Thèse n° 8847

Control of Contact Tasks in Autonomous and Human-Robot Collaborative Scenarios: A Dynamical System Approach

Présentée le 10 décembre 2021

Faculté des sciences et techniques de l'ingénieur
Laboratoire d'algorithmes et systèmes d'apprentissage
Programme doctoral en robotique, contrôle et systèmes intelligents

pour l'obtention du grade de Docteur ès Sciences

par

Walid AMANHOUD

Acceptée sur proposition du jury

Prof. F. Mondada, président du jury
Prof. A. Billard, directrice de thèse
Prof. P. Robuffo Giordano, rapporteur
Prof. F. Ferraguti, rapporteuse
Dr R. Boulic, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

"L'espérance brille dans la victoire, et il n'y a pas de victoire sans combat."
— Charles Péguy

À mes parents et mon frère

Acknowledgements

This thesis would not have been possible without the support and help of many great people.

First of all, I would like to thank my thesis supervisor Prof. Aude Billard for giving me the opportunity to join her lab. Thanks for the advices and feedback provided throughout the thesis, in particular during the writing of the papers, the thesis, and the preparation of the oral exam.

I would like to express my gratitude to the members of the jury who participated to the private defense: Dr. Ronan Boulic, Prof. Francesco Mondada, Prof. Paolo Giordanno, and Prof. Federica Ferraguti. Thanks for the valuable feedback received during the exam.

I would like to extend my gratitude to Dr. Mohamed Bouri for our collaboration in the Hasler project. Thanks for having always been involved and motivated during this project.

I would also like to thank the secretary of the lab Joanna Erfani and the secretary of the doctoral program Corinne Lebet who both took care of the administrative tasks during the PhD. A special thanks goes to Corinne for her support and encouragement since the beginning of this journey.

During the PhD, there are two things that I particularly loved. The first one is working with robots and we are lucky to be well equipped in LASA for that. The second one is my colleagues. I was fortunate enough to meet only brilliant people at LASA who all brought something to me and whom I consider like friends. Therefore, I would like to express my gratitude to everyone I have met since my first day in the lab. Obviously, during four years and a half, among these people, there were some who were more important than others and I would like to highlight them now. First of all, I would like to thank my first officemate Mahdi Khoramshahi who took me under his wing when I joined the lab. Mahdi was the mentor I needed at the beginning to put me on the right way and I appreciated a lot collaborating with him. I would also like to thank Bernardo Fichera, Farshad Khadivar, Jacob Hernandez Sanchez, and Leonardo Urbano who were like brothers to me in the lab. Their presence was great to have and I really loved all the time we spent together whether inside or outside of work. A special thanks goes to Jacob for our collaboration in the Hasler project, in particular for his enthusiasm and smart advices. I am also grateful to Iason Batzianoulis who was like a big brother to me in

the lab. Thanks for the encouragement and precious advices during all these years. Finally, I would like to thank Carolina Correia for being so kind and supportive since she joined the lab.

At this stage, I would like to acknowledge all the students who participated to a project with me: Maxime Bonnesoeur, Sthithpragya Gupta, Arthur Ponchon, Caroline Savio. A special thanks goes to Maxime who contributed to my PhD research.

Outside LASA, I would like to sincerely thank my friends Thibault Asselborn and Benoit Desbiolles for their support not only during the PhD but since we started together the bachelor at EPFL in 2010. Besides being brilliant people, Thibault and Benoit are the kind of person who pushes you to move forward despite the difficulties and to always give the best of yourself.

I would also like to thank my two closest friends in France, Ali Arshed and Fabien Bouillaguet. Thanks for the encouragement during the PhD and for the valuable and sincere friendship during all these years since we met together in secondary school.

Finally, and most importantly, I would like to deeply thank my family and more specifically my brother and parents for the endless support throughout the PhD. There are no words to express how grateful I am to my parents for everything they have done for me. In particular, for having always tried to make my life easier and allow me to fully focus on my studies. They are my models in life, models of courage, abnegation, and success.

Lausanne, November 8, 2021

Walid Amanhoud

Abstract

The ability of robots to control forces when interacting with their environment is a major topic in robotics. Especially nowadays where robots are asked to perform contact tasks in unstructured environment while guaranteeing safe human-robot interaction and collaboration. Although the design of new compliant robots (mechanically-generated) is indeed a first step towards this safe collaborative environment, it should be accompanied with new control strategies that exploit the new capabilities of the novel robot design. In this thesis, we propose to develop such control strategies by exploiting the time-invariant Dynamical System (DS) framework.

Dynamical systems have been used extensively to generate robot motions. In the first part of the thesis, we present a modulation of a dynamical system that enables the generation of desired contact forces in addition to the robot trajectories. We evaluate this strategy in two scenarios; 1) a polishing task on a non-flat surface and 2) a bi-manual reaching, grasping and manipulation of an object. The results demonstrate the ability of the controller to generate a smooth contact with both the surface and the object, track the desired contact force and be robust to external disturbances, such as those exerted by humans.

Focusing further on the robot's interaction with the environment, we investigate the tracking of the contact forces, a critical component of any contact tasks. Uncertainties, coming either from the robot's dynamics or the environment, can limit the force tracking accuracy and potentially affect the task execution. In the second part of the thesis, we propose to address this limitation through online adaptation of a state-dependent force correction model. We evaluate this method on a polishing task with various levels of contact forces and robot velocities. By integrating our method with existing works, we also showcase a collaborative cleaning task where force, motion, and task adaptation occur at the same time. Due to its reactivity and adaptability, we highlight the ability of the DS framework to be used in collaborative environments subject to uncertainties.

In the last two parts of the thesis, we focus on collaborative teleoperation scenarios where supernumerary robotic arms, controlled through foot interfaces, assist a human in performing tasks that would require more than two hands. More specifically, in the third part, we consider an industrial scenario, where two robotics arms assist a human operator in supporting and moving an object while the human/user is free to work on the object with his hands. To ease the control of the robotic arms, shared control strategies are derived from the design of two assistance modalities enabled upon contact with the object: autonomous contact force generation and auto-coordination of the robotic arms. In the fourth and final part, we consider a laparoscopic surgical scenario where two robotic assistants, one holding a camera, the other holding a retractor, assist a surgeon during the operation. We present a control framework to ensure safe collaboration with the surgeon including an inverse kinematics solver to satisfy all safety restrictions and task-related constraint, an assistance modality to ease the control of the camera through the foot and robot-to-human haptic cues to increase the surgeon situational awareness. For both scenarios, user studies are conducted to evaluate the proposed control strategies.

Keywords: Dynamical Systems, Force control, Physical Human-Robot Interaction, Human-Robot Collaboration, Shared Control, Supernumerary Robotic, Robotic Surgery.

Résumé

La capacité des robots à contrôler les forces de contact lorsqu'ils interagissent avec leur environnement est un sujet majeur en robotique. En particulier de nos jours, où les robots doivent réaliser des tâches en contact dans des environnements non structurés tout en garantissant une interaction et une collaboration en toute securité avec l'humain. Bien que la conception de nouveaux robots (mécaniquement) conformes soit un premier pas vers cet environnement collaboratif sûr, elle doit s'accompagner de nouvelles stratégies de contrôle qui exploitent les nouvelles capacités des robots. Dans cette thèse, nous proposons de développer de telles stratégies de contrôle en exploitant le cadre des systèmes dynamiques (DS) invariant dans le temps.

Les systèmes dynamiques ont été largement utilisés pour générer les mouvements de robot. Dans la première partie de la thèse, nous présentons une modulation d'un système dynamique qui permet la génération de forces de contact désirées en plus des trajectoires du robot. Nous évaluons cette stratégie dans deux scénarios ; 1) une tâche de polissage sur une surface non plane et 2) l'atteinte, la saisie et la manipulation bi-manuelle d'un objet. Les résultats démontrent la capacité du contrôleur à générer un contact en douceur à la fois avec la surface et l'objet, à suivre la force de contact souhaitée et à être résistant aux perturbations externes, telles que celles exercées par les humains.

Se focalisant davantage sur l'interaction du robot avec l'environnement, nous étudions le suivi des forces de contact, un élément essentiel de toute tâche en contact. Les incertitudes, provenant soit de la dynamique du robot ou de l'environnement, peuvent limiter la précision du suivi de la force et potentiellement affecter l'exécution de la tâche. Dans la deuxième partie de la thèse, nous proposons d'aborder cette limitation par l'adaptation en ligne d'un modèle de correction de force dépendant de l'état (la position) du robot. Nous évaluons cette méthode sur une tâche de polissage avec différents niveaux de forces de contact et de vitesses de robot. En intégrant notre méthode à des travaux existants, nous présentons également une tâche de nettoyage collaborative où la force, le mouvement et l'adaptation de la tâche se produisent en même temps. De par sa réactivité et son adaptabilité, nous soulignons la capacité des systèmes dynamiques à être utilisés dans des environnements collaboratifs soumis à des incertitudes.

Dans les deux dernières parties de la thèse, nous nous concentrons sur des scénarios de téléopération collaboratifs où des bras robotiques surnuméraires, contrôlés via des interfaces de pied, aident un humain à effectuer des tâches qui nécessiteraient plus de deux mains. Plus précisément, dans la troisième partie, nous considérons un scénario industriel, où deux bras robotiques aident un opérateur humain à soutenir et à déplacer un objet tandis que l'homme/utilisateur est libre de travailler sur l'objet avec ses mains. Pour faciliter le contrôle des bras robotiques, des stratégies de contrôle partagées sont construites à partir de la conception de deux modalités d'assistance activées au contact de l'objet : la génération autonome de force de contact et l'auto-coordination des bras robotiques. Dans la quatrième et dernière partie, nous considérons un scénario de chirurgie laparoscopique où deux assistants robotiques, l'un tenant une caméra, l'autre une pince, assistent un chirurgien pendant l'opération. Nous présentons un cadre de contrôle pour assurer une collaboration sûre avec le chirurgien incluant un solveur de cinématique inverse satisfaisant toutes les contraintes liées à la tâche et la sécurité, une modalité d'assistance pour faciliter le contrôle de la caméra via le pied, et des retours haptiques de robot à humain pour augmenter l'appréciation de la situation du chirurgien. Pour les deux scénarios, des études utilisateurs sont menées pour évaluer les stratégies de contrôle proposées.

Mots-clés : Systèmes Dynamiques, Contrôle en Force, Interaction Physique Homme-Robot, Collaboration Homme-Robot, Contrôle Partagé, Robotique Surnuméraire, Chirurgie Robotique.

Contents

Acknowledgements **iii**

Abstract **v**

1 Introduction **1**

1.1 Motivation . . . 1
1.2 A Dynamical System Approach . . . 3
1.3 Thesis Contributions . . . 4
1.4 Force Control in Contact Tasks . . . 4
1.5 Teleoperated Shared Control . . . 7
1.6 Robotic Assisted Laparoscopic Surgery . . . 10
1.7 Thesis Outline . . . 13
1.8 Publications, Multimedia, and Source code . . . 14

2 Technical Background **16**

2.1 Dynamical Systems . . . 16
2.2 Robot Motion Control with Dynamical Systems . . . 17
    2.2.1 Robot Dynamics . . . 17
    2.2.2 Dynamical Systems as Motion Generators . . . 19
    2.2.3 DS-based Impedance Controller . . . 20
2.3 Task-space Orientation Control . . . 22

3 Motion and Contact Force Generation with Dynamical Systems **23**

3.1 Introduction . . . 23
3.2 Method . . . 24
    3.2.1 Nominal DS Definition . . . 24
    3.2.2 Force-based DS Modulation . . . 24
    3.2.3 Ensuring Passivity . . . 25
3.3 Extension to a Bi-Manual Reaching, Grasping, and Manipulation Task . . . 27
    3.3.1 Formulation . . . 27
    3.3.2 Passivity Analysis . . . 29
3.4 Experimental Evaluations . . . 30
    3.4.1 Polishing Task on a Non-Flat Surface . . . 31
    3.4.2 Bi-Manual Reaching, Grasping and Manipulation Tasks . . . 34

3.4.2.1 Reaching and Grasping an Object 34
3.4.2.2 Reaching, Grasping, and Moving an Object 35
3.5 Discussion and Conclusion 37

4 Contact Force Adaptation in Uncertain Environments 38
4.1 Introduction 38
4.2 Method 39
4.2.1 Force Adaptation with Dynamical Systems 39
4.2.2 Design of the State-Dependent Force Correction Model 40
4.2.3 Convergence Behaviour 40
4.3 Illustrative Example in Simulation 41
4.4 Experimental Evaluations 43
4.4.1 Force Adaptation on a Non-Flat Surface 43
4.4.2 Collaborative Cleaning of a Non-Flat Surface 46
4.5 Discussion and Conclusion 50

5 Contact-Initiated Shared Control Strategies for Four-Arm Manipulation 51
5.1 Introduction 51
5.2 Method 53
5.2.1 Control Strategies Overview 53
5.2.2 Uncoordinated Robot Control Without Force Assistance (strategy *U*) . . 55
5.2.3 Uncoordinated Robot Control With Force Assistance (strategy *U*+*F*) . . . 56
5.2.4 Coordinated Robot control With Force Assistance (strategy *C*+*F*) 57
5.2.5 Coordinated Robot Control Without Force Assistance (strategy *C*) 60
5.3 Experimental Evaluations 60
5.3.1 Experimental Setup and Task 60
5.3.2 Evaluation Metrics and Statistical Studies 62
5.4 Results 65
5.5 Discussion and Conclusion 68

6 Four-Arm Robotic Assisted Laparoscopic Surgery 70
6.1 Introduction 70
6.2 Four-Arm Robotic Assisted Surgical System 71
6.3 Method 72
6.3.1 Control Framework Overview 72
6.3.2 Robotic Arms’ Control Law 74
6.3.3 QP-based Inverse Kinematics 75
6.3.4 Task Planning 82
6.3.4.1 Desired Tools’ Tip Dynamics 82
6.3.4.2 Description of the Control Phases of the Robotic Tools 82
6.3.4.3 Reference Camera Motion Dynamics 83
6.3.4.4 Camera Assistance: Adaptive Autonomous Tool Tip Following . 84
6.3.4.5 Reference Gripper Motion Dynamics 85

6.3.4.6 Reference Admittance Velocity Computation 86
6.3.4.7 Fulcrum Effect Consideration 87
6.3.4.8 Task Modulation Gain Computation 88
6.3.5 Haptic Cues Design 88
6.3.5.1 Guidance Wrench Computation 89
6.3.5.2 Haptic Guidance Torques Computation 91
6.4 Experimental Evaluations 91
6.4.1 User Study 91
6.4.1.1 Tasks Description 91
6.4.1.2 Protocol 94
6.4.1.3 Evaluation Metrics and Statistical Studies 94
6.4.1.4 Results 96
6.4.2 Control Framework Validation 97
6.4.2.1 Trocar and Tool Tip Task Constraints Achievement 97
6.4.2.2 Compliance to Physical Human Interaction 98
6.4.2.3 Workspace Collision Avoidance 101
6.4.2.4 Tool Collision Avoidance 103
6.4.2.5 End-effector Collision Avoidance 105
6.4.2.6 Camera's Adaptive Tool Tip Following Assistance 107
6.5 Discussion and Conclusion 109

7 Conclusion 111
7.1 Main Contributions 111
7.2 Outlook 112
7.2.1 Chapter 3 112
7.2.2 Chapter 4 113
7.2.3 Chapter 5 113
7.2.4 Chapter 6 114

A Appendix Chapter 3 116
A.1 Common Experimental Details 116
A.2 Polishing Task on a Non-Flat Surface 117
A.3 Bi-manual Reaching and Grasping Task 120
A.4 Bi-manual Reaching, Grasping, and Manipulation Task 120

B Appendix Chapter 4 122

C Master Haptic Foot Interface 123

D Appendix Chapter 5 125
D.1 Experimental Details 125
D.2 Results of the Statistical Studies 128

E Appendix Chapter 6 134

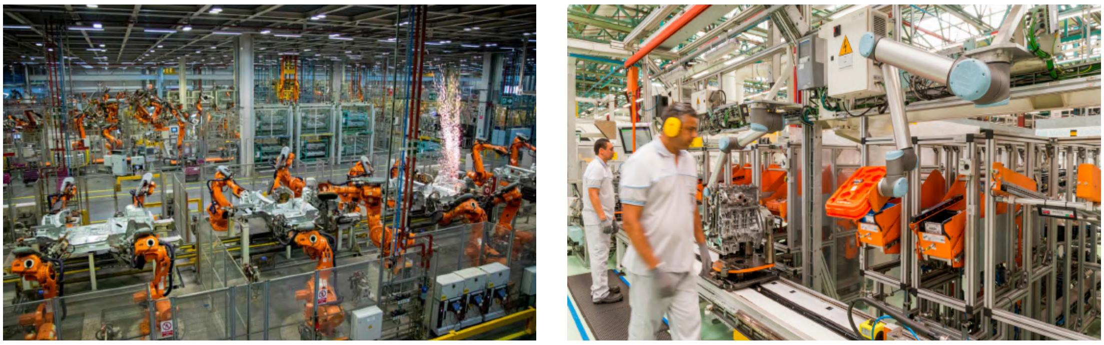
E.1 Camera-Based Tool Tip Tracking Strategy 134
E.2 Technical Details 135
E.3 Results of the Statistical Studies 138

Bibliography **144**

Curriculum Vitae **154**

1 Introduction

1.1 Motivation



(a) ©The Guardian (b) ©FCA Group

Figure 1.1 – Examples of a traditional (Figure 1.1a) and collaborative (Figure 1.1b) robotic assembly line. In Figure 1.1b, the human operators can safely interact with the robots which assist their human co-workers in the delivery and assembly operations of parts.

With respect to the nature of the interaction between a robot and its environment, robotic tasks can be classified into *non-contact tasks* and *contact tasks* (Vukobratovic (2009)). While non-contact tasks require a robot to move freely in its environment in an unconstrained manner, contact tasks require the robot to make a physical contact with its environment and eventually exert specific *contact forces*. Physically interacting with the environment remains one of the main challenges in robotics especially nowadays where robots are asked to operate in unstructured environment. For many years, robots were mainly used in structured environment where everything is defined and known (see Figure 1.1a). This is still the case today, in the manufacturing industry for instance, where robots are often pre-programmed to perform repetitive and cumbersome tasks such as pick and place or polishing. This way of using robots is not possible anymore in unstructured environment where contact tasks should be performed under the uncertainty of the real world (e.g., uncertainties in the location of the objects, surfaces to interact with, the presence of humans etc.). While in structured environment robots are stiff and rigid to deliver the required precision (i.e., they are controlled in position at the low level), in unstructured environment, we need *compliant* robots; i.e., robots that are able to react to interaction forces. Compliance capabilities can be obtained through a force feedback to control the interaction forces and achieve a proper reaction behaviour. Force feedback is not only useful at the contact point but also on the full robot (i.e., at the different links) to provide full body compliance which can be obtained with torque sensing. These last years, the new generation of robots is equipped with such sensing capabilities such as the KUKA or Franka robots, allowing to perform interaction, manipulation, and cooperation tasks while providing safety and performance. However, these robots are not always used at their full capabilities and remain often controlled at the position level. Another aspect about physically interacting with an unstructured environment is that robots are still far from full autonomy, i.e., humans need to be in the loop. Having humans working with robots is desirable for many reasons such as to perform collaborative tasks (i.e., robotic assistance), help in the decision making process, or for safety regulations etc. There are many possibilities for a human to interact with a robot such as speech, physical interaction or haptics. Effective human robot collaboration should ensure safety of the interaction, allow a robot to understand and interpret the human intention and vice versa, and provide intuitive control strategies to combine the human input with the robot commands to achieve the collaborative task (see Figure 1.1b).

This evolution toward more complex robotic tasks involving collaboration with humans and physical interaction with environments that are less structured and predictable is part of industry 4.0 and motivates the main objective of this thesis: *Developing control strategies to perform contact tasks in unstructured environment with/without human collaboration.*

To this end, several scenarios are considered in this thesis where robotic manipulators have to physically interact with objects or surfaces, while interacting and/or collaborating with humans in uncertain environments. More specifically, the first half of the thesis focuses on autonomous contact task scenarios such as one robot polishing/cleaning a surface or two robots reaching, grasping, and manipulating an object. The robots should come in contact with the target surfaces, apply desired contact forces while moving on the surfaces, and be robust to large disturbances such as those introduced by humans. This robustness should translate into compliant, reactive, and adaptive robot behaviours in response to the disturbances. The second half of the thesis is dedicated to collaborative scenarios involving supernumerary robotics arms teleoperated with the feet of the human. It participates into a new trend in human-robot interaction/collaboration aiming to augment human capabilities through supernumerary devices and allow humans to perform tasks with higher degree of precision and/or tasks requiring more than one person (see Figure 1.2). In this thesis, two scenarios are considered where two robotic manipulators supplement the two human biological arms resulting into four-arm collaboration; one industrial scenario where the robots assist a human into supporting and moving an object, and one surgical scenario where the robots assist a surgeon during laparoscopic surgery. In both scenarios, the robots should be provided with partial autonomy to ease the control through the feet. In addition, they should understand the human intent, act proactively under the supervision of the human while being robust to disturbances from the environment. To achieve the aforementioned scenarios and attain the main thesis objective, we take an approach based on the *time-invariant Dynamical System (DS) framework*.

Figure 1.2 – Laparoscopic surgery as a real-life example where multiple hands are needed (©brisbanesurgeon).

1.2 A Dynamical System Approach

One of the problems in traditional task planning algorithms is the assumption that the environment is fixed and that the robot cannot be disturbed during the execution of the task. This is an invalid assumption if we want robots to operate in uncertain environment, interact and collaborate with humans. For that, it is required to have a task planning strategy that is intrinsically robust to uncertainties and changes in the environment. Time-invariant dynamical systems has emerged as a general and flexible solution to that problem. They have been particularly used in the literature to model (e.g., from human demonstrations; Khansari-Zadeh and Billard (2011); Neumann et al. (2013)) and generate robot motions (Koditschek (1989); Rimon (1990); Sanner and Slotine (1995); Feder and Slotine (1997); Khansari-Zadeh et al. (2012); Mirrazavi Salehian et al. (2016)). Given the position of the robot and a target to reach, time-invariant dynamical systems embed in one function, that is strictly state-dependent, all possible solutions to reach a target. Controllers driven by such DS are robust to real-time disturbances as they provide on-the-fly correction and adaptation to changes in the environment making them suitable to operate in unstructured environment (Kronander and Billard (2015)). Furthermore, the framework offers great adaptation capabilities suitable to provide adaptive motion generation. This is achieved through external signals modulating the dynamics of the DS; e.g., Huber et al. (2019) achieves obstacle avoidance by modulating a nominal DS based on the distance to the obstacles, while Khoramshahi et al. (2018) and Khoramshahi and Billard (2019) use the tracking error resulting from physical human interaction to respectively refine a task and switch from one task to another. Note that, contrary to other task planners, a DS does not usually generate optimal trajectories (e.g., with minimum time, distance). However, if trained with trajectories obtained from optimal control, it could embed optimal path. We decided to adopt this framework in this thesis as we are more interested in its robustness to real-time disturbances, in particular in its reactivity and adaptability.

1.3 Thesis Contributions

Our *first contribution* extends the use of dynamical systems to enable generating contact forces in addition to motion through the DS. The idea is to use a desired contact force profile to modulate a nominal DS responsible of motion, and use the modulated DS to drive the robot. We show that this simple strategy allows to perform contact tasks with reasonable force tracking accuracy while offering robustness to large environmental or human applied disturbances such as breaking the contact with a surface/object. In tasks where accurate force tracking is required, we propose, in a *second contribution*, online adaptation of a state-dependant force correction model to counteract the uncertainties due to the surface geometry or robot dynamics for example, that limit the force tracking accuracy. By integrating such a model inside our DS modulation strategy, we can increase the force tracking accuracy while preserving the reactivity to real-time disturbances. In a *third contribution*, we propose shared control strategies for a four-arm manipulation task where two supernumerary robotic arms assist a human in supporting and moving an object while the human is working on the object with the hands. To this end, the two robotic arms are partially controlled with the feet of the human through haptic foot interfaces, while the shared control strategies are derived from our first contribution using the time-invariant DS framework. In a *fourth and last contribution*, we target solo laparoscopic surgery, where two supernumerary robotic arms, also controlled through haptic foot interfaces, assist a surgeon during the operation. A control framework is presented to achieve all the task and safety-related constraints and shared control strategies are designed to ease the control of the robotic assistants and increase the situational awareness of the surgeon. In the next sections, we review the literature related to the contributions above.

1.4 Force Control in Contact Tasks

Force control is a major topic in robotics especially nowadays where various tasks require robots to interact with humans and/or arbitrary surfaces, in dynamically changing environments, such as in object manipulation, surface operation or tele-manipulation in surgery. Besides following accurately the desired motion and force profile, robots should also be compliant to deal with inaccurate modelling of the surfaces (e.g., stiffness and location) and real-time disturbances such as those introduced by humans or unexpected changes in the environment (Lefebvre et al. (2005)). This requires the development of appropriate control techniques to regulate the robot movement and interaction forces with the environment. From a historical perspective, approaches to force control can be divided into two broad categories, namely *direct force control* where a target force (and position) is achieved through a force feedback loop and *indirect force control* where interaction forces are controlled through an impedance control law (Villani and De Schutter (2016)).

Direct force control is still very much influenced by the hybrid position/force control (Raibert and Craig (1981)). This approach decomposes the task in two orthogonal decoupled subspaces, that are task-specific (Mason (1981)), and where position and force are controlled separately. Force is controlled along the constrained directions imposed by the environment while position is controlled in the unconstrained ones. A selection matrix is usually used to define which degrees of freedom (DOFs) are either position or force controlled which implies an explicit knowledge of the interaction tasks. Transition from free motion to motion in contact is performed in stages. In free motion, a position controller controls the robot in all control directions. Once the robot is in contact, the position controller switches to the hybrid force/position controller. The strength of such approach is that it ensures accurate position and force tracking. However, by neglecting the interaction dynamics between the robot and the environment, it lacks robustness in the face of disturbances and uncertainties, and may lead to instabilities at contact or contact loss (Vukobratovic (2009)).

Impedance control was offered as a solution to some of these drawbacks (Hogan (1984)). Impedance control belongs to the indirect force control category and was the first control strategy to explicitly consider the interaction dynamics with the environment. The control law is defined by a mechanical mass-spring-damper system simulating the dynamical relationship between motion deviations and interaction forces. The relationship is called impedance when force is generated from motion deviation while it is called admittance when motion is generated from force error. Specific control behaviours can be obtained by modifying the general impedance law. In particular, stiffness (Salisbury (1980)) or damping control (Whitney (1977)) can be respectively achieved when only the pose or velocity deviation is considered in that relationship. The target impedance matrices should be chosen in accordance with the given task. Low stiffness is usually needed in directions where small interaction forces have to be kept while high stiffness is selected where position accuracy is mandatory and the environment is compliant. Impedance controllers are suitable for providing a compliant behavior in all phases of a contact task; i.e., non-contact, contact transition and contact (Hogan (1987)); but are limited in their ability to track forces, mainly due to partial knowledge of the environment (e.g., location and stiffness). To overcome this limitation, two distinct strategies are usually employed in the literature: *impedance and set-point adaptation. Impedance adaptation* adjusts the impedance parameters online (e.g., inertia, damping, and stiffness) to improve tracking in response to force, position, or velocity measurements. This strategy is particularly inspired from the ability of humans to control the force they exert on objects/surfaces by modulating their arm stiffness to compensate for environment forces and instabilities. For instance, Lee and Buss (2008) proposed an impedance controller with a variable stiffness modulated according to the force tracking error. Similarly, Ferraguti et al. (2013) developed a stable impedance controller with variable stiffness, where in the context of a puncturing task, the desired stiffness is adapted depending on the characteristics of the tissues to puncture, thereby imitating surgeons' behaviour. While more recently, Duan et al. (2018) designed a variable impedance controller with zero stiffness where the damping is modulated through adaptive control law based on the force error. *Set-point adaptation* improves force tracking by adjusting the impedance set-point (e.g., reference position) based on force tracking error or on real-time estimation of the environment's change in stiffness. For example, such strategy is exploited in Roveda et al. (2013) for assembly/insertion task in uncertain environment. The set-point is designed to track a desired deformation of the environment indirectly controlling for the interaction force. In Roveda et al. (2016) the set-point of the impedance controller is calculated through an external admittance controller whose gains are optimized for accuracy and stability purposes. An admittance controller is also used in Shahriari et al. (2017) and Kramberger et al. (2018) to adapt a reference position trajectory learned with Dynamic Movement Primitives (DMPs). Finally, other strategies use impedance control, but in combination with direct force control to provide accurate force tracking and a compliant robot behavior in response to external forces (Schindlbeck and Haddadin (2015); Scherzinger et al. (2017)). Even though the aforementioned works provide accurate motion and force tracking performance, their robustness to real-time disturbances is either not mentioned or limited to small disturbances, such as in the surface location or in the environment stiffness (Lee and Buss (2008); Roveda et al. (2013); Shahriari et al. (2017); Kramberger et al. (2018)). Large disturbances such as those introduced by human interactions are often not addressed. For instance, let us consider a scenario where a robot cleans/polishes a non-flat surface in collaboration with a human operator. At any moment, the human might want to change the working area by physically interacting with the robot or stop the task by breaking the contact with the surface. In such situations, the robot should not only damp the disturbances using the impedance control (which is only effective for small disturbances) but also ”react” and ”adapt” its behaviour on the fly. To properly react and adapt, the robot is required to re-plan the execution of the task from the disturbed state and modify the task based on the interaction with the environment. These reactivity and adaptability need to be continuous, smooth, and robust toward human highly dynamic behaviour and other uncertainties in the environment. Representing tasks with time-indexed references for position and force profiles is the main drawback in current approaches in achieving fast reactivity toward large disturbances; see Shahriari et al. (2017); Kramberger et al. (2018) as examples where a time-dependent representation of the task was used. In contrast, in a state-dependent and time-invariant task representation, interactions with the environment can be captured by changes in the robot's state which can be used in the modification and re-planning of the task. In this thesis, to obtain the reactivity and adaptability needed nowadays to perform contact tasks in unstructured environment where a robot can interact/collaborate with humans, we design control strategies based on the time-invariant dynamical system framework (see chapter 3 and 4).

An important aspect of controlling a robot interacting with unknown environments is to make sure that the interaction is stable for both performance and safety purposes. A sufficient condition to achieve stability is to ensure passivity of the whole system (Van der Schaft and Van Der Schaft (2000)). Passivity is an essential factor to ensure a stable interaction with any passive environment (Stramigioli (2015)). It implies that the system never generates extra energy or in other words, that the total energy of the system is bounded by the initial stored energy plus the one injected in the system from the interaction with the environment. Proposing controllers that ensure passivity with respect to external forces and robot velocity has become a rule of thumb in interaction control in order to get a stable behaviour both in free motion and in contact with the environment. As a mean for this objective, energy tanks first proposed in Secchi et al. (2006) have been quite exploited as they provide a very flexible way to keep track of the energy flow in the system and prevent instabilities coming from some control actions. An energy tank is a reservoir of energy with given initial and maximum (allowed) levels that can be connected to any system. The energy dissipated by the system is used to fill the tank from where energy is extracted to implement potential non-passive actions, without violating the passivity of the whole system (i.e, including the tank). The energy stored in the tank can be freely used as long as the storage is not depleted offering great flexibility to the user. For theses reasons energy tanks have been used in many fields including interaction/force control (Ferraguti et al. (2013); Schindlbeck and Haddadin (2015); Shahriari et al. (2017)) or teleoperation (Franken et al. (2011); Ferraguti et al. (2015b)). In this thesis, we also exploit exploit energy tanks to guarantee passivity of our control approaches.

