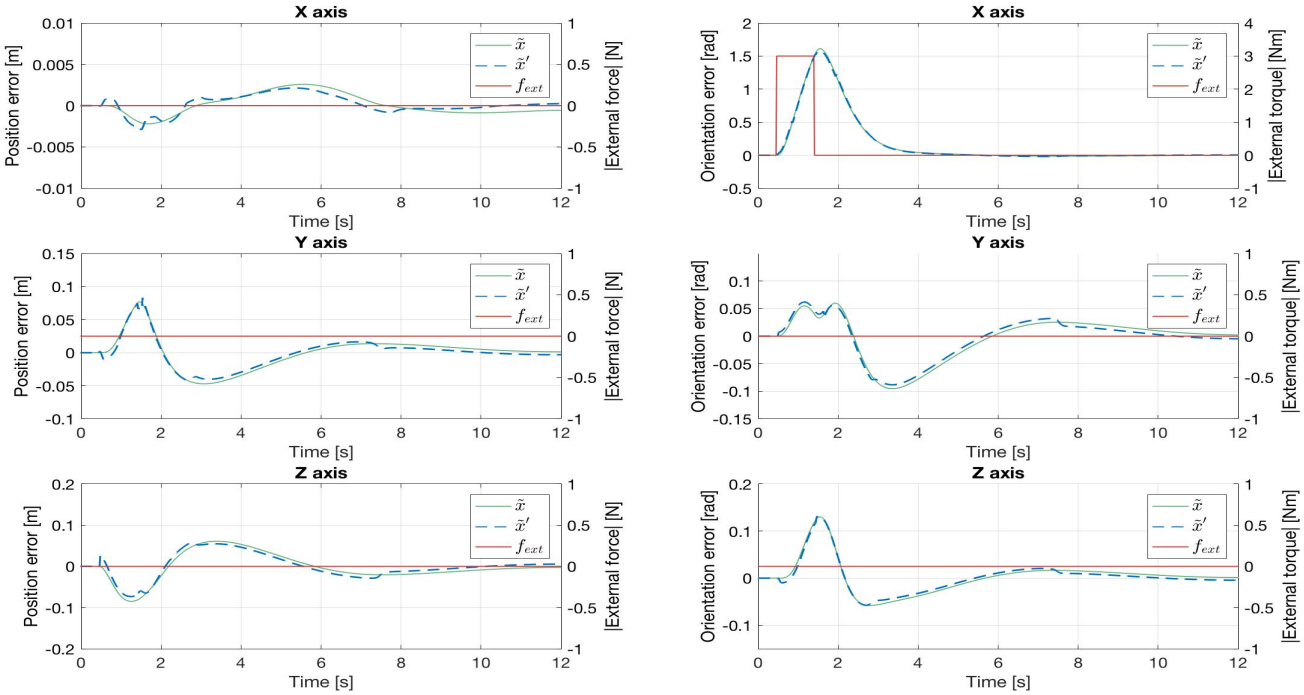


Figure 4.14: Actual and calculated (from impedance model) end-effector position and orientation error, with a compliant behavior. A $3Nm$ torque was applied on the X axis.

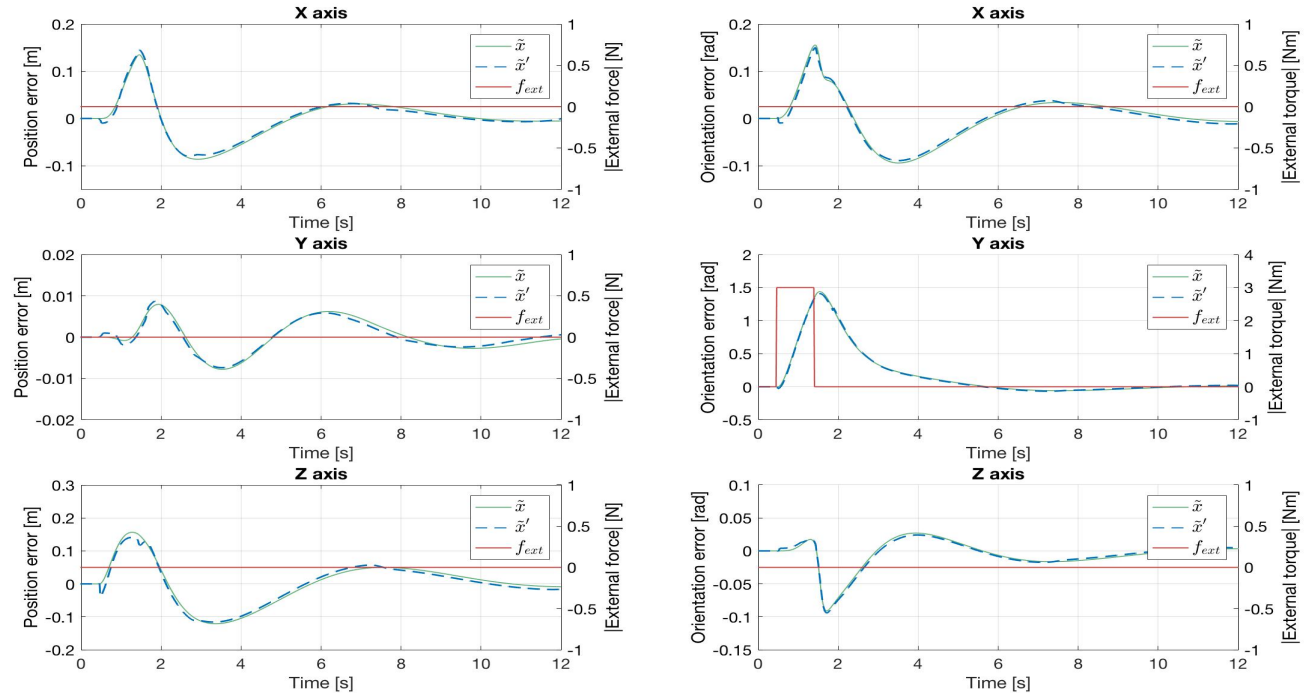


Figure 4.15: Actual and calculated (from impedance model) end-effector position and orientation error, with a compliant behavior. A $3Nm$ torque was applied on the Y axis.

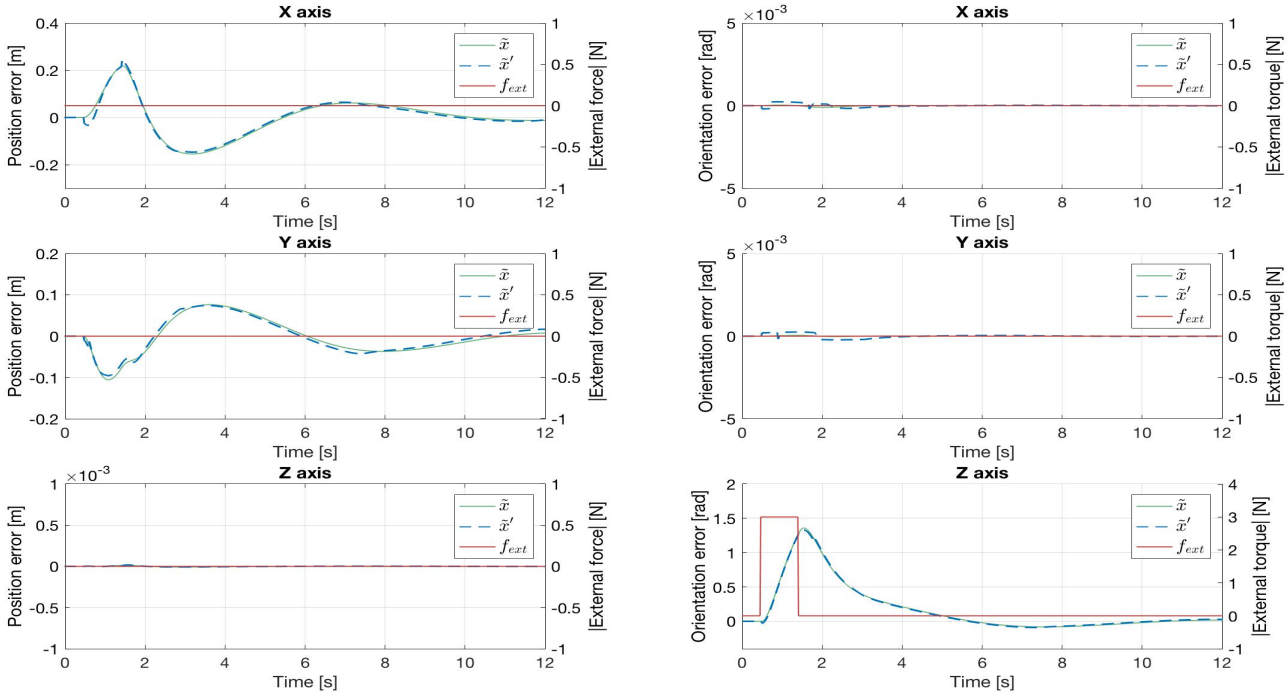


Figure 4.16: Actual and calculated (from impedance model) end-effector position and orientation error, with a compliant behavior. A $3Nm$ torque was applied on the Z axis.

Discussion

Analysing the previous presented results, several conclusions can be drawn.

The first one concerns the clear motion distinction between the two behaviors: with a rigid behavior and for the same external force or torque, the robot translation and rotation is visibly less than with a compliant behavior. For an accurate example, Figures 4.5-4.7 and 4.11-4.13 can be compared: the perturbation of $5N$ in the X, Y and Z axis generates a maximum position error 10 times higher for the compliant behavior. Additionally, all the other motion directions present a higher error value when compared to the ones in which the robot has a rigid behavior. This indicates that, as expected, a higher stiffness value in Equation (3.26) generates a rigid behavior where the end-effector tends to reach the desired rest state with a lower position/orientation error. Although a rigid behavior was successfully accomplished, the robot's motion rigidness is limited due to the Astrobee's propulsion power. Regarding the compliant behavior, this behavior is generated due to a lower stiffness value and the robot rather moves passively to the external perturbation, diverging more from the desired state.

Furthermore, the robot not only reaches for desired/rest state for both behaviors while an external force acts on the end-effector, but also in the absence of external perturbations.

Lastly, the end-effector behaves accordingly to desired impedance model expressed by Equation (3.26). This can be concluded due to the overlap of the actual simulated end-effector error motion, $\tilde{x}$, and the calculated end-effector error motion from the impedance model, $\tilde{x}'$. Additionally, the difference between the actual and calculated motion error never exceeds $7mm$ and $0.02rad$ which is considerably low when compared to the overall motion and thus can be neglected. Nevertheless, it can be explained due to formulation independent factors such as the robot's actuator response time, actuator's precision or small randomized effects existent on the ISS simulation that aim to mimic the ISS real environment.

4.2 Human Interaction Implementation

The algorithm proposed assumes that two agents are involved in the handover: a robot and a human.

As the formulated algorithm implementation was done using the Astrobee's simulator, the user interaction must be simulated as well. Thus, the current section aims to describe the user interaction interface implementation and validation.

Instead of simply pre-defining trajectories and actions for a simulated user hand, the hand model is rather controlled by a person, i.e., an user outside the simulation, by means of an hand-tracking device. This setup delivers a virtual reality experience for the user and enables real user interaction with the algorithm proposed for microgravity robots.

4.2.1 Simulated Hand Model

The main requirement concerning the simulated hand model relayed on its the ability to perform grasping motions similar to the human hand. To produce these motions, the hand should have five fingers. Therefore, the iCub hand was selected.

Given that loading the entire robot would be highly computational expensive, only the hand was implemented on the Gazebo simulator. Therefore, to implement the hand model, existing *Rviz* iCub meshes were used. Furthermore, it was implemented on Gazebo 7.0 and Ubuntu 16.04 LTS. The simulated hand model is shown in Figure 4.17.

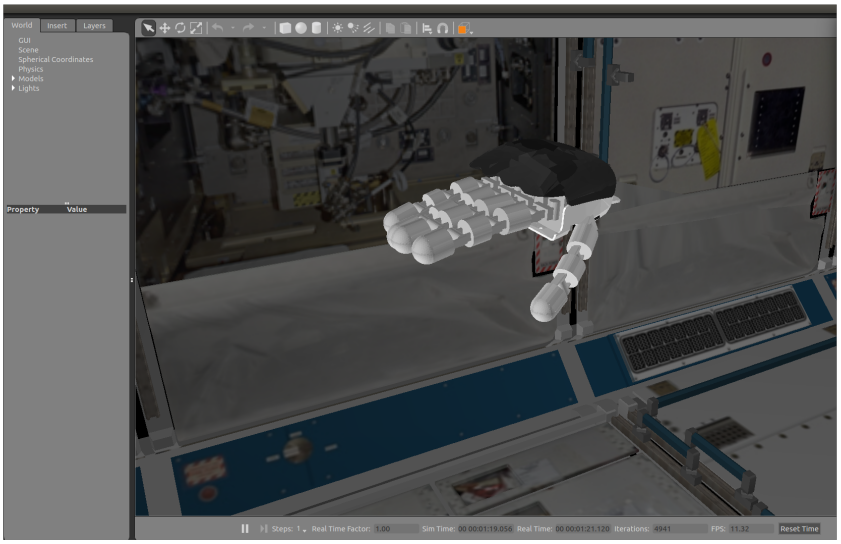


Figure 4.17: Simulated user hand model on Gazebo.