1.5 Teleoperated Shared Control

The integration of robotic systems in collaborative scenarios has seen an extensive growth over the last two decades. The control of such robotic systems intended for providing assistance can be divided into four main categories, namely, *direct*, *shared*, *supervisory* and *autonomous* control. In direct control, the human fully controls the robot, either by physical contact or via teleoperation. Teleoperation usually involves master-slave systems (Niemeyer et al. (2016)) where master interfaces can range from simple joysticks to voice systems, foot interfaces, or sensors measuring muscular or brain activities. Teleoperation is used in many applications to allow interaction with scaled, hazardous, or inaccessible environments such as for space exploration (Sheridan (1993)), minimally invasive surgery (Guthart and Salisbury (2000)), or search and rescue (Hirche et al. (2003)). In conventional teleoperation, the robot has no autonomy or intelligence. Conversely, autonomous control refers to robots with full autonomy, executing tasks without the need of human intervention (Brett et al. (2007); Parietti et al. (2015)). Robot autonomy has progressed a lot in parallel to the advances in hardware capabilities (e.g, sensors, computational power) and software architectures (e.g., vision algorithms, machine learning, artificial intelligence). While autonomous robots provide satisfactory performance in structured environment, they are still limited in their flexibility and dexterity when operating in uncertain and complex environments; i.e., humans should be in the loop. Minimum human intervention characterizes supervisory control (Niemeyer et al. (2016)), where the human is only involved in the decision-making process. The human provides high level commands to the robot that still completes the (sub-)task autonomously (Shirkhodaie (2002)). In shared control however, the human and the robot are both involved in the execution of the task.

Shared control is often presented as an intermediate level between direct and supervisory control. Shared control teleoperation appears in various forms in the literature, but according to Abbink and Mulder (2010), one can distinguish two main categories: input-mixing shared control and haptic shared control. In the input-mixing category, the inputs of the human and the autonomous system are mixed at the level of the slave controller. They are usually mixed either in an overlapping or a complementary way. The overlapping way is mainly represented by blending strategies where a weighted combination of the inputs defines the arbitration between the human and the autonomous system and allows to achieve an adaptive assistance level (Hansson and Servin (2010); Goil et al. (2013); Storms and Tilbury (2014); Dragan and Srinivasa (2013); Gopinath et al. (2016); Muelling et al. (2017)). Conversely, in the complementary way, the human and the robot control different subspaces of the task without affecting each other. This can be achieved by acting on different subspaces of a task, such as on different degrees of freedom (Abi-Farraj et al. (2016)) or using nullspace control strategies exploiting robot redundancies (Wang et al. (2015)). Haptic shared control follows a different approach where the human and the robot share the control of a task, and the human is assisted by applying forces at the master device. Through a haptic interaction, the operator can be fully aware of the robot's intentions and express her/his own control intentions by choosing to conform or not with the robot. The concept of virtual fixtures introduced by Rosenberg (1993) is often considered as the first instance of haptic shraed control in the literature. Virtual fixtures can be seen as virtual elements, like walls, velocity fields, or tubes providing passive/active guidance to the human during the task execution (e.g., to follow a desired trajectory). This work aroused a high interest in the field of teleoperation leading to further research in the use of virtual guidance (Selvaggio et al. (2016); Abi-Farraj et al. (2016, 2018)). In particular, Boessenkool et al. (2012) highlighted the benefits of haptic shared control in terms of task performance (e.g. time completion, accuracy, contact force), control effort and cognitive workload, especially in free-motion phases. However, these benefits usually rely on an accurate task model and sensors' information. Inaccuracies between the task model and the real environment can lower the performance of the shared control system, especially in constrained contact tasks (Smisek et al. (2015), van Oosterhout et al. (2015)).

As shown through the aforementioned works, various shared control strategies have been proposed in the literature. They show that developing a proper strategy seems to be mainly task-dependent (Passenberg et al. (2010)). However, including the human aspect is also essential when designing the strategies. In particular, criteria such as intuitiveness, fatigue, cognitive load, or comfort have to be considered. This is particularly true in cases where the human should be directly involved in the execution of the task (e.g., through physical interaction) while controlling the slave arms, and/or control a complex system with multiple degrees of freedom such as a multi-arm system. In this thesis, we are interested in such collaborative scenarios, and more specifically in four-arm collaborative tasks where the two human arms and two supernumerary robotics arms are involved in the execution of the tasks. To control the robotics arms, we explore the use of the feet to offer additional degrees of freedom extra to those of the human hands and achieve a richer human-robot collaboration. Foot interfaces are indeed less prone to distractions or interruptions that can cause the use of the head, speech or eyes-based interfaces, and can provide more natural mapping (Punt et al. (2005); Yang et al. (2020)). Also, several studies demonstrated the possibility of controlling foot and hands simultaneously (Abdi et al. (2015, 2016); Yang et al. (2020)). The use of foot interfaces to control additional robotic arms to those of the human is expected to increase cognitive load and fatigue. Therefore, in this thesis, we devise shared control strategies to ease the control of the robots though the feet (see the results of the user studies conducted in sections 5.4 and 6.4.1.4).

In chapter 5, we focus on an industrial application where two robotic arms assist a worker in supporting and moving an object while the worker is working on the object. Shared control teleoperation of a multi-arm system for manipulation tasks has been studied in a few works in the literature, especially recently such as in Ferraguti et al. (2015a) and Selvaggio et al. (2018), where input mixing and/or haptic shared control strategies are used to control a dual-arm system. Multi-arm system can also benefit from shared control to enable coordination behaviours between the arms, which is particularly suitable for symmetric tasks such as supporting and moving a load. This is the case in Laghi et al. (2018) where a dual-arm system, teleoperated with hands' gestures, switches to a coordinated behaviour controlled with one hand, once a particular gesture is detected. Similarly, in Rakita et al. (2019), assistance is provided to perform various coordinated tasks with the arms of a humanoid robot teleoperated with the arms of the user. The assistance is activated when the intention to perform a certain type of coordinated task is recognized from both hands' motion (based on a predefined bi-manual action vocabulary). The coordination strategies proposed in Laghi et al. (2018) and Rakita et al. (2019) are enabled based on the human input and in particular gestures of the hands or recognizable bi-manual patterns during distant (teleoperated) actions. If the hands of the human are busy with part of the tasks, the coordination strategies for the robotic manipulators need to be adapted to convey the intention of the human to the robots through other input signals. For instance, the sensed interaction forces with the environment (i.e., the contact information), which are not considered in the related aforementioned works, constitute a suitable alternative to understand when assistance should be provided and to design proactive shared control strategies. Additionally, by incorporating the contact information to trigger the robot assistance needed, special conditions of the operator, such as limited situational awareness or sensory-motor disorders, could be addressed. In chapter 5, we exploit the interaction forces and design shared control strategies based on time-invariant dynamical systems to provide assistance once the contact with the object is made.

1.6 Robotic Assisted Laparoscopic Surgery

As part of a research project entitled "Four-Armed Manipulation with Robot Assisted Laparoscopic Surgery", this thesis also aims to achieve solo Laparoscopic Surgery (LS). Laparoscopic surgery is a type of Minimally Invasive Surgery (MIS), where the surgery is performed through small incisions in the abdomen (Spaner and Warnock (1997)). This operation usually involves one surgeon manipulating the surgical instruments and one or two assistants. The assistant(s) control(s) a camera, commonly referred as an endoscope, and a retractor which is a tool used to keep the organs, muscles, and other soft tissues out of the way of the surgeon's instruments. There are several difficulties encountered in conventional laparoscopic surgery. For instance, holding the camera for the whole operation duration (which can last several hours) is often tiring for the assistant who has to keep the image stable for the surgeon. Moreover, communication errors and delays can occur in the surgical team which can reduce the efficiency of the operation and affect safety (Nurok et al. (2011); Elprama et al. (2013)). To deal with these limitations, in chapter 6 we target robot-assisted solo surgery, where only the surgeon is involved in the execution of the task by controlling up to four instruments simultaneously thanks to robotic assistants (see Figure 1.3). Since the introduction of robotic surgery in the mid 1980's (Hockstein et al. (2007)), robots have become more and more present in the operating room. Surgeons can benefit from robots not only to increase their autonomy, but

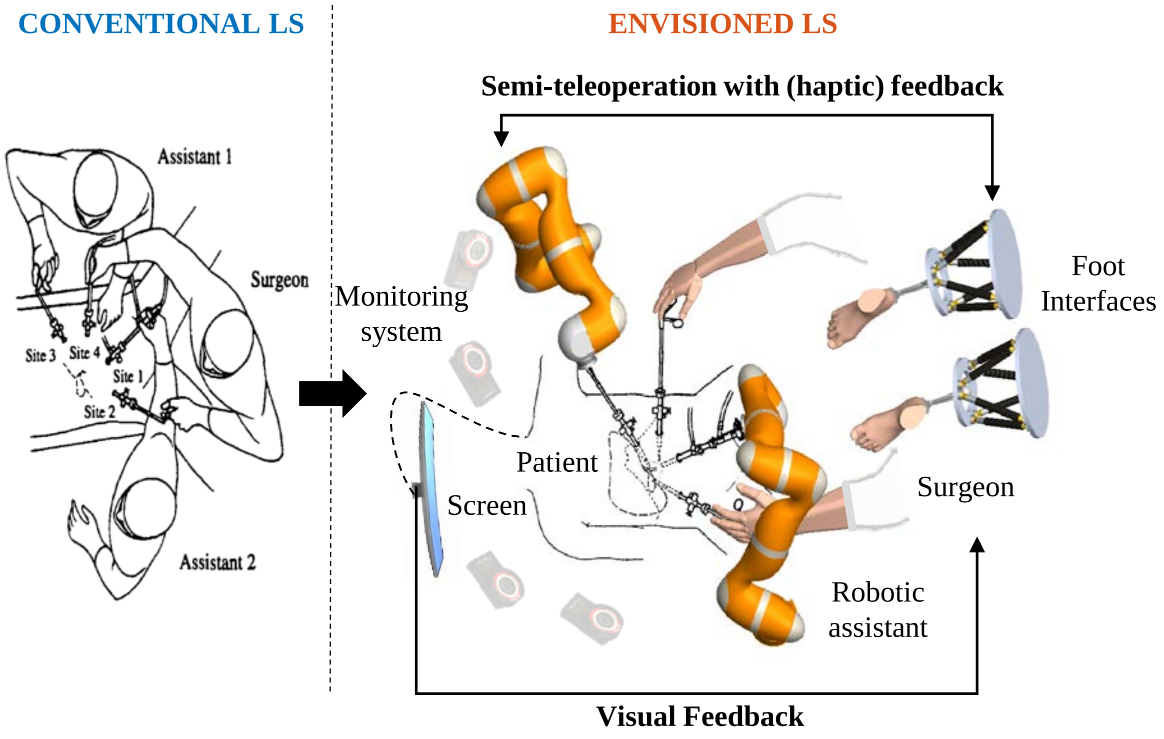


Figure 1.3 – Conventional vs Envisioned Robotic-Assisted Laparoscopic Surgery (©Jacob Hernandez Sanchez)

also to improve their flexibility, dexterity, and precision, especially in the face of fatigue and hand tremors arising during long operations. Similarly to section 1.5, depending on the level of assistance, robot-assisted surgery can be classified into three main categories: *direct control*, *shared control* and *supervisory control* (Moustris et al. (2011)).

In *direct control*, the surgeon has the full control of the robot, either manually (hands-on surgery) or by teleoperating the robotic manipulators through master interfaces. Telesurgical systems represent most of the surgical robots used nowadays, where the da Vinci robot (see Figure 1.4) is the leader in the field and most successful example (Bodner et al. (2004)). These systems are usually based on a remote teleoperation scheme, where the surgeon telemanipulates the robots in a sitting position through hand interfaces and can switch instruments via foot pedals. While higher dexterity and accuracy are provided to the surgeon, several issues are faced with most of the existing commercial systems. First, by remotely controlling the robotic arms, the surgeon's situation awareness; i.e., the perception, comprehension of the environment, and ability to predict its future state; may be affected (Randell et al. (2016)). Moreover, the interaction forces between the surgical tools and the environment (soft tissues) are often lost. This lack of feedback forces can increase the overall applied forces and tissue trauma (Wagner et al. (2002)). To cope with this issue, recent systems try to use force/torque sensors on the slave devices and reflect the measured interactions forces through haptic feedback such as the MiroSurge system (Tobergte et al. (2011)). However, challenges still remain, especially, due to the price, size, geometry and sterilization constraints imposed by MIS (Okamura (2009)). Furthermore, controlling more than two instruments at the same time is not possible, while many operations require three or more instruments to move in coordination.

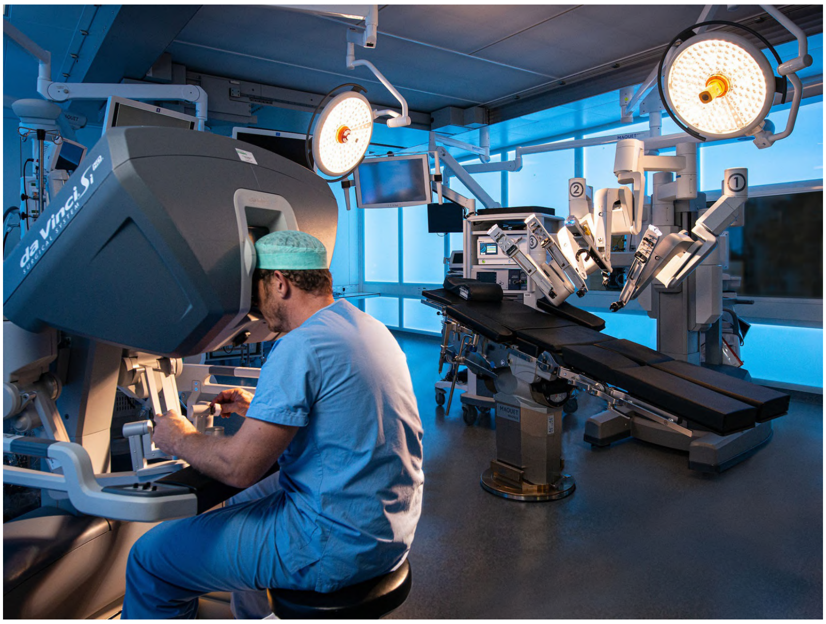


Figure 1.4 – Da Vinci robot (©Hirslanden)

To allow the surgeon controlling more than two surgical instruments simultaneously, recent works focus on developing autonomous robotic assistants, i.e., robots that are able to perform intended tasks based on the current state and sensing of the system without human intervention (SARAS project, De Rossi et al. (2019)). Achieving full autonomy is challenging and mainly depends on three factors (Moustris et al. (2011)): the difficulty of the task, the difficulty of the environment (e.g., static vs dynamic), and the dependence to the human. Indeed, human input is often necessary for the robot to complete its (sub-)task successfully, the robot being then qualified as semi-autonomous. Minimum surgeon input is used in *supervisory control*, where the surgeon only intervenes to provide high-level commands (e.g., preoperative plans) to the robot that takes care of the execution of the task (Dieterich and Gibbs (2011)). While in *shared control*, the surgeon and the robot are both involved in the execution of the task by sharing parts of it. Shared control has been particularly exploited in robotics surgery especially within teleoperation schemes. Input-mixing shared control strategies are for example used to interact with dynamic soft tissues (e.g, beating heart surgery), by compensating the motion of the moving organs through additional commands blended with the surgeon ones (Yuen et al. (2009)). They are also used to filter and scale surgeon's motion to increase accuracy and reduce tremor. Similarly, haptic-based shared control strategies are used to constrain the surgeon's movement or provide guidance (Meli et al. (2017); Xiong et al. (2017)). In chapter 6, we exploit teleoperated shared control strategies to ease the robot-assisted solo surgery by helping the surgeon controlling two robotic assistants via haptic foot interfaces. The proposed control framework not only provides assistance but also takes into account all the safety and task-related constraints. Indeed, laparoscopic surgery first constraints the surgical tools to always move through a fixed insertion point referred as the trocar. The trocar imposes a Remote Center of Motion (RCM) constraint on the tools. To satisfy the RCM constraint, passive and active strategies have been used in the literature. While passive strategies achieve the RCM constraint physically through mechanical design (Kuo et al. (2012)), the active ones programmatically enforce the constraint through control. The latter have been particularly used with serial robots by exploiting the joint redundancies to satisfy both the RCM constraint and the desired tool tip task. In particular null-space control strategies (Dahroug et al. (2016); Sandoval et al. (2017); Sadeghian et al. (2019); Su et al. (2019)) or task augmentation method (Aghakhani et al. (2013); Sandoval et al. (2017)) have been recently used in the literature. In chapter 6, we propose to achieve the RCM constraint and the desired tool tip task by solving the inverse kinematics of the robotic assistants through Quadratic Programming (QP). The QP formulation allows to incorporate safety constraints related to joint limits, joint and end-effector velocity limits; as well as workspace, end-effector, and tool collision avoidance. Dynamical systems are used to express the desired tool tip dynamics in order to provide smooth collaborative behaviour and allow physical interactions with humans.

1.7 Thesis Outline

This thesis is composed of different chapters which are related to the contributions introduced previously. A brief overview of each chapter is provided below:

Chapter 2 - Technical Background

This chapter presents the technical preliminaries upon with this thesis is built. It introduces the concept of (time-invariant) dynamical systems along with their properties and describes how to use a DS to control the motion of a robot.

Chapter 3 - Motion and Contact Force Generation with Dynamical Systems

In this chapter, we present our first contribution on modulating a DS to generate contact forces in addition to motion. Using a desired contact force profile, we construct a modulation term that we add to a nominal DS responsible of bringing a robot in contact with a surface (and eventually moving the robot along the surface). We test our control strategy and its robustness in two scenarios using 7-DoF robotic arms: 1) a polishing task on a non-flat surface, 2) a bimanual grasping and manipulation task. In particular, we show that our approach 1) provides smooth contact with a surface while applying large forces, 2) maintains a desired contact force with reasonable accuracy while the robot is moving on the non-flat surface, and 3) allows to comply to strong external disturbances when reaching, grasping, and moving an object around.

Chapter 4 - Contact Force Adaptation in Uncertain Environements

This chapters describes our second contribution which extends the work of the first one by focusing on the force tracking accuracy. This accuracy is often limited by the uncertainties in the robot dynamics and the environment. To deal with this limitation, we propose online adaptation of a state-dependent force correction model encoded with Radial Basis Functions (RBFs). The model is adapted online using the feedback of the measured interaction forces. We evaluate our approach with a 7-DoF robot and show its efficiency to reduce the force error to a negligible amount with different target forces and velocities. We also showcase a collaborative cleaning with a human where in addition to force adaptation, motion and task adaptation are achieved using previous works through physical human interaction with the robot. Thereby we highlight the benefits of using the time-invariant DS framework in unstructured environment where we need reactive and adaptive behaviours in response to interactions with the environment.

Chapter 5 - Contact-Initiated Shared Control Strategies for Four-Arm Manipulation

This chapters presents our third contribution on providing robotic assistance for a four-arm manipulation task where two supernumerary robotic arms are controlled with the feet of the human via bi-pedal haptic interfaces. The robotic arms complement the tasks of the human arms, in supporting and moving an object while the human is working on the object with the hands. In order to reduce fatigue, cognitive workload and ease the execution of the feet manipulation, we propose two types of assistance that get enabled upon contact with the object (i.e., based on the interaction forces): autonomous contact force generation and auto-coordination of the robotic arms. Four (shared) control strategies are derived from the combinations (absence/presence) of both assistance modalities, and compared to each other through a user study (with 12 participants) on a four-arm manipulation task. We show that force assistance positively improves human-robot fluency in the four-arm task, ease of use and usefulness, as well as reduces fatigue. It is also a crucial factor when controlling both robots with one foot, to make this dual assistance approach the preferred and more successful among the proposed control strategies.

Chapter 6 - Four-Arm Robotic Asissted Laparoscopic Surgery

This chapter describes our fourth and last contribution where we aim to achieve solo robotic assisted laparoscopic surgery. To this end, two robotic arms, controlled via two haptic foot interfaces, assist a surgeon during the operation; one is holding an endoscope/camera and the other one is manipulating a laparoscopic gripper. To guarantee a safe collaboration with the surgeon, a control framework is proposed where an inverse kinematics problem is formulated for the robotic assistants to satisfy all the safety restrictions and task-related constraints. An adaptive autonomous tool tip following method is moreover proposed as an assistance modality for the camera to ease the control through the foot. Furthermore, haptic feedback strategies are designed to increase the situation awareness of the surgeon during the execution of the task. The different control features of the framework are validated through a series of small experiments. Camera assistance is further assessed through a user study, jointly with a grasping assistance modality for the gripper. The study consists in uni-pidal, bi-pedal and four-handed tasks where target-seeking/following and peg-transfer are performed. The results show the benefits of robotic assistance in each task for increasing performance, perception of fluency and reducing task load. Furthermore, the use of shared control mitigates the degradation in mental load and ease of use perceived when increasing the task complexity (i.e., the number of limbs to control).

1.8 Publications, Multimedia, and Source code

The contents of the chapters related to the contributions are either published or submitted to peer-reviewed conferences and journals which are listed thereafter:

- Amanhoud, W., Khoramshahi, M., and Billard, A. (2019). A dynamical system approach to motion and force generation in contact tasks. In Proceedings of Robotics: Science and Systems, Freiburg-im-Breisgau, Germany (Amanhoud et al. (2019), chapter 3).

- Amanhoud,W., Khoramshahi, M., Bonnesoeur, M., and Billard, A. (2020). Force adaptation in contact tasks with dynamical systems.In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 6841–6847. IEEE (Amanhoud et al. (2020), chapter 4).

- Amanhoud, W., Hernandez Sanchez, J., Bouri, M., and Billard, A. (2021). Contact-initiated shared control strategies for four-arm supernumerary manipulation with foot interfaces. The International Journal of Robotics Research, 40(8–9), 986–1014 (Amanhoud et al. (2021), chapter 5).
- Hernandez Sanchez, J., Amanhoud, W., Billard, A., and Bouri, M. Enabling Hybrid robotic-assisted solo-Surgery with Four Arms via Haptic Foot Interfaces. Submitted in October 2021 to The International Journal of Robotics Research (Hernandez Sanchez et al. (2021), chapter 6).

Links to videos of the experiments related to the contributions can be found in Table 1.1, while links to source code are available in Table 1.2:

Chapter	**Multimedia link**
3	https://youtu.be/lz0uxUEVc3g
4	https://youtu.be/pNJ2tLjxLac
5	https://youtube.com/playlist?list=PLs3zEsp7m08Vdn-7oemYUlpN-Tk0bX--z
6	https://youtube.com/playlist?list=PLs3zEsp7m08XRrBd7fq9rApVjwhaz1kce

Table 1.1 – Links to multimedia contents

Chapters	**Source code link**
3, 4	https://github.com/epfl-lasa/ds_based_contact_tasks
5, 6	https://github.com/epfl-lasa/hasler_project

Table 1.2 – Links to source code

2 Technical Background

2.1 Dynamical Systems

This section aims to provide some properties and definitions related to dynamical systems which are widely used in this thesis. In its generic form, a *continuous-time dynamical system* can be written as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}) \tag{2.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $t \in \mathbb{R}^+$ and $\boldsymbol{u} \in \mathbb{R}^m$ respectively denote the system's state vector (with n the state's space dimension), the time, and the control input (with m the input's space dimension), while the function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ governs the evolution of the system. A continuous-time dynamical system which has no input signal is called *autonomous* and takes the simplified form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}) \tag{2.2}$$

In this thesis, we are exclusively working with autonomous dynamical systems which are *time-invariant* or strictly state-dependent; i.e., of the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{2.3}$$

The time-invariance property means that the relationship between the system's state and state derivative is constant with respect to time.

A state $\bar{\boldsymbol{x}}$ that satisfies $\boldsymbol{f}(\bar{\boldsymbol{x}}) = \boldsymbol{0}$ is called an *equilibrium* of the system. Stability of the equilibrium point is commonly studied using *Lyapunov's theory*. Lyapunov's indirect method focuses on the local stability properties of the linearized system around the equilibrium which are fully described by the Jacobian matrix of the system $\boldsymbol{J}(\bar{\boldsymbol{x}}) = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}}$. If all eigenvalues λ_i of $\boldsymbol{J}(\bar{\boldsymbol{x}})$ satisfy $\Re\{\lambda_i\} < 0$, $\bar{\boldsymbol{x}}$ is asymptotically stable. Conversely, if at least one eigenvalue of $\boldsymbol{J}(\bar{\boldsymbol{x}})$ satisfies $\Re\{\lambda_i\} > 0$, $\bar{\boldsymbol{x}}$ is unstable. As an alternative to the indirect method, Lyapunov's direct method provides a more generic way to establish the stability properties of a system. It consists to study how certain scalar functions, taking as input the system's state, evolve with the system's state itself. In particular, let us suppose that there exists a function $V(\boldsymbol{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ such that:

- $V(\boldsymbol{x})$ is continuously differentiable.
- $V(\boldsymbol{x}) > 0 \ \forall \boldsymbol{x} \in \mathbb{R}^n \backslash \{\bar{\boldsymbol{x}}\}$ and $V(\bar{\boldsymbol{x}}) = 0$.

The equilibrium state is:

- stable if $\dot{V}(\boldsymbol{x}) \leq 0 \ \forall \boldsymbol{x} \in \mathbb{R}^n$.
- asymptotically stable if $\dot{V}(\boldsymbol{x}) < 0 \ \forall \boldsymbol{x} \in \mathbb{R}^n \backslash \{\bar{\boldsymbol{x}}\}$.
- globally asymptotically stable if $\dot{V}(\boldsymbol{x}) < 0 \ \forall \boldsymbol{x} \in \mathbb{R}^n \backslash \{\bar{\boldsymbol{x}}\}$ and $\|\boldsymbol{x}\| \rightarrow \infty \implies V(\boldsymbol{x}) \rightarrow \infty$.

A function $V(\boldsymbol{x})$ that satisfies the stability conditions above is referred as a *Lyapunov function.*

A particular class of dynamical systems that we will use in this thesis are *gradient systems*:

$$\dot{\boldsymbol{x}} = -\nabla_{\boldsymbol{x}} V_c(\boldsymbol{x}) \tag{2.4}$$

with $V_c(\boldsymbol{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ a twice continuously differentiable function. Selecting $V_c(\boldsymbol{x})$ as a Lyapunov function (assuming that $\bar{\boldsymbol{x}}$ is an equilibrium of the system and $V_c(\boldsymbol{x}) > 0$), it follows from Eq. 2.4 that $V_c(\boldsymbol{x})$ is non-increasing. Indeed, $\dot{V}_c(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} V_c(\boldsymbol{x}) \dot{\boldsymbol{x}} = -\|\nabla_{\boldsymbol{x}} V_c(\boldsymbol{x})\|^2 < 0$, proving the stability of $\bar{\boldsymbol{x}}$. These dynamical systems are often referred as *conservatives* while $V_c(\boldsymbol{x})$ is often called a potential energy function.

2.2 Robot Motion Control with Dynamical Systems

Dynamical systems have been particularly exploited as motion generators, especially in our group (Khansari-Zadeh et al. (2012); Mirrazavi Salehian et al. (2016)). One of the main thesis contributions is to extend the use of dynamical systems to perform contact tasks, i.e., generating motion and contact forces. Before presenting the thesis' contributions, in this section, we describe how to use DS to control a robot motion while being robust to real-time disturbances.

2.2.1 Robot Dynamics

Let us first consider the dynamical model of a N degrees of freedom robotic manipulator in the joint space:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{\boldsymbol{c}} + \boldsymbol{\tau}_{\boldsymbol{ext}} \tag{2.5}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^N$ denote the joint state position, velocity, and acceleration vectors, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{N \times N}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{N \times N}$ the Coriolis/Centrifugal matrix, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^N$ the gravity torques, while $\boldsymbol{\tau}_{\boldsymbol{c}} \in \mathbb{R}^N$ and $\boldsymbol{\tau}_{\boldsymbol{ext}} \in \mathbb{R}^N$ are the control and external torques respectively.

Most of the time, a robotic task is formulated in the Cartesian space (or task space) by specifying the desired end-effector behaviour. The end-effector state $\boldsymbol{X} \in \mathbb{R}^M$ is described using M coordinates. One of the most common representation uses $M = 6$, where $\boldsymbol{X}$ includes both the position $\boldsymbol{x} \in \mathbb{R}^3$ and orientation $\boldsymbol{\eta} \in \mathbb{R}^3$ of the end-effector, with $\boldsymbol{\eta}$ usually selected to be the vector of Euler angles. The relationship between the end-effector and joint state position vectors is commonly described by the robot forward kinematics through the function $\boldsymbol{f_{kin}} : \mathbb{R}^N \to \mathbb{R}^M$ such that:

$$\boldsymbol{X} = \boldsymbol{f_{kin}}(\boldsymbol{q}) \tag{2.6}$$

From the forward kinematics, one can derive the differential kinematics which relates the end-effector and joint state velocities. The mapping can be obtained in two ways. The first way uses direct differentiation of the forward kinematics with respect to the joint state position such that:

$$\dot{\boldsymbol{X}} = \begin{pmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\eta}} \end{pmatrix} = \boldsymbol{J_a}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.7}$$

where $\boldsymbol{J_a}(\boldsymbol{q}) = \dfrac{\partial \boldsymbol{f_{kin}}(\boldsymbol{q})}{\partial \boldsymbol{q}} \in \mathbb{R}^{M\times N}$ is referred as the analytical Jacobian matrix. This matrix is dependent on the representation of the end-effector state, and particularly the representation of the orientation. If a different representation is chosen (e.g., quaternion, column vectors of the orientation matrix etc.), the analytical Jacobian changes.

The second way exploits geometric considerations from the forward kinematics to construct a unique Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{6\times N}$, known as the geometric Jacobian, relating the linear $\dot{\boldsymbol{x}} \in \mathbb{R}^3$ and angular $\boldsymbol{\omega} \in \mathbb{R}^3$ end-effector velocity to $\dot{\boldsymbol{q}}$:

$$\dot{\boldsymbol{X}} = \begin{pmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{\omega} \end{pmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.8}$$

The geometric Jacobian is of great interest in robotics as it also provides the relationship between forces applied at the end-effector and torques needed at the joints to support these forces:

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^{\boldsymbol{T}}\boldsymbol{W} \tag{2.9}$$

where $\boldsymbol{W} \in \mathbb{R}^6$ is a task-space wrench while $\boldsymbol{\tau} \in \mathbb{R}^N$ are the corresponding joint space torques.