The hand model encompasses joints in each finger. One DOF was given to all of them, although the joints that connect the fingers to the palm are composed by more DOF in a human hand. This means that the user must not spread the fingers apart from each other when performing the interaction, with exception of the thumb.

Furthermore, the maximum reachable angle for each finger joint was defined as $90°$ and the minimum as $0°$ that occurs when the finger is not bending. As for the hand attitude angles, the roll-pitch-yaw Euler angles were limited by $-90°$ to $90°$. Concerning the hand position, the X, Y and Z values were not limited in the simulation.

Regarding the control interface for moving the finger joints, hand position and attitude, controllers were simulated in the Gazebo hand model using *ros_control*. This interface creates multiple ROS topics that connect the commanded values to the controllers. To do so, transmission elements were added to the hand model URDF. In addition to the transmission tags, the *gazebo_ros_* control plugin was also added to the URDF. This plugin parses the transmission tags and loads the appropriate hardware interfaces and controller manager. Additionally, an *.yaml* configuration file that contains the controller settings is created. Lastly, this file was loaded to the parameter server via a *roslaunch* file that included the *world* where the model is spawn. Given the desired interaction with the Astrobee, the same world as the robot (*ISS*) was selected to launch the hand model.

Lastly, the values given to the control interface implemented will further be described in the next section.

4.2.2 Simulated Hand Model Control through a Leap Motion Device

The current section aims to describe the mechanism of obtaining the control values for the simulated hand and fingers from a real user hand using a Leap Motion device and the integration of this interface with the simulator.

Leap Motion Brief Explanation

The Leap Motion is a tracking device which main purpose is the tracking of human hand and fingers.

The Leap Motion system employs a right-handed Cartesian coordinate system. The origin is centered at the top of the Leap Motion Controller. The X and Z axes lie in the horizontal plane, with the X axis running parallel to the long edge of the device. The Y axis is vertical, with positive values increasing upwards. The Z axis has positive values increasing toward the user. This coordinate system is presented in Figure 4.18

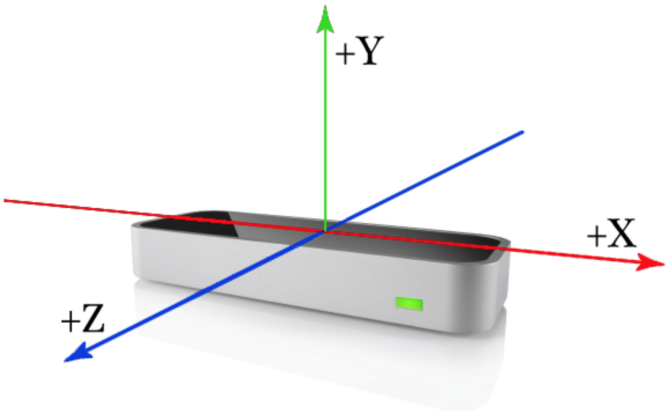


Figure 4.18: Leap motion axis.

The Leap Motion consists of two stereo cameras and three infrared LEDs. It tracks infrared light with a wavelength of $850 \times 10^{-9}\ m$, which is outside of the visible light spectrum. The sensors are directed along the Y axis – upward when the controller is in its standard operating position.

Positional data is provided for each finger and hand. In fact, an advantage of this device as compared to its predecessors is that it can distinguish each finger individually. The identification algorithm attaches to each finger, gesture and hand an unique identifier, making it easy for each entity to be tracked throughout the motion flow. This data is within a sub-millimetre accuracy (0.01 mm).

Additionally, the device has the advantage of having an attached programming interface called Leap Motion SDK (Software Development Kit), allowing to intuitively develop small applications APIs (Application

Programming Interface) in a wide variety of either high or medium level programming languages such as C++, Java, Python or JavaScrip.

Summarizing, the Leap Motion device provides a hand tracking method that is precise in its capture and digitization of the human fingers and hand movements.

Implementation Architecture

Given the multiple advantages of the Leap Motion sensor, this device was selected has the user hand tracking method.

In this manner, the hand and the fingers data were acquired from the device. To achieve this, the Leap Motion SDK that interfaces the device with Ubuntu 16.04 was used. This software provides a native interface to obtain the tracking data from the Leap Motion device, transforming it into intuitive vectors and classes, such as arm, hands and fingers. Fingers are identified by type name, i.e., thumb, index, middle, ring, and pinky and each finger contains four bones ordered from base to tip which are also identified my name, i.e., metacarpal, proximal, intermediate and distal. Both fingers and bones are represented in Figure 4.19. Furthermore, the native interface provided by SDK is a dynamic library that can be accessed through an available language wrapper libraries. Having this into consideration, Python was the selected language for this integration implementation.

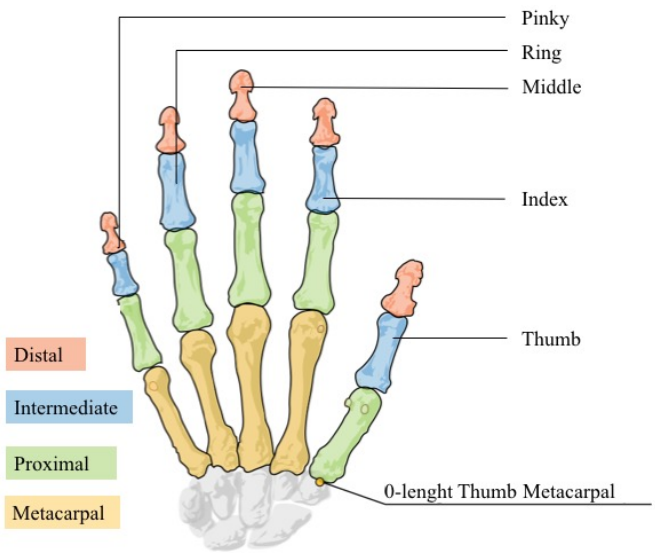


Figure 4.19: Representation of a human hand with reference to the relevant bones for the implemented interface, in color, and the finger's names in black.

Moreover, an integration with ROS had to be implemented. Thus, a ROS node (*data_to_topic*) that fetched the fingers position and the hand position/attitude from the SDK and published it as ROS topic (*/leapmotion/data*) was integrated. Lastly, another ROS node (*/leap_motion_subspubs*) that processed the tracking data was created. This node subscribes to */leapmotion/data* topic to access the hand tracking data and calculates the joint's angle, θ_{joint}. The calculation is done by taking the dot product of the vector aligned with the longitudinal axis of each bone, B_{ij}, and the vector aligned with the longitudinal axis of the previous bone to that one, $B_{(i-1)j}$, with $i = 1, 2, ..., nb$, where nb represents the bone index that takes value 4 except for the thumb where the value is 3 and $j = 1, ..., 5$ representing the finger index.

$$\theta_{joint} = cos^{-1} \frac{B_{ij} \cdot B_{(i-1)j}}{|B_{ij}| \times |B_{(i-1)j}|} \tag{4.5}$$

Lastly, the node publishes the calculated angles to the respective joint control command topic described on the previous section and publishes the hand palm position and attitude to the control topic concerning the hand motion. In total, the node publishes the data to 20 control topics (14 for the finger's joints and 6 for the hand motion).

Figure 4.20 displays schematically the described implementation architecture.

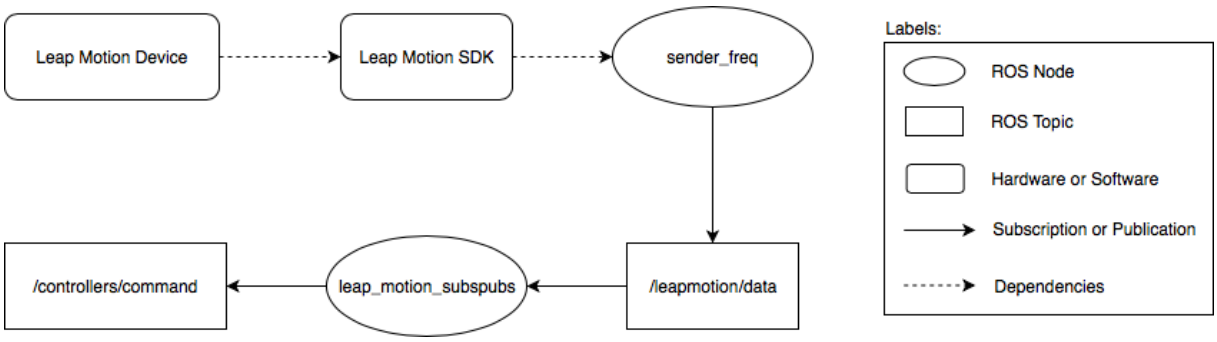


Figure 4.20: Block diagram of the implementation architecture of a Gazebo simulated hand model control through a Leap Motion device, using ROS environment. The topic concerning */controllers/command/* represents a simplified display of the 20 actual control command topics regarding the finger's joints and hand motion (position and attitude).

4.2.3 Results

With the aim of validating the previous described implementation, several tests were carried out with the simulated hand model and the Leap Motion device where hand translation/rotation motions and finger displacement were evaluated.

On the one hand, Figure 4.21-4.23 presents the X, Y and Z tracked user hand position, in blue, and the simulated hand model position, in green, respectively. On these tests, the hand was moved twice to the maximum and minimum tracked position without loss of visibility, e.g., position values higher or lower were not precisely tracked by the device. On the other hand, Figure 4.24-4.26 present the same information as the previous figures but regarding the hand roll, pitch and yaw, respectively. Lastly, Figure 4.27 displays all the calculated joint angles from the tracked fingers and the simulated angles when a fist gesture is performed twice.

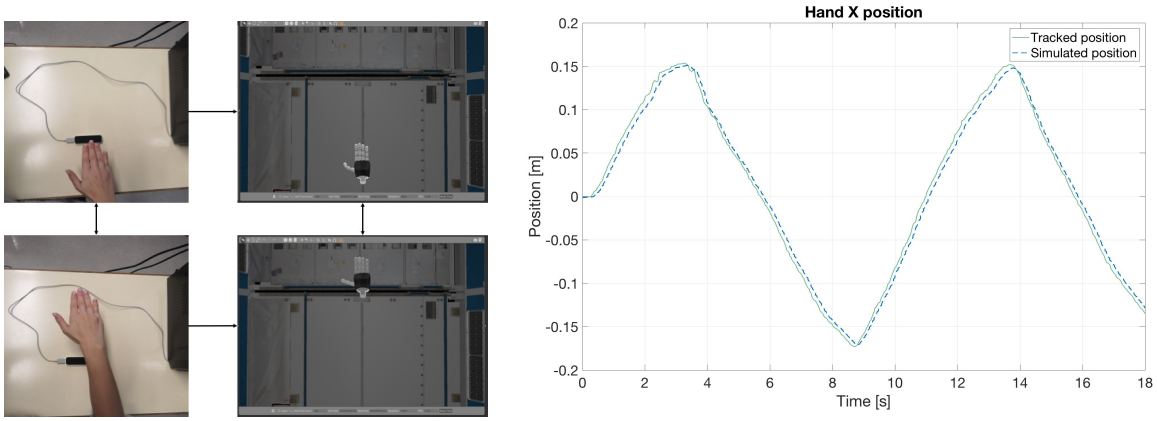


Figure 4.21: The four figures on the left represent the user hand translation on the X axis and the respective simulated hand motion. The plot on the right encompasses that motion data.

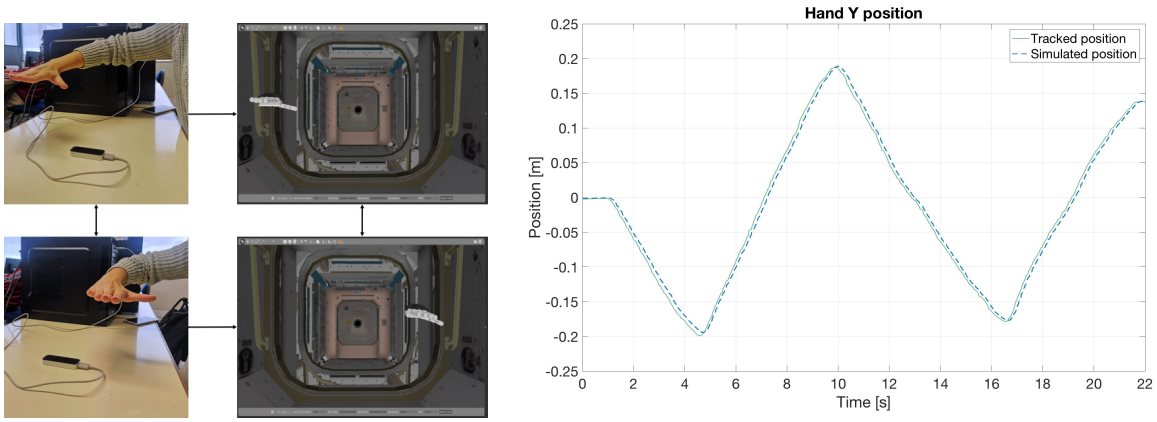


Figure 4.22: The four figures on the left represent the user hand translation on the Y axis and the respective simulated hand motion. The plot on the right encompasses that motion data.

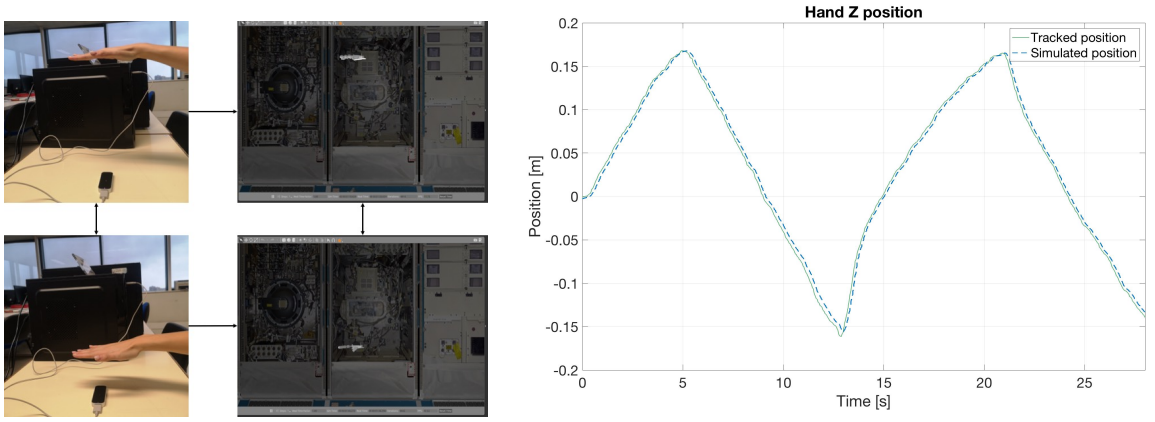


Figure 4.23: The four figures on the left represent the user hand translation on the Z axis and the respective simulated hand motion. The plot on the right encompasses that motion data.

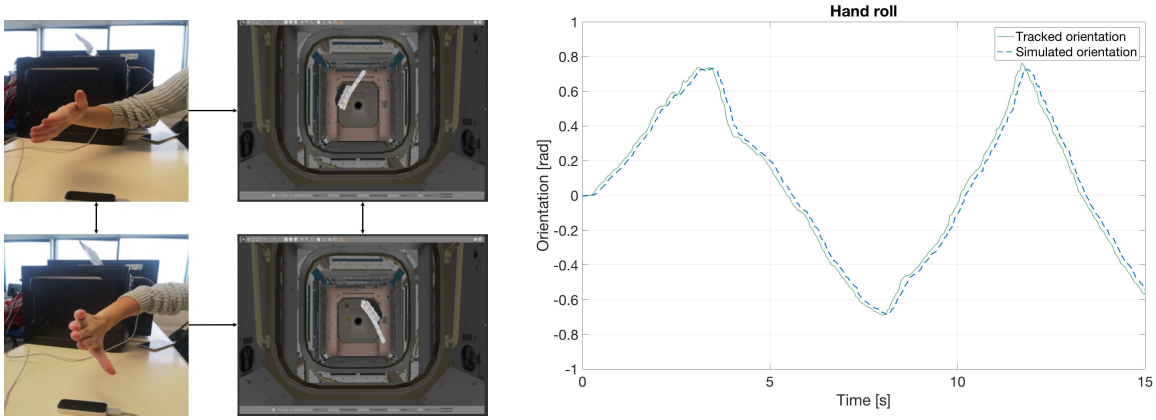


Figure 4.24: The four figures on the left represent the user hand roll and the respective simulated hand motion. The plot on the right encompasses that motion data.

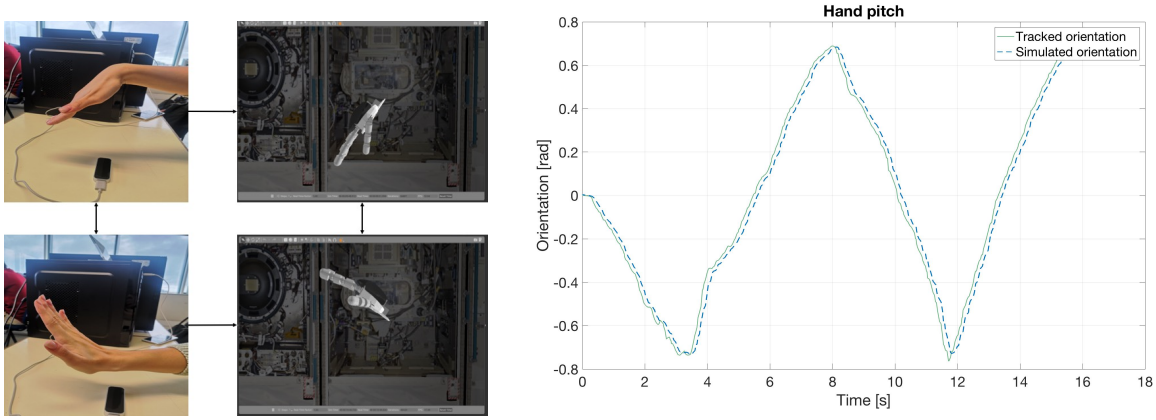


Figure 4.25: The four figures on the left represent the user hand pitch and the respective simulated hand motion. The plot on the right encompasses that motion data.

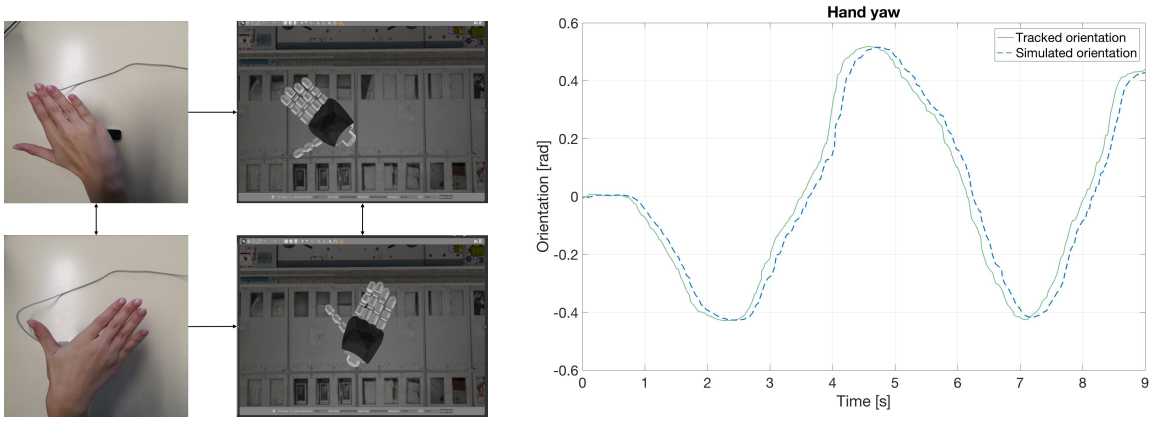


Figure 4.26: The four figures on the left represent the user hand yaw and the respective simulated hand motion. The plot on the right encompasses that motion data.

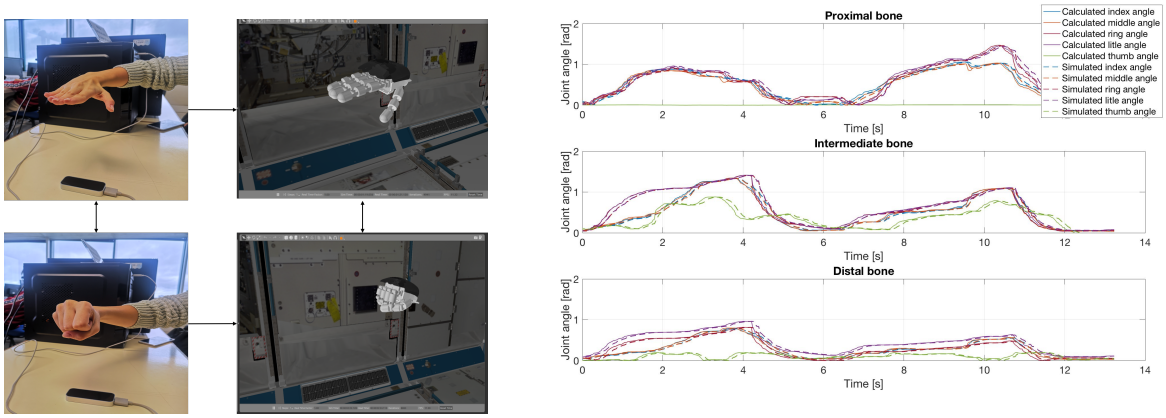


Figure 4.27: The four figures on the left represent a fist gesture performed by the user and the respective simulated hand gesture. The plot on the right encompasses that joint motion data.

Discussion

As is possible to observe from the previous figures, the simulated hand position and orientation and fingers are successfully controller through the Leap Motion device following the same movements as the user hand, with a processing time delay inferior to $0.1s$ which, accordingly to Miller's research [72], is small enough for users to perceived the simulated hand response as immediate and thus making the implemented interface as an accurate VR test environment.

4.3 Handover Algorithm Results

Following the formulated implementation and with the aim of validating the proposed handover algorithm, a robot-to-human and human-to-robot handover task were performed with a robot rigid and compliant behavior and using the user interaction interface developed.

Figure 4.28 displays the user hand, the Leap Motion device and the simulation environment.

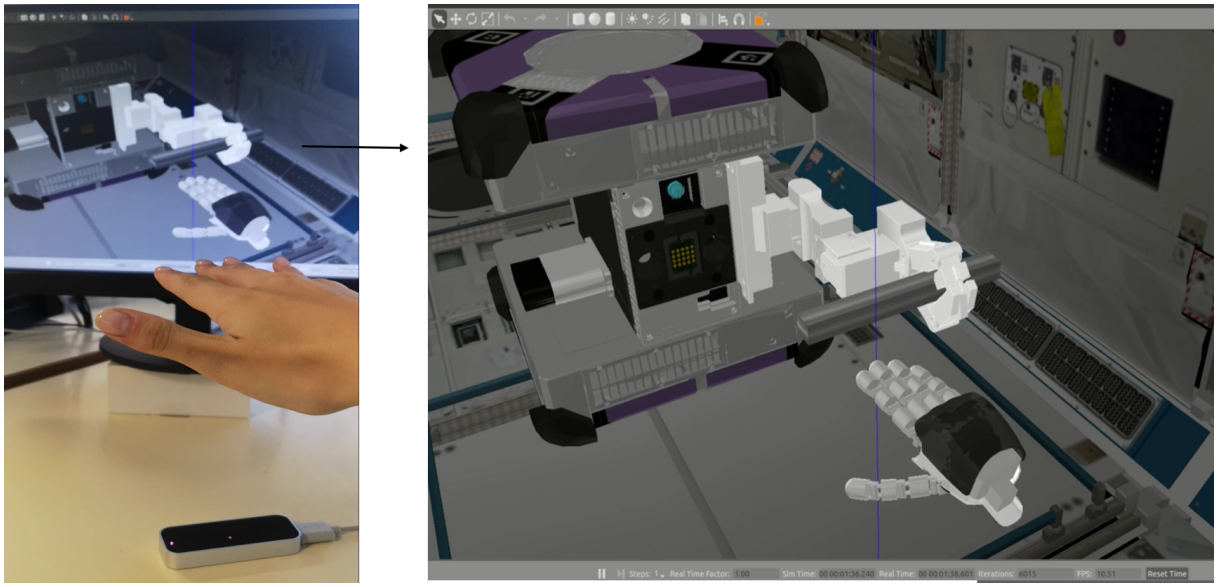


Figure 4.28: Representation of the user hand controlled via Leap Motion and the simulation environment.

4.3.1 Robot-to-Human

The first validation tests were performed concerning the robot-to-human handover where the robot acts as the giver and the user as the receiver.

The motion results are shown in Figure 4.29 and 4.31 for the rigid and compliant behavior, respectively. Additionally, the gripper angle and linear velocity is represented in Figure 4.30 and 4.32 for both behaviors.