To design a controller for achieving a desired robot behaviour in the task space, it is convenient to express the robot dynamics also in the task space, as done in the operational space formulation (Khatib (1987)). To this end, one needs to relate the end-effector and joint state accelerations. This is achieved through the second-order differential kinematics which can be obtained from Eq. 2.8:

$$\ddot{\boldsymbol{X}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.10}$$

Substituting $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{X}}$ from Eq. 2.8, $\ddot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\big(\ddot{\boldsymbol{X}} - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}\big)$ from Eq. 2.10, and both $\boldsymbol{\tau_c} = \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{W_c}$ and $\boldsymbol{\tau_{ext}} = \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{W_{ext}}$ from Eq. 2.9, into Eq. 2.5 leads to:

$$\boldsymbol{M_X}(\boldsymbol{q})\ddot{\boldsymbol{X}} + \boldsymbol{C_X}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{X}} + \boldsymbol{g_X}(\boldsymbol{q}) = \boldsymbol{W_c} + \boldsymbol{W_{ext}} \tag{2.11}$$

where the Cartesian space inertia matrix $\boldsymbol{M_X}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$, Coriolis/centrifugal matrix $\boldsymbol{C_X}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{6\times 6}$, and gravity forces $\boldsymbol{g_X}(\boldsymbol{q}) \in \mathbb{R}^6$ are given by:

$$\begin{cases} \boldsymbol{M_X}(\boldsymbol{q}) = \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{-1} \\ \boldsymbol{C_X}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{J}(\boldsymbol{q})^{-T}\big(\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{J}}(\boldsymbol{q})\big)\boldsymbol{J}(\boldsymbol{q})^{-1} \\ \boldsymbol{g_X}(\boldsymbol{q}) = \boldsymbol{J}(\boldsymbol{q})^{-T}\boldsymbol{g}(\boldsymbol{q}) \end{cases} \tag{2.12}$$

while $\boldsymbol{W_c} = \big[\boldsymbol{F_c^T}, \boldsymbol{T_c^T}\big]^T$ and $\boldsymbol{W_{ext}} = \big[\boldsymbol{F_{ext}^T}, \boldsymbol{T_{ext}^T}\big]^T$ are the task-space control and external wrench respectively, which can be decomposed into linear forces ($\boldsymbol{F_c}, \boldsymbol{F_{ext}} \in \mathbb{R}^3$) and torques ($\boldsymbol{T_c}, \boldsymbol{T_{ext}} \in \mathbb{R}^3$).

Eq. 2.11 represents the robot (or end-effector) dynamics in the six-dimensional Cartesian space. Sometimes in this thesis, for simplification purposes, we will only consider the three-dimensional linear Cartesian space:

$$\boldsymbol{M_x}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{C_x}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}} + \boldsymbol{g_x}(\boldsymbol{q}) = \boldsymbol{F_c} + \boldsymbol{F_{ext}} \tag{2.13}$$

where $\boldsymbol{M_x}(\boldsymbol{q}) \in \mathbb{R}^{3\times 3}$, $\boldsymbol{C_x}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{3\times 3}$, and $\boldsymbol{g_x}(\boldsymbol{q}) \in \mathbb{R}^3$ are the linear inertia matrix, Coriolis/-centrifugal matrix, and gravity forces.

2.2.2 Dynamical Systems as Motion Generators

To use time-invariant DS as motion generators, two main strategies can be considered. The first one uses the DS in an open-loop configuration, i.e., without the feedback of the current robot state. The DS provides a desired linear velocity which is integrated over time to get a desired position:

$$\begin{cases} \dot{\boldsymbol{x}}_{\boldsymbol{d}}(t) = \boldsymbol{f}(\boldsymbol{x_d}(t)) \\ \boldsymbol{x_d}(t) = \boldsymbol{x_d}(0) + \displaystyle\int_0^t \boldsymbol{f}(\boldsymbol{x_d}(\tau))d\tau \end{cases} \tag{2.14}$$

where $\boldsymbol{x_d}(t), \dot{\boldsymbol{x}}_{\boldsymbol{d}}(t) \in \mathbb{R}^3$ represent the time-indexed desired position and velocity with $t \in \mathbb{R}^+$ the time and $\boldsymbol{x_d}(0)$ the initial desired position obtained from the initial robot state. Such configuration is suitable to be used with standard position/impedance controller. However, the time dependency reduces the robustness property to real-time disturbances as the current robot state is not taken into account.

In this thesis, we mainly use dynamical systems in a closed-loop configuration by providing the current robot position as input feedback to the DS in order to produce a desired velocity:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}} = \boldsymbol{f}(\boldsymbol{x}) \tag{2.15}$$

In this way, any disturbances affecting the robot (end-effector) are captured by the robot state and taken into account by the DS which can re-plan the execution of the task from the disturbed state once the disturbances disappear.

2.2.3 DS-based Impedance Controller

A time-invariant DS used in a closed-loop configuration can be tracked with a velocity field controller. In this thesis, we take advantage of the DS-based impedance controller (Kronander and Billard (2015)), previously developed in our group. It computes a control force $\boldsymbol{F_c}$ similarly to a damping controller with a task-varying damping matrix $\boldsymbol{D}(\boldsymbol{x}) \in \mathbb{R}^{3\times3}$ and a gravity cancellation term such that:

$$\boldsymbol{F_c} = \boldsymbol{D}(\boldsymbol{x})(\dot{\boldsymbol{x}}_{\boldsymbol{d}} - \dot{\boldsymbol{x}}) + \boldsymbol{g_x}(\boldsymbol{q}) = d_1\dot{\boldsymbol{x}}_{\boldsymbol{d}} - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} + \boldsymbol{g_x}(\boldsymbol{q}) \tag{2.16}$$

$\boldsymbol{D}(\boldsymbol{x})$ is decomposed such that the first eigenvector is aligned with the desired dynamics:

$$\begin{cases} \boldsymbol{D}(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x})\boldsymbol{\Lambda}\boldsymbol{U}(\boldsymbol{x})^{\boldsymbol{T}} \\ \boldsymbol{U}(\boldsymbol{x}) = [\boldsymbol{e_1}(\boldsymbol{x})\ \boldsymbol{e_2}(\boldsymbol{x})\ \boldsymbol{e_3}(\boldsymbol{x})] \\ \boldsymbol{e_1}(\boldsymbol{x}) = \dfrac{\dot{\boldsymbol{x}}_{\boldsymbol{d}}}{\|\dot{\boldsymbol{x}}_{\boldsymbol{d}}\|} \\ \boldsymbol{\Lambda} = \mathrm{diag}\big([d_1\ d_2\ d_3]\big) \end{cases} \tag{2.17}$$

where $\boldsymbol{e_2}(\boldsymbol{x}) \in \mathbb{R}^3$ and $\boldsymbol{e_3}(\boldsymbol{x}) \in \mathbb{R}^3$ are selected such that $\boldsymbol{U}(\boldsymbol{x}) \in \mathbb{R}^{3\times3}$ forms an orthonormal basis while $d_1, d_2, d_3 \in \mathbb{R}^+$ are the eigenvalues/damping gains ensuring the positive semi-definite property of $\boldsymbol{D}(\boldsymbol{x})$. This construction of $\boldsymbol{D}(\boldsymbol{x})$ first allows to selectively damp disturbances that are orthogonal to the desired velocity (through the selection of d_2 and d_3). Second, it guarantees passivity for desired conservative DS (i.e., DS that derives from a potential function). Indeed, let us consider a storage function $W \in \mathbb{R}$ including the linear kinetic energy of a robot and the potential function $V_c(\boldsymbol{x})$ from which derives the DS (i.e., $\dot{\boldsymbol{x}}_{\boldsymbol{d}} = -\nabla_{\boldsymbol{x}} V_c(\boldsymbol{x})$):

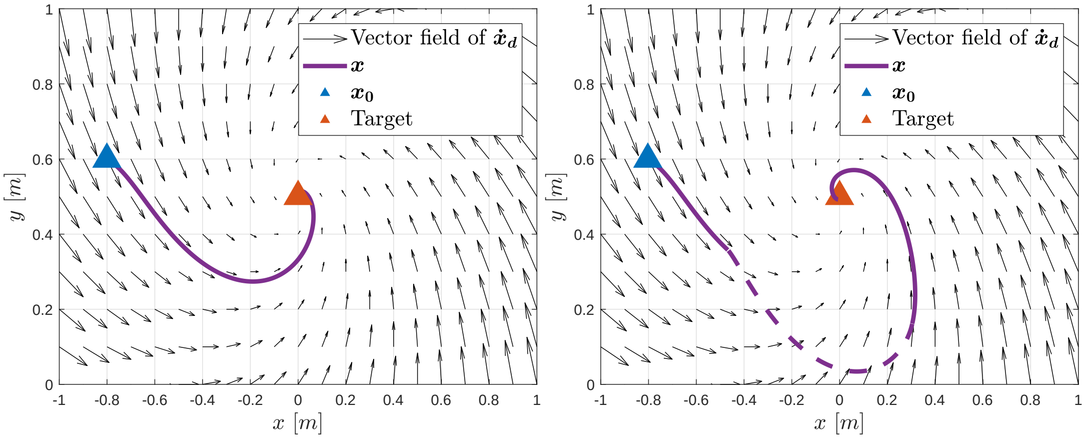
$$W = \frac{1}{2}\dot{\boldsymbol{x}}^{\boldsymbol{T}}\boldsymbol{M_x}(\boldsymbol{q})\dot{\boldsymbol{x}} + d_1 V_c(\boldsymbol{x}) \tag{2.18}$$

The rate of change of W becomes:

$$\dot{W} = \dot{\boldsymbol{x}}^{\boldsymbol{T}}\boldsymbol{M_x}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \frac{1}{2}\dot{\boldsymbol{x}}^{\boldsymbol{T}}\dot{\boldsymbol{M}}_{\boldsymbol{x}}(\boldsymbol{q})\dot{\boldsymbol{x}} + d_1\dot{\boldsymbol{x}}^{\boldsymbol{T}}\nabla_{\boldsymbol{x}} V_c(\boldsymbol{x}) \tag{2.19}$$

Substituting $\boldsymbol{M_x}(\boldsymbol{q})\ddot{\boldsymbol{x}}$ from Eq. 2.13 gives:

$$\dot{W} = \dot{\boldsymbol{x}}^{\boldsymbol{T}}\big(\boldsymbol{F_c} + \boldsymbol{F_{ext}} - \boldsymbol{C_x}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}} - \boldsymbol{g_x}(\boldsymbol{q})\big) + \frac{1}{2}\dot{\boldsymbol{x}}^{\boldsymbol{T}}\dot{\boldsymbol{M}}_{\boldsymbol{x}}(\boldsymbol{q})\dot{\boldsymbol{x}} - d_1\dot{\boldsymbol{x}}^{\boldsymbol{T}}\dot{\boldsymbol{x}}_{\boldsymbol{d}} \tag{2.20}$$



(a) Without disturbances

(b) With disturbances

Figure 2.1 – 2D illustration of a robot tracking a time-invariant linear DS to reach a target when there is no disturbances applied to the robot (Figure 2.1a) and with the presence of disturbances (Figure 2.1b). In Figure 2.1b, the dashed part of the robot trajectory is the result of a negative disturbance force applied to the robot along the y axis.

Using the skew-symmetric property of $\frac{1}{2}\dot{\boldsymbol{M}}_{\boldsymbol{x}}(\boldsymbol{q}) - \boldsymbol{C}_{\boldsymbol{x}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and substituting $\boldsymbol{F_c}$ by Eq. 2.16 finally leads to:

$$\dot{W} = -\dot{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{F_{ext}} \tag{2.21}$$

As $\dot{W} < \dot{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{F_{ext}}$, the system is passive with respect to the environment. Note that the conservative Lyapunov constraints on the DS forces the dynamics to keep decreasing energy, i.e., that the path starts and ends from one place to another. The path types belonging to this category can range from simple straight line to very complex curvy shapes which can be learned from data (Perrin and Schlehuber-Caissier (2016); Figueroa and Billard (2018); Rana et al. (2020)). Furthermore, for non-conservative DS, Kronander and Billard (2015) also proves passivity. It assumes that any DS can be decomposed into a conservative and non-conservative part and uses energy tanks to guarantee passivity of the system. This demonstration will not be shown here as it will be used as a basis (and described in details) later when proving passivity of our modulation strategy to generate motion and contact forces with a DS (see section 3.2.3).

As an illustrative example, in Figure 2.1, we simulate the dynamics of a robot in 2D, tracking a linear DS $\dot{\boldsymbol{x}}_{\boldsymbol{d}} \in \mathbb{R}^2$ defined by:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}} = \boldsymbol{A}(\boldsymbol{x_a} - \boldsymbol{x}) \qquad \text{with } \boldsymbol{A} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \ \boldsymbol{x_a} = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} \tag{2.22}$$

where $\boldsymbol{x_a}$ is the target attractor. To control the robot, a 2D version of the DS-impedance controller is used. Two scenarios are presented: DS tracking without disturbances applied to the robot (Figure 2.1a) and with disturbances (Figure2.1b). The robot starts from the same initial position in both cases $\boldsymbol{x_0} = [\text{-}0.8, 0.6]^T$. In the absence of disturbances (Figure 2.1a), the robot reaches the DS while accurately following the desired dynamics. When applying a negative disturbance force along the y axis during a short amount of time (Figure 2.1b), the robot does not follow the flow of the DS anymore (dashed part of the trajectory) but complies to the disturbance until it disappears. At that moment, the robot just re-plans its motion from the last disturbed state and follows again the flow of the DS to reach the target. This robustness to real-time disturbances is the result of the time-invariance property and the use of the current robot position as input to the DS. It is a critical requirement when operating in unstructured environment where many disturbances of this kind can occur.

2.3 Task-space Orientation Control

In general, controlling the robot orientation in task-space is more complex than the position. It first requires to choose a representation for specifying the desired and measured orientation. For that purpose, the minimal orientation representation embodied by Euler angles is frequently used in robotics. From the measured $\boldsymbol{R_o} = [\boldsymbol{x_o y_o z_o}] \in \mathbb{R}^{3\times3}$ (where $\boldsymbol{x_o}, \boldsymbol{y_o}, \boldsymbol{z_o} \in \mathbb{R}^3$ are column vectors) and desired rotation/orientation matrix $\boldsymbol{R_{o,d}} \in \mathbb{R}^{3\times3}$ of the frame attached to the end-effector, one can extract the corresponding set of measured $\boldsymbol{\phi} \in \mathbb{R}^3$ and desired $\boldsymbol{\phi_d} \in \mathbb{R}^3$ Euler angles, and compute the Euler-based orientation error $\tilde{\boldsymbol{\phi}} = \boldsymbol{\phi_d} - \boldsymbol{\phi}$. This representation suffers however from singularities. A better approach consists to directly compute the orientation error from the rotation matrices $\tilde{\boldsymbol{R}} = \boldsymbol{R_{o,d}} \boldsymbol{R_o^T}$ and convert it into the axis-angle representation $\tilde{\boldsymbol{\zeta}} \in \mathbb{R}^3$. It allows to parametrize the error by a unit vector $\boldsymbol{e} \in \mathbb{R}^3$ indicating a rotation axis and an angle $\theta \in \mathbb{R}$ representing the rotation (error) angle around this axis such that $\tilde{\boldsymbol{\zeta}} = \theta \boldsymbol{e}$. This approach is the one adopted in this thesis to control the robot orientation in the task-space as it avoids singularities. From the error signal $\tilde{\boldsymbol{\zeta}}$, we compute the task-space control torque $\boldsymbol{T_c}$ (in Eq. 2.11) as:

$$\boldsymbol{T_c} = \boldsymbol{K_\zeta} \tilde{\boldsymbol{\zeta}} + \boldsymbol{D_\omega}(\boldsymbol{\omega_d} - \boldsymbol{\omega}) \tag{2.23}$$

where $\boldsymbol{K_\zeta} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{D_\omega} \in \mathbb{R}^{3\times3}$ are the rotational stiffness and damping gain matrices respectively, while $\boldsymbol{\omega_d} \in \mathbb{R}^3$ is the desired angular velocity. The same strategy can be used when representing the measured and desired orientation with unit quaternion $\boldsymbol{Q} \in \mathbb{R}^4$ which provides a non-minimal orientation representation with several advantages. Indeed, it is singularity free, has a more compact representation than a rotation matrix, and the conversion to axis-angle representation is straightforward. Quaternion is also a suitable representation to use for tasks requiring smooth orientation changes. In this thesis, depending on the tasks and their needs, we either use quaternions or rotation matrices.

3 Motion and Contact Force Generation with Dynamical Systems

Note: The material presented in this chapter is adopted from:
Amanhoud, W., Khoramshahi, M., and Billard, A. (2019). A dynamical system approach to motion and force generation in contact tasks. In Proceedings of Robotics: Science and Systems, Freiburg-im-Breisgau, Germany.

3.1 Introduction

In this chapter, we propose a control strategy to perform contact tasks with robustness to large real-time disturbances. The envisioned strategy should react to human interactions (e.g., stopping the robot, breaking the contact with the surface/object, or moving the robot arbitrarily), and also to unexpected changes in the environment (e.g., the position and orientation of the surface/object). To this end, we take advantage of the time-invariant DS framework. DS provide very fast reactivity and enable on-the-fly re-planning of trajectories. In this work, we extend this framework to perform contact tasks by generating contact forces in addition to motion. Our strategy consists to encode contact tasks by combining the desired motions and contact force profiles in a single mathematical expression, by means of time-invariant DS. The combination is based on local modulation of a nominal DS (responsible of motion) to generate contact forces when the robot is close to the surface. As a result, this strategy offers:

- Stable and accurate motion and contact force generation.
- Compliant behavior in all phases of the contact task (in free motion, upon making contact and when in contact).
- Robustness to real-time disturbances.

The related work for this chapter is reviewed in section 1.4. In the following, we first describe our method in sections 3.2 and 3.3. We evaluate it in section 3.4 on two real-world scenarios: a polishing task on a non-flat surface and a bi-manual reaching, grasping and manipulation task. We conclude with a discussion about the method and results obtained in section 3.5.

3.2 Method

3.2.1 Nominal DS Definition

In chapter 2, we presented the technical background related to motion control using time-invariant DS. With this background in mind, let us assume the existence of a nominal DS $\boldsymbol{f}(\boldsymbol{x})$ that brings the robot in contact with a surface and moves the robot along the surface, as illustrated in Figure 3.2. We suppose that the contact surface is non-penetrable and that we have an explicit expression for the normal vector $\boldsymbol{n}(\boldsymbol{x})$ and distance to the surface $\Gamma(\boldsymbol{x})$ at all points in space. The nominal DS should satisfy:

$$\begin{cases} \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}(\boldsymbol{x}) = 0 & \text{in contact} \\ \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{n}(\boldsymbol{x}) > 0 & \text{in free motion} \end{cases} \tag{3.1}$$

To meet these constraints, such dynamics can be simply designed (see Appendix A.2 as an example), but also learned from human demonstrations and/or locally modulated $\big($Khansari-Zadeh and Billard (2011); Kronander et al. (2015)$\big)$. Thereafter, we show how to modulate the nominal DS to generate contact forces in addition to motion.

3.2.2 Force-based DS Modulation

To achieve the desired motion and force profile with a single DS, we decompose the system as follows:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{f}_{\boldsymbol{n}}(\boldsymbol{x}) \tag{3.2}$$

with $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ the modulated DS, $\boldsymbol{f}(\boldsymbol{x})$ the nominal DS responsible of motion, and $\boldsymbol{f}_{\boldsymbol{n}}(\boldsymbol{x})$ the modulation term that applies only along the direction normal to the surface to generate contact forces:

$$\boldsymbol{f}_{\boldsymbol{n}}(\boldsymbol{x}) = \frac{F_d(\boldsymbol{x})}{d_1}\boldsymbol{n}(\boldsymbol{x}) \tag{3.3}$$

where $F_d(\boldsymbol{x}) \in [0, F_{d,+}]$ with $F_{d,+} > 0$ is the state-dependent desired contact force profile. $F_d(\boldsymbol{x})$ is time-invariant to provide robustness to real-time disturbances and should be designed to generate contact forces only when the robot is close to the surface, hence the state dependency. Providing Eq. 3.2 as input to the DS-based impedance controller (Eq. 2.16) gives as result the control force:

$$\boldsymbol{F_c} = d_1\boldsymbol{f}(\boldsymbol{x}) + d_1\boldsymbol{f}_{\boldsymbol{n}}(\boldsymbol{x}) - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} + \boldsymbol{g}(\boldsymbol{x}) \tag{3.4}$$

The first term represents the driving force along the nominal dynamics while the second term denotes the modulation contact force along the normal direction to the surface.

The modulation strategy is illustrated in Figure 3.3 where the nominal DS presented in Figure 3.2 is modulated to generate a contact force when the robot becomes close to the surface. Before contact with the surface, the desired and nominal DS are aligned and identical. Close to contact, the normal modulation component gets generated and modulates the nominal DS to

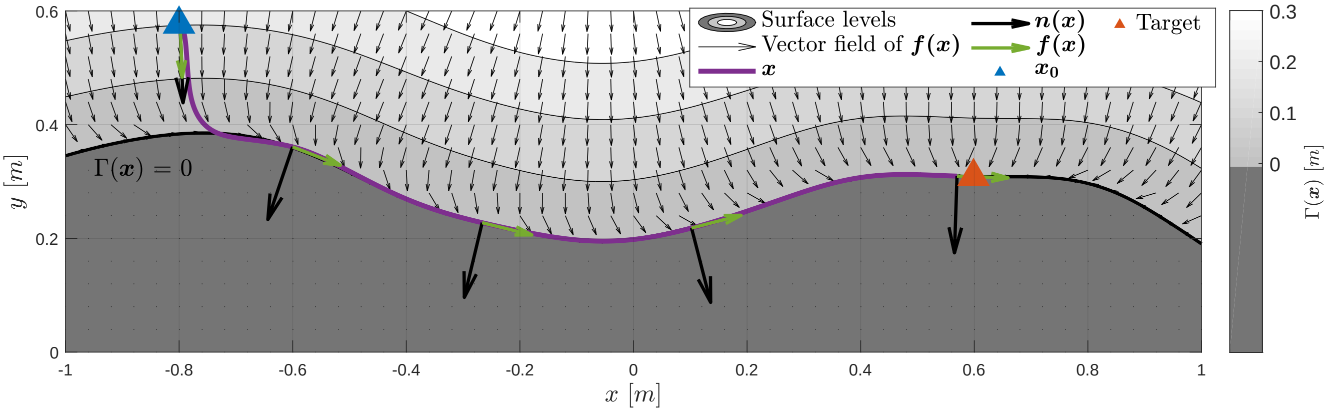


Figure 3.2 – Illustration of a robot driven by a nominal DS to come in contact with a surface and move towards a target, starting from an initial position $\boldsymbol{x_0}$. The normal distance $\Gamma(\boldsymbol{x})$ and vector $\boldsymbol{n}(\boldsymbol{x})$ to the surface can be learned using different learning algorithms such as Support Vector Regression (SVR) (Steinke et al. (2005)) or Gaussian Process Regression (GPR) (Li et al. (2016)). Here, we use SVR with a Gaussian kernel ($C = 100$, $\epsilon = 0.01$, $\sigma = 0.20$).

produce the desired force. To illustrate the robustness of our approach in face of disturbances, an external force disturbs the robot away from the surface while the robot is moving. The modulated DS reacts to the disturbance by realigning with the nominal one. Once the disturbance disappears, the robot reaches the surface and moves toward the target while applying the desired contact force.

3.2.3 Ensuring Passivity

In background section 2.2 we saw that if a conservative DS is given as input to the DS-based impedance controller (Eq. 2.16), then the robot system described by Eq. 2.13 is passive with respect to the environment. To analyze passivity of our force-based DS modulation strategy, let us proceed similarly to Kronander and Billard (2015) and first decompose the nominal DS into a conservative part $\boldsymbol{f_c}(\boldsymbol{x}) \in \mathbb{R}^3$ and a non-conservative part $\boldsymbol{f_r}(\boldsymbol{x}) \in \mathbb{R}^3$ such that:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f_c}(\boldsymbol{x}) + \boldsymbol{f_r}(\boldsymbol{x}) \tag{3.5}$$

where $\boldsymbol{f_c}(\boldsymbol{x})$ derives from the potential function $V_c(\boldsymbol{x})$ (i.e., $\boldsymbol{f_c}(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} V_c(\boldsymbol{x})$). The modulation strategy in Eq. 3.2 modifies the rate of change of the storage function W used in Eq. 2.18 to:

$$\dot{W} = d_1 \dot{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{f_r}(\boldsymbol{x}) + d_1 \dot{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{f_n}(\boldsymbol{x}) - \dot{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{D}(\boldsymbol{x}) \dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{F_{ext}} \tag{3.6}$$

which can be rewritten into:

$$\dot{W} = p_r + p_c - p_d + \dot{\boldsymbol{x}}^{\boldsymbol{T}} \boldsymbol{F_{ext}} \tag{3.7}$$

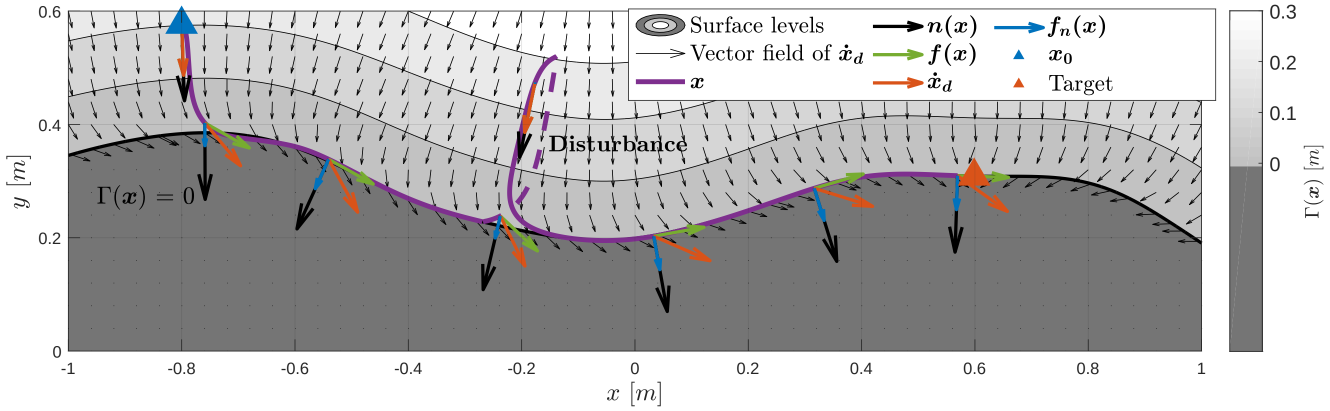


Figure 3.3 – Illustration of the modulation approach on the task of reaching and moving on a non-flat surface. The robot is driven by the modulated DS and undergoes a disturbance normal to the surface (dashed line).

with $p_d = \dot{x}^T D(x)\dot{x}$, $p_r = d_1 \dot{x}^T f_r(x)$ and $p_n = d_1 \dot{x}^T f_n(x)$ respectively denoting the dissipated power, the power due to the non-conservative part of the nominal DS and the power generated by the normal modulation term. Thanks to the definition of $D(x)$ (Eq. 2.17) we can ensure that $p_d \geq 0$ while the sign of the first two terms in Eq. 3.7 is undefined. Therefore, we cannot guarantee passivity of the system with respect to the environment. To restore passivity, we consider an approach based on energy tanks (see section 1.4). The goal is to use the energy dissipated by the system to fill the tank and extract energy from the tank to temporarily execute non-passive actions.

Thus, let us introduce a virtual tank state s that stores the dissipated energy in the system which mainly comes from the damping term p_d. This energy aims to modulate the nominal DS without violating passivity. The resulting energy flow is governed by the tank's dynamics, which is coupled with the robot's state (x and $\dot{x}$) as follows:

$$\dot{s} = \alpha(s) p_d - \beta_r(s, p_r) p_r - \beta_n(s, p_n) p_n \tag{3.8}$$

The scalar functions $\alpha(s)$, $\beta_r(s, p_r)$ and $\beta_n(s, p_n)$ control the energy flow between the virtual tank and the robot. Let us define s_+ as the maximum energy level allowed to be stored in the tank. To ensure that s remains bounded between 0 and s_+, we define the scalar functions such that:

$$\begin{aligned} \alpha(s) &= \Lambda^{-}_{s_+-\delta_s, s_+}(s) \\ \beta_i(s, p_i) &= \begin{cases} 0 & \text{if } s < 0 \text{ and } p_i > 0 \\ 0 & \text{if } s > s_+ \text{ and } p_i < 0 \qquad i = r, n \\ 1 & \text{otherwise} \end{cases} \end{aligned} \tag{3.9}$$

where $\delta_s \in [0, s_+]$ while the function $\Lambda^-_{a,b}(u)$ provides a smooth transition from 1 to 0 as u transits from a to b:

$$\Lambda^-_{a,b}(u) = \begin{cases} 1 & u < a \\ \frac{1}{2}\left(1 + \cos\left(\pi \frac{u-a}{b-a}\right)\right) & a \leq u \leq b \\ 0 & u > b \end{cases} \tag{3.10}$$

with $u, a, b \in \mathbb{R}$ and $a < b$. If the tank is depleted, the controller should not generate the potential non-passive actions. The control law should be corrected accordingly by taking the state of the tank and the power variables into account. To this end, we correct the modulation law in Eq. 3.2 as follows:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}} = \boldsymbol{f'}(\boldsymbol{x}) + \boldsymbol{f'_n}(\boldsymbol{x}) \tag{3.11}$$

with:

$$\begin{cases} \boldsymbol{f'}(\boldsymbol{x}) = \boldsymbol{f_c}(\boldsymbol{x}) + \beta'_r(s, p_r)\boldsymbol{f_r}(\boldsymbol{x}) \\ \boldsymbol{f'_n}(\boldsymbol{x}) = \beta'_n(s, p_n)\boldsymbol{f_n}(\boldsymbol{x}) \end{cases} \tag{3.12}$$

where $\beta'_r(s, p_r)$ and $\beta'_n(s, p_n)$ are scalar functions satisfying:

$$\beta'_i(s, p_i) = \begin{cases} 1 & \text{if } p_i < 0 \\ \beta_i(s, p_i) & \text{otherwise} \end{cases} \qquad i = r, n \tag{3.13}$$

Let us now define the final storage function $W(\boldsymbol{x}, \dot{\boldsymbol{x}}, s)$ taking the tank's dynamics into account:

$$W(\boldsymbol{x}, \dot{\boldsymbol{x}}, s) = \frac{1}{2}\dot{\boldsymbol{x}}^{\boldsymbol{T}}\boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{x}} + d_1 V_c(\boldsymbol{x}) + s \tag{3.14}$$

Substituting $\dot{s}$ by Eq. 3.8 and $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ by Eq. 3.11, the rate of change of $W(\boldsymbol{x}, \dot{\boldsymbol{x}}, s)$ becomes:

$$\dot{W}(\boldsymbol{x}, \dot{\boldsymbol{x}}, s) = \left(\beta'_r(s, p_r) - \beta_r(s, p_r)\right)p_r + \left(\beta'_n(s, p_n) - \beta_n(s, p_n)\right)p_n - \left(1 - \alpha(s)\right)p_d + \dot{\boldsymbol{x}}^{\boldsymbol{T}}\boldsymbol{F_{ext}} \tag{3.15}$$

The first two terms are now both non-positives, while the third one remains dissipative since $1 - \alpha(s) \geq 0$. As a result, the full system is passive with respect to $\dot{\boldsymbol{x}}^{\boldsymbol{T}}\boldsymbol{F_{ext}}$.

3.3 Extension to a Bi-Manual Reaching, Grasping, and Manipulation Task

3.3.1 Formulation

In this section, we investigate how the modulation strategy can be applied to the challenging task of reaching, grasping and manipulating an object using two robotic arms. To start, let us consider the scenario illustrated in Figure 3.4. The main variables used to describe the problem are provided in Table 3.1. For the rest of this section, superscript L and R will refer to the left and right robot respectively.

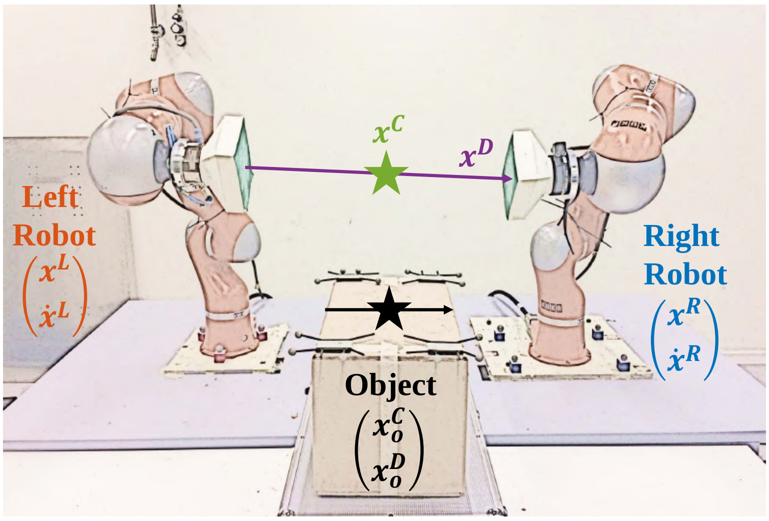

x^C
x^D
Left Robot
$\begin{pmatrix} x^L \\ \dot{x}^L \end{pmatrix}$
Right Robot
$\begin{pmatrix} x^R \\ \dot{x}^R \end{pmatrix}$
Object
$\begin{pmatrix} x_o^C \\ x_o^D \end{pmatrix}$

Figure 3.4 – Scenario for reaching, grasping and manipulating an object with two robotic arms.

$\boldsymbol{x}^L, \dot{\boldsymbol{x}}^L, \dot{\boldsymbol{x}}_d^L$	Left robot tool position, velocity and desired dynamics
$\boldsymbol{x}^R, \dot{\boldsymbol{x}}^R, \dot{\boldsymbol{x}}_d^R$	Right robot tool position, velocity and desired dynamics
$\boldsymbol{x}_o^C$ and $\boldsymbol{x}_o^D$	Measured object center position and dimension vector
$\boldsymbol{x}^C$ and $\boldsymbol{x}^D$	Measured center position and distance vector between the two robots
$\boldsymbol{x}_d^C$ and $\boldsymbol{x}_d^D$	Desired center position and distance vector
$\dot{\boldsymbol{x}}_d^C$ and $\dot{\boldsymbol{x}}_d^D$	Desired center position and distance vector dynamics

Table 3.1 – Main variables used to describe the reaching, grasping and manipulation task with two robots.

The robots' center position $\boldsymbol{x}^C$ and distance vector $\boldsymbol{x}^D$ are computed from their tool tip positions $\boldsymbol{x}^R$ and $\boldsymbol{x}^L$:

$$\begin{cases} \boldsymbol{x}^C = \dfrac{\boldsymbol{x}^L + \boldsymbol{x}^R}{2} \\ \boldsymbol{x}^D = \boldsymbol{x}^R - \boldsymbol{x}^L \end{cases} \tag{3.16}$$

from where we can derive the relations below:

$$\begin{cases} \dot{\boldsymbol{x}}^R = \dot{\boldsymbol{x}}_C + \dfrac{\dot{\boldsymbol{x}}^D}{2} \\ \dot{\boldsymbol{x}}^L = \dot{\boldsymbol{x}}_C - \dfrac{\dot{\boldsymbol{x}}^D}{2} \end{cases} \tag{3.17}$$

To reach and manipulate the object during the task, we choose to couple the robots' motion by controlling for a desired robots' centre position $\boldsymbol{x}_d^C$ and distance vector $\boldsymbol{x}_d^D$, using simple linear dynamics:

$$\begin{cases} \dot{\boldsymbol{x}}_d^C = A_C \left(\boldsymbol{x}_d^C - \boldsymbol{x}^C\right) \\ \dot{\boldsymbol{x}}_d^D = A_D \left(\boldsymbol{x}_d^D - \boldsymbol{x}^D\right) \end{cases} \tag{3.18}$$

where $\boldsymbol{A_C}$ and $\boldsymbol{A_D}$ are positive gain diagonal matrices. Basically, $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}}$ specifies the desired positioning behavior of the robots' center while $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}}$ defines the desired closing behavior on the object's surface. $\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{C}}$ and $\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}}$ can be respectively set to $\boldsymbol{x}_{\boldsymbol{o}}^{\boldsymbol{C}}$ and $\boldsymbol{x}_{\boldsymbol{o}}^{\boldsymbol{D}}$ during the reaching phase and modified during the manipulation phase.

To do the grasping part of the task, we use the modulation strategy presented in section 3.2.2. First, we introduce the nominal DS $\boldsymbol{f}^{\boldsymbol{R}}(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}})$ and $\boldsymbol{f}^{\boldsymbol{L}}(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}})$ which bring both robots in contact with the target surface (e.g., the object's surface):

$$\begin{cases} \boldsymbol{f}^{\boldsymbol{R}}(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}}) = \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}} + \dfrac{\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}}}{2} \\ \boldsymbol{f}^{\boldsymbol{L}}(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}}) = \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}} - \dfrac{\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}}}{2} \end{cases} \tag{3.19}$$

Once the robots reach the object's surface, they should generate the desired contact force profile $F_d(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}}) \geq 0$ which is assumed to be the same for both of them. To this end, the normal modulation terms are defined as follows:

$$\boldsymbol{f}_{\boldsymbol{n}}^{\boldsymbol{i}}(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}}) = \frac{F_d(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}})}{d_1^i}\boldsymbol{n}^{\boldsymbol{i}} \quad i = L,R \tag{3.20}$$

The force application directions $\boldsymbol{n}^{\boldsymbol{R}}$ and $\boldsymbol{n}^{\boldsymbol{L}}$ are derived from the desired distance vector $\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}}$. For a box (with two parallel surfaces), they are opposite for the two robots:

$$\boldsymbol{n}^{\boldsymbol{L}} = \text{-}\boldsymbol{n}^{\boldsymbol{R}} = \frac{\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}}}{\|\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}}\|} \tag{3.21}$$

From there, the desired robots' velocity can be finally expressed:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} = \boldsymbol{f}^{\boldsymbol{i}}(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}}) + \boldsymbol{f}_{\boldsymbol{n}}^{\boldsymbol{i}}(\boldsymbol{x}^{\boldsymbol{L}},\boldsymbol{x}^{\boldsymbol{R}}) \quad i = L,R \tag{3.22}$$

The modulated DS are then tracked with the DS impedance controller (Eq. 2.16).

3.3.2 Passivity Analysis

Guaranteeing passivity of the bi-manual system is difficult in particular if the desired dynamics of both robots are coupled. However, let us assume that the desired dynamics $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}}$ and $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}}$ are both conservatives (such as the linear dynamics specified in Eq. 3.18), i.e.:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} = -\nabla_{\boldsymbol{x}^{\boldsymbol{i}}} V_i(\boldsymbol{x}^{\boldsymbol{i}}) \quad i = C,D \tag{3.23}$$

where $V_C(\boldsymbol{x}^{\boldsymbol{C}})$ and $V_D(\boldsymbol{x}^{\boldsymbol{D}})$ are potential functions. Setting the same impedance gains for both robots: $d_1^L = d_1^R = d_1$, one can consider similarly to Eq. 2.18 a global storage function W taking into account both robots' kinetic energy and the potential functions such that:

$$W = \frac{1}{2}\left(\dot{\boldsymbol{x}}^{\boldsymbol{R}^T} \boldsymbol{M}_{\boldsymbol{x}}^{\boldsymbol{R}}(\boldsymbol{q}^{\boldsymbol{R}})\dot{\boldsymbol{x}}^{\boldsymbol{R}} + \dot{\boldsymbol{x}}^{\boldsymbol{L}^T} \boldsymbol{M}_{\boldsymbol{x}}^{\boldsymbol{L}}(\boldsymbol{q}^{\boldsymbol{L}})\dot{\boldsymbol{x}}^{\boldsymbol{L}}\right) + 2d_1 V_C(\boldsymbol{x}^{\boldsymbol{C}}) + \frac{d_1}{2} V_D(\boldsymbol{x}^{\boldsymbol{D}}) \tag{3.24}$$

where $\boldsymbol{M}_{\boldsymbol{x}}^{\boldsymbol{R}}(\boldsymbol{q}^{\boldsymbol{R}})$ and $\boldsymbol{M}_{\boldsymbol{x}}^{\boldsymbol{L}}(\boldsymbol{q}^{\boldsymbol{L}})$ are the robots' cartesian space linear inertia matrices, while $\boldsymbol{q}^{\boldsymbol{R}}$ and $\boldsymbol{q}^{\boldsymbol{L}}$ are the joint state position vectors.

Differentiating Eq. 3.24 using Eq. 3.23 leads to:

$$\dot{W} = \frac{1}{2}\frac{d}{dt}\left(\dot{\boldsymbol{x}}^{\boldsymbol{R}^T} \boldsymbol{M}_{\boldsymbol{x}}^{\boldsymbol{R}}(\boldsymbol{q}^{\boldsymbol{R}})\dot{\boldsymbol{x}}^{\boldsymbol{R}} + \dot{\boldsymbol{x}}^{\boldsymbol{L}^T} \boldsymbol{M}_{\boldsymbol{x}}^{\boldsymbol{L}}(\boldsymbol{q}^{\boldsymbol{L}})\dot{\boldsymbol{x}}^{\boldsymbol{L}}\right) - 2d_1 \dot{\boldsymbol{x}}^{\boldsymbol{C}^T} \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}} - \frac{d_1}{2} \dot{\boldsymbol{x}}^{\boldsymbol{D}^T} \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}} \tag{3.25}$$

From the expressions of $\dot{\boldsymbol{x}}^{\boldsymbol{C}}$ and $\dot{\boldsymbol{x}}^{\boldsymbol{D}}$ in Eq. 3.17, it becomes:

$$\dot{W} = \frac{1}{2}\frac{d}{dt}\left(\dot{\boldsymbol{x}}^{\boldsymbol{R}^T} \boldsymbol{M}_{\boldsymbol{x}}^{\boldsymbol{R}}(\boldsymbol{q}^{\boldsymbol{R}})\dot{\boldsymbol{x}}^{\boldsymbol{R}} + \dot{\boldsymbol{x}}^{\boldsymbol{L}^T} \boldsymbol{M}_{\boldsymbol{x}}^{\boldsymbol{L}}(\boldsymbol{q}^{\boldsymbol{L}})\dot{\boldsymbol{x}}^{\boldsymbol{L}}\right) - d_1 \left(\dot{\boldsymbol{x}}^{\boldsymbol{R}} + \dot{\boldsymbol{x}}^{\boldsymbol{L}}\right)^{\boldsymbol{T}} \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}} - \frac{d_1}{2} \left(\dot{\boldsymbol{x}}^{\boldsymbol{R}} - \dot{\boldsymbol{x}}^{\boldsymbol{L}}\right)^{\boldsymbol{T}} \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}} \tag{3.26}$$

Expanding the time derivative of the robots' kinetic energy similarly to Eq. 3.6 results in:

$$\begin{aligned} \dot{W} = {} & d_1 \dot{\boldsymbol{x}}^{\boldsymbol{R}^T} \boldsymbol{f}^{\boldsymbol{R}}(\boldsymbol{x}^{\boldsymbol{L}}, \boldsymbol{x}^{\boldsymbol{R}}) + d_1 \dot{\boldsymbol{x}}^{\boldsymbol{R}^T} \boldsymbol{f}_{\boldsymbol{n}}^{\boldsymbol{R}}(\boldsymbol{x}^{\boldsymbol{L}}, \boldsymbol{x}^{\boldsymbol{R}}) - \dot{\boldsymbol{x}}^{\boldsymbol{R}^T} \boldsymbol{D}^{\boldsymbol{R}}(\boldsymbol{x}^{\boldsymbol{L}}, \boldsymbol{x}^{\boldsymbol{R}}) \dot{\boldsymbol{x}}^{\boldsymbol{R}} + \dot{\boldsymbol{x}}^{\boldsymbol{R}^T} \boldsymbol{F}_{\boldsymbol{ext}}^{\boldsymbol{R}} \\ & + d_1 \dot{\boldsymbol{x}}^{\boldsymbol{L}^T} \boldsymbol{f}^{\boldsymbol{L}}(\boldsymbol{x}^{\boldsymbol{L}}, \boldsymbol{x}^{\boldsymbol{R}}) + d_1 \dot{\boldsymbol{x}}^{\boldsymbol{L}^T} \boldsymbol{f}_{\boldsymbol{n}}^{\boldsymbol{L}}(\boldsymbol{x}^{\boldsymbol{L}}, \boldsymbol{x}^{\boldsymbol{R}}) - \dot{\boldsymbol{x}}^{\boldsymbol{L}^T} \boldsymbol{D}^{\boldsymbol{L}}(\boldsymbol{x}^{\boldsymbol{L}}, \boldsymbol{x}^{\boldsymbol{R}}) \dot{\boldsymbol{x}}^{\boldsymbol{L}} + \dot{\boldsymbol{x}}^{\boldsymbol{L}^T} \boldsymbol{F}_{\boldsymbol{ext}}^{\boldsymbol{L}} \\ & - d_1 \left(\dot{\boldsymbol{x}}^{\boldsymbol{R}} + \dot{\boldsymbol{x}}^{\boldsymbol{L}}\right)^{\boldsymbol{T}} \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}} - \frac{d_1}{2} \left(\dot{\boldsymbol{x}}^{\boldsymbol{R}} - \dot{\boldsymbol{x}}^{\boldsymbol{L}}\right)^{\boldsymbol{T}} \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}} \end{aligned} \tag{3.27}$$

Cancelling the terms with $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}}$ and $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}}$ using Eq. 3.19 finally leads to:

$$\dot{W} = \dot{W}^R + \dot{W}^L \tag{3.28}$$

where:

$$\dot{W}^i = d_1 \dot{\boldsymbol{x}}^{\boldsymbol{i}^T} \boldsymbol{f}_{\boldsymbol{n}}^{\boldsymbol{i}}(\boldsymbol{x}^{\boldsymbol{L}}, \boldsymbol{x}^{\boldsymbol{R}}) - \dot{\boldsymbol{x}}^{\boldsymbol{i}^T} \boldsymbol{D}^{\boldsymbol{i}}(\boldsymbol{x}^{\boldsymbol{L}}, \boldsymbol{x}^{\boldsymbol{R}}) \dot{\boldsymbol{x}}^{\boldsymbol{i}} + \dot{\boldsymbol{x}}^{\boldsymbol{i}^T} \boldsymbol{F}_{\boldsymbol{ext}}^{\boldsymbol{i}} \tag{3.29}$$

$\dot{W}^i$ with $i = \{L, R\}$ is equivalent to Eq. 3.6 without the non-conservative term. Thus, to guarantee the stability of the bi-manual system, we can use the energy tank approach derived in section 3.2.3. To this end, two tanks s_R and s_L can be defined to make $\dot{W}^R$ and $\dot{W}^L$ passive with respect to $\dot{\boldsymbol{x}}^{\boldsymbol{R}^T} \boldsymbol{F}_{\boldsymbol{ext}}^{\boldsymbol{R}}$ and $\dot{\boldsymbol{x}}^{\boldsymbol{L}^T} \boldsymbol{F}_{\boldsymbol{ext}}^{\boldsymbol{L}}$.

3.4 Experimental Evaluations

In this section, we evaluate the DS-based strategy proposed in two real world scenarios: a) polishing of a non-flat surface using a single robotic arm and b) reaching, grasping and manipulating an object with two robotic arms. We assess the ability of the approach to generate the desired force profile and to do so in the different types of disturbances, by moving

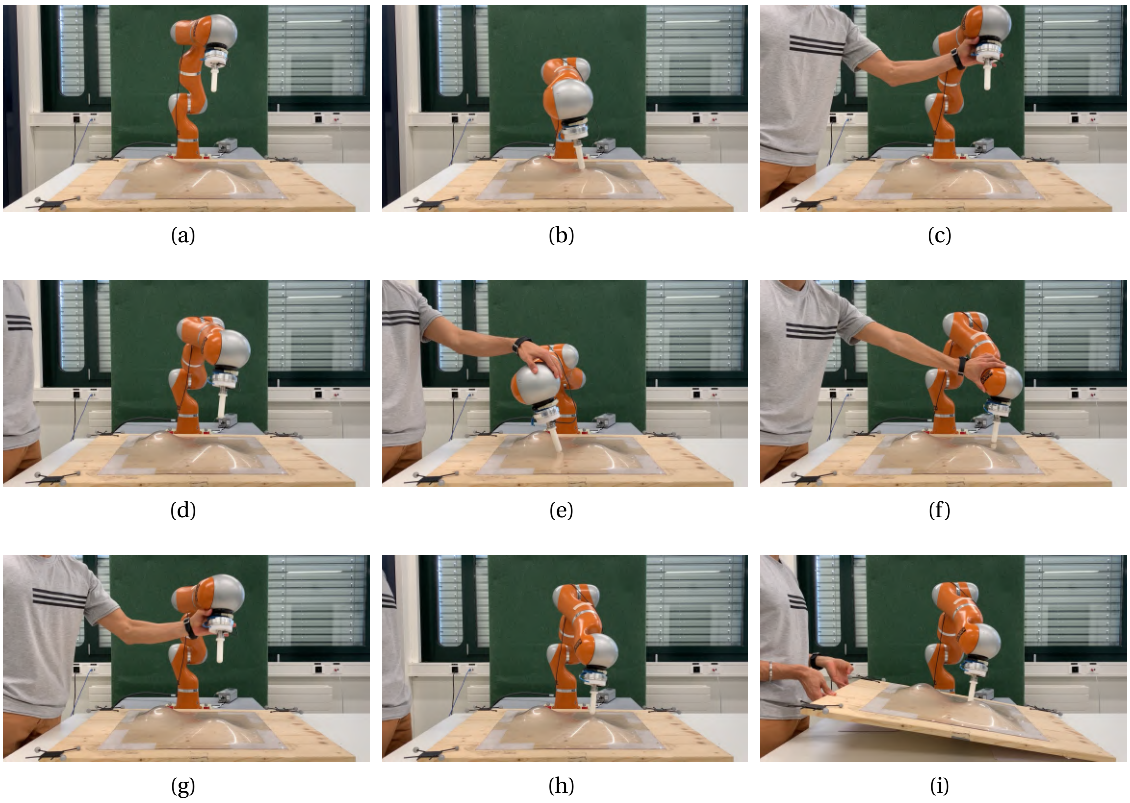

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 3.5 – Snapshots of the circular polishing task on a non-flat surface. Starting from an initial position (Figure 3.5a), the robot comes in contact with a surface to perform a circular motion on it (Figure 3.5b). After a bit, the human breaks the contact with the surface and pulls the robot away from the surface (Figure 3.5c). Once the disturbance disappears, the robot comes back to the surface (Figure 3.5d). Then, the human applies disturbances to the robot while the robot is moving in contact with the surface (Figure 3.5e and 3.5f). Finally, after breaking the contact (Figure 3.5g) and letting the robot reaching the surface again (Figure 3.5h), the human applies disturbances to the surface by changing its inclination (Figure 3.5i).

unexpectedly the surface/object prior and during contact, or breaking the contact. These experimental evaluations can be watched in a video available online (see the link in section 1.8)

3.4.1 Polishing Task on a Non-Flat Surface

The DS modulation strategy is first tested in a circular polishing task on a non-flat surface. A 7-DoF robotic arm (KUKA LWR IV+) is used to perform the task. The robot is equipped with joint torque sensors at the actuators and can be torque-controlled. A 6-axis ATI force-torque sensor is also mounted on the end-effector on which a 3D printed finger tool is attached. The non-flat and rigid surface is fabricated by deforming a Plexiglas sheet using heat. It is attached on a wooden plate whose pose is tracked by a motion capture system. The robot's behavior is

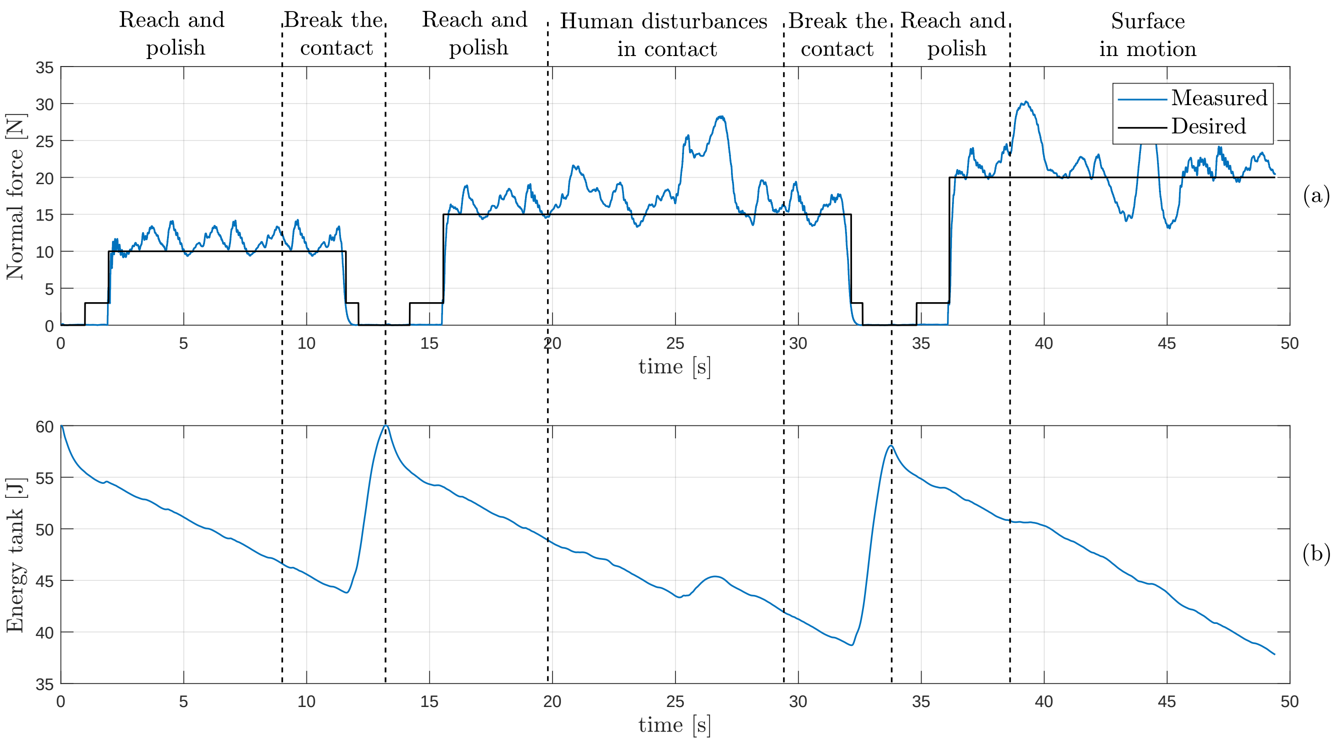


Figure 3.6 – Polishing task under various human disturbances: (a) Measured versus desired normal force / (b): Robot's energy tank s (section 3.2.3).

evaluated in a simple scenario: the robot comes in contact with the target surface to perform a circular motion on the surface while applying the desired contact force, and experiences disturbances from a human. Snapshots of the experiment are provided in Figure 3.5. The technical details of the implementation are provided in Appendix A.1 and A.2).

Figure 3.6a shows the measured and desired force profiles recorded during the experiment. The robot first reaches the surface to perform the polishing task without experiencing any disturbances. The force generation is relatively accurate with a RMS force error of around 1.9 N (19% of the desired force) during this period. After a while, the human intentionally breaks the contact with the surface by pulling the robot away from the surface. No instabilities are observed during this phase. Once released by the human, the robot simply returns back to the surface following the flow of the DS to perform the task. Then, the human interacts with the system by pushing and stalling the robot while the robot is in contact with the surface. The measured force remains smooth denoting a safe and stable interaction. Finally, after breaking the contact a second time, the robot reaches the surface again and the human momentarily changes the inclination of the surface. The robot smoothly complies to the disturbance without getting unstable.

Figure 3.6b depicts the time evolution of the energy tank during the experiment. The energy tank is initialized to 60 J which is defined as the maximum allowed level for this task. The

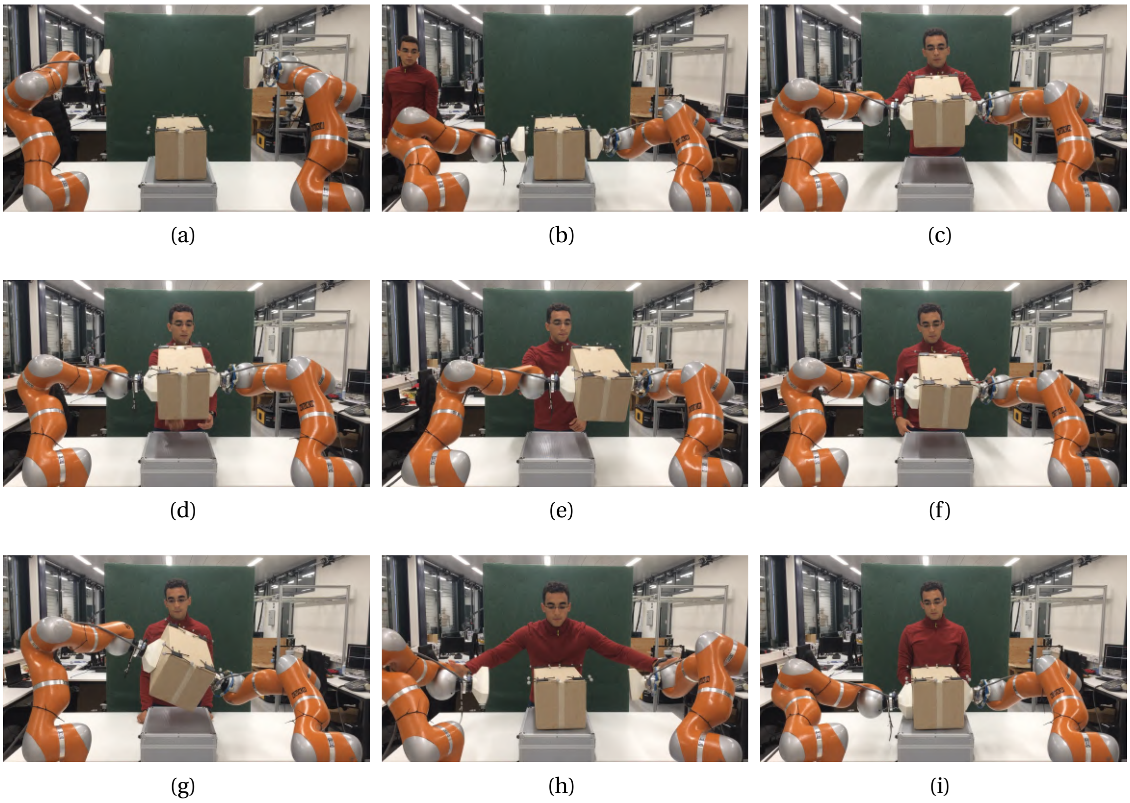


Figure 3.7 – Snapshots of the bi-manual reaching and grasping task. Starting from their initial positions (Figure 3.7a), both robots reach the object to grasp it (Figure 3.7b). Once the object is grasped, the human lifts the robots to move the object up (Figure 3.7c). From there, the human applies fast shocks on the object (Figure 3.7d). After that, the human physically interacts with the robots to move the object to the left (Figure 3.7e), then to the right (Figure 3.7f), and to change its orientation (Figure 3.7g) Finally, the human breaks the grasp by pushing the robots away from each other (Figure 3.7h) before letting them reaching and grasping the object again (Figure 3.7i).

reason for this high value is that the nominal DS is considered to be purely non-conservative (see the explicit definition in Appendix A.2), i.e., the energy tank is used both to generate the desired motion and contact forces. In other words, to reach the surface and move on it while generating contact forces, energy should be always extracted from the tank. Therefore, we chose the initial tank value relatively high so that the robot could perform the task during at least 1 min in the case there are no human disturbances. As can observed in Figure 3.6b, the energy tank is mostly decreasing during the experiment. However, when the human applies disturbances to the robot, such as pushing the robot while it is moving on the surface or pulling the robot away from the surface, large amounts of energy are dissipated and directly used to fill the tank.

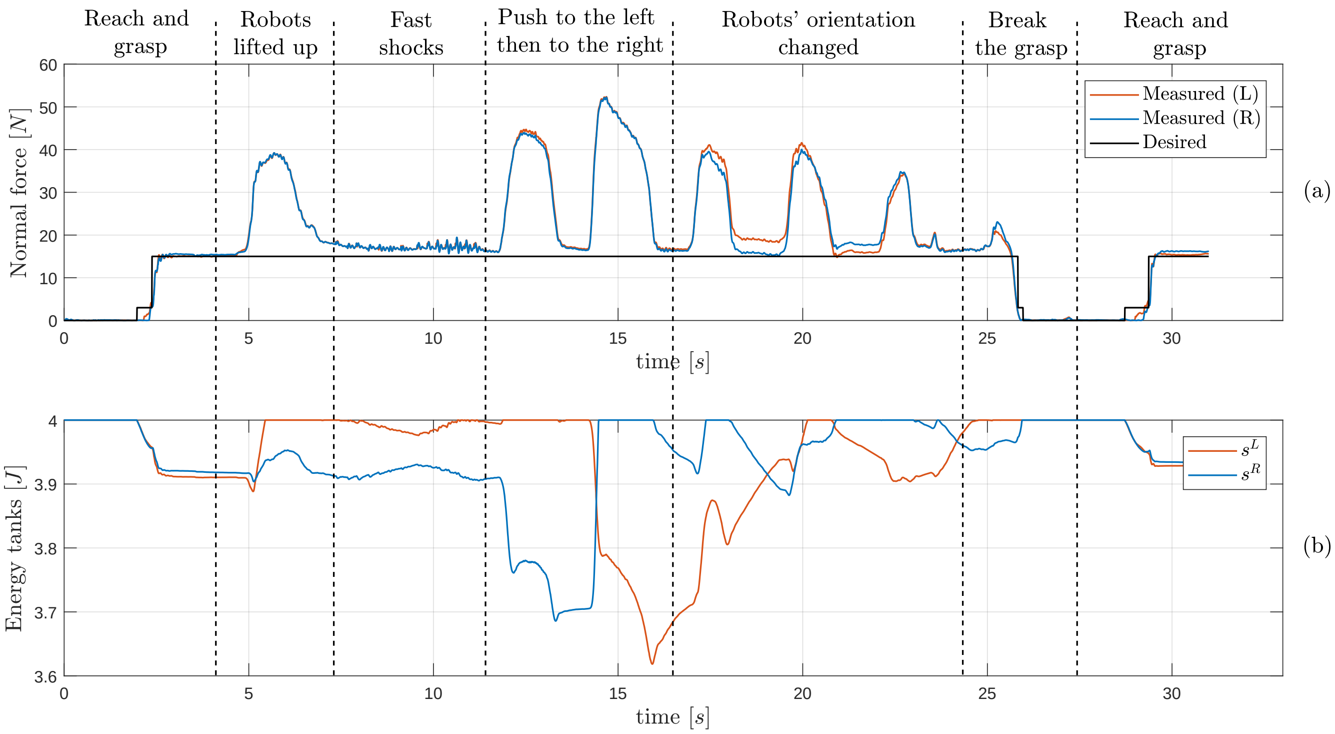


Figure 3.8 – Reaching and grasping task under various human disturbances: (a): Measured and desired normal force / (b): Robots' energy tank s_L and s_R (refer to section 3.3.2).