Rigid Behavior

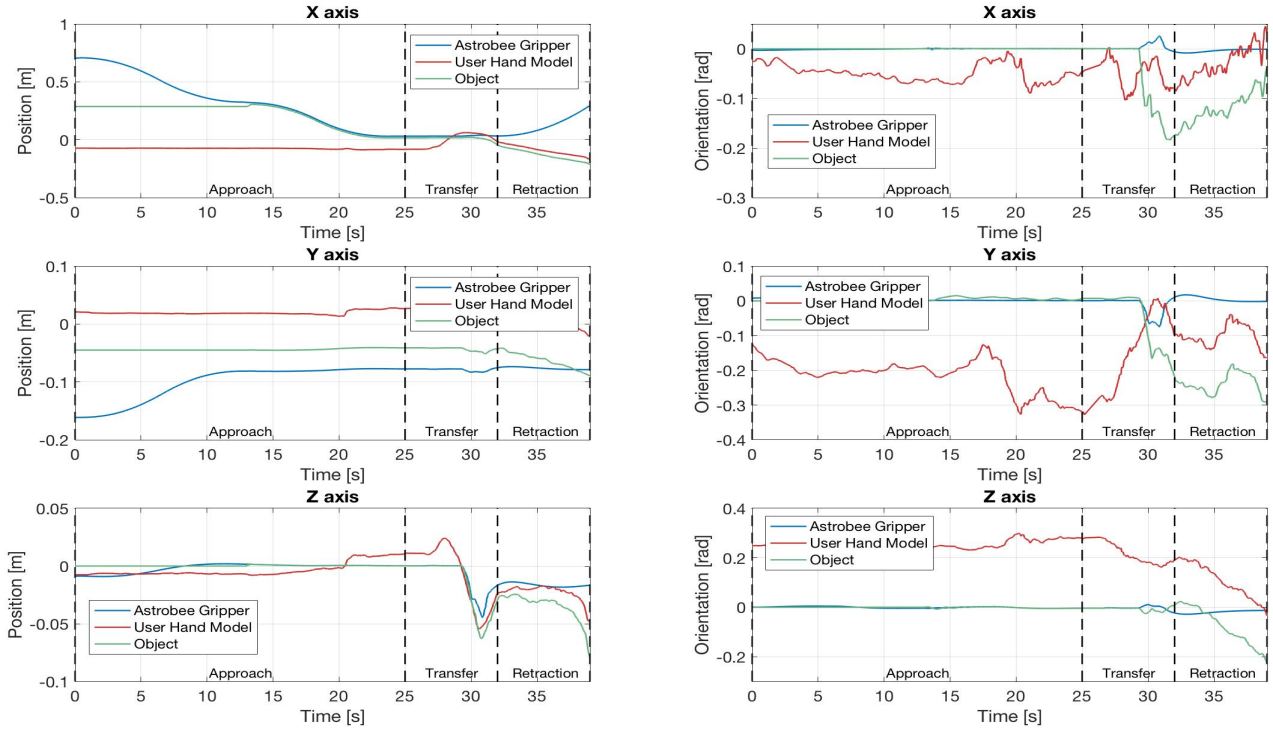


Figure 4.29: Position and orientation of the Astrobee's gripper, the simulated user hand model CM and simulated object CM during a robot-to-human object handover with a rigid behavior.

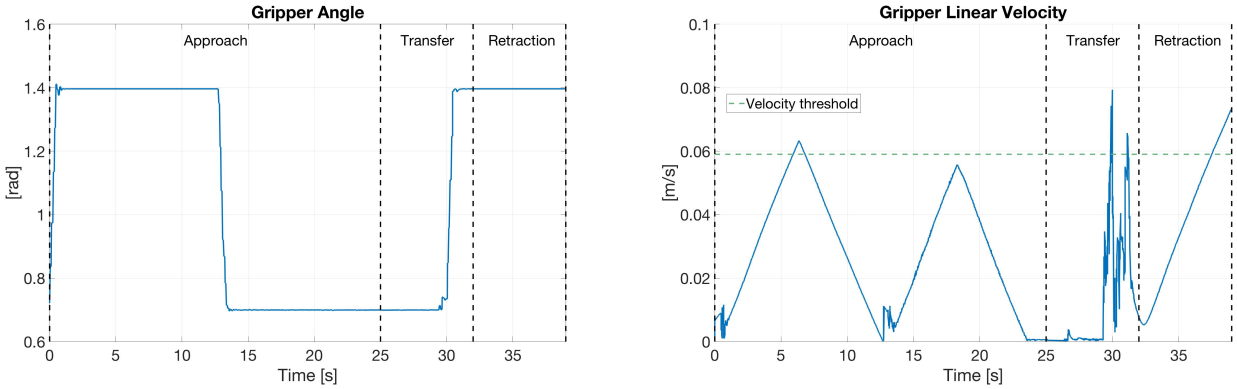


Figure 4.30: Astrobee's gripper angle (left) and linear velocity (right) during an robot-to-human object handover with a rigid behavior.

Compliant Behavior

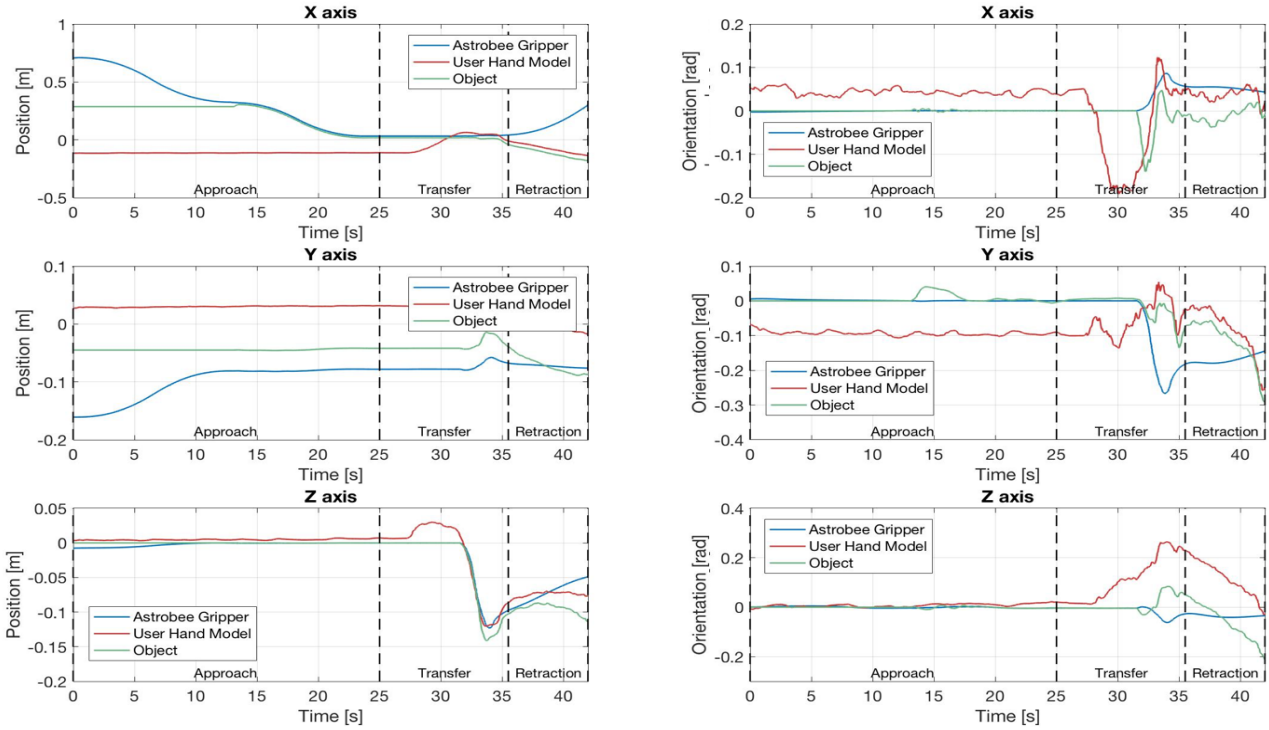


Figure 4.31: Position and orientation of the Astrobee's gripper, simulated user hand model CM and simulated object CM during a robot-to-human object handover with a compliant behavior.

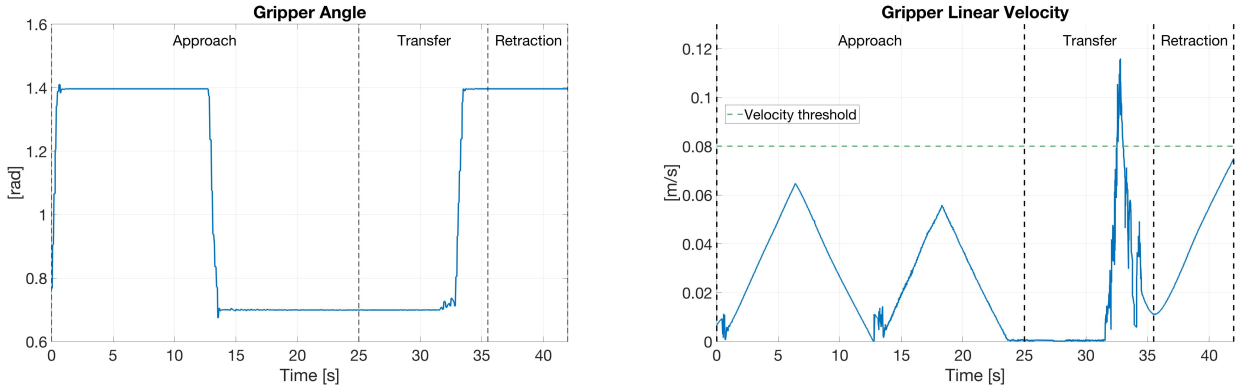


Figure 4.32: Astrobee's gripper angle (left) and linear velocity (right) during an robot-to-human object handover with a compliant behavior.

Discussion

The handover sequence is well perceived by analysing the X error position in Figure 4.29 and 4.31. As formulated, the robot initiates the handover with a closed gripper and without the object. It then opens the gripper, gets closer to the object and grabs it 13 seconds after initiating the task. Following, the robot moves to the handover location ($x = 0,\ y = -0.07,\ z = 0$) with 0 orientation in all axis. After activating the IC, the robot is ready to deliver the object and the Transfer phase initiates in which the robot signals the user and waits for the gripper velocity threshold. When this occurs, the robot opens its gripper and the object is transferred to the user. Lastly, both move away from the handover location, during the Retraction phase.

Furthermore, it is interesting to scrutinize the Z position representation for the same figures as this direction encompasses the higher robot's movement during the transfer. On the rigid behavior simulation, the gripper moves $0.04m$ when the user interaction is happening, whereby on the compliant case the robot reaches $0.14m$ away from the handover location. Therefore, during the transfer, the gripper motion is in accordance to the impedance control results, given that, for a similar user external interaction, the gripper's movement is minimum for a rigid behavior and it follows the simulated user hand for the compliant case. In this last case, a 10 times higher Z position error is generated when compared to the rigid behavior.

4.3.2 Human-to-Robot

The second validation tests were performed concerning the human-to-robot handover where the robot acts as the receiver and the user as the giver.

As in the previous section, the motion results are shown in Figure 4.33 and 4.35 for the rigid and compliant behavior, respectively and the gripper angle and linear velocity is represented in Figure 4.34 and 4.36.

The last validation test carried out, concerned the failure reception module. In this test, the Astrobee was firstly pushed without the object being placed on its gripper, followed by a successful reception. The results concerning the algorithm validation for this task are presented in Figure 4.37 and the gripper angle and linear velocity in Figure 4.38.

Rigid Behavior

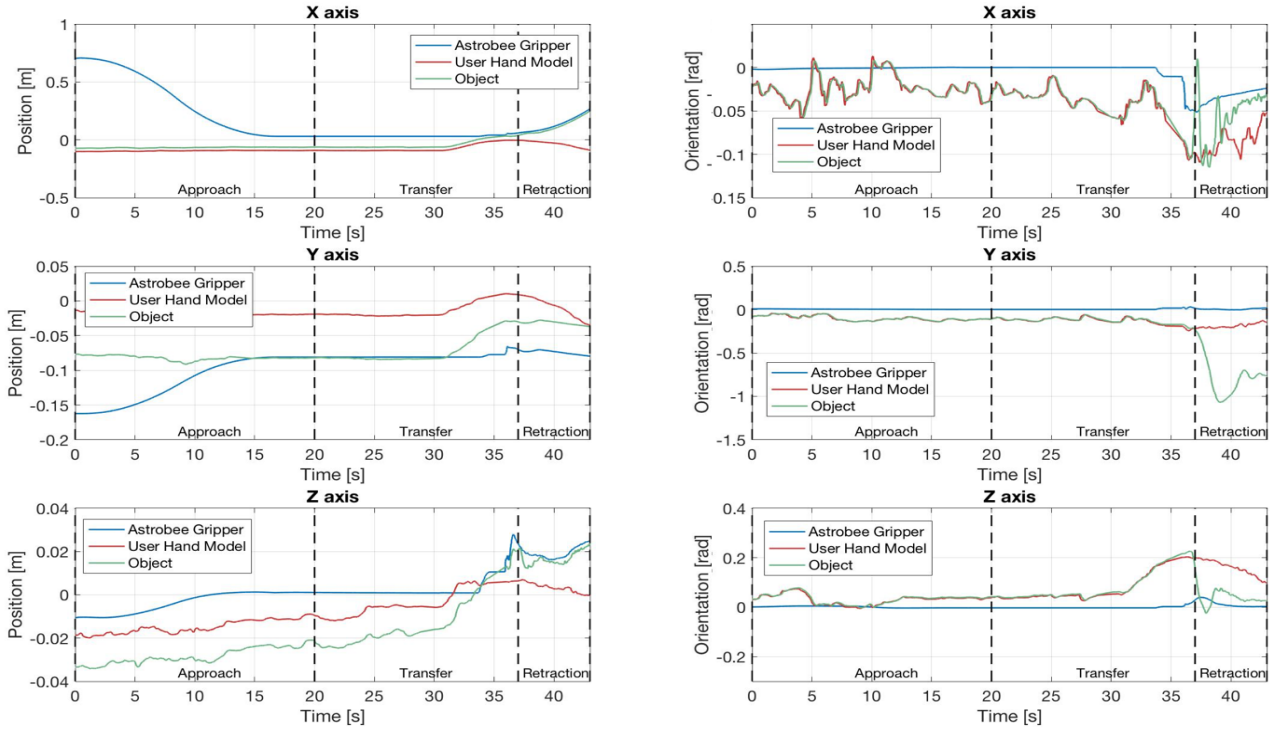


Figure 4.33: Position and orientation of the Astrobee's gripper, simulated user hand model CM and simulated object CM during a human-to-robot object handover with a rigid behavior.

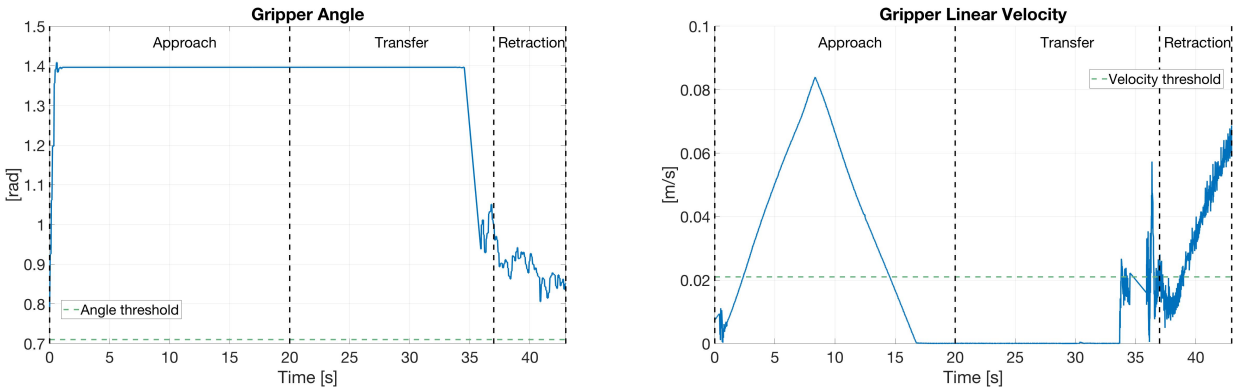


Figure 4.34: Astrobee's gripper angle (left) and linear velocity (right) during a human-to-robot object handover with a rigid behavior.

Compliant Behavior

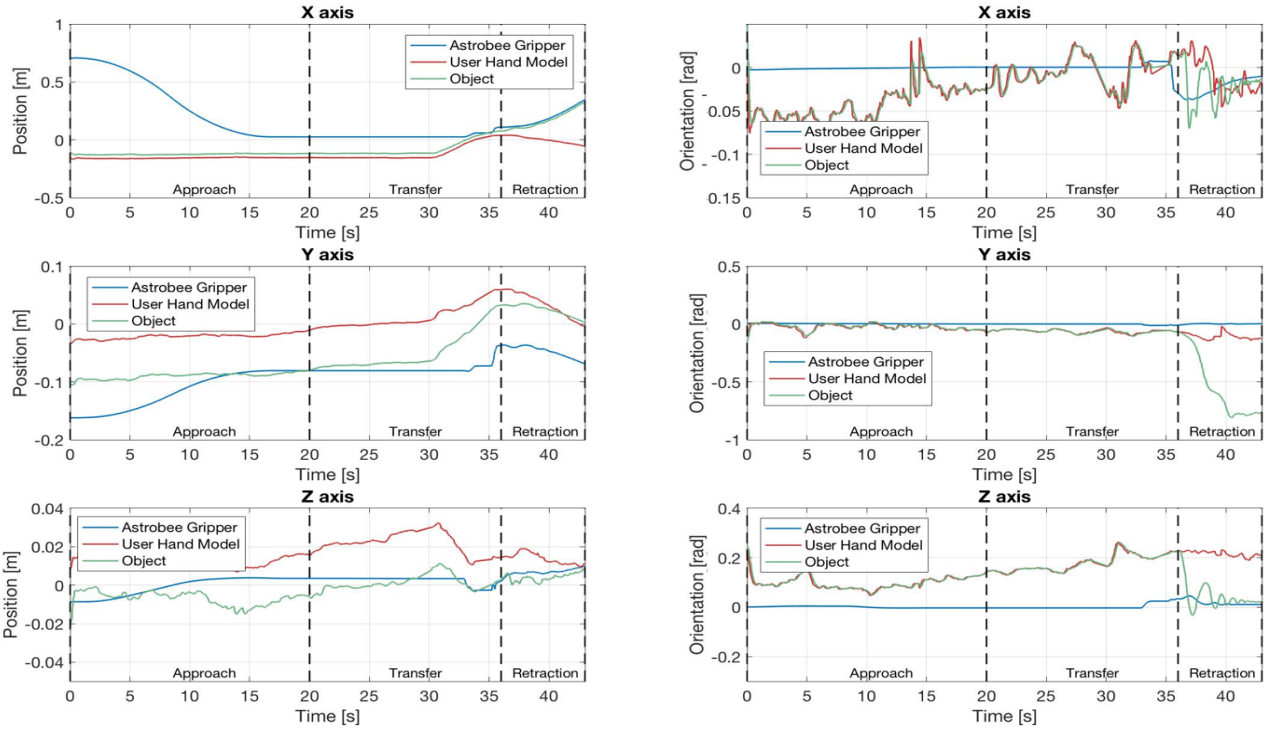


Figure 4.35: Position and orientation of the Astrobee's gripper, simulated user hand model CM and simulated object CM during a human-to-robot object handover with a compliant behavior.

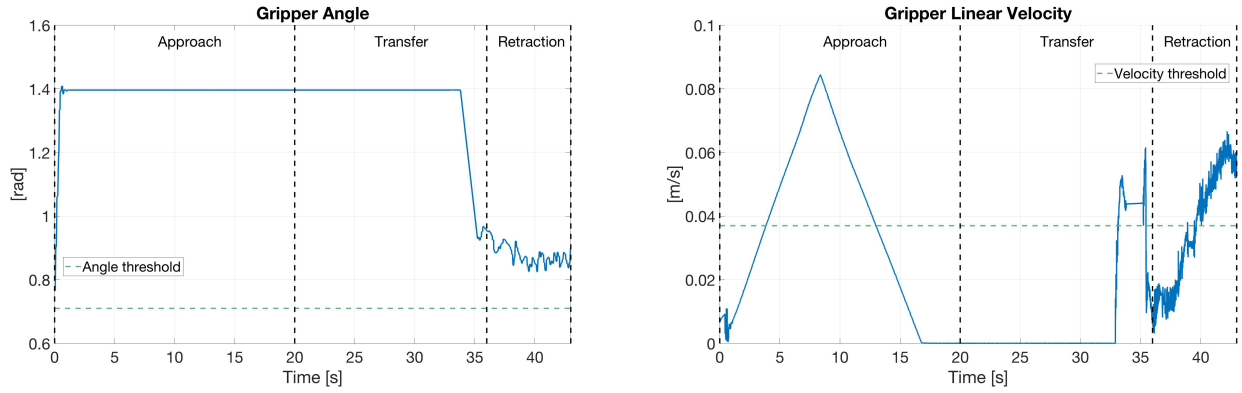


Figure 4.36: Astrobee's gripper angle (left) and linear velocity (right) during a human-to-robot object handover with a compliant behavior.

Failure Reception Module

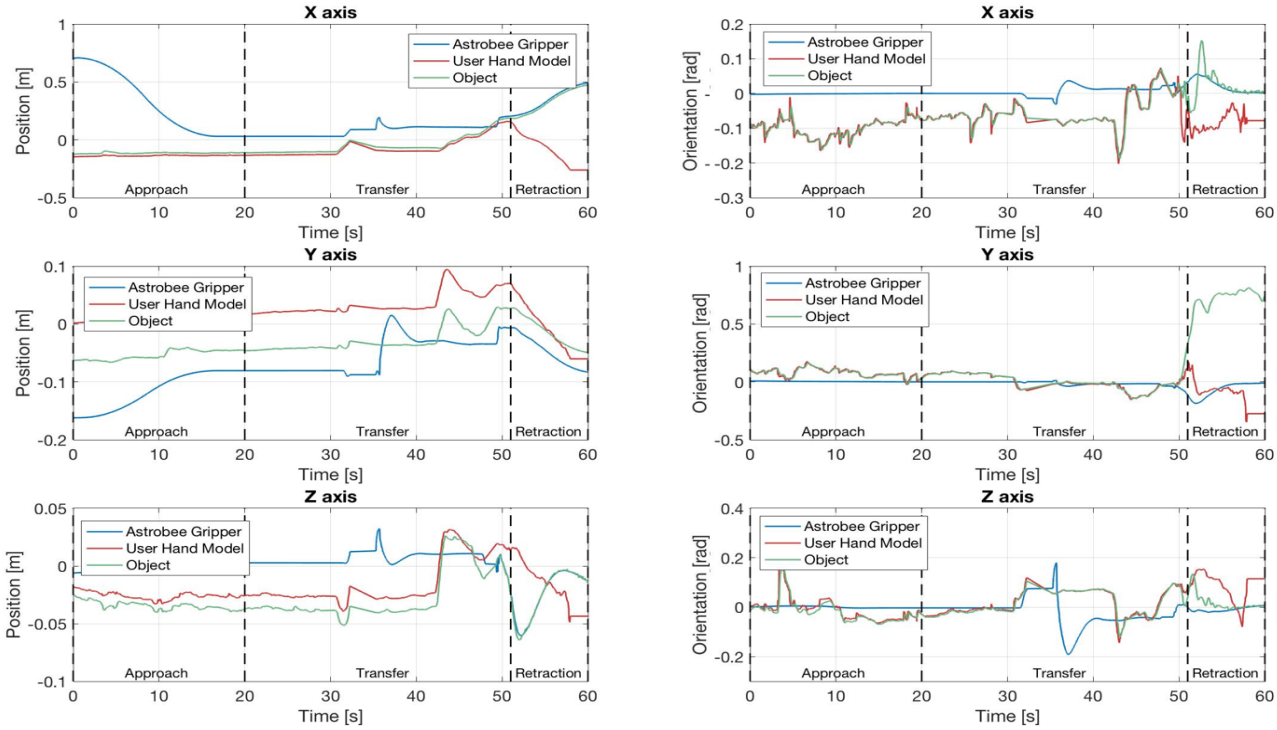


Figure 4.37: Position and orientation of the Astrobee's gripper, simulated user hand model CM and simulated object CM for a first failed reception and a second successful reception during a human-to-robot object handover with a compliant behavior.

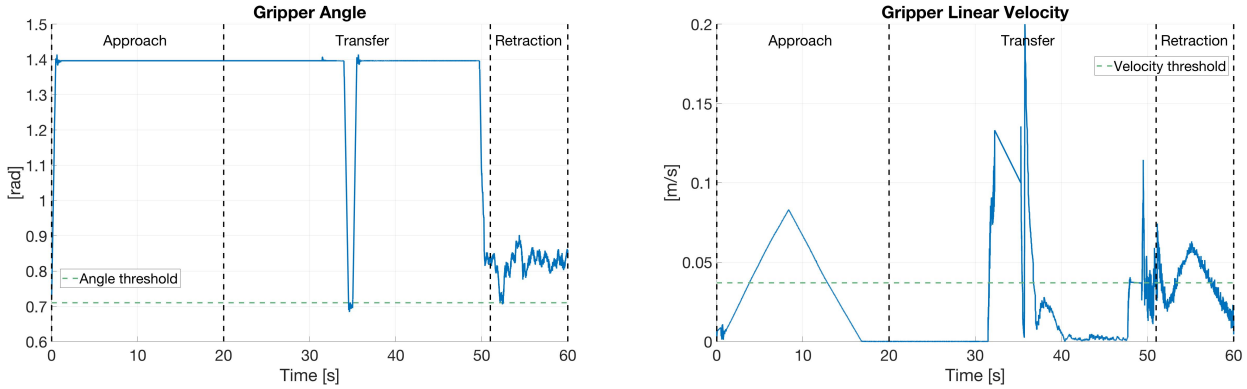


Figure 4.38: Astrobee's gripper angle (left) and linear velocity (right) for a first failed reception and a second successful reception during a human-to-robot object handover with a compliant behavior.