3.4.2 Bi-Manual Reaching, Grasping and Manipulation Tasks

3.4.2.1 Reaching and Grasping an Object

A second experimental evaluation is done with two KUKA LWR IV+ robots to reach and grasp a cardboard box. The box has a mass of 0.65 ± 0.05 kg and is tracked by the motion capture system to get its pose. Both robots are equipped with a 6-axis ATI force-torque sensor at the end-effector on which a flat palm is mounted for grasping. The evaluation scenario is designed such that the two arms reach and grasp the object before a human comes and interacts with the system by moving it around, changing its orientation or even breaking the grasp. Snapshots of the experiment are provided in Figure 3.7. The implementation is based on section 3.3.1 while the technical details are provided in Appendix A.1 and A.3.

Figure 3.8a illustrates the measured and desired contact forces. The RMS force error when the object is grasped and without human disturbances is around 1.7 N (11.3% of the desired force) for both robots. The non-contact/contact transition in the reaching and grasping phases is smooth and no instability is observed in the force profiles when the human intentionally breaks the grasp. Similarly, despite disturbances applied on the system after grasping (e.g., fast shocks on the box, changing the system pose), the measured forces remain smooth guaranteeing the stability and delivering a satisfactory compliant behavior.

Figure 3.8b depicts the behavior of the energy tanks for both robots. The tanks are initialized at the maximum allowed level which is set to 4.0 J. Note that this level is much lower than for the polishing task as here energy is only extracted from the tanks to implement the non-passive actions due to the generation of the desired grasping forces (see section 3.3.2). When the robots are initially moving toward the object, energy is mainly dissipated. However, this dissipated energy cannot be stored in the tanks since they are already full. Close to contact, a desired contact force starts to be generated while the robots are still slightly moving. These non-passive actions are implemented by extracting energy from the tanks. Once the object is grasped, the tanks levels remain constant until the human moves the robots to lift the object. This dissipated energy is stored in the tanks but in a non-symmetrical way due to the interaction. When the human applies fast shocks on the object, the tanks level is barely changing as the robots barely move. Then, moving the system to the left direction (from the human's point of view) causes the right arm to generate extra energy as it moves in the direction where it applies the force while the left robot dissipates energy. A high amount of energy is extracted from the tank of the right robot to execute this non-passive action and maintain the grasp. When pushing the system to the right, the opposite behavior happens with energy being generated by the left robot and dissipated by the right one, leading their associated tanks to be respectively drained and filled. A similar reasoning can be applied to the other disturbance phases where the human moves the arms to change the object's orientation or breaks the grasp.

3.4.2.2 Reaching, Grasping, and Moving an Object

A final experimental evaluation is done with the bi-manual setup where both robotic arms first reach, grasp and lift the cardboard box to a predefined attractor and with a predefined orientation. Then, through a graphical interface, the human changes the location of the attractor to move the object from left to right and back to the center. Robustness to physical disturbances is also demonstrating through the human physically interacting with the robots while they are holding the object to the desired location. Snapshots of the experiment are provided in Figure 3.9 while the technical details of the implementation are provided in Appendix A.1 and A.4.

Figure 3.10 shows the measured and desired contact forces during the task (Figure 3.10a) as well as the behaviour of the energy tanks (Figure 3.10b). In comparison to before, the force tracking accuracy is reduced. This can be partly attributed to the desired dynamics for the center and distance vector which are subject to the uncertainties in the dimension and pose of the object as well as in the pose of the robots. These uncertainties in the nominal DS can create additional holding/repulsing forces which result in the robots applying more or less contact forces than desired. This behaviour can be observed when changing the attractor (i.e., $\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{C}}$ in Eq. 3.18) on the fly. Nonetheless, the force tracking accuracy still remains acceptable. Indeed, from the time where the object is grasped and lifted until the time where disturbances

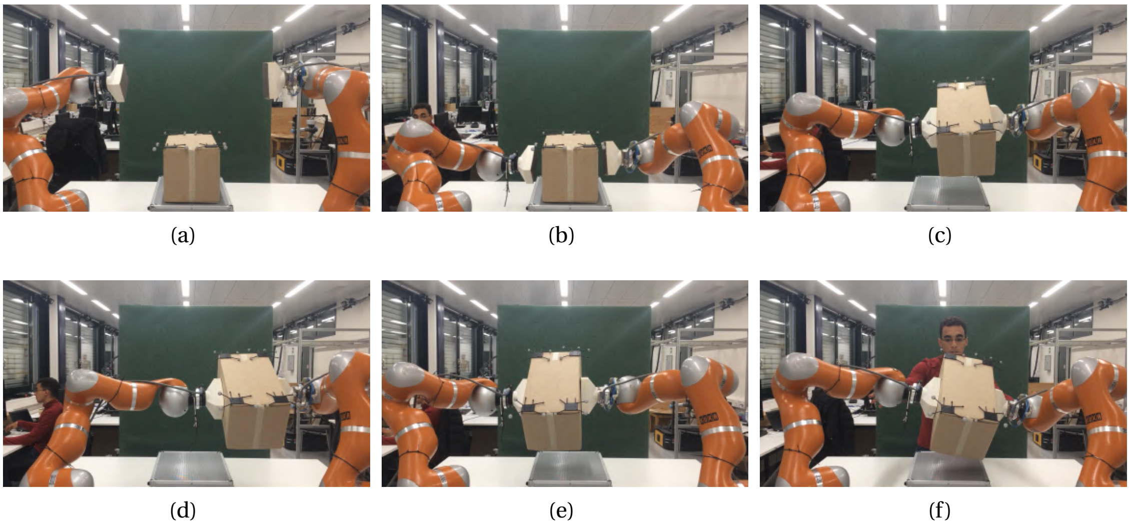


Figure 3.9 – Snapshots of the bi-manual reaching and grasping task. Starting from their initial positions (Figure 3.9a), both robots reach the object (Figure 3.9b), grasp it, and lift to a predefined attractor (Figure 3.9c). Through a graphical interface, the human changes the attractor position to move the object from left (Figure 3.9d) to right (Figure 3.9e) and back to the center. The human finally disturbs the robots through physical interaction (Figure 3.9f).

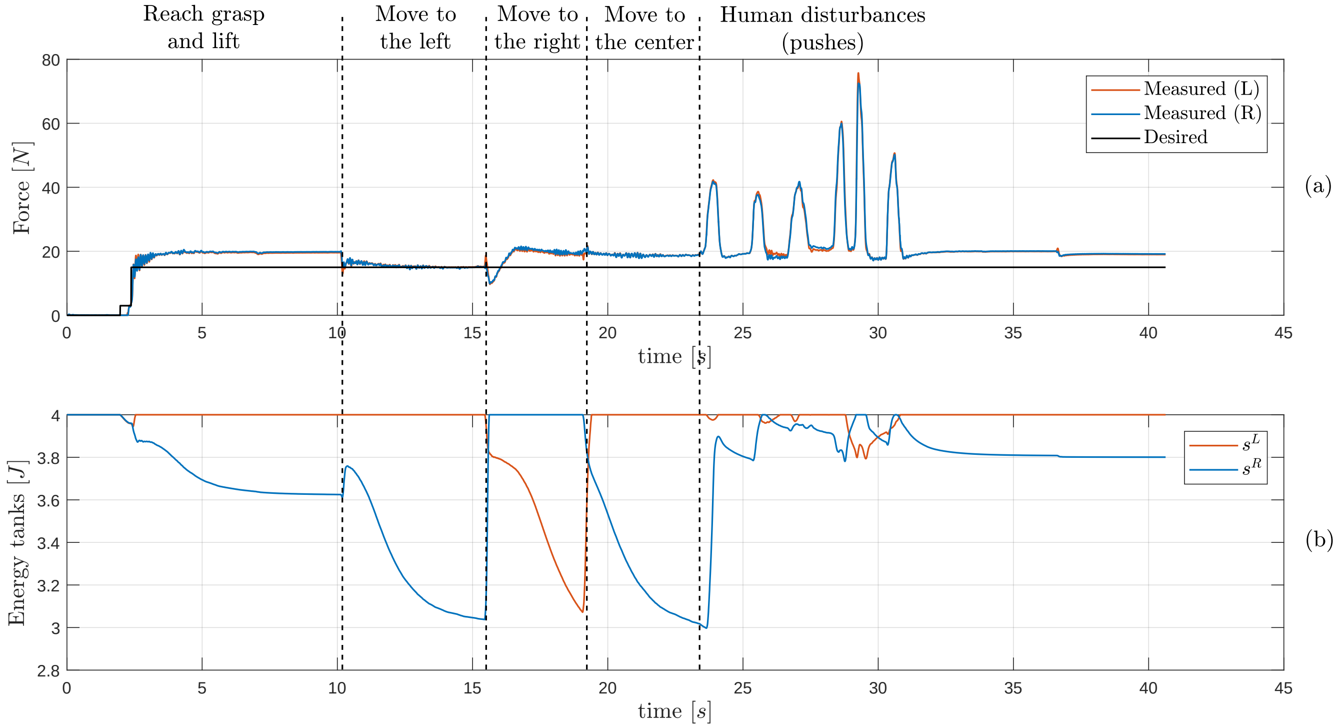


Figure 3.10 – Reaching, grasping and manipulation task under human disturbances: (a): Measured and desired normal force / (b): Robots' energy tank s_L and s_R (refer to section 3.3.2).

start (i.e., from 4 to 23 *s* approximately), the RMS force error is around around 3.79 *N* (25.3% of the desired force) for both robots. Otherwise, we can see that the grasp is maintained even in case of human disturbances (pushes). The system can keep the grasp while providing stability and compliance against external disturbances.

Concerning the energy tanks, they behave similarly to Figure 3.8b. In particular, the energy tank of one robot decreases when moving in the direction where it applies contact forces, while it decreases when moving in the opposite direction.

3.5 Discussion and Conclusion

In this chapter, we proposed a Dynamical System approach to generate motion and force in contact tasks with robotic manipulators. The strategy is based on local modulation of a nominal DS that allows to reach a target surface and move on it. The use of the DS framework provides flexibility and smoothness in the motion and force generation as well as robustness to real-time disturbances. As in traditional impedance controller, the generation of the contact forces is achieved implicitly and results in a relatively accurate force tracking performance as demonstrated in the experimental results, despite the absence of force feedback. To preserve passivity, we used the concept of energy tanks to monitor the energy flow and execute non-passive actions as long as energy is available in the tanks. The experiments confirmed our theoretical proofs for stability and passivity where the robots' behavior were smooth and compliant under drastic human disturbances (such as breaking the contact).

The method assumes in return that the dynamics of the robot (gravity etc.) are compensated and the contact surface is approximately known. Assuming such conditions in practice is not always possible and is reasonable as long as the errors in motion and force are small with respect to the task tolerances. This is the case in the two real-world evaluation scenarios presented in this chapter. However, if the errors are too large, the task cannot be achieved properly anymore. For contact tasks, these errors mainly come from uncertainties in the environment (e.g, location of the surface, normal of the surface, friction), robot model, measurement noises and other unmodeled dynamics. Most of these uncertainties are however structural and can be modelled or corrected for. The robot can learn or adapt online to these uncertainties through the interaction with the surface. In the next chapter, we explore such directions and investigate how we can adapt our DS-based strategy for contact tasks to deal with uncertainties and improve the tracking of the task.

4 Contact Force Adaptation in Uncertain Environments

Note: The material presented in this chapter is adopted from:
Amanhoud,W., Khoramshahi, M., Bonnesoeur, M., and Billard, A. (2020). Force adaptation in contact tasks with dynamical systems. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 6841–6847. IEEE.

4.1 Introduction

Many tasks require robots to enter in contact with arbitrary surfaces, move on them while applying desired contact forces. For finishing operations such as polishing or grinding, the contact force exerted on the surface is a key process variable, as the amount of force applied directly affects the material removal rate (Erlbacher (2000)). Besides following accurately the desired motion and force profile, the robots should also be compliant to deal with inaccurate modelling of the surfaces (e.g., stiffness and location) and real-time disturbances such as those introduced by humans or unexpected changes in the environment (Lefebvre et al. (2005)). This requires the development of suitable control strategies to regulate the robot movement and interaction forces with the environment. For this purpose, in chapter 3, we proposed a simple control strategy based on the time-invariant dynamical system framework. This strategy allows a robot to perform contact tasks while reacting to human interactions (e.g., stopping the robot, breaking the contact, and moving the robot arbitrarily), or unexpected changes in the environment (e.g., the position and orientation of the surface/object). However, the generation of the desired contact force is done in open loop without any force feedback. To be accurate, it assumes that the robot's dynamics (gravity, inertia, etc) are compensated for, the contact surface is known (stiffness, shape, location, friction) and the measurement noises little. In practice, these entities are subject to uncertainties which result in errors in the force produced at the end-point (see the force tracking plots of the previous chapter in section 3.4 as examples). These errors can often present a structural shape that can be learned and compensated for. This is particularly true for repetitive tasks where the robot needs to repeat the same motion while maintaining a desired force profile (e.g., see the polishing task in section 3.4.1 where a force error pattern can be identified in Figure 3.6a from 2 *s* to 10 *s* approximately). The robot can use the force error measured through the interaction with the surface to adapt online a compensation model.

In this chapter, we extend the approach presented in chapter 3 to deal with uncertainties and improve the force tracking performance through online learning of a state-dependent force correction model. The advantage of learning a state-dependent force compensation is that the online adaptation can be deactivated once the error has been sufficiently reduced and the learned profile can be re-used later without the need to re-learn. We present our method in section 4.2, we evaluate it in section 4.4, and we conclude with a discussion about the method and results obtained in section 4.5.

4.2 Method

4.2.1 Force Adaptation with Dynamical Systems

Let us consider a state-dependent force correction model $\hat{F}(\boldsymbol{x},\boldsymbol{\theta}) \in [\hat{F}_{-},\hat{F}_{+}]$ with $\hat{F}_{-} < 0$ and $\hat{F}_{+} > 0$ where $\boldsymbol{\theta} = [\theta_1,\theta_2,...,\theta_K]^T$ is the associated set of parameters of size K. This correction is added to the normal modulation term $\boldsymbol{f_n(x)}$ introduced in Eq. 3.3 such that:

$$\boldsymbol{f_n}(\boldsymbol{x}) = \frac{F_d(\boldsymbol{x}) + \hat{F}(\boldsymbol{x},\boldsymbol{\theta})}{d_1}\boldsymbol{n}(\boldsymbol{x}) \tag{4.1}$$

Substituting Eq. 4.1 in the modified force control law of Eq. 3.4 provides:

$$\boldsymbol{F_c} = d_1\boldsymbol{f}(\boldsymbol{x}) + \big(F_d(\boldsymbol{x}) + \hat{F}(\boldsymbol{x},\boldsymbol{\theta})\big)\boldsymbol{n}(\boldsymbol{x}) - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} + \boldsymbol{g}(\boldsymbol{x}) \tag{4.2}$$

Inserting Eq. 4.2 in the three dimensional linear cartesian space dynamics (Eq. 2.13) gives:

$$\boldsymbol{M_x}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{C_x}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}} = d_1\boldsymbol{f}(\boldsymbol{x}) + \big(F_d(\boldsymbol{x}) + \hat{F}(\boldsymbol{x},\boldsymbol{\theta})\big)\boldsymbol{n}(\boldsymbol{x}) - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} + \boldsymbol{F_{ext}} \tag{4.3}$$

Let us now introduce the measured contact forces $\boldsymbol{F_m} \in \mathbb{R}^3$ acting in the opposite direction of $\boldsymbol{F_{ext}}$. Eq. 4.3 can be re-written as:

$$F_d(\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x}) - \boldsymbol{F_m} = \boldsymbol{M_x}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{C_x}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}} + \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} - \hat{F}(\boldsymbol{x},\boldsymbol{\theta})\boldsymbol{n}(\boldsymbol{x}) \tag{4.4}$$

Projecting Eq. 4.4 along $\boldsymbol{n}(\boldsymbol{x})$, and considering that the robot is in contact (Eq. 3.1 holds) leads to:

$$F_d(\boldsymbol{x}) - \boldsymbol{n}(\boldsymbol{x})^{\boldsymbol{T}}\boldsymbol{F_m} = \boldsymbol{n}(\boldsymbol{x})^{\boldsymbol{T}}\big(\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x},\dot{\boldsymbol{x}})\dot{\boldsymbol{x}}\big) - \hat{F}(\boldsymbol{x},\boldsymbol{\theta}) + \boldsymbol{n}(\boldsymbol{x})^{\boldsymbol{T}}\boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{4.5}$$

Our objective is to minimize the error $\tilde{F}$ between the desired and measured contact force along $\boldsymbol{n}(\boldsymbol{x})$, defined as:

$$\tilde{F} = F_d(\boldsymbol{x}) - \boldsymbol{n}(\boldsymbol{x})^T\boldsymbol{F_m} \tag{4.6}$$

To this end, one can use the cost function $J = \frac{1}{2}\tilde{F}^2$ whose negative gradient can be used to update the parameters $\boldsymbol{\theta}$ of $\hat{F}(\boldsymbol{x},\boldsymbol{\theta})$:

$$\dot{\boldsymbol{\theta}} = -\epsilon_r \frac{\partial J}{\partial \boldsymbol{\theta}} = -\epsilon_r \frac{\partial J}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial \hat{F}} \frac{\partial \hat{F}}{\partial \boldsymbol{\theta}} = \epsilon_r \tilde{F} \frac{\partial \hat{F}}{\partial \boldsymbol{\theta}} \tag{4.7}$$

where $\epsilon_r > 0$ is the adaptation rate.

4.2.2 Design of the State-Dependent Force Correction Model

To design our state-dependent force correction, we opt for a normalized linear combination of Gaussian Radial Basis kernel functions which is a well-established method in machine learning (Bishop (2006)). Radial Basis Functions (RBF) are often used in applications needing approximation or interpolation especially for their smoothness and nice convergence properties, making them suitable for online adaptation (Buhmann (2003)). Thus, we design $\hat{F}(\boldsymbol{x},\boldsymbol{\theta})$ as follows:

$$\hat{F}(\boldsymbol{x},\boldsymbol{\theta}) = \frac{\sum_{i=1}^{K} \theta_i \phi(\boldsymbol{x} - \boldsymbol{c_i})}{\sum_{j=1}^{K} \phi(\boldsymbol{x} - \boldsymbol{c_j})} \qquad \text{with } \phi(\boldsymbol{x}) = \exp\left(\frac{-||\boldsymbol{x}||^2}{2\sigma^2}\right) \tag{4.8}$$

where $\theta_i \in \mathbb{R}$ and $\boldsymbol{c_i} \in \mathbb{R}^3$ respectively denote the weight and center position of gaussian i, while $\sigma > 0$ defines the kernel width for the K gaussians. Given Eq. 4.8, we can express the gradient of $\hat{F}(\boldsymbol{x},\boldsymbol{\theta})$ with respect to θ_i, which can be used in Eq. 4.7 to update $\boldsymbol{\theta}$:

$$\frac{\partial \hat{F}(\boldsymbol{x},\boldsymbol{\theta})}{\partial \theta_i} = \frac{\phi(\boldsymbol{x} - \boldsymbol{c_i})}{\sum_{j=1}^{K} \phi(\boldsymbol{x} - \boldsymbol{c_j})} \tag{4.9}$$

4.2.3 Convergence Behaviour

Let us first assume that the environment dynamics change much slower than the convergence rate. We also assume that the K gaussians are uniformly and disjointly activated over a period T; i.e., each kernel is visited periodically T/Ks. The convergence behaviour of our method can be studied by considering the following form for the force error:

$$\tilde{F} = \hat{F}(\boldsymbol{x},\boldsymbol{\theta}^*) - \hat{F}(\boldsymbol{x},\boldsymbol{\theta}) - \eta(t) \tag{4.10}$$

where $\boldsymbol{\theta}^*$ is a unique optimal set of parameters modelling the force error, while $\eta(t)$ accounts for unmodelled dynamics. Let us linearize $\hat{F}(\boldsymbol{x},\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ around $\boldsymbol{\theta}^*$:

$$\hat{F}(\boldsymbol{x},\boldsymbol{\theta}) = \hat{F}(\boldsymbol{x},\boldsymbol{\theta}^*) + \left.\frac{\partial \hat{F}}{\partial \boldsymbol{\theta}}^T\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} (\boldsymbol{\theta} - \boldsymbol{\theta}^*) + O(\boldsymbol{\theta},\boldsymbol{\theta}^*) \tag{4.11}$$

where $O(\boldsymbol{\theta},\boldsymbol{\theta}^*)$ denotes the high-order terms of the Taylor expansion. Substituting $\hat{F}(\boldsymbol{x},\boldsymbol{\theta})$ by Eq. 4.11 in Eq. 4.10 gives:

$$\tilde{F} = -\boldsymbol{S}(\boldsymbol{\theta} - \boldsymbol{\theta}^*) + d(t) \tag{4.12}$$

with:

$$\begin{cases} S = \left.\dfrac{\partial \hat{F}}{\partial \boldsymbol{\theta}}^T\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} \\ d(t) = \text{-}O(\boldsymbol{\theta},\boldsymbol{\theta}^*) - \eta(t) \end{cases} \tag{4.13}$$

Assuming that the disturbance term $d(t)$ is negligible, the parameters' dynamics in Eq. 4.7 can be approximated by:

$$\dot{\boldsymbol{\theta}} = \text{-}\epsilon_r \frac{\partial J}{\partial \boldsymbol{\theta}} = \text{-}\epsilon_r \frac{\partial J}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial \boldsymbol{\theta}} \approx \text{-}\epsilon_r \boldsymbol{S}^{\boldsymbol{T}} \boldsymbol{S}(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \tag{4.14}$$

The matrix $\boldsymbol{S}^{\boldsymbol{T}}\boldsymbol{S}$ being positive semi-definite, $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ is a stable equilibrium. Convergence occurs if the force error signal is rich enough, which is often referred as the *Persistent Excitation (PE)* condition in the literature of adaptive control (Åström and Wittenmark (2013)). Here, PE condition for θ_i is satisfied if the area associated with the i-th RBF kernel is visited by the robot; i.e., $\partial\hat{F}(\boldsymbol{x},\boldsymbol{\theta})/\partial\theta_i \neq 0$. If convergence of the correction term is proven, stability of the full closed-loop system needs to be considered to ensure stable interaction with the environment. The energy tank-based passivity formulation described in section 3.2.3 still holds for this work.

4.3 Illustrative Example in Simulation

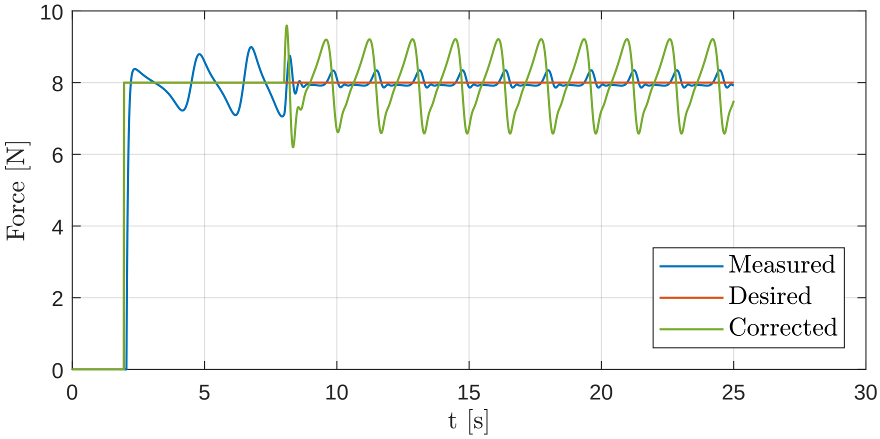
In this section, we evaluate our approach in a simple 3D simulation environment in Matlab, where a point mass robot comes in contact with a planar surface ($\boldsymbol{n}(\boldsymbol{x}) = [0,0,\text{-}1]^T \forall \boldsymbol{x}$) to perform a circular motion (of radius $0.05m$) around a fixed point while generating a desired contact force of $8N$ normal to the surface. A spring model with a non-linear state-dependent stiffness is used to simulate the reaction force $\boldsymbol{F_m}$ of the surface when the robot enters in contact with it:

$$\boldsymbol{F_m} = \begin{bmatrix} 0 \\ 0 \\ (500 + 100000x^2 + 400000y^2) * z \end{bmatrix} \tag{4.15}$$

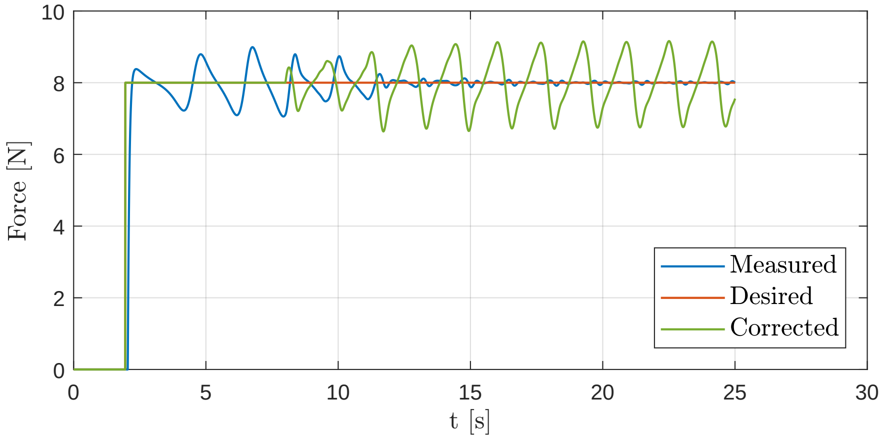
Two force correction models are evaluated:

- A constant model or integrator, i.e. $\hat{F}(\boldsymbol{x},\boldsymbol{\theta}) = c$ where $c \in \mathbb{R}$ is initialized to zero and adapted online.
- Our state-dependent model proposed in Eq. 4.8, where we employ $K = 100$ gaussians (with $\sigma = 0.011m$) uniformly distributed on a fixed $D \times D$ $2D$ grid ($D = 16cm$) whose center matches the center of the desired circular motion on the planar surface. The weights of the gaussians are initialized to zero.

For both models, the simulation lasts $25s$, the adaptation starts at $t = 8s$ and we use the same adaptation rate $\epsilon_r = 30$ and time step $dt = 0.005s$. The measured, desired, and corrected desired (i.e., desired + the learned correction) force profiles are presented in Figures 4.2a and Figure 4.2b for the constant and state-dependent correction models respectively. As can be



(a) Constant correction model



(b) State-dependent correction model

Figure 4.2 – Force adaptation on a planar surface in simulation

observed in these figures, before starting the adaptation, a large error pattern is visible in the measured force signals with a RMS force error of about $0.69N$. The constant correction model can quickly reduce this error to $0.13N$ (computed over the last $5s$ of the simulation) but a small error pattern is still visible at the end and cannot be corrected (see Figure 4.2a). While the state-dependent model takes a bit more time to correct but drastically reduces the RMS force error to a negligible amount of $0.021N$ at the end of the simulation (see Figure 4.2b). Basically, with the same adaptation rate, the state-dependent correction model outperforms a simple integrator whose performance is particularly constrained by the adaptation rate and the dynamics of the force error. Selecting a higher rate can improve the performance of the integrator but it is prone to instabilities, especially in practice. If our state-dependent correction model can correct and capture more non-linearities than an integrator, its efficiency also depends on the selection of the hyper-parameters. This is discussed in more details in the next section where we evaluate our approach in real world experiments.

4.4 Experimental Evaluations

We evaluate our method in two real-world scenarios. In the first scenario, a robot comes in contact with a non-flat surface, moves on it while generating a desired contact force profile. We assess the ability of the system to improve the force tracking under different conditions and analyze the effect of the choice of hyper-parameters (K and σ) on the granularity and precision of the force modulation. We also show the robustness of our method to real-time disturbances. In the second scenario, we perform a collaborative task with a human where the human asks the robot to clean the surface at different locations. To achieve this, we combine the proposed force adaptation with a mechanism to adapt the attractor of a nominal limit cycle proposed in former works of the lab $\big($ Khoramshahi et al. (2018); Khoramshahi and Billard (2019)$\big)$. We show that the force modulation can adapt fast enough to cope with the change in dynamics. Technical details for both tasks can be found in Appendix B.

4.4.1 Force Adaptation on a Non-Flat Surface

Similarly to section 3.4.1, a KUKA LWR IV+ robotic arm is used for the task and equipped with a 6-axis ATI force-torque sensor on which a 3D printed finger tool is attached. The rigid non-flat surface the robot interacts with is also the same.

The robot's behavior is systematically evaluated in a simple task: starting from a fixed initial position, the robot comes in contact with the surface to perform a circular motion (with a fixed center) on the surface while applying a desired contact force. The experiment is repeated 18 times under different choices for hyper-parameters, target force and motion velocity. For each run, force adaptation starts after $9s$ approximately, the experiment lasts $50s$, $\epsilon_r = 20$, while $\hat{F}_+ = -\hat{F}_- = 10N$. To learn $\hat{F}(\boldsymbol{x},\boldsymbol{\theta})$, we should define K, σ, and the spatial distribution of the gaussians. The latter mainly depends on the desired robot motion on the surface. Alike section 4.3, we distribute the gaussians uniformly on the surface in a $D \times D$ 2D grid ($D = 16cm$) to cover the desired circular motion of radius $5cm$. K and σ need to be picked in relationship to each other to reach accurate and smooth interpolation. To enforce reasonable overlap between the gaussians, we propose a kernel width of $\sigma_{th} = \frac{D}{\sqrt{K}}$ (i.e., the gaussians will intercept at $0.5\frac{K}{K-1}\sigma_{th}$). Finally, the complexity and variability of the unmodeled dynamics need to be considered in selecting K; i.e., higher/lower number of gaussians results in higher/lower approximation power, but higher/lower convergence time.

Table 4.1 evaluates the performance of our method by comparing the force tracking error before and after adaptation in terms of Root Mean Square error (RMS). In Table 4.1a, different combinations of K and σ are tested, while keeping the same target force and desired robot velocity in contact. Before adaptation, the RMS error reaches in average $3.18 \pm 0.06N$. At the end of the experiments, the lowest RMS value ($0.36N$) is obtained by using $K = 100$ gaussians and setting σ to σ_{th}. We can observe that the RMS error is usually higher when using a signifi-

σ K	$0.5\sigma_{th}$	σ_{th}	$2\sigma_{th}$
9	**1.45** (3.19)	**0.89** (3.21)	**1.55** (3.23)
100	**0.58** (3.06)	**0.36** (3.19)	**0.79** (3.20)
900	**0.49** (3.19)	**0.69** (3.19)	**1.25** (3.22)

(a)

v_0 F_T	$0.2m/s$	$0.25m/s$	$0.3m/s$
$10N$	**0.29** (2.69)	**0.32** (3.04)	**0.37** (3.52)
$15N$	**0.31** (2.82)	**0.38** (3.27)	**0.41** (3.46)
$20N$	**0.30** (2.77)	**0.38** (3.09)	**0.48** (3.53)

(b)

Table 4.1 – Force adaptation on the non-flat surface: RMS values of the force error in N obtained over the last $10s$ of the experiments (in bold). The values in parentheses are the RMS errors obtained in the last $5s$ before starting the adaptation. In Table 4.1a, different combinations of numbers of RBFs and kernel width are tested while keeping the same target force after contact ($F_T = 15N$) and desired robot velocity ($v_0 = 0.25m/s$). In Table 4.1b, the target velocity and contact force are changed while keeping the same adaptation settings ($K = 100$, $\sigma = \sigma_{th}$).

cantly smaller or larger σ as expected. Indeed, approximation power of larger kernels suffers from specificity, while small kernels lack generalization/interpolation. Furthermore, with small kernels, high adaptation rates are prone to fluctuations and instabilities. These results also show that lower number of gaussians ($K = 9$) provides less effective correction.

This is illustrated in the measured force profile in Figure 4.3a (left) which still has a significant error pattern that cannot be captured by the model. This also translates visually in Figure 4.3a (right) where only three distinct regions are visible in the learned force correction map. When increasing K to 100, the approximation of the model improves as shown in Figure 4.3b. Higher K is expected to result in lower RMS error. However, it requires more time for convergence as illustrated in the measured force profiles. This is why, higher RMS errors are obtained for $K = 900$ than for $K = 100$, as the model parameters are still converging at the end of the experiment. The higher specificity of $K = 900$ is visible in the force correction map (Figure 4.3c (right)) where the correction is only learned along the robot trajectory.

In Table 4.1b, our method is evaluated when targeting different target forces and velocities, while keeping the same adaptation settings. The method shows robustness to desired motion velocity and force where in average, the RMS error is reduced to $0.36 \pm 0.06N$. The lower performance when targeting simultaneously higher desired velocities and forces is expected due to higher frictions. Also, comparing the average results across the two tables ($0.90 \pm 0.42N$ in Table 4.1a) shows that our method is less sensitive to the desired behavior than the hyper-parameters. This delineates the importance in choosing the hyper-parameters

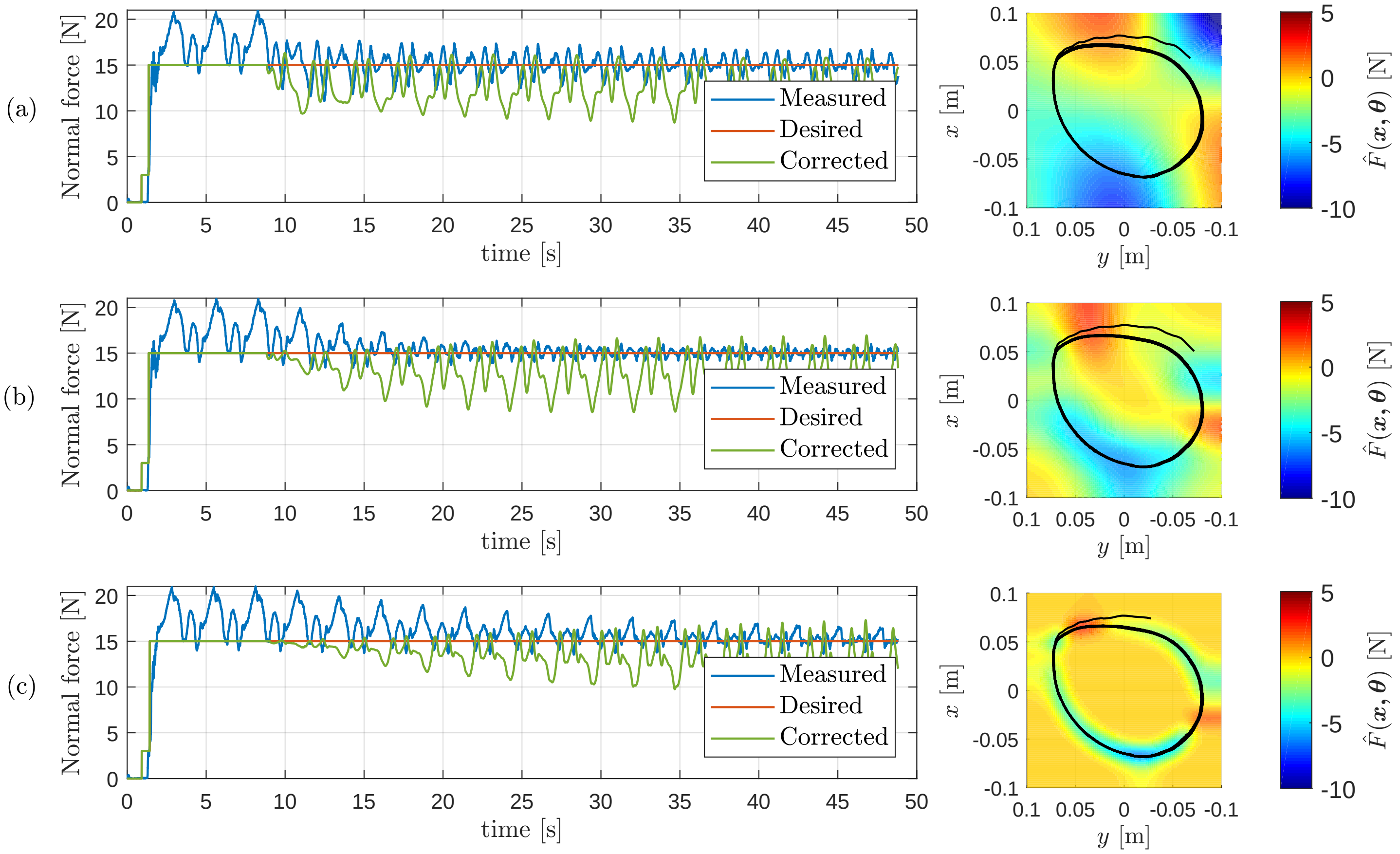


Figure 4.3 – Force adaptation on the non-flat surface when using different number of RBFs with $\sigma = \sigma_{th}$, $F_T = 15N$, and $v_0 = 0.25m/s$: (a) $K = 9$, (b) $K = 100$, (c) $K = 900$. On the left side, the measured, desired and corrected desired force profile are depicted. On the right side, the force compensation map learned at the end of the experiments is drawn with respect to the robot's relative position to the attractor, along with the robot path on the surface.

that can be achieved following the proposed guideline.

Moreover, Figure 4.4 highlights the influence of the number of repetitions (i.e., circles) performed (after starting the adaptation) on the convergence and tracking performance of our method. As expected, for a fixed adaptation rate, higher K or robot velocity needs more repetitions to converge. In general, the number of repetitions needed is mainly affected by the adaptation rate, the number of RBFs and the robot velocity.

Finally in Figure 4.5, an example of force adaptation under disturbances is provided. Between 27 and 32s, a human pushes the robot away from the attractor while the robot is moving in contact. The algorithm compensates for the generated force errors by adapting the correction limited by a lower bound. However, once the disturbance disappears, the force error is

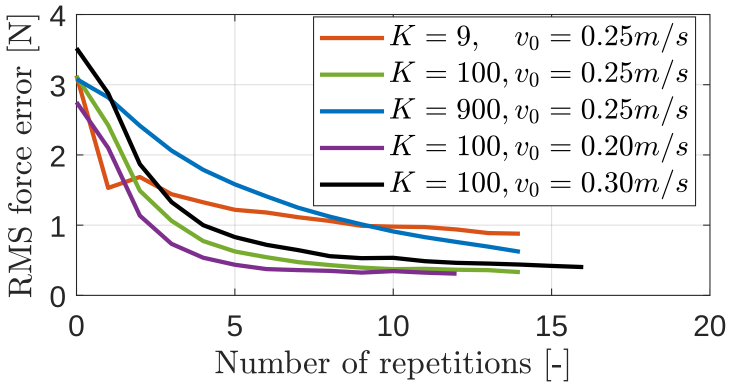


Figure 4.4 – Force tracking error as a function of the number of repetitions (e.g., circles) with $\sigma = \sigma_{th}$, $F_T = 15N$.

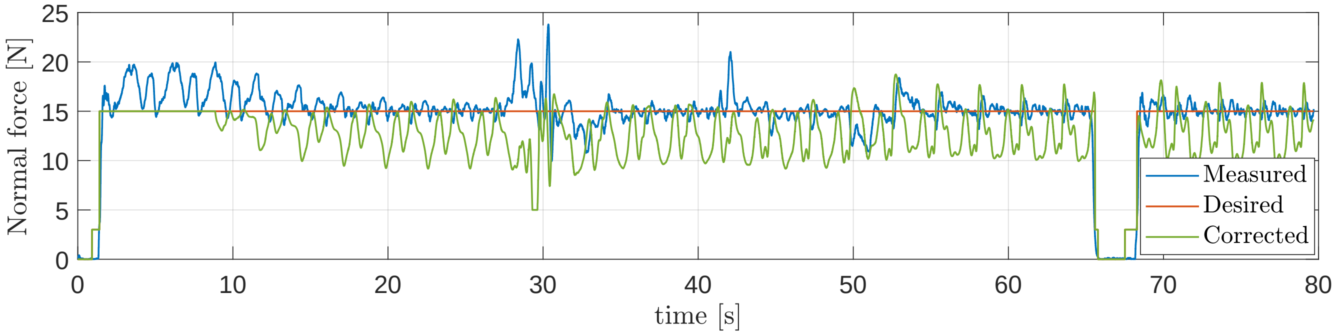


Figure 4.5 – Force adaptation on the non-flat surface under disturbances with $K = 100$, $\sigma = \sigma_{th}$, $F_T = 15N$, and $v_0 = 0.25m/s$.

reduced quickly; i.e. less than 6s. At 42s, the human changes the surface inclination, keeps it tilted, and puts it back at 50s. The correction pattern is adapted quickly (2 to 5s) to the inclination changes. Our adaptation mechanism only considers the interaction with the surface captured by the force-torque sensor. For instance, when the human takes the robot away from the surface at 65s by interacting with the body, the force error is compensated instantaneously once the robot reaches the surface; i.e., the previously adapted model remains unchanged.

4.4.2 Collaborative Cleaning of a Non-Flat Surface

In this second scenario, we perform a cooperative cleaning task of the non-flat surface with a human. The same setup as in section 4.4.1 is used with the exception of the robot tool, replaced with a cleaning pad (see Figure 4.6). Here, we highlight the benefits of using dynamical systems to perform contact tasks with a reactive and adaptive behaviour while interacting both with the surface and a human. To this end, we combine:

- The work in Khoramshahi and Billard (2019) to switch across different tasks. This work proposes a mechanism that smoothly switches the desired robot task to the one intended by the human (identified through the robot's real velocity that captures the physical interaction with the human). For the cleaning of the surface we define two tasks:

- The homing task ($i = 1$) defined by $\boldsymbol{f^1(x)} = \boldsymbol{x_{a,h}} - \boldsymbol{x}$ and $F_d^1(\boldsymbol{x}) = 0\ \forall \boldsymbol{x}$, where the robot should reach a fixed attractor above the surface, with $\boldsymbol{x_{a,h}}$ the attractor.
- The circular cleaning task ($i = 2$) defined by $\boldsymbol{f^2(x)}$ and $F_d^2(\boldsymbol{x})$ similarly to section 4.4.1 (see Appendix B).

From these tasks, we express the adapted nominal DS as:

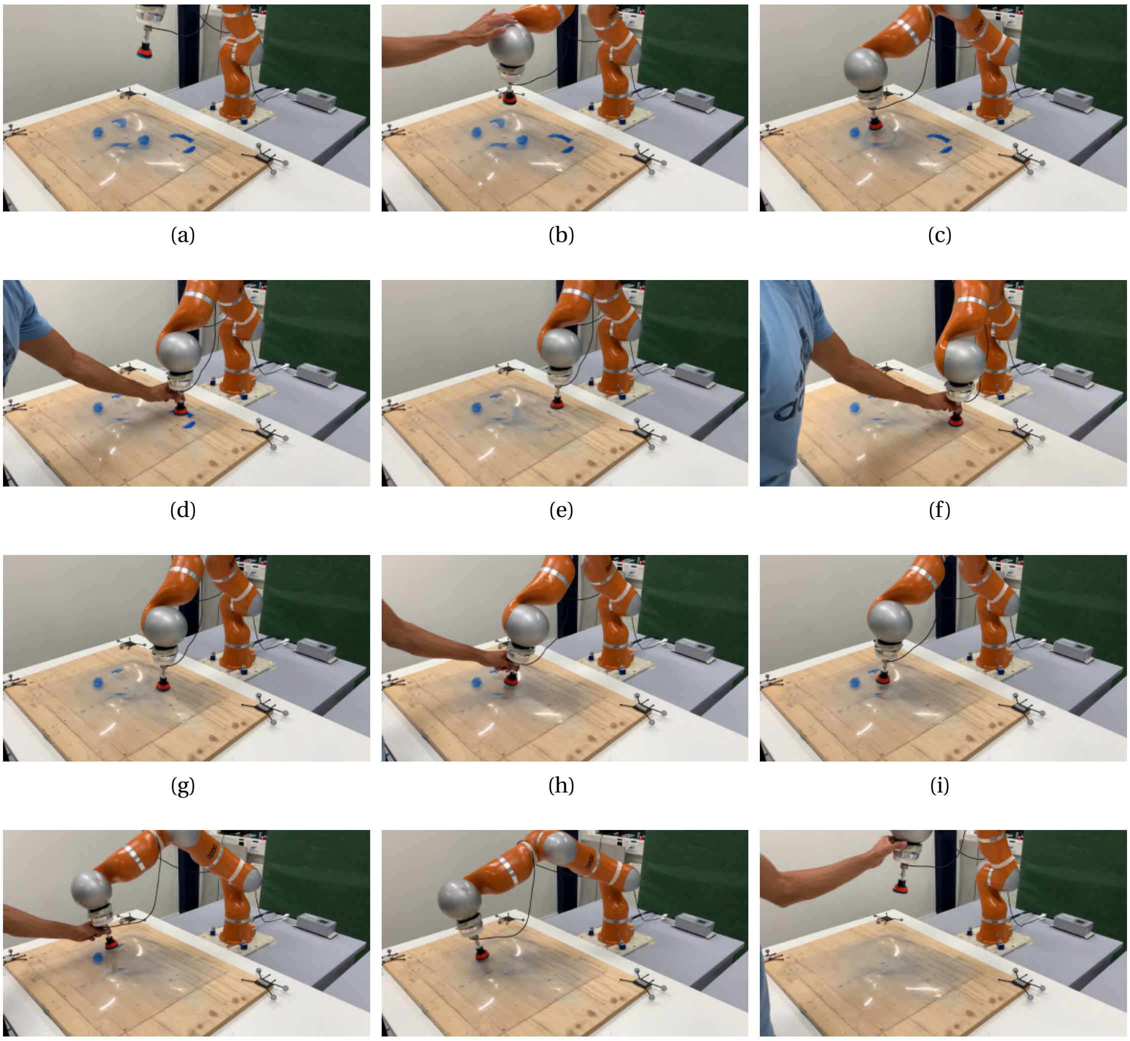
$$\boldsymbol{f(x)} = b_1 \boldsymbol{f^1(x)} + b_2 \boldsymbol{f^2(x)} \tag{4.16}$$

where $\boldsymbol{b} = [b_1, b_2]^T$ is the belief vector whose update rule ensures that $b_1 + b_2 = 1$ and that one $b_i \rightarrow 1$ in finite time (see Khoramshahi and Billard (2019)). Concerning the overall desired force profile $F_d(\boldsymbol{x})$, we set it to:

$$F_d(\boldsymbol{x}) = \begin{cases} F_d^i(\boldsymbol{x}) & \text{if } b_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

- The work in Khoramshahi et al. (2018) to adapt the cleaning motion by adjusting the location of the attractor in $\boldsymbol{f^2(x)}$ through the physical interaction with the human.
- The method in section 4.2.1 to adapt the desired force.

Thus, three adaptation levels occur in this scenario: task, motion, and force. Each of them is activated under simple conditions. Task adaptation is active when the robot is in free motion; i.e., no contact with the surface, or $F_d(\boldsymbol{x}) = 0$. Motion adaptation is activated when the robot is cleaning the surface (i.e., $b_2 = 1$ and the robot is in contact), and the tangential forces (w.r.t to the surface) perceived at the end-effector exceed a predefined threshold $\varepsilon_{F,t}$. To this end, we assume that the human grabs the robot tool to change the cleaning area and that large forces would result from this interaction. Finally, force adaptation is enabled when the robot is cleaning the surface and the tangential forces remain below $\varepsilon_{F,t}$.



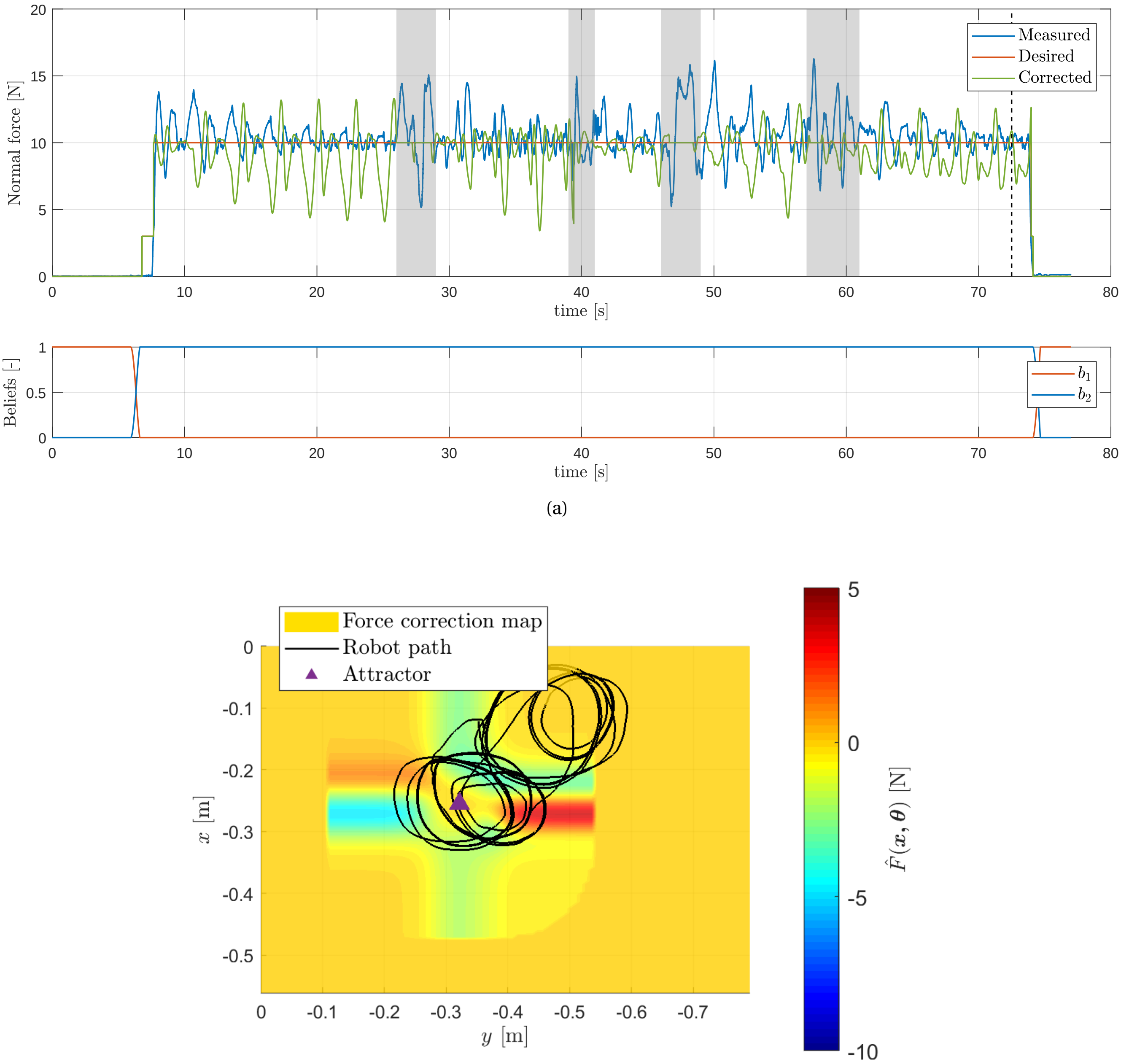
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.6 – Snapshots of the collaborative cleaning task. Starting from its home position (Figure 4.6a), the robot switches to the circular cleaning task thanks to the physical interaction with the human (Figures 4.6b). During the task, at multiple occasions (Figure 4.6d, 4.6f, 4.6d, and 4.6h) the human physically interacts with the robot to change the cleaning area. After each interaction the robot keeps cleaning the new zone (Figure 4.6e, 4.6g, 4.6i, and 4.6k). The human finally stops the cleaning task by pulling the robot away from the surface which switches to the homing task (Figure 4.6l).



(a)

(b)

Figure 4.7 – Collaborative cleaning of the non-flat surface: Force adaptation is illustrated in Figure 4.7a top with the measured, desired, and corrected desired force profile. At the bottom, task adaptation is represented by the beliefs' evolution. Motion adaptation is visible in Figure 4.7b with the whole robot path projected on the surface's plane during the cleaning task. The gray shared areas in Figure 4.7a highlight the interaction periods with the human to adapt the motion. Figure 4.7b also shows the attractor and force correction map captured at 72.5s (i.e., vertical dashed line in Figure 4.7a).

Snapshots of the collaborative scenario can be observed in Figure 4.6. The robot starts from its home position ($b_1 = 1$). Following the physical human interaction, the robot switches to the cleaning task ($b_2 = 1$) and comes in contact with the surface to start cleaning it around a predefined initial attractor. The beliefs' profile illustrated in Figure 4.7a (bottom) shows the switching across the two tasks. The activation of the force adaptation occurs after upon contact with the surface at $8s$; see Figure 4.7a (top). The force error quickly reduces to a negligible amount; i.e., RMS error of $0.43N$ between 20 and $25s$. The robot cleans the same area until approximately $25s$ when the human starts to physically interact with the robot with the intention to change the cleaning area. This is illustrated by the large peaks in the measured force profile; see Figure 4.7a (top). Figure 4.7b illustrates the adaptation of the motion (i.e., the attractor) which occurs several times between 25 and $74s$. During that period, force adaptation is active when the tangential forces are small; i.e., the absence of human interaction. For example, one can see the force correction map and adapted attractor captured at $72.5s$ in Figure 4.7b. Finally, at around $74s$, the human stops the cleaning task through physical interaction resulting in the robot retreating; i.e., $b_1 = 1$.

4.5 Discussion and Conclusion

In this chapter, we used dynamical systems for force adaptation in contact tasks. Our method uses online adaptation of a state-dependent force correction model encoded with gaussian RBF kernels. It is particularly suitable for repetitive tasks as shown in section 4.4.1 where the robot should repeat the same motion while generating a desired contact force. For non-repetitive tasks which do not require to learn a model (no structural errors), our method could be simply modified to online adaptation of an offset. Our results showed that the force tracking accuracy is significantly improved for different desired motion-force profiles even in the face of real-time disturbances. For the correction to be effective, the hyper-parameters should be set properly; in particular the number of gaussians and kernel width. In our experiments, we distributed the RBFs locally on a small area around the attractor instead of covering the whole surface. The alternative requires a large number of gaussians which increases the computational cost and convergence time. The latter can be reduced using higher adaptation rate which, however, is prone to fluctuations and instabilities.

Overall, our experimental results suggest that many robotic applications can benefit from force adaptation with dynamical systems. As demonstrated in the collaborative cleaning task in section 4.4.2, the time-invariant DS framework provides reactive and adaptive robotic behaviour. This enables robots to perform tasks in uncertain environment where the robot is required to physically interact with humans, objects, and surfaces.

5 Contact-Initiated Shared Control Strategies for Four-Arm Manipulation

Note: The material presented in this chapter is adopted from:
Amanhoud, W., Hernandez Sanchez, J., Bouri, M., and Billard, A. (2021). Contact-initiated shared control strategies for four-arm supernumerary manipulation with foot interfaces. The International Journal of Robotics Research, 40(8–9), 986–1014.

5.1 Introduction

Whether in industrial or surgical settings, many tasks require at least two people to be achieved successfully. To this end, robotic assistance could be used to enable a single person to perform such tasks alone, with the help of robots. In this chapter, we are interested in a four-arm manipulation scenario where two supernumerary robotic arms assist an operator to carry and move an object on which she/he is working with both hands, as illustrated in Figure 5.2. To this end, the feet of the operator are used to control each robotic arm through haptic foot interfaces (see Appendix C). Foot interfaces are indeed less prone to distractions or interruptions than head, speech or eyes-based interfaces, and can give more natural mapping. To enable and ease the four-arm solo manipulation of the object and reduce the cognitive workload, we exploit the time-invariant DS framework to develop (shared) control strategies, providing assistance upon contact with the object based on the contact forces. Two types of assistance are proposed:

- Autonomous contact force generation to hold the object and prevent it from falling. By delegating the robustness of the grasp to the autonomy, the force assistance is intended to simplify both mentally and physically the task to the operator so she/he can focus just in moving the object with the feet and prioritize the task of the hands.

- Auto-coordination of the robotic arms to ease the manipulation of the object and compensate for the lack of coordination of the robots' motion driven by the human's feet. The coordination is intended to mentally simplify the task by directly controlling the two robotic

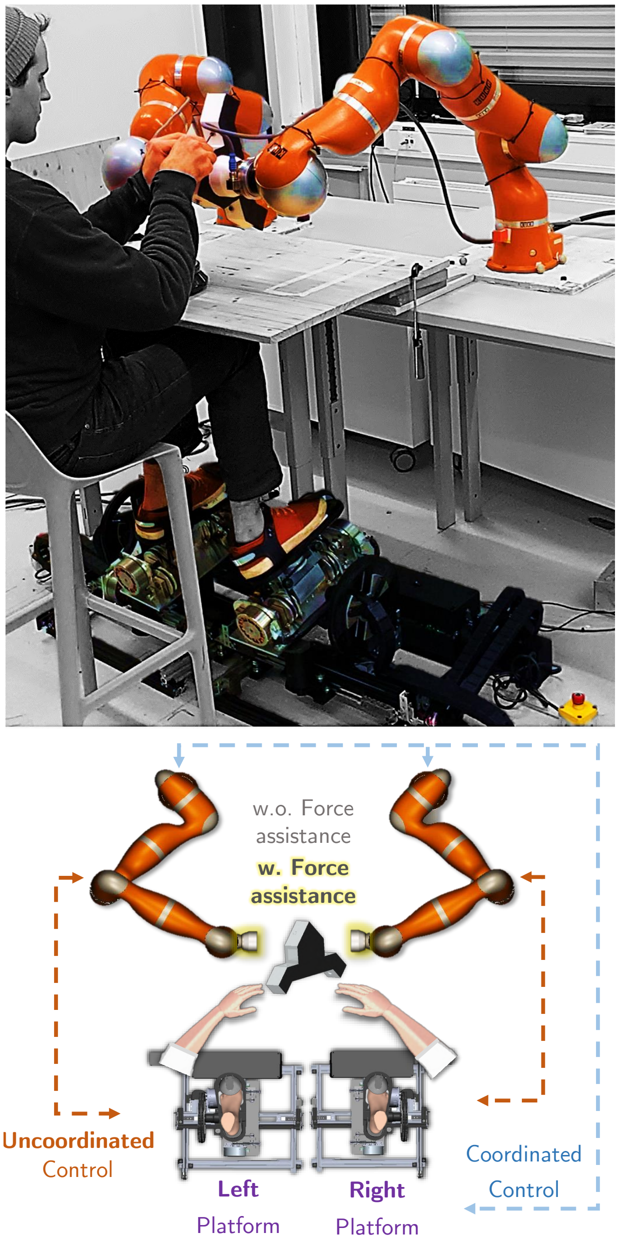


Figure 5.2 – Four-arm shared manipulation of an object (©Jacob Hernandez Sanchez). Two robotic arms assist a human operator in supporting and moving an object while the human is working on the object with the hands. To this end, the robots are partially controlled with the feet of the human using haptic foot interfaces. We design four control strategies from the combinations (absence/presence) of two assistance modalities: force and coordination assistance.

arms (and indirectly the pose of the object) with a single foot.

To evaluate the benefits of these two assistance modalities, four control strategies are designed and compared with each other through a user study on a four-arm manipulation task with

12 participants. The user study quantifies the assistance modalities in terms of efficiency and success rate and evaluates their influence on the task fluency and user perceived task load. The work presented in this chapter is published in Amanhoud et al. (2021) with Jacob Hernandez Sanchez as co first author. Jacob focused on the design and control of the foot haptic interfaces. Jacob is also the main contributor to the analysis and interpretation of the experimental data gathered during the user study. In this chapter, only the theoretical material developed by the thesis author is reported. The main results obtained from the data analysis are briefly summarized to validate the theoretical development. Regarding the design and conduct of the experimental study, it was performed collaboratively. For more details, please refer to Amanhoud et al. (2021) and Appendix D.

Related work for this chapter is reviewed in section 1.5. The control strategies are presented in section 5.2 while the experimental study is described in section 5.3. Results are reported in section 5.4 and a conclusion about this work is finally provided in section 5.5.

5.2 Method

5.2.1 Control Strategies Overview

The four (shared) control strategies result from the combinations (presence/absence) of the two assistance modalities considered in this work, namely, force and coordination assistance, and are illustrated in Figure 5.3. The assistance modalities are activated once the contact with the object is made. Prior to contact, all control strategies behave the same with each foot controlling its ipsilateral robotic arm. The operator is responsible of positioning and orienting the robotic arms to perform the grasp. As soon as the interaction forces exceed a specific threshold, i.e., after contact, four control strategies can be distinguished:

- Uncoordinated Control Without Force Assistance (*U*): This is the baseline strategy where the operator still uses their two feet to control the robotic arms. The operator should maintain the grasp while moving and orientating the object. The grasping forces are rendered in the feet.

- Uncoordinated Control With Force Assistance (*U*+*F*): The operator controls the two robotic arms with their two feet but a force assistance is provided to keep a robust grasp. The force rendered to the feet is the error between the assistive contact forces and the measured end-effector forces.

- Coordinated Control Without Force Assistance (*C*): The control of both robots switches from two feet to a single foot (the dominant one) after the contact is done. The operator controls the center position between both robots' end-effectors as well as the opening/closing (distance between both end-effectors). The opening/closing is controlled in the fifth degree of freedom of the foot platform (foot's self rotation) and the net grasping force is rendered in this axis as well.