Discussion

As in the previous task results, the handover sequence is well perceived by analysing the X error position of Figure 4.33 and 4.35. In this case, the robot initiates the handover with a closed gripper and without the object as the user is grabbing it. It then opens the gripper and moves to the same handover location (x=0, y=-0.07, z=0) maintaining an open gripper. After activating the IC, the robot is ready to receive the object and thus, the Transfer phase begins in which the robot signals the user and waits to detect the object placement on its end-effector through the exceed of the gripper velocity threshold. When this occurs, the transfer of the object is performed and the user hand model and the Astrobee move away from the handover position.

Moreover, the X error position representation also shows twice the end-effector displacement for the compliant behavior when compared to the rigid behavior during the object's transfer. Thus, as expected, the Astrobee moves more passively to the user interaction on the compliant case.

Concerning the failure reception module, the first false reception attend was performed after 32 seconds of the task initiation. The robot closed the gripper and rapidly re-open it after checking the gripper angle and realizing that no object was delivered, followed by the user re-signaling. The object was then successfully transferred and, once again, the robot and the user hand moved away from the handover location.

Chapter 5

User Study

The previous Chapter described the implementation and results of the formulated algorithm and an user interaction interface with the simulation.

Given that an interaction interface was developed, the current chapter aims to describe the systematic user study conducted.

5.1 Aim of study

Based on the results of Kupcsik's study [73] it is known that for static handover tasks using cartesian compliant control, compliance parameters are less important for success and high stiffness is always preferred and highly rated. Gasparri in [74] shows that when using impedance handover dynamics the optimal manipulator stiffness is high in the case of perfect knowledge of the framework.

As described in the previous chapter, an impedance controller was implemented on a free-flying robot equipped with a fixed joints manipulator, designed to actuate on a microgravity environment. To emulate the human hand in the simulation, a model of a five finger hand was implemented and controlled through a Leap Motion device. Having that into consideration, the user study conducted aimed to explore the subjective outcomes effects on the user concerning the implemented impedance controller parameters. In particular, the preferences between the implemented behaviors in the transfer phase (rigid and compliant) were evaluated. Furthermore, the handover success of the two behaviors was also studied.

5.2 Hypothesis

Two hypothesis were in advance proposed for the experimental study:

H1 - "The impedance control parameters will affect the participant's perception of the object handover task with high stiffness (rigid behavior) being the most fluent, desirable and cooperative and low stiffness

(compliant behavior) the less fluent, desirable and cooperative".

H2 - "The impedance control parameters will affect the object handover task success with high stiffness (rigid behavior) being the most successful and low stiffness (compliant behavior) the less successful".

5.3 Study Location

The study was conducted in the 5^{th} floor of the North Tower on Instituto Superior Técnico located in Lisbon, Portugal.

5.4 Study Population and Eligibility Criteria

Ten people with ages between 21-30 participated in this experiment (6 female and 4 male). 7 participants had an engineering background, 2 had a health background and 1 had an arts background.

Since the implemented hand model on the simulation was the right hand it was important that all the participants were right-handed. This eligibility criterion avoided negatively influenced results due to that factor.

Furthermore, it was required participants that reported little to non previous familiarity with controlling simulated models via Leap Motion in order to have all the participants on the same experience circumstances when initiating the task.

5.5 Tasks and Conditions

To test the hypothesis proposed, a study with participants was conducted in which three different tasks were tested: robot-human, human-robot handover and a collaborative task that encompassed both handovers.

Moreover, the controller parameters conditions were manipulated to achieve rigid robot behavior or compliant robot behavior during each task.

The paragraphs below describe the three tasks performed during the study:

T1: Robot-to-human handover - The robot approaches the simulated hand with an object between its gripper. It stops at a pre-defined location and gives a signal (flashlight on and off once) when the controller has been activated and thus it is ready to perform the handover. The user must approach the simulated hand model to the object, grab it, wait for the robot's gripper to open and move the hand model

away from the robot with the object.

T2: Human-to-robot handover - The robot approaches the simulated hand without an object at a predefined location. When there, the robot opens the gripper and as in the previous described task, gives a signal (flashlight on and off once) when it is ready to perform the handover. The user must place the object on the robot's gripper that will then close. Lastly, the user must move the hand model away without the object.

T3: Collaborative Task - Rather then only receiving or giving an object, a robot may be involved on a task with a higher demand of collaboration such as holding an object when the astronaut does not need it. In this situation, both handover directions are performed. This collaborative task aims to mimic such cases. Therefore, a robot-to-human handover is firstly performed like previously described but after receiving the object, the robot does not move away but rather waits for the object once again like described in human-to-robot handover. In this manner, when it is ready to receive the object it gives the flashlight signal. After giving the object, the user must move the simulated model hand away from the robot.

5.6 Measures

It were used subjective measures to assess the user's preferences between the two conditions for each task. In particular three scales were evaluated — fluency, satisfaction and team work. The fluency scale extended a previously proposed measure of subjective fluency in [75] and [76], the satisfaction measure proposed in the tests performed in [77] and lastly it was possible to analyse the positive teammate traits also in [75]. Furthermore, the participants provided evaluation ratings after each interaction by giving a score between 1 (fully disagree) and 9 (fully agree), as proposed in [77], to the following statements:

S1 - *The robot contributed for the fluency of the interaction.*
S2 - *I was satisfied with the interaction.*
S3 - *The robot was committed to the task.*

Moreover, an objective measure was used to perceive which condition is more successful. This measure concerned the number of non-successful object handover in the three tasks. It was considered a successful object handover when the object was transferred between the giver and the receiver being always in contact with at least one of the agents.

5.7 Experimental Setup and Procedure

The participants were sited approximately 40 cm from the computer screen and the Leap Motion device will be placed on the table in front of the participant and 20 cm from screen. Figure 5.1 presents an example of an user during the experiment.

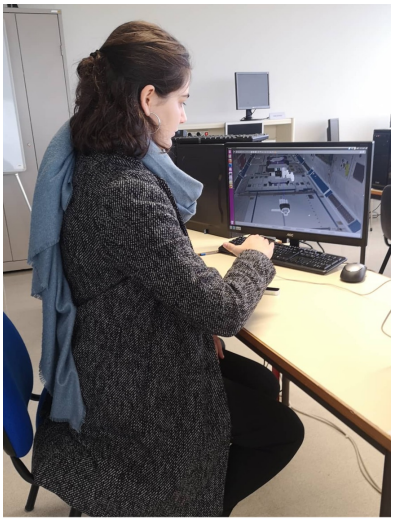


Figure 5.1: User performing the experiment.

With the goal of familiarization with the Leap Motion and the simulated hand integration, each participant performed different manoeuvres of their choice like doing a fist and experimenting with different attitudes and positions of the hand. Furthermore, the same object used on each task was also placed on the simulation with the aim of enabling the participant to experience grabbing it before doing it in the task itself. This section of the procedure lasted 10 minutes.

The second section of the experiment was the handover tasks. The three tasks proposed with each two different robot behavior comprised six experimental conditions. Therefore, the study involved 12 rounds of interaction for each participant – two for each experimental condition with randomized controlled trials. After each round of interaction, participants filled out the questionnaire presented on the measures section regarding their experience perception of the handover. Lastly, all the participants received the instructions that encompassed the description of the tasks and the experimental procedure. This is presented in Appendix B.

5.8 Results

The questionnaire's results are presented in Figure 5.2, 5.3 and 5.4, for the robot-to-human handover, human-to-robot handover and collaborative task, respectively. Additionally, Table 5.1 displays the total number of failed transfers for each task.

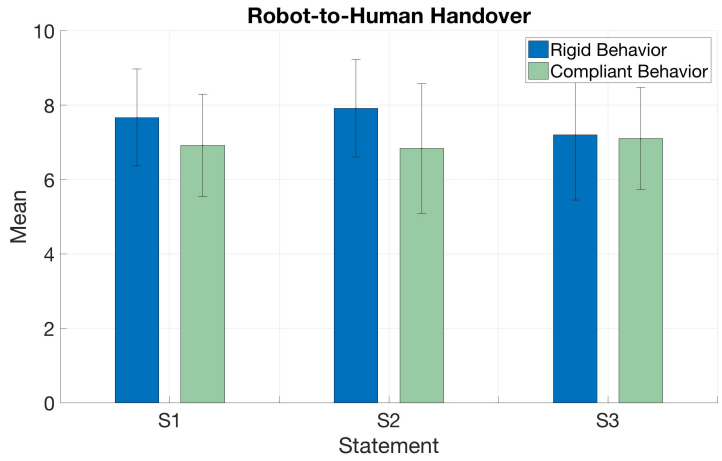


Figure 5.2: Representation of the mean and standard deviation of the questionnaire's results for a robot-to-human handover.

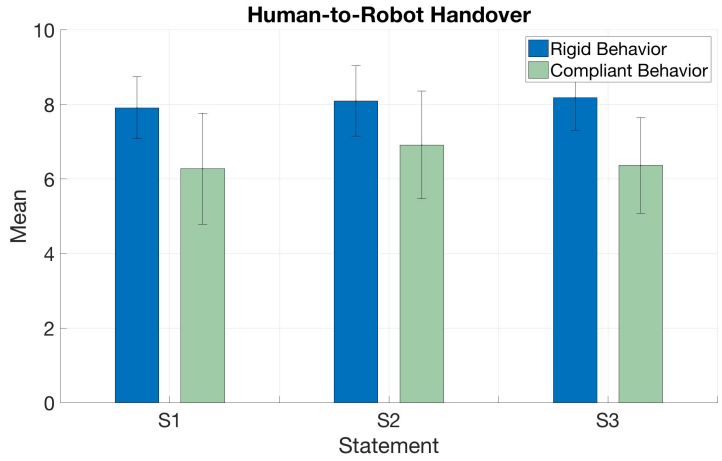


Figure 5.3: Representation of the mean and standard deviation of the questionnaire's results for a human-to-robot handover.

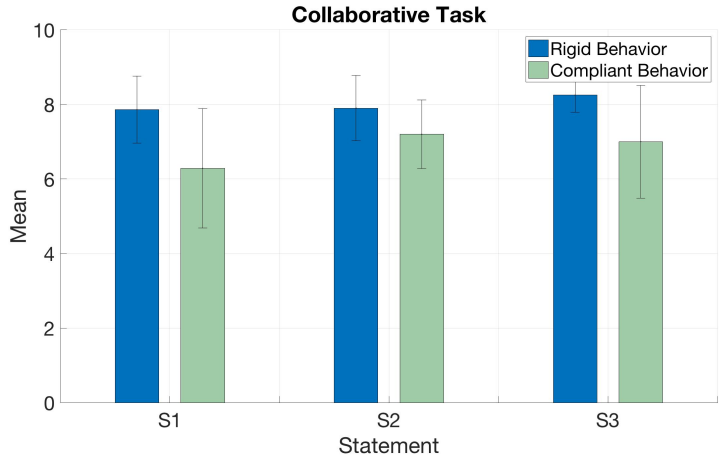


Figure 5.4: Representation of the mean and standard deviation of the questionnaire's results for a collaborative task.

		Rigid Behavior	Compliant Behavior
Task	Robot-to-Human Handover	3	3
	Human-to-Robot Handover	0	2
	Collaborative	2	3

Table 5.1: Total number of failed handovers on the three performed tasks.

Discussion

Concerning the user's responses to the proposed statements, the results presented on Figure 5.2 indicate that in a robot-to-human handover scenario users perceived higher fluency for a rigid behavior (p-value $= 0.1195$, $\alpha = 0.05$) and were more satisfied with the interaction also for the rigid behavior (p-value $= 0.0589$, $\alpha = 0.05$). Moreover, no substantial difference between both behaviors was felt regarding the robot commitment to the task (p-value $= 0.8880$, $\alpha = 0.05$) and thus concerning the cooperation perceived.

In Figure 5.3 shows higher distinction between the answers regarding the two behaviors in the human-to-robot handover scenario, as users perceived more fluency, satisfaction and cooperation for a rigid behavior (p-value of 0.0132, 0.0401 and 0.0057, respectively and with $\alpha = 0.05$).