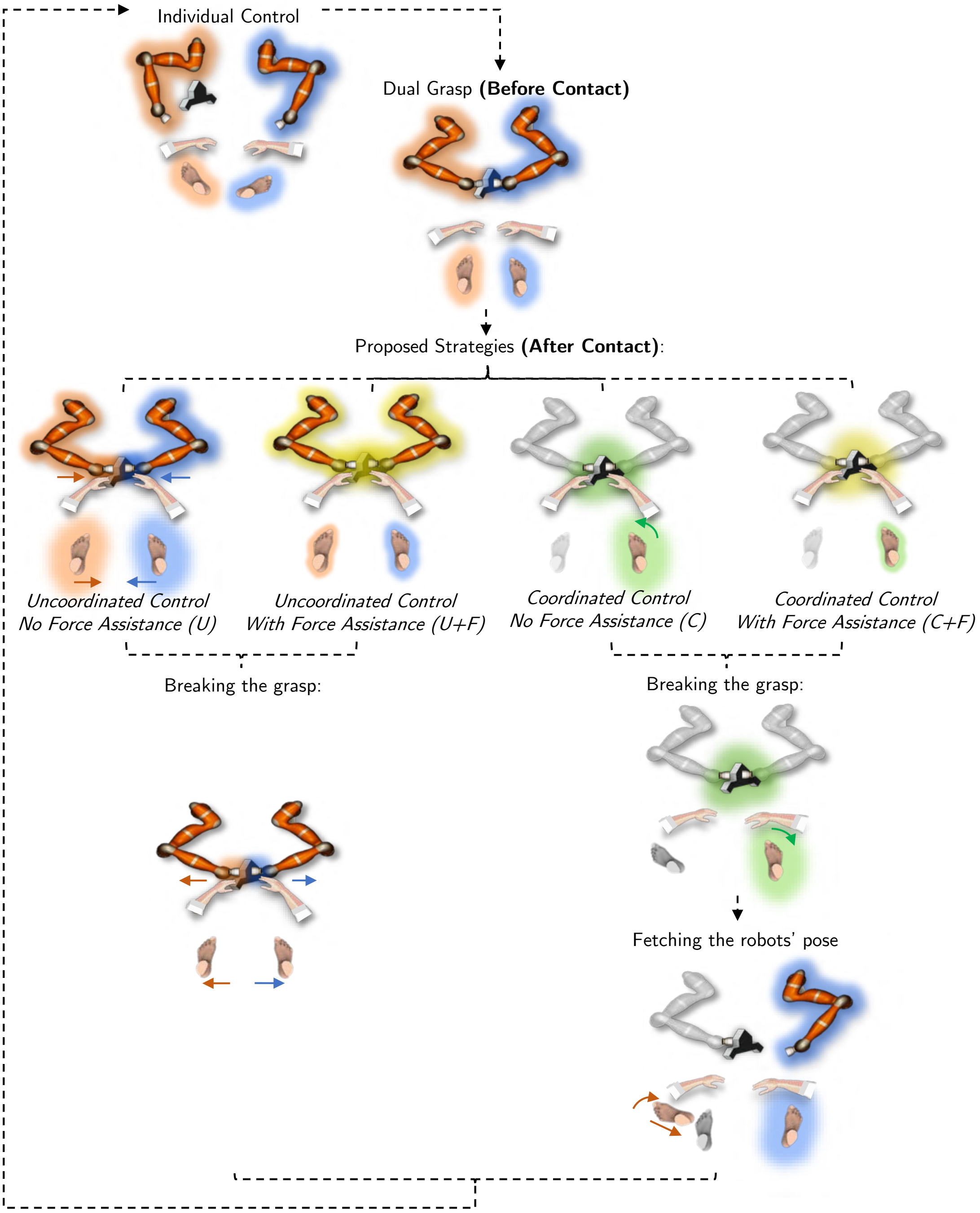

Individual Control
Dual Grasp **(Before Contact)**
Proposed Strategies **(After Contact)**:
Uncoordinated Control
No Force Assistance (U)
Uncoordinated Control
With Force Assistance (U+F)
Coordinated Control
No Force Assistance (C)
Coordinated Control
With Force Assistance (C+F)
Breaking the grasp:
Breaking the grasp:
Fetching the robots' pose

Figure 5.3 – Overview of the proposed control strategies (©Jacob Hernandez Sanchez). The uncoordinated strategies (strategies U and U+F) are represented by orange and blue glows where each foot controls one robot. After contact, force assistance is represented by a yellow glow (strategies U+F and C+F), while coordination assistance is represented by a green glow (strategies C and C+F).

- Coordinated Control With Force Assistance (*C+F*): The worker also controls both robotic arms by using a single foot, but in this case a force assistance is provided to keep the grasp while the object is moved, and the force error is reflected. This with the goal of simplifying the task both mentally and physically.

In the uncoordinated control strategies (i.e., *U* and *U+F*), to break the grasp, the operator has to move his feet in the opposite direction. For strategy *U+F* in particular, this implies fighting against the haptic feedback (i.e., the force error). In that case, once the measured contact forces are lower than a threshold, force assistance stops and the grasp is broken. Regarding the coordinated control strategies (i.e., C or $C+F$), the grasp can be broken with the fifth DOF of the foot platform (i.e, the one responsible for opening/closing). The robotic arms remain coordinated until the distance between the two end-effectors exceeds a threshold. At that moment, the feet have to fetch the robotic arms in the workspace of the foot platforms to assure a stable transition. This fetching phase is accompanied with haptic cues that push the feet to the required pose. When a distance threshold is met, then the individual robot control is regained for each foot.

The four control strategies overviewed above are all implemented through the time-invariant DS framework. Similarly to the previous chapters, the desired linear robots dynamics are specified by a DS which is tracked with the DS-impedance controller (Eq. 2.16 in section 2.2). The robots' end-effector orientation is controlled using Eq. 2.23 in section 2.3 while the haptic foot interfaces are driven using the control law in Eq. C.2 (see Appendix C). In the following sections, we describe for each control strategy, how we construct the desired DS, end-effector orientation and angular velocity as well as the reflected torques at the master foot interfaces. Superscript $i = \{L, R\}$ will be used to represent the left or right robot/master foot interface (with respect to the human point of view, see Figure 5.2).

5.2.2 Uncoordinated Robot Control Without Force Assistance (strategy *U*)

In this control strategy, the two foot interfaces are used to control both robots. Each robot tracks a linear dynamical system whose attractor is obtained by mapping the associated foot pose ($\boldsymbol{\xi}^{\boldsymbol{i}}$, see Appendix C) to a desired attractor in the robots' world frame. To this end, we define $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}}$ as follows:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} = \boldsymbol{A}^{\boldsymbol{i}}(\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{i}} - \boldsymbol{x}^{\boldsymbol{i}}) \tag{5.1}$$

where $\boldsymbol{A}^{\boldsymbol{i}} \in \mathbb{R}^{3\times3}$ is a positive definite diagonal matrix with constant values while $\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{i}} \in \mathbb{R}^3$ denotes the desired robot attractor obtained through:

$$\begin{pmatrix} \boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{i}} \\ \omega_h^i \end{pmatrix} = \boldsymbol{\Upsilon}_{\boldsymbol{p}}^{\boldsymbol{U}} \boldsymbol{\xi}^{\boldsymbol{i}} + \begin{pmatrix} \boldsymbol{x}_{\boldsymbol{0}}^{\boldsymbol{i}} \\ 0 \end{pmatrix} \tag{5.2}$$

$\boldsymbol{x}_{\boldsymbol{0}}^{\boldsymbol{i}} \in \mathbb{R}^3$ is a fixed attractor offset, while $\boldsymbol{\Upsilon}_{\boldsymbol{p}}^{\boldsymbol{U}} \in \mathbb{R}^{4\times5}$ is a telefunctioning matrix mapping the foot's pose (measured in the platform joint space) to the desired degrees of freedom in the robots' task space, when the robots are uncoordinated. As can be appreciated in Appendix D, this matrix considers transformations in the reference frames between the master device (foot platform) and the telemanipulator, as well as scaling factors. Depending on the defined mappings for the foot gestures, $\boldsymbol{\Upsilon_p^U}$ can include position:position and/or position:velocity mappings between the degrees of freedom. Here, $\omega_h^i \in [\omega_{h,-}, \omega_{h,+}]$, with $\omega_{h,-} < 0$ and $\omega_{h,+} > 0$, is an angular velocity input allowing self-rotation of the end-effector along $\boldsymbol{z_o^i}$. A position:velocity mapping is used to control the self-rotation of the end-effector, since its range of motion is significantly larger than the range of motion of the human input (see Figure 5.7).

Concerning the orientation, the axis-angle error $\boldsymbol{\tilde{\zeta}^i}$ (in Eq. 2.23) is computed to align the end-effector axis $\boldsymbol{z_o^i}$ with a desired entity $\boldsymbol{z_{o,d}^i}$, fixed to a predefined direction for both robots. While the desired angular velocity $\boldsymbol{\omega_d^i}$ is defined as the sum of two terms, one term to help aligning $\boldsymbol{z_o^i}$ with $\boldsymbol{z_{o,d}^i}$ and a second term ($\omega_h \boldsymbol{z_o^i}$) to allow self-rotation of the end-effector (see Appendix D for more details).

Moreover, this control strategy allows the measured robot end-effector forces (obtained with force/torque sensors) to be reflected to the master interfaces such that the human can perceive the robots' interaction with the environment. Therefore, we define $\boldsymbol{\tau_d^i}$ from Eq. C.2 as follows:

$$\boldsymbol{\tau_d^i} = \boldsymbol{\Upsilon_F^U F_{ext}^i} \tag{5.3}$$

where $\boldsymbol{\Upsilon_F^U} \in \mathbb{R}^{5\times3}$ is the force telefunctioning matrix that maps the interaction forces of the environment ($\boldsymbol{F_{ext}^i}$) to the foot through haptic feedback (rendered in the joint space of the foot platform). Depending on the choice of mapping for the foot gestures (see Figure 5.7), this matrix may include force:force or force:torque transformations between the joints of the foot platform and the robot end-effector. Moreover, as can be seen in Appendix D, the interaction forces are conveyed to the user's feet with a scaling factor that changes depending on the control strategy.

Note that the haptic feedback works selectively for the user-controlled degrees of freedom that are relevant for keeping the grasp (i.e., linear translations). This is to evoke a feeling of telepresence while performing and keeping the grasp. On the other hand, a joystick-like virtual spring-damper is rendered in the axis of the platform that controls the self-rotation of the robots' end-effector. Joystick-like means that a force of the spring-damper will push the user's foot to the resting position. This is achieved by setting specifically the values of $\boldsymbol{K_M^i}$ and $\boldsymbol{D_M^i}$ in Eq. C.2 (see Appendix D for more details).

5.2.3 Uncoordinated Robot Control With Force Assistance (strategy $U{+}F$)

In this control strategy, each robot is controlled through its corresponding foot interface as in section 5.2.2. However, force assistance is provided to support the object once the object is grasped. To this end, using the work presented in chapter 3 (section 3.2.2), Eq. 5.1 is modified as follows:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} = \boldsymbol{A}^{\boldsymbol{i}}(\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{i}} - \boldsymbol{x}^{\boldsymbol{i}}) + \frac{F_d^{U,i}}{d_1^i}\boldsymbol{z}_{\boldsymbol{o}}^{\boldsymbol{i}} \tag{5.4}$$

where the desired contact force F_d^i is set to:

$$F_d^{U,i} = \begin{cases} F_T^{U,i} & \text{grasped} \\ 0 & \text{not grasped} \end{cases} \tag{5.5}$$

with $F_T^{U,i} > 0$ the target force of robot i when the object is grasped; i.e., persistent contact forces are detected from both robots' force-torque sensors. This grasp condition is given below:

$$\begin{cases} \text{grasped} & \mu_F^L \geq \varepsilon_F \ \wedge \ \mu_F^R \geq \varepsilon_F \\ \text{not grasped} & \text{otherwise} \end{cases} \tag{5.6}$$

where μ_F^i with $i = \{L, R\}$ is the mean value over a sliding window of size n of the measured forces along the $\boldsymbol{z}_{\boldsymbol{o}}^{\boldsymbol{i}}$ axis of the robot's end-effector, while $\varepsilon_F \geq 0$ is a force threshold.

Moreover, the haptic feedback in Eq. 5.3 is modified to reflect the force error between the measured and desired contact forces at the master interface such that:

$$\boldsymbol{\tau}_{\boldsymbol{d}}^{\boldsymbol{i}} = \boldsymbol{\Upsilon}_{\boldsymbol{F}}^{\boldsymbol{U}}(\boldsymbol{F}_{\boldsymbol{ext}}^{\boldsymbol{i}} + F_d^{U,i}\boldsymbol{z}_{\boldsymbol{o}}^{\boldsymbol{i}}) \tag{5.7}$$

This reduces the risks for the operator to unintentionally break the grasp, as the feet can compensate for this force error.

5.2.4 Coordinated Robot control With Force Assistance (strategy C+F)

In this control strategy, initially, when the object is not grasped, the two robots are controlled individually in an uncoordinated manner (i.e., as in section 5.2.2). Once the object is grasped, both force assistance and auto-coordination of the robotic arms are activated to ease the support and movement of the manipulated object. During the coordination, a single foot interface is used to control both robots at the same time. Figure 5.4 highlights the switching mechanisms between uncoordinated and coordinated robot control with force assistance. In particular, to switch to the coordinated control, the grasp condition is used. While to switch back to the uncoordinated control, a fetching mechanism is developed to increase safety and will be detailed later. First, let us design $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}}$ to allow auto-coordination of the arms and autonomous contact force generation when the object is grasped. Using our approach in chapter 3 (section 3.3.1), we express the desired robots' dynamics $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}}$ as:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} = \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}} + \tilde{\dot{\boldsymbol{x}}}_{\boldsymbol{d}}^{\boldsymbol{D}^{\boldsymbol{i}}} + \frac{F_d^{C,i}}{d_1^i}\boldsymbol{n}^{\boldsymbol{i}} + v_h\boldsymbol{n}^{\boldsymbol{i}} \tag{5.8}$$

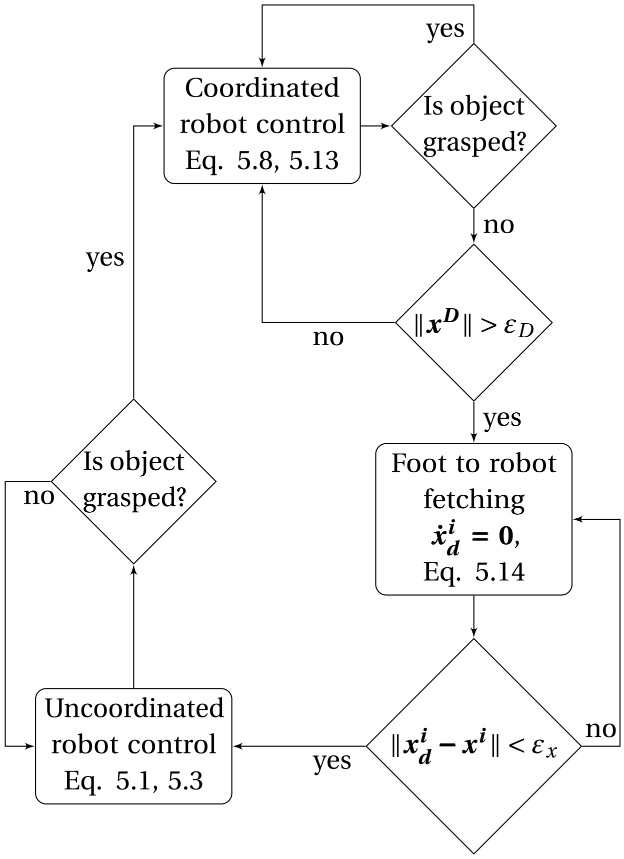

yes
Coordinated robot control Eq. 5.8, 5.13
Is object grasped?
no
yes
$\|\boldsymbol{x}^{\boldsymbol{D}}\| > \varepsilon_D$
no
yes
Is object grasped?
no
Foot to robot fetching $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} = \boldsymbol{0}$, Eq. 5.14
Uncoordinated robot control Eq. 5.1, 5.3
yes
$\|\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{i}} - \boldsymbol{x}^{\boldsymbol{i}}\| < \varepsilon_x$
no

Figure 5.4 – Flow chart representation of *C+F* control strategy with the switching mechanisms between coordinated and uncoordinated robot behaviours.

with:

$$\tilde{\dot{\boldsymbol{x}}}_{\boldsymbol{d}}^{\boldsymbol{D}^{\boldsymbol{R}}} = -\tilde{\dot{\boldsymbol{x}}}_{\boldsymbol{d}}^{\boldsymbol{D}^{\boldsymbol{L}}} = \frac{\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}}}{2} \qquad \boldsymbol{n}^{\boldsymbol{L}} = -\boldsymbol{n}^{\boldsymbol{R}} = \frac{\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}}}{\|\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}}\|} \tag{5.9}$$

where $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}}$ and $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{D}}$ are the desired dynamics for the center position and distance vector between the two robots' end-effector as defined previously in Eq. 3.18, while $\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}}$ is the desired distance vector.

The first two terms in Eq. 5.8 allow the robots to move in coordination. The third term generates contact force along the desired grasping direction $\boldsymbol{n}^{\boldsymbol{i}}$ obtained from $\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}}$, such that:

$$\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{D}} = \begin{cases} \boldsymbol{x}_{\boldsymbol{0}}^{\boldsymbol{D}} & \text{grasped} \\ \boldsymbol{P}\boldsymbol{x}^{\boldsymbol{D}} & \text{not grasped} \end{cases} \tag{5.10}$$

where $\boldsymbol{x}_{\boldsymbol{0}}^{\boldsymbol{D}}$ is the initial distance vector measured after grasping the object while $\boldsymbol{P} \in \mathbb{R}^{3\times 3}$ is a projection matrix used to constrain the distance vector along a specific direction when the object is not grasped. The desired contact force $F_d^{C,i}$ is set to:

$$F_d^{C,i} = \begin{cases} F_T^{C,i} & \text{grasped} \\ 0 & \text{not grasped} \end{cases} \tag{5.11}$$

where $F_T^{C,i} > 0$ is the target force when the object is grasped. The last term in Eq. 5.8 is the human input velocity along the desired grasping direction where $v_h \in [v_{h,-}, v_{h,+}]$ with $v_{h,-} < 0$ and $v_{h,+} > 0$. It allows the human to break the grasp or squeeze more the object.

Intending to reduce human cognitive workload and fatigue, a single master foot interface is used to control the two robots; i.e., five degrees of freedom are available as human input. One is used to compute v_h while three are defining the desired robots' center position $\boldsymbol{x_d^C}$ used in Eq. 3.18. The remaining one is used to rotate the end-effectors along a specific direction. These transformations are described in Eq. 5.12:

$$\begin{pmatrix} \boldsymbol{x_d^C} \\ v_h \\ \omega_h \end{pmatrix} = \boldsymbol{\Upsilon_p^C} \boldsymbol{\xi^j} + \begin{pmatrix} \boldsymbol{x_0^C} \\ 0 \\ 0 \end{pmatrix} \tag{5.12}$$

where, similarly to $\boldsymbol{\Upsilon_p^U}$ in Eq. 5.2, $\boldsymbol{\Upsilon_p^C} \in \mathbb{R}^{5\times5}$ is the pose telefunctionning matrix when the robots are coordinated, while $\boldsymbol{\xi^j}$ is the pose of the master interface j used to control both robots. $\boldsymbol{x_0^C} \in \mathbb{R}^3$ is a fixed attractor offset. $\omega_h \in [\omega_{h,-}, \omega_{h,+}]$, with $\omega_{h,-} < 0$ and $\omega_{h,+} > 0$, is the angular velocity input used to rotate the object around the grasping direction $\boldsymbol{n^j}$, alike section 5.2.2.

Regarding the orientation, the desired end-effector orientation axis $\boldsymbol{z_{o,d}^i}$ is set to $\boldsymbol{n^i}$, while $\boldsymbol{\omega_d^i}$ is here also the sum of two components, one to align $\boldsymbol{z_o^i}$ with $\boldsymbol{z_{o,d}^i}$ and another one ($\omega_h \boldsymbol{n^j}$) to rotate the object along $\boldsymbol{n^j}$ (see Appendix D for more details).

Similarly to section 5.2.3, haptic feedback is also provided to avoid unintentional break of the grasp by reflecting the force error between the measured and desired contact force to the master interface:

$$\boldsymbol{\tau_d^i} = \begin{cases} \boldsymbol{\Upsilon_F^C}(\boldsymbol{F_{ext}^i} + F_d^{C,i} \boldsymbol{n^i}) & i = j \\ \boldsymbol{0_{5\times1}} & \text{otherwise} \end{cases} \tag{5.13}$$

where $\boldsymbol{\Upsilon_F^C} \in \mathbb{R}^{5\times3}$ is the force telefunctionning matrix acting as $\boldsymbol{\Upsilon_F^U}$ in Eq. 5.3.

Finally, we previously discussed how the transition from uncoordinated to coordinated robot control occurs using the grasp condition in Eq. 5.6. To switch back to uncoordinated control, we proceed as illustrated in Figure 5.4. First, the human should break the grasp through v_h. From there, the robots remain in coordinated control as long as the distance between them (i.e., $\|\boldsymbol{x^D}\|$, see section 3.3.1 for the definition of $\boldsymbol{x^D}$) does not exceed a threshold $\varepsilon_D > 0$. If the threshold is exceeded, both robots stop moving ($\boldsymbol{\dot{x}_d^i} = \boldsymbol{0_{3\times1}}$) until the human fetches them with the foot interfaces. The fetching phase consists to move each foot until the distance between the desired attractor commanded through the foot (i.e., $\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{i}}$ in Eq. 5.2) and the current robot position $\boldsymbol{x}^{\boldsymbol{i}}$ becomes lower than a threshold $\varepsilon_x > 0$. To ease the fetching phase and help the human 'finding' the robots, haptic cues are provided using Eq. 5.14:

$$\boldsymbol{\tau}_{\boldsymbol{d}}^{\boldsymbol{i}} = \boldsymbol{\Omega}_{\boldsymbol{F}}^{\boldsymbol{C}}(\boldsymbol{x}^{\boldsymbol{i}} - \boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{i}}) \tag{5.14}$$

where $\boldsymbol{\Omega}_{\boldsymbol{F}}^{\boldsymbol{C}} \in \mathbb{R}^{5\times3}$ is a telefunctionning matrix mapping the position error to forces at the foot interface. This transition mechanism improves safety by avoiding too fast robot motions that would occur if the robots immediately switch to uncoordinated control after breaking the grasp.

5.2.5 Coordinated Robot Control Without Force Assistance (strategy *C*)

This control strategy only enables auto-coordination of the robotic arms upon contact with the object. It is the same than section 5.2.4, except Eq. 5.8 which gets simplified to:

$$\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} = \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{C}} + \tilde{\dot{\boldsymbol{x}}}_{\boldsymbol{d}}^{\boldsymbol{D}^{\boldsymbol{i}}} + v_h \boldsymbol{n}^{\boldsymbol{i}} \tag{5.15}$$

Moreover, Eq. 5.13 is modified to reflect as feedback to the master interface the measured interaction forces instead of the force error:

$$\boldsymbol{\tau}_{\boldsymbol{d}}^{\boldsymbol{i}} = \begin{cases} \boldsymbol{\Upsilon}_{\boldsymbol{F}}^{\boldsymbol{C}} \boldsymbol{F}_{\boldsymbol{ext}}^{\boldsymbol{i}} & i = j \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{5.16}$$

5.3 Experimental Evaluations

In this section, we evaluate the four control strategies described above in a four-arm manipulation (4A) task. Our objective is to evaluate the benefits of the two types of assistance considered in this work: autonomous contact force generation and auto-coordination of the robotic-arms. To this end, a user study was conducted. The details about the technical implementation can be found in Appendix D.

5.3.1 Experimental Setup and Task

In the 4A task, the human uses his two biological arms to work on an object, while being assisted by two robotic arms for supporting and moving the object. The two robotic arms are composed of two KUKA LWR IV+ robots. A 6-axis ATI force torque sensor is mounted on each end-effector on which a 3D printed flat palm tool is attached for grasping the object. Two foot interfaces are used as master interfaces to share the control of the robotic arms with the human. The object to manipulate is illustrated in Figure 5.5. The 4A task consists in reaching, grasping and moving the object between different locations, to inspect and tighten/loosen

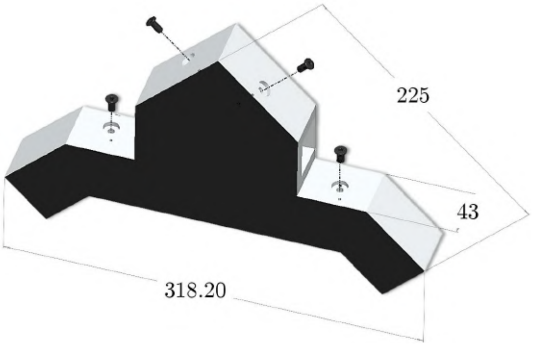


Figure 5.5 – Object's description (©Jacob Hernandez Sanchez). The object to manipulate is a 3D printed piece with a mass of 0.525 kg. It has four thread inserts at different positions, in which M5 hex-head screws have to be loosened/tightened. The relevant object's dimensions are highlighted in the figure (units are in mm).

the screws depending on their status, and to weight the object. The object is initially placed at a fixed position on a wooden board serving as a table (see Figure 5.6a). On one side of the table, the user is sitting on a chair with both feet positioned on the foot interfaces. The initial distance between the feet is approximately 0.5 m. Each foot starts in the center of each platform's workspace. The type of footwear is not constrained, so users are free to use their own shoes. Moreover, the foot-rest of the platform is adjusted in length to fit the participant shoes. Then, the feet are strapped and secured to the platforms. Both platforms are horizontally aligned. The distance between the robot's bases is approximately 0.9 m. The initial resting position of the robots' end-effectors is approximately at the center of the working table, and at a height matching the level of the user's chest. The chair position is adjusted differently for individuals to start always in a neutral position of 90° angles in hip-knee-ankle (w.r.t. horizontal axis). However, the chair's height and the distance between the foot interfaces are kept fixed for all participants. If some participants do not not meet approximately 90° angle in the knee because of being too tall, then extra cushions are used upon the chair, to help them meet this initial condition. The two robotic arms are placed in front of the human on the other side of the table. From there, the user is asked to complete the 4A task by achieving successively the following eight sub-tasks which are illustrated in Figure 5.6:

1. Starting from their initial pose (Figure 5.6a), control the robotic arms to reach the object and grasp it (Figure 5.6b). The grasp is encouraged by coordinating the position of the end-effectors as diametrically opposed. Thereby, the holding force is enabled by the normal forces of each robotic arm on the object as well as the friction they provoke by the interaction with gravity.

2. Lift the object and bring it in front of him/her (Figure 5.6c).

3. Use their hands to tighten/loosen the screws while the object is supported by the robot (Figure 5.6d). If possible, the human should use the robots to rotate the object for a better access to the screws.

4. Bring and position the object on a balance (Figure 5.6e).
5. Break the grasp (Figure 5.6f) and weight the object (Figure 5.6g).
6. Reach and grasp the object again.
7. Move the object to the final position (Figure 5.6h) and break the grasp (Figure 5.6i).
8. Bring the robots to their initial pose (Figure 5.6j).

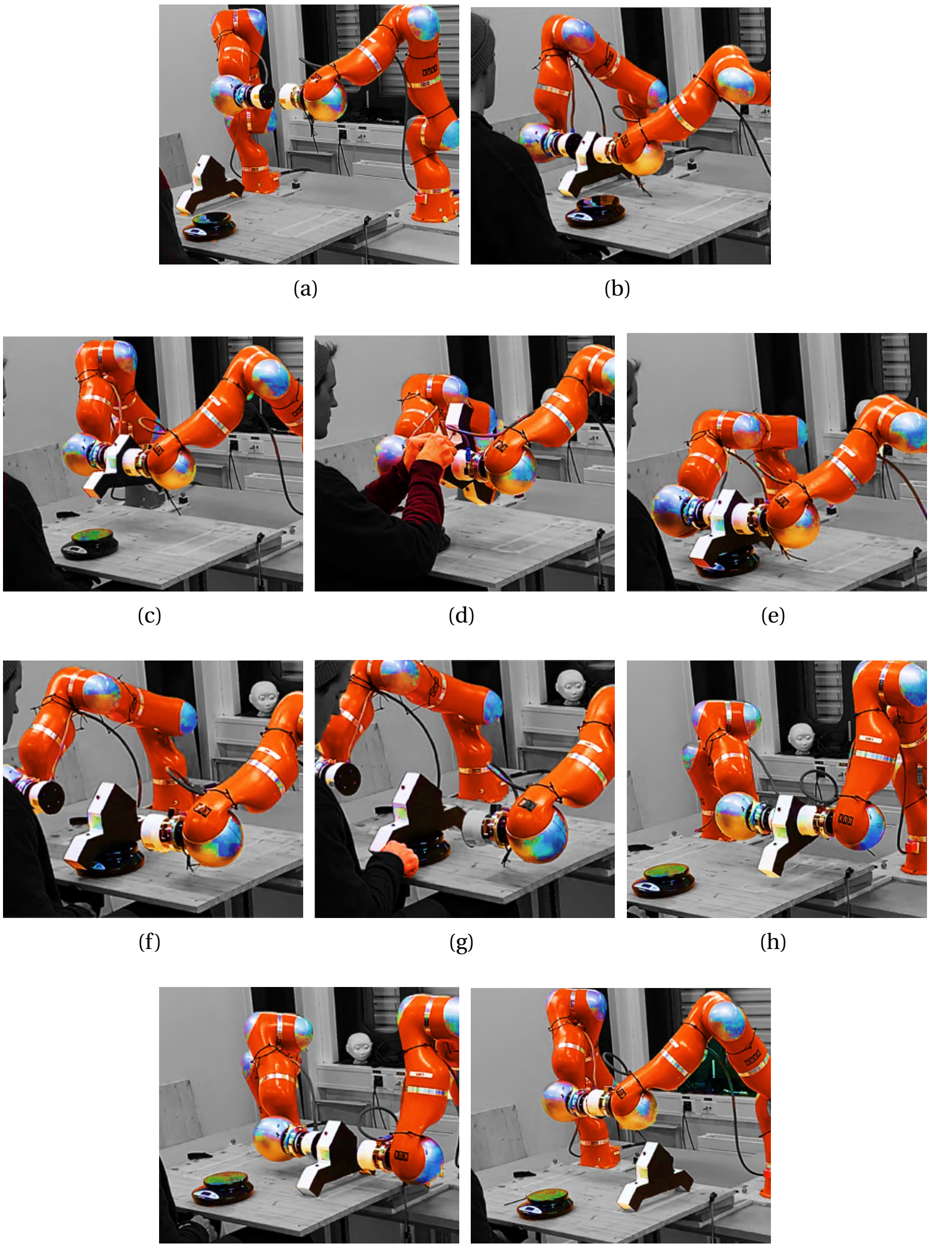
Twelve subjects performed the 4A task four times, to test the four control strategies. All subjects were male lab members, of 28.00 ± 2.93 years old and initially naive in the use of foot interfaces to control robotic arms. They were however familiar with robotic systems and/or work with robot technologies. For each subject, the order of the control strategies was randomized. For each control strategy, each participant had a preliminary training session lasting up to 5 min (depending on the participant's confidence), to get familiar with the control mode including the mapping, scaling and haptic feedback from the foot interfaces.