As expected, for the collaborative task, the results were also higher for the rigid behavior concerning S1, S2 and S3 (p-value of 0.0375, 0.0445, 0.0492, respectively and with $\alpha = 0.05$). Nevertheless, the results were closer to the compliant behavior when compared to the results obtained for the human-robot handover. This can be explained to the robot-to-human handover part in the task.

The results presented on Table 5.1 show that although users perceived higher fluency and were more satisfied with the rigid behavior on the robot-to-human handover, the same number of failed transfers happen for both behaviors representing a 15% of failure. While in a human-to-robot handover task no failed transfers were recorded for a rigid behavior and two happened for the compliant resulting in 10% of failure. This results are in accordance to the questionnaire results regarding this scenario. Concerning the collaborative task, two failed transfers were recorded for the rigid behavior, while for the compliant behavior were registered three, generating a 10% of failure and 15% of failure, respectively.

Summarizing, the results indicate that H1 concerning the influence of the impedance control parameters on the participant's perception of fluency, satisfaction and cooperation was verified for the first two factors, given that no considerable difference was register regarding the perceived cooperation between both behaviors on the robot-to-human handover task. Additionally, the data supported H2 for the human-to-robot handover and collaborative task as more successful object transfers were performed for the rigid behavior.

Chapter 6

Conclusions

6.1 Conclusion

The emerging field of collaborative space robotics has a high impact on space exploration. This dissertation aimed at advancing that field by formulating an algorithm that enabled a free-flyer robot to perform an object handover with a human in a fluent, dynamic and successful manner, using force control during the transfer phase.

To achieve the proposed goal, the State-of-the-Art was explored and, to the best of the author knowledge, no record of this dissertation problem was found in the literature.

Following, the proposed algorithm was formulated with resource to a FSM approach where all relevant aspects of the robot-to-human and human-to-robot handover were analysed. Additionally and given the desired fluency and dynamic of the handover, a compliant control method was designed, more specifically, an impedance control approach.

Furthermore, the formulated algorithm was implemented on the NASA Astrobee's simulator and two impedance behaviors were studied concerning the transfer phase: rigid and compliant. As the results portrait, the robot followed the desired state and behaved accordingly to the derived impedance model for both cases.

Having into consideration the goal of conducting an user study, an user interaction interface was also developed. This interface enabled an user to control accurately a hand model implemented on the same simulator as the Astrobee through a tracking device, Leap Motion. In this manner, a virtual reality perception environment was generated enabling an user interaction with the robot.

Additionally, the formulated algorithm was successfully validated for both tasks, robot-to-human handover and human-to-robot handover, with both robot behaviors, rigid and compliant.

Given the successful validation of the proposed algorithm and the user interface, a systematic user experiment was conducted. This study aimed to not only understand the user's preferences between the two behaviors but also understand the impact of that behavior on the success of the task. Thus, three tasks were performed: robot-to-human handover, human-to-robot handover and a collaborative task that encompassed both handovers.
As the results showed, users perceived higher fluency, higher cooperation and were more satisfied with the interaction for the rigid behavior on the human-to-robot handover and on the collaborative task. Additionally, users perceived only higher fluency and were more satisfied with the interaction for the rigid behavior on the robot-to-human handover as no considerable difference was found concerning the cooperation factor. Lastly, more successful handovers were register for the rigid behavior during the human-to-robot handover and collaborative task. Therefore, the rigid behavior was overall more preferable and registered higher transfer success during the proposed tasks.

6.2 Contributions

The main contributions of the current dissertation were the following:

1. The work developed contributed to the research field of collaborative space robotics by providing a state-machine based algorithm that enables a free-flying robot to perform an object handover with a human in microgravity.

2. An in-depth insight of the several aspects of the handover phases between robots and humans were studied and summarized, contributing for the HRC field.

3. Concerning the contribution for the field of compliant control, this thesis formulated, implemented and validated an impedance controller for a free-flying robot equipped with a manipulator designed to actuate in a microgravity environment. Additionally, two types of control behaviors were studied: rigid and compliant.

4. The algorithm implementation was performed on the Astrobee simulator, contributing with a new possible human-robot interaction functionality for the recently launched robot.

5. An user interaction interface that encompassed a simulated hand controlled via Leap Motion was developed enabling a VR testing environment not only for the current research but also for future applications using the Gazebo simulator.

6. Lastly, this work also contributed with an user study on how impedance control parameters impact qualitatively and quantitatively an object handover transfer phase between a free-flying robot and a human on a microgravity environment.

6.3 Future Work

Concerning future work, several aspects of the dissertation were not in-depth explored due to lack of time or dispersion from the thesis main concerns. Additionally, other applications or functionalities could be newly formulated and implemented following the developed research.

Among this aspects is the grasping algorithm. For the work developed and as mentioned during the state-machine formulation, no grasping algorithm was applied as this field was considered out the dissertation goals scope. Furthermore, the object shape was assumed to be constant and known. It would be interesting to integrate a grasping algorithm to accurately calculate the gripper motion and force accordingly to the object shape. Additionally, a gradual grasping algorithm that adjusted the robot's gripper angle to the user hand motion could also be implemented increasing the perceived fluency during the transfer phase.

Another assumption done was concerning a fixed joint scenario for the implementation on the Astrobee. As the formulation for controllable joints was presented on Chapter 3, the impedance controller with a non-fixed manipulator can be eventually validated, perhaps on one of the recent arms being developed for Astrobee or on another microgravity free-flying robot equipped with a manipulator.

Furthermore, the force control module could be substituted for another controller and user tests could be performed to understand the user's preferences between both.

Having into consideration larger objects present on the ISS, the algorithm could be implemented on two robots where the object could be simultaneously carried. Consequently, an interface interaction that included both hands could be developed for tests.

Lastly, validating the work developed on a microgravity environment with astronauts would constitute the ultimate future work for this dissertation.

Bibliography

[1] E. Seedhouse. *Tim Peake and Britains Road to Space*, volume 540. Cham, Switzerland: Springer, 1^{nd} edition, 2017.

[2] M. Duggan and T. Moseman. Deep space gateway architecture to support multiple exploration demonstration goals. *IEEE Aerospace Conference*, 2018.

[3] A. Bauer, D. Wollherr, and M. Buss. Human–robot collaboration: A survey. *International Journal of Humanoid Robotics*, 5(01):47–66, Nov. 2008.

[4] B. J. Grosz. Collaborative systems. *AAAI Presidential Address*, (2):67–85, Nov. 1996.

[5] A. Ajoudani, A. M. Zanchettin, A. A.-S. S. Ivaldi, K. Kosuge, , and O. Khatib. Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42(5):957–975, 2017. doi: 10.1007/s10514-017-9677-2.

[6] N. Sebanz, H. Bekkering, and G. Knoblich. Joint action: bodies and minds moving together. *Trends in Cognitive Sciences*, 10(2):70–76, 2006. doi:10.1016/j.tics.2005.12.009.

[7] T. Ariyoshi, K. Nakadai, and H. Tsujino. Effect of facial colors on humanoids in emotion recognition using speech. *13th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, 2014. doi:10.1109/roman.2004.1374730.

[8] K. Park, J. S. Lee, H. Jung, and Y. Lee. Human-robot interface using robust speech recognition and user localization based on noise separation device. *The 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2009. doi: 10.1109/roman.2009.5326264.

[9] X. Zhao, A. M. Naguib, and S. Lee. Kinect based calling gesture recognition for taking order service of elderly care robot. *The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2014.

[10] H. I. Lin and Y. P. Chiang. Understanding human hand gestures for learning robot pick-and-place tasks. *International Journal of Advanced Robotic Systems*, 12(5):49, 2015. doi:10.5772/60093.

[11] K. Kosuge and N. Kazamura. Control of a robot handling an object in cooperation with a human. *6th IEEE International Workshop on Robot and Human Communication (RO-MAN)*. doi:10.1109/roman.1997.646971.

[12] D. J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar. Collaborative human-humanoid carrying using vision and haptic sensing. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 607–612, 2014. doi:10.1109/icra.2014.6906917.

[13] D. Feth. Haptic human–robot collaboration: Comparison of robot partner implementations in terms of human-likeness and task performance. *Presence: Teleoperators and Virtual Environments*, 20 (2):173–189, 2011. doi: 10.1162/presa00042.

[14] W. Weitschat, J. Ehrensperger, M. Maier, and H. Aschemann. Safe and efficient human-robot collaboration part i: Estimation of human arm motions. *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. doi: 10.1109/icra.2018.8461190.

[15] A. Dragan, K. Lee, and S. Srinivasa. Legibility and predictability of robot motion. *8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308, 2013.

[16] K. Iwatsuka, K. Yamamoto, and K. Kato. Development of a guide dog system for the blind with character recognition ability. *IEEE Computer Society*, pages 401–405, 2004. doi: 10.1109/cc-crv.2004.1301475.

[17] I.Werry, K. Dautenhahn, B. Ogden, and W. Harwin. Investigating a robot as a therapy partner for children with autism. *European Conference for the Advancement of Assistive Technology (AAATE)*, 2001.

[18] B. J. You, M. Hwangbo, S. O. Lee, S. Oh, Y. D. Kwon, and S. Lim. Development of a home service robot 'issac'. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2630–2635, 2003.

[19] M. W. Kadous, R. K.-M. Sheh, and C. Sammut. Caster: A robot for urban search and rescue. *Australasian Conference on Robotics and Automation (ARAA)*, 2005.

[20] R. Regele. Cimon: A visual navigation system for flying through the international space station. *International Aeronautical Federation*, 2018.

[21] L. Fluckiger, K. Browne, B. Coltin, J. Fusco, T. Morse, and A. Symington. Astrobee robot software: Enabling mobile autonomy on the iss. *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2018.

[22] P. Roque and R. Ventura. Space cobot: Modular design of an holonomic aerial robot for indoor microgravity environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[23] R. Shadmehr and F. Mussa-Ivaldi. Adaptive representation of dynamics during learning of a motor task. *The Journal of Neuroscience*, 14(5):3208–3224, 1994. doi: 10.1523/jneurosci.14-05-03208.1994.

[24] D. W. Franklin, G. Liaw, T. E. Milner, R. Osu, E. Burdet, and M. Kawato. Endpoint stiffness of the arm is directionally tuned to instability in the environment. *The Journal of Neuroscience*, 27(29): 7705–7716, 2007.

[25] M. Krutky, R. Trumbower, and E. Perreault. Effects of environmental instabilities on endpoint stiffness during the maintenance of human arm posture. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, 2009. doi: 10.1109/iembs.2009.5334751.

[26] A. Calanca, R. Muradore, and P. Fiorini. A review of algorithms for compliant control of stiff and fixed-compliance robots. *IEEE/ASME Transactions on Mechatronics*, 21(2):613–624, 2016. doi: 10.1109/tmech.2015.2465849.

[27] P. Neranon and R. Bicker. Force/position control of a robot manipulator for human-robot interaction. *Thermal Science*, 20(2):537–548, 2016. doi: 10.2298/tsci151005036n.

[28] M. Chumacher, J. Wojtusch, P. Beckerle, and O. V. Stryk. An introductory review of active compliant control. *Robotics and Autonomous Systems*, 119:185–200, 2019. doi: 10.1016/j.robot.2019.06.009.

[29] G. Zeng and A. Hemami. An overview of robot force control. *Robotica*, 15(5):473–482, 1997.

[30] J. Cazalilla, M. Vallés, Valera, V. Mata, and M. Díaz-Rodríguez. Hybrid force/position control for a 3-dof 1t2r parallel robot: Implementation, simulations and experiments. *Mechanics Based Design of Structures and Machines*, 44(1-2):16–31, 2016. doi: 10.1080/15397734.2015.1030679.

[31] B. Siciliano and L. Villani. *The Kluwer International Series in Engineering and Computer Science*, volume 540. Kluwer Academic Publishers, 1^{nd} edition, 1999.

[32] M. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.

[33] H. Bruyninckx and J. D. Schutter. Specification of force-controlled actions in the task frame formalism: A survey. *IEEE Transactions on Robotics and Automation*, 12(5):581–589, 1996.

[34] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. Gersem. Active compliant motion: a survey. *Advanced Robotics*, 19(5):479–499, 1996. doi: 10.1163/156855305323383767.

[35] H. M. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981. doi: 10.1115/1.3139652.

[36] S. Chiaverini, B. Siciliano, and L. Villani. Parallel force/position control with stiffness adaptation. *in IEEE International Conference on Robotics and Automation (ICRA)*, 1997. doi: 10.1109/robot.1997.614290.

[37] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993. doi: 10.1109/70.246048.

[38] N. Hogan. Impedance control: An approach to manipulation: Part i—theory. *Journal of Dynamic Systems, Measurement and Control*, 107(1):1–7, 1985. doi: 10.1115/1.3140702.

[39] N. Hogan. Impedance control: An approach to manipulation: Part ii—implementation. *Journal of Dynamic Systems, Measurement and Control*, 107(1):8–16, 1985. doi: 10.1115/1.3140713.

[40] M. Tufail and C. de Silva. Impedance control schemes for bilateral teleoperation. *International Conference on Computer Science Education (CSEIT)*, pages 44–49, 2014.

[41] T. Yoshikawa. Force control of robot manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (06):220–2269, 2000.

[42] D. Gorinevsky, A. Formalsky, and A. Schneider. *Force Control of Robotics Systems*. CRC Press, Boca Raton, 1^{nd} edition, 1997.

[43] M. Mendoza-Gutierrez, F. Reyes, I. Bonilla-Gutierrez, and E. Gonzalez-Galvan. Proportional-derivative impedance control of robot manipulators for interaction tasks. *Proceedings of the Institution of Mechanical Engineers*, 225(3):315–329, 2011. doi: 10.1080/15397734.2015.1030679.

[44] R. Volpe and P. Khosla. The equivalence of second-order impedance control and proportional gain explicit force control. *The International Journal of Robotics Research (IJRR)*, 14(06):574–589, 1995. doi: 10.1080/15397734.2015.1030679.

[45] C. Ott, R. Mukherjee, and Y. Nakamura. Unified impedance and admittance control. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 554–561, 2010. doi: 10.1109/robot.2010.5509861.

[46] A. Albu-Schäffer and G. Hirzinger. Cartesian impedance control techniques for torque controlled light-weight robots. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 657–663, 2002.

[47] A. H. Mason and C. L. MacKenzie. Grip forces when passing an object to a partner. *Experimental brain research*, 163(2):173–287, Nov. 2005. doi:10.1007/s00221-004-2157-x.

[48] B. Siciliano and O. Khatib. *Robotics and the Handbook*. Springer Handbook of Robotics, 1^{nd} edition, 1997.

[49] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard. A human-inspired controller for fluid human-robot handovers. *IEEE-RAS 16th International Conference on Humanoid Robots*, 2016. doi: 10.1109/humanoids.2016.7803296.

[50] I. Kim and H. Inooka. Hand-over of an object between human and robot. *IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, 1992. doi: 10.1109/roman.1992.253888.

[51] L. Takayama and C. Pantofaru. Influences on proxemic behaviors in human–robot interaction. *Intelligent robots and systems*, pages 5495–5502, 2009. doi: 10.1109/IROS.2009.5354145.

[52] V. M. J. Aleotti and S. Caselli. Comfortable robot to human object hand-over. *Robot and Human Interactive Communication (RO-MAN)*, pages 771–776, 2012. doi: 10.1109/ROMAN.2012.6343845.

[53] E. A. Sisbot and R. Alami. A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(5):1045–1057, 2012.

[54] M. Cakmak, S. S. Srinivasa, M. K. Lee, S. Kiesler, and J. Forlizzi. Using spatial and temporal contrast for fluent robot-human hand-overs. *International Conference on Human-Robot Interaction (HRI)*, pages 489–496, 2011.

[55] K. Koay, E. Sisbot, and D. Syrdal. Exploratory studies of a robot approaching a person in the context of handing over an object. *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, pages 18–24, 2007.

[56] A. Edsinger and C. C. Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. *The 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1167–1172, 2007. doi: 10.1109/roman.2007.4415256.

[57] W. P. Chan, C. A. Parker, H. Loos, M. V. der, and A. E. Croft. Grip forces and load forces in handovers: implications for designing human–robot handover controllers. *Human–Robot Interaction*, pages 9–16, 2012. doi: 10.1145/2157689.2157692.

[58] A. Dragan and S. Srinivasa. Generating legible motion. *Proceedings Robotics: Science and Systems (RSS)*, 2013.

[59] E. C. Grigore, K. Eder, A. G. Pipe, C. Melhuish, and U. Leonards. Joint action understanding improves robot-to-human object handover. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4622–4629, 2013.

[60] N. Hendrich, H. Bistry, J. Liebrecht, and J. Zhang. Natural robot-human handover combining force and tactile sensor. *Workshop on Assistive and Service Robotics, IROS*, 2014.

[61] K. W. Strabala, M. K. Lee, J. L. F. A. D. Dragan, S. Srinivasa, M. Cakmak, and V. Micelli. Towards seamless human-robot handovers. *Human–robot interaction*, 2(1):112–132, 2013. doi: 10.5898/jhri.2.1.strabala.

[62] K. Nagata, Y. Oosaki, M. Kakikura, , and H. Tsukune. Delivery by hand between human and robot based on fingertip force-torque information. *Intelligent robots and systems*, 2(1):750–757, 1998. doi: 10.1109/IROS.1998.727283.

[63] J. Bohren. Towards autonomous robotic butlers: Lessons learned with the pr2. *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 5568–5575, 2011.

[64] V. Micelli, K. Strabala, and S. S. Srinivasa. Perception and control challenges for effective human robot handoffs. *Conference in Robotics: Science and Systems (RSS)*, 2011.

[65] M. Prada, A. Remazeilles, A. Koene, and S. Endo. Dynamic movement primitives for human-robot interaction: Comparison with human behavioral observation. *International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[66] A. Kupcsik, D. Hsu, and W. S. Lee. Learning dynamic robot-to-human object handover from human feedback. *Springer Proceedings in Advanced Robotics Robotics Research*, pages 161–176, 2017. doi: 10.1007/978-3-319-51532-8-10.

[67] W. P. Chan, I. Kumagai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Implementation of a robot-human object handover controller on a compliant underactuated hand using joint position error measurements for grip force and load force estimations. *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. doi: 10.1109/icra.2014.6907004.

[68] J. Hermsdörfer. Moving weightless objects grip force control during microgravity. *Experimental Brain Research*, 132(1):52–64, 2000.

[69] H. Clark. Using language. *Cambridge University Press*, 1996.

[70] A. Mitrevski and P. G. Plöger. Reusable specification of state machines for rapid robot functionality prototyping. *RoboCup 2019: Robot World Cup XXIII Lecture Notes in Computer Science*, pages 408–417, 2019. doi: 10.1007/978-3-030-35699-6-32.

[71] V. Lippiello and F. Ruggiero. Cartesian impedance control of a uav with a robotic arm. *IFAC Proceedings Volumes*, 45(22):704–709, 2012. doi: 10.3182/20120905-3-hr-2030.00158.

[72] R. B. Miller. Response time in man-computer conversational transactions. *International Business Machines Corporation*, pages 267–277, 1968.

[73] D. K. Hsu and W. S. Lee. Learning dynamic robot-to-human object handover from human feedback. *Springer Proceedings in Advanced Robotics Research*, page 161–176, 2017.

[74] G. M. Gasparri, F. Fabiani, M. Garabini, L. Pallottino, G. G. M. Catalano, R. Persichin, and A. Bicchi. Robust optimization of system compliance for physical interaction in uncertain scenarios. *IEEE-RAS 16th International Conference on Humanoid Robots*, 2016.

[75] G. Hoffman. Evaluating fluency in human-robot collaboration. *IEEE Transactions on Human-Machine Systems*, 49(3):209–218, June 2019.

[76] C. M. Huang, M. Cakmak, and B. Mutlu. Adaptive coordination strategies for human-robot handovers. *Robotics: Science and Systems XI (RSS)*, 2015.

[77] M. Koene, S. Endo, A. Remazeilles, M. Prada, and A. M. Wing. Experimental testing of the coglaboration prototype system for fluent human-robot object handover interactions. *The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2014.

Appendix A

Impedance Model Derivation

Derivation of the impedance dynamic relationship between the end-effector motion in the inertial Cartesian coordinates and external generalized forces with resource to the derived dynamics of the system in the generalized joint space and to the proposed control law.

Substituting (3.25) into (3.23) yields:

$$B(\xi)\,\ddot{\xi} + C(\xi,\dot{\xi})\,\dot{\xi} = J^T\,(-k_B\,\dot{x} + k_D\,\tilde{x}) + u_{ext} \tag{A.1}$$

Multiplying both sides by J^{-T} gives:

$$J^{-T}(B(\xi)\,\ddot{\xi} + C(\xi,\dot{\xi})\,\dot{\xi}) = \underbrace{J^{-T}J^T}_{I}(-k_B\,\dot{x} + k_D\,\tilde{x}) + u_{ext}J^{-T} \tag{A.2}$$

Re-arranging (3.10) and (3.13) in order to $\dot{\xi}$ and $\ddot{\xi}$, respectively:

$$\dot{\xi} = J^{-1}\dot{x} \tag{A.3}$$

and

$$\ddot{\xi} = J^{-1}(\ddot{x} - \dot{J}\dot{\xi}) \tag{A.4}$$

Substituting (A.3) into (A.4) gives:

$$\ddot{\xi} = J^{-1}(\ddot{x} - \dot{J}J^{-1}\dot{x}) \tag{A.5}$$

Knowing (A.3) and (A.5), (A.2) can be expressed by:

$$J^{-T}\ B(\xi)\ [J^{-1}(\ddot{x} - \dot{J}J^{-1}\dot{x})] + J^{-T}\ C(\xi,\dot{\xi})\ J^{-1}\ \dot{x} = -k_B\ \dot{x} + k_D\ \tilde{x} + u_{ext}\ J^{-T} \Leftrightarrow$$

$$\Leftrightarrow J^{-T}\ B(\xi)\ J^{-1}\ \ddot{x} - [J^{-T}\ B(\xi)\ \dot{J}\ J^{-1}\ J^{-1}\ \dot{x}] + J^{-T}\ C(\xi,\dot{\xi})\ J^{-1}\ \dot{x} = -k_B\ \dot{x} + k_D\ \tilde{x} + u_{ext}\ J^{-T} \Leftrightarrow$$

$$\Leftrightarrow J^{-T}\ B(\xi)\ J^{-1}\ \ddot{x} + [-J^{-T}\ B(\xi)\ \dot{J}\ J^{-1}\ J^{-1} + J^{-T}\ C(\xi,\dot{\xi})\ J^{-1}]\ \dot{x} = -k_B\ \dot{x} + k_D\ \tilde{x} + u_{ext}\ J^{-T} \Leftrightarrow$$

$$\Leftrightarrow \underbrace{J^{-T}\ B(\xi)\ J^{-1}}_{B_x}\ \ddot{x} + \underbrace{J^{-T}\ [C(\xi,\dot{\xi}) - B(\xi)\ J^{-1}\ \dot{J}]\ J^{-1}}_{C_x}\ \dot{x} = -k_B\ \dot{x} + k_D\ \tilde{x} + u_{ext}\ J^{-T} \Leftrightarrow$$

$$\Leftrightarrow B_x\ \ddot{x} + C_x\ \dot{x} = -k_B\ \dot{x}\ +\ k_D\ \tilde{x} + f_{ext} \Leftrightarrow$$

$$\Leftrightarrow B_x\ \ddot{x} + C_x\ \dot{x} + k_B\ \dot{x}\ -\ k_D\ \tilde{x} = f_{ext} \Leftrightarrow$$

$$\Leftrightarrow B_x\ \ddot{x} + (C_x + k_B)\ \dot{x}\ -\ k_D\ \tilde{x} = f_{ext}$$

(A.6)

As proposed, an impedance dynamic model was derived.

Appendix B

User Study - Instructions

This experimental study is integrated in my MSc degree thesis in “Object handover between humans and robots in microgravity” in Aerospace Engineering.

In this study you’ll be able to control a hand model on a simulation through a Leap Motion device (sensor). Your main goal is to give/receive an object to/from the robot presented in a simulation.

The first phase of the study is a familiarization with the simulated hand using the Leap Motion sensor. You will have 10 minutes to play freely with the hand and to do any movements of your choice like doing a fist, closing only one or more fingers at a time and moving or rotating the hand in the virtual space. Furthermore, an object is placed on the simulation environment and you can grab it.

The second phase is the tasks performed. The paragraphs below describe the three tasks considered in the study:

- Robot-to-human handover – The robot will approach the simulated hand with an object in the gripper It will stop at a pre-defined location and give a signal (flashlight on and off once) when it is ready to make the handover. You must approach the object, grab it, wait for the gripper to open and move away from the robot with the object in the virtual hand.
- Human-to-robot handover - The robot will approach the simulated hand at a pre-defined location. When there, the robot will open the gripper and as in the previous described task, it will give a signal (flashlight on and off once) when it is ready to perform the handover. You must hand the object to the robot gripper. It is important to touch the gripper with the object for it to close. Lastly move the virtual hand away from the robot.
- Collaborative Task – Firstly a robot-to-human handover will be made like previously described but after receiving the object, the robot will not leave and it will wait again for the object like described in human-to-robot handover so when it’s ready to receive it will give the flashlight signal. After giving the object you must move the hand away from the robot.

In total, 12 interactions will take place in the present experiment, 4 trials for each task described previously.

After each interaction, a questionnaires will be given.

Thank you for your participation,
Adriana Fernandes