The mapping according to the control strategy is explained in Figure 5.7. Regarding haptic control, for those axes using a position-velocity mapping (i.e., yaw and $roll$), a joystick is simulated via virtual spring-damper. Whereas in axes using a position-position mapping (i.e., x, y, $pitch$), damping is used to smooth down the motion of the feet. All these gains are experimentally tuned for comfort of the user, based on a exploratory study of two people (both males, age 26.5 ± 0.5), to be uniform across control strategies. Moreover, the range of forces allowed to be reflected to the feet are carefully chosen based on findings from related works for adequate perception and to avoid startling the user (Abbink and van der Helm (2004); Ichinose et al. (2013); Geitner et al. (2018)). After training, the participant was asked to perform the real task within 5 min. The experimental protocol was approved by the Human Research Ethics Committee of EPFL and the methods were carried out in accordance with the approved protocol. Furthermore, the procedure adhered to basic ethical practice from the Nuremberg Code.

5.3.2 Evaluation Metrics and Statistical Studies

To study the differences between the four control strategies, subjective and objective metrics are defined. Subjective metrics include two questionnaires that were filled for each control strategy:

- The first questionnaire measures perception of fluency of the interaction through eight categories including: usefulness, predictability, fluency, trust, goal perception, robot intelligence and satisfaction (Hoffman (2019)). Figure D.1 in Appendix D shows the list of questions/items for each category with the associated Cronbach's alpha value as a measure of internal consistency.
- The second questionnaire is the NASA Task Load Index (NASA-TLX) measuring the participant's performance demand (i.e., how satisfied is the user with her/his performance



(a) (b)

(c) (d) (e)

(f) (g) (h)

(i) (j)

Figure 5.6 – Snapshots of the four-arm manipulation task performed by a human using the $U+F$ control strategy. Starting from their initial resting position (5.6a), the human moves the robots to reach and grasp the object located at the initial position on his left. Once the grasp is secured (5.6b), the human lifts the object to bring it in front of him (5.6c), and starts tightening/loosening the screws on the object with his hands (5.6d). In this phase, the human can control the robots to rotate the object to have a better access to the screws. After that, the human brings the object on a weight scale in front of him (5.6e). Once done, the human breaks the grasp (5.6f) and measures the object's weight (5.6g). Then, the object should be grasped again and transported to the final position on the right (5.6h). Finally, after breaking the grasp (5.6i), the human moves the robots to their initial position approximately (5.6j).

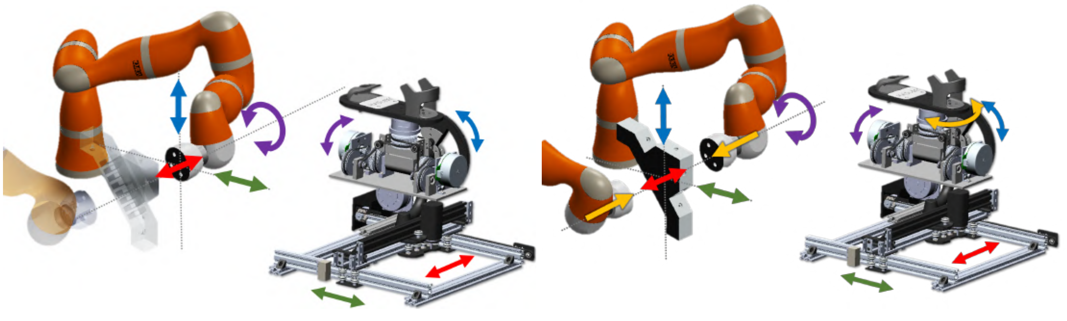

(a) (b)

Figure 5.7 – Color code mapping of the degrees of freedom between the robotic arms and the foot interfaces (©Jacob Hernandez Sanchez). Figure 5.7a describes the **uncoordinated strategies** (U and U+F,) while Figure 5.7b) represents the **coordinated strategies** (C and C+F). In both cases, the linear axes of the foot interfaces (red and green) are mapped to linear motions for the robots, while the $pitch$ axis is transformed in a linear vertical (z) motion (blue). On the other hand, the $roll$ motion (purple) is transformed in the self-rotation of the robot(s') end-effector(s). Note that for the uncoordinated strategies, the end-effectors rotate in the same direction when both feet do a $roll$ motion in opposite directions (bio-mechanically the same muscle groups are used). Furthermore, the coordinated strategies (Figure 5.7b) include the yaw rotation (yellow) for controlling the robots' spacing along the grasping direction (linear motion). In general, we use a position-velocity mapping when the range of motion of the robot is larger than the human input (Kim et al. (1987)), such as for the $roll$ and yaw axes.

-in reversed scale-), effort (i.e., how hard -physically and mentally- the user has to work to accomplish her/his level of performance), frustration (i.e., how frustrated the user felt), mental demand (i.e., how much mental activity was required), physical demand (i.e., how much physical activity was required), and temporal demand (i.e., pace of the execution of the task) of the task (Hart and Staveland (1988)).

The answers to these questionnaires are assessed using a 5 Point Likert Scale from 1 = *strong disagreement* to 5 = *strong agreement.*

On the other hand, the objective metrics are composed by:

- The individual sub-tasks' efficiency σ_k defined as:

$$\sigma_k = \frac{t_{max} - t_k}{t_{max}} \tag{5.17}$$

  where $\sigma_k \in [0,1]$ is the relative efficiency metric for the sub-task k, with $k \in [1, s_{max}]$, where $s_{max} = 8$ is the maximum number of sub-tasks. This metric considers timeliness since t_k is the time needed to perform sub-task k, and $t_{max} = 5\ min$ is the maximum time allowed for the overall task. Consequently, when a sub-task was not accomplished, t_k is considered to be equal to t_{max} and hence the efficiency (σ_k) is zero.

- The number of object falls N_f that counts the number of times the object unintentionally falls during the task (e.g., while being grasped or after collision with one of the robotic arms).

- The number of sub-tasks achieved $s \in [0, s_{max}]$ measuring completeness of the whole 4A task.

- The overall success metric β_T encompassing the sub-tasks' efficiency, the total number of sub-tasks to achieve and the number of object falls, that we define as:

  $$\beta_T = \frac{\sum_k^s \sigma_k}{s_{max} \exp\left(\frac{N_f}{s_{max}}\right)}. \tag{5.18}$$

  with $\beta_T \in [0, 1]$. Note that replacing Eq. 5.17 in Eq. 5.18 yields:

  $$\beta_T = \frac{\left(s - \frac{t_{total}}{t_{max}}\right)}{s_{max} \exp\left(\frac{N_f}{s_{max}}\right)}. \tag{5.19}$$

  where, t_{total} is the total time used to complete the whole task. Thereby, this formula considers efficiency, completeness and penalizes object falls.

The evaluation metrics presented above are used in two statistical studies. Study I analyses the effect of the four different control strategies (*U*, *U+F*, *C+F*, and *C*) while study II analyzes the individual effect of the assistance modalities: force assistance (*F.A*) and auto-coordination assistance (*C.A*), as well as their interaction. In the next section, we report the main results obtained from these two studies (see Amanhoud et al. (2021) for more details).

5.4 Results

Results from Study I can be observed in Figures D.2 and D.3 analyzing the responses to all subjective and objective metrics respectively, for all the control strategies. A first point to highlight is the absence of significant differences among the control strategies regarding the elements of task load, except for physical demand where the control strategies with force assistance ($C + F$ and $U + F$) outperform the two other ones. Strategies with force assistance generally provide better perception of fluency where significant differences are obtained for ease of use, usefulness, satisfaction, human-robot fluency and goal perception. The same observations can be made regarding objective metrics, where strategy *C+F* is particularly efficient into reducing the number of object falls (median of 0) when compared to the baseline strategy *U* (median of 2), thereby providing the highest overall success. The ability of control strategy *C+F* (and *U+F*) to keep a robust grasp can be explained through Figure 5.8 analyzing the averages and standard deviations of the contact forces applied by both robotic arms during the task. Indeed, it can be noticed that control strategy *C+F* achieves the second strongest grasp (right: 26.64 ± 1.59 *N*, left: 26.57 ± 1.54 *N*) while keeping the second lowest median of standard deviations (left and right: 4.37 *N*). This is thanks to the autonomous contact

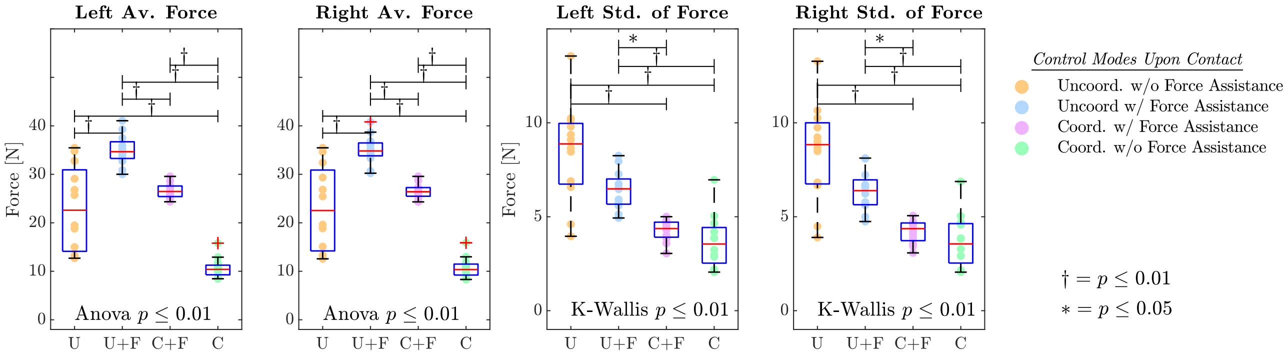
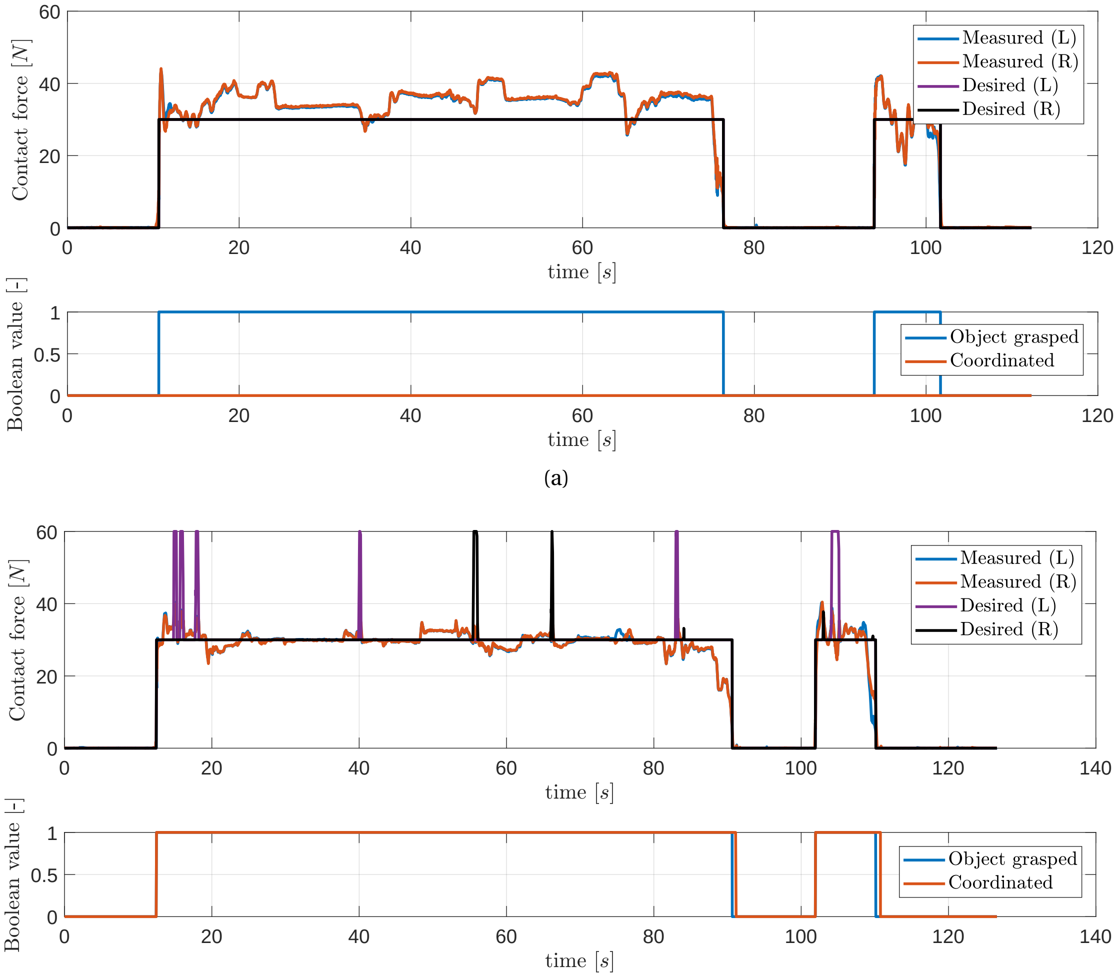


Figure 5.8 – Averages and standard deviations of the contact forces for each robot and control strategy (©Jacob Hernandez Sanchez). Twelve participants were involved in this study. The Tukey plots are overlaid on the data points for clarity. The type of statistical test (one-Way Anova with repeated measures or Kruskal-Wallis, depending on the normality of the data) is indicated in each subplot with its p-value. Furthermore, the pair-wise significant differences under 1% and 5% are indicated with the symbols explained in the legend of the figure.

force generation provided once the object is grasped (see Figure 5.9 as examples of measured and desired contact forces when force assistance is used), allowing participants to delegate the application of the grasping force to the robots. In contrast, for control strategy C, the average contact force applied by the participants is the lowest (right robot: $26.64 \pm 1.59\ N$, left robot: $10.70 \pm 2.08\ N$), and probably not sufficient to keep a robust grasp, as reflected in the high amount of object falls (median of 4.5). On top of that, having only coordination assistance results in the lowest median standard deviation along the task ($C : 3.55\ N$ for both arms) because of the difficulty that many participants found when holding and keeping the grasp with a single foot without force assistance. This difficulty is reflected in the subjective results where control strategy C first depicts the lowest predictability among all strategies. It is moreover significantly worse in terms of ease of use, usefulness, human robot fluency, goal perception and satisfaction than the two strategies $U{+}F$ and $C{+}F$. In general, for these metrics, when using a single foot, force assistance is crucial to have responses similar to two foot control.

The positive impact of force assistance is particularly highlighted in Study II through Figures D.4 and D.5 analyzing the effect of the two assistance modalities on the subjective and objective metrics respectively. Indeed, force assistance significantly reduces physical and performance demand, as well as frustration. It also reduces the perceived effort and mental demand, but not notably, probably because the task is not challenging enough to significantly increase the mental demand and effort when no force assistance is provided. Similarly, force assistance increases the perception of fluency, providing significant differences in all categories except trust. Moreover, study II confirms the significant role of force assistance into reducing the number of object falls and completing the overall task with success.



(a)

(b)

Figure 5.9 – Examples of measured and desired interaction forces obtained with the shared control strategies including force assistance, namely *U*+*F* (Figure 5.9a), and *C*+*F* (Figure 5.9b). The "object grasped" signal informs about the object's grasping state based on the grasp condition (Eq. 5.6). The "coordinated" signal reflect whether the robots move in coordinated manner or not. (*L*) and (*R*) refer to the left and right robot respectively. Note that for control strategy *U*+*F* both robots have the same desired force profile when the object is grasped. For *C*+*F*, the desired force profiles are designed as explained in Appendix D. These examples are obtained from one participant to the user study.

Regarding coordination assistance, Study II shows no benefits in terms of task load. The absence of positive impact in metrics such as mental workload can be explained by the unfamiliar gesture of grasping with the foot (using abduction/adduction of the foot or *yaw*) as people are more used to grasp objects using at least two end-effectors (e.g., two fingers, two hands, fingers against the palm, etc). Similarly, during the experiments, we noticed that subjects had a consistent tendency to continue moving the non-dominant foot, even after the auto-coordination is activated, a tendency that was receding as the task evolves. This implies an extra mental effort to consciously stop the motion of the non-dominant foot, which can cause the user to struggle in the switching phase. Furthermore, foot rotation are biomechanically coupled and cannot be controlled completely independently. This is particularly true for the foot motion required to close the robots' grip and rotate the object, since yaw and $roll$ of the foot are coupled in the talocrurar/ankle joint. Therefore, it is more difficult to simultaneously apply force, keep the foot pose (to hold the grip), and move the object. This potential extra physical effort could affect metrics such as physical demand or perceived effort. The reasons discussed above may also explain the absence of positive benefits in all categories of fluency, where in particular, a significant negative impact is obtained for predictability and goal perception. Nevertheless, the individual negative effect of coordination assistance can be sometimes compensated by the interaction with force assistance as demonstrated by the results for the categories of ease of use, usefulness, and goal perception. Concerning the objective metrics, coordination assistance has no effect on the number of object falls (as this is more related to force) but it significantly reduces task completeness. Indeed, completing the whole task when only having coordination assistance (i.e., control strategy C) was not possible for several participants (5 out of 12), who after grasping the object the first time, had difficult to keep it secure, affecting the efficiency of the next sub-tasks and hence the whole task completeness.

5.5 Discussion and Conclusion

In this chapter, we proposed four (shared) control strategies for proactive assistance of feet telemanipulation to perform four-arm manipulation tasks. The goal was to perform robust collaborative contact tasks with the human. To this end, we used the contact state information to proactively facilitate the task for the human. In our scenario, the hands are free to physically interact with the system and to perform high dexterity tasks upon the object. To assist the human operator in supporting and moving the manipulated object, two assistance modalities are proposed: autonomous contact force generation and auto-coordination of the robotic arms. A user study evaluated the proposed strategies, the two assistance modalities, and their interactions, on a four-arm manipulation task. We measured perceived workload, subjective human-robot fluency and (custom) objective metrics such as task efficiency, completeness, and success.

Overall, this work provides insights on the pertinence of shared control strategies in four-arm manipulation tasks with the particularity of using the feet as user inputs. Our main finding in the user study is that force assistance is in general perceived as better than coordination assistance in a bilaterally isomorphic task during bipedal telemanipulation where grasping is involved. Indeed, force assistance provides more robust manipulations and improves the efficiency of a four-arm task where hands and feet are involved. However, combining a single foot control (instead of two feet) with force assistance resulted in a higher success in the execution of the task. Moreover, the user expressed better ease of use, usefulness, impression of robot intelligence and satisfaction during the collaborative task when using the dual assistance approach.

Putting aside the assistance modalities, intrinsically, the tele-operation scheme used to control both robots is a standard force-reflecting bilateral tele-operation architecture (Niemeyer et al. (2016)), where the interaction forces measured by a slave robot are reflected to the master interface, whereas the slave robot tracks a reference position/velocity controlled through the master interface. Although these position-force architectures often select a PD controller for the tracking of the telemanipulator's reference commands, in this work we use the DS-based impedance controller specifically to control the robots' linear motion, according to the task (reaching, dual arm grasping and rotating the object in the sagital plane). We used the DS-based framework because it opens the door to (shared) tasks for the robot with an increasing level of complexity encoded in the time-invariant dynamical system. Indeed, DS can be exploited to generate complex smooth behaviours, for instance to reach a target while avoiding obstacles (Khansari-Zadeh and Billard (2012); Huber et al. (2019)) or to switch between multiple tasks (Khoramshahi and Billard (2019)). In this work, we chose to start with a simple linear DS for the uncoordinated control or baseline (i.e., pure tele-operation) and increased the complexity for the other assistance modalities. For example in $U{+}F$, we merge the motion input of the human with the autonomous-force generation in a unified DS formulation in Eq. 5.4 that is tracked with the same impedance control as the baseline. This is also the case for the strategies with coordination assistance, in which we couple the dynamical systems that both robotic arms are tracking. Therefore, the reactivity and adaptability of this framework makes it suitable for developing shared-control strategies in human-robot collaborative scenarios where the robot has to interact both with humans and its environment. Note finally that, in the baseline strategy in which the attractor is the mapping of the foot's position in relation to a desired robot end effector's position, the formulation is approximately equal to a PD controller. The only difference is that the $\boldsymbol{P}$ matrix is no longer diagonal, but it becomes state varying and task dependent. Specifically, from Eq. 5.1 and 2.16, the apparent stiffness (the gradient of the damping force in Eq. 2.16 with respect to $\boldsymbol{x^i}$) can be approximated as $\boldsymbol{P^i(x^i) = D^i(x^i)A^i}$, where $\boldsymbol{P^i(x^i)} \in \mathbb{R}^{3\times 3}$ is a virtual stiffness matrix if the simple linear DS tracking equation had to be re-formulated as a PD controller. Indeed, the state-varying damping matrix $\boldsymbol{D^i(x^i)}$ of the DS-impedance law changes according to the task; specifically, the robot enables a selective compliance in the directions not relevant to the task, e.g., low gains in the directions orthogonal to the grasping direction. For this reason, considering the conditions in which the task was developed (no significant external disturbances), we believe that the results obtained and the conclusions drawn from the experimental study are agnostic to the dynamical system framework. In turn, we explain its implementation on its known robustness and flexibility for physical human robot interaction.

6 Four-Arm Robotic Assisted Laparoscopic Surgery

Note: The material presented in this chapter is adopted from:
Hernandez Sanchez, J., Amanhoud, W., Billard, A., and Bouri, M. (2021). Enabling hybrid robotic-assisted solo-surgery with four arms via haptic foot interfaces. Under review at The International Journal of Robotics Research.

6.1 Introduction

Laparoscopic surgery is a Minimally Invasive Surgery (MIS) performed through small holes in the abdomen. It is done by a surgeon and one or two assistants usually controlling an endoscope (or camera) that provides visual feedback for the surgeon, and a retractor used to make space for the surgeon's instruments. Several problems have been observed and reported during the surgical procedures such as fatigue and miscommunications among the surgical team. To improve surgeons and assistants' daily life, Robotic-Assisted Laparoscopic Surgery (RALS) is a promising direction. Following that direction, we aim to achieve solo laparoscopic surgery where the surgeon is provided two robotic assistants manipulating the camera and the retractor, and controlled with the feet of the surgeon thanks to two haptic foot interfaces (see Figure 1.3). In this chapter, we present a control framework to achieve safe collaborative RALS with the envisioned system. The framework first includes an inverse kinematics solver, formulated as a Quadratic Programming (QP) problem and solved in real time, to achieve cooperative human-robot surgical tasks while satisfying all the safety restrictions (e.g, collision, joint limits etc.) and task-related constraints (e.g. the remote center of motion (RCM) constraint). Dynamical systems are exploited to define the desired task-specific tool tip dynamics together with admittance control to allow hand-guiding motion from persistent physical human interactions. Thereby, the robotic assistants can achieve their desired tasks precisely, comply to human disturbances, and re-plan their tasks from the disturbed states, while respecting safety and task constraints. To ease the control of the camera through the foot, an adaptive autonomous tool tip following is presented as an assistance modality. Furthermore, haptic cues are designed to increase the surgeon's situational awareness.

The work in this chapter is under preparation for submission to a scientific journal with Jacob Hernandez Sanchez as co first author. In terms of theoretical development, Jacob focused on the control of the foot interfaces and the actuated gripper/retractor for which an autonomous force-controlled grasping was designed as an assistance modality to provide a stable and secure grasp with the foot. Jacob is also the main contributor to the statistical analysis of the data gathered during the user study conducted to evaluate the assistance modalities. In this chapter, the theoretical material developed by the thesis author is exclusively reported. Only the main results obtained from the user study are briefly presented to validate part of the theoretical development. Regarding the experimental study, it was designed and conducted collaboratively.

The related work for this chapter is reviewed in section 1.6. The four-arm robotic assisted surgical system is described in section 6.2. The method is presented in section 6.3 and evaluated in section 6.4. A discussion about the method and results obtained is finally provided in section 6.5.

6.2 Four-Arm Robotic Assisted Surgical System

The four-arm robotic assisted surgical system used in this chapter is illustrated in Figure 6.2 and consists in:

- Two 7 DoFs robotic arms (Franka Emika Panda), one holding a camera ($640 \times 480\ px$) and the other one holding a gripper mounted on a 6-axis ATI force torque sensor.
- A 1 DoF actuated gripper. This gripper is current-controlled and provides force-feedback estimation of the interaction with the objects being grasped.
- Two 5 DoFs robotic foot interfaces. These platforms are world-grounded kinesthetic haptic devices and include the leg's dynamics in their control to alleviate fatigue.
- A seat for the operator. Contrary to conventional laparoscopic surgery, the operator is envisioned to be in a sitting position, to be able to use comfortably their four limbs.
- A phantom with holes mimicking a body where the surgical tasks for training are performed.
- A screen providing visual feedback to the participant.

In this system, the surgeon utilizes conventional laparoscopic tools with their hands and uses the feet to control two robotic assistant manipulators holding surgical tools.

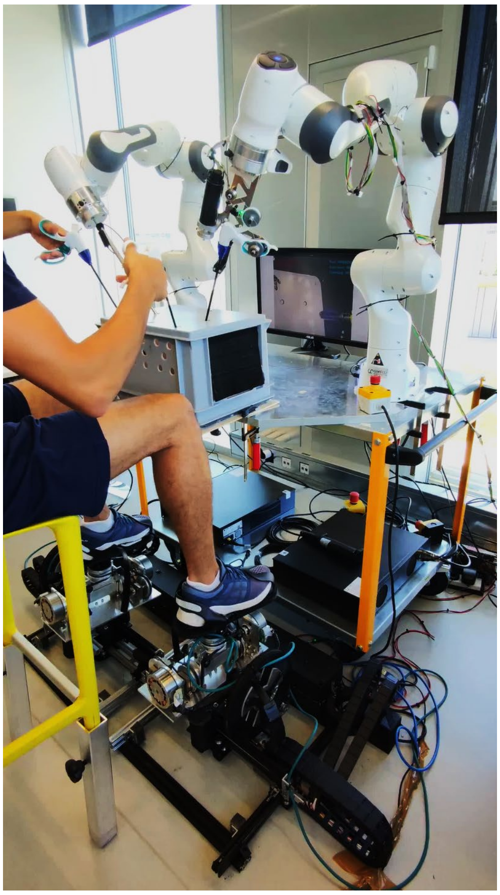


Figure 6.2 – Four-arm robotic assisted surgical system.

6.3 Method

In this section, we present the methodologies of control developed for the two robotic arms that are part of our four-arm setup. We use superscript $i = C, G$ to represent the robotic assistant holding the camera or the gripper and superscript $m = C, G$ to represent the corresponding foot interfaces.

6.3.1 Control Framework Overview

The architecture of the control framework is illustrated in Figure 6.3. A QP-based inverse kinematics is at the center of the framework (see section 6.3.3). It is used to satisfy all the task and safety related constraints imposed by our scenario and imposed by laparoscopic surgery in general, such as the RCM constraint. The QP outputs desired joint velocities which are integrated forward to get desired joint positions. The desired joint positions are tracked with a joint impedance controller outputting control torques for the robot (see section 6.3.2).

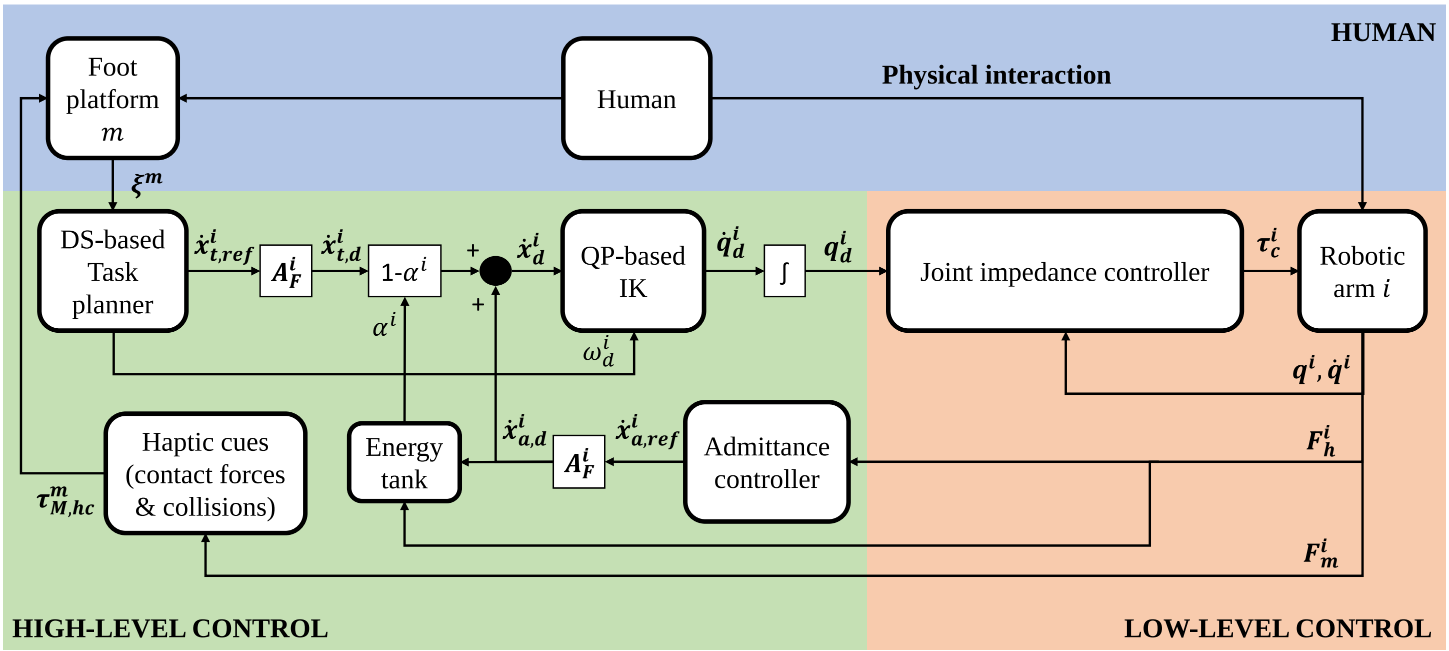


Figure 6.3 – Control framework architecture.

Because of the RCM constraint, the task of the tool tip is limited to four degrees of freedom. In this work, the tool tip task is defined by a desired linear velocity $\dot{\boldsymbol{x}}^i_d$ and a self-rotation velocity ω^i_d. The derivation of these two terms is described in section 6.3.4. In particular, $\dot{\boldsymbol{x}}^i_d$ is a weighted combination of two terms: a desired task specific velocity $\dot{\boldsymbol{x}}^i_{t,d}$ and a desired admittance velocity $\dot{\boldsymbol{x}}^i_{a,d}$ (see section 6.3.4.1). Both of them are obtained after scaling their respective reference velocities. The scaling matrix $\boldsymbol{A}^i_F$ is used to take into account the fulcrum effect (see section 6.3.4.7). The reference admittance velocity $\dot{\boldsymbol{x}}^i_{a,ref}$ allows the robot to comply to physical human interaction. It is obtained from an admittance controller taking as input the estimated human forces $\boldsymbol{F}^i_h$ (see section 6.3.4.6). $\boldsymbol{F}^i_h$ and $\dot{\boldsymbol{x}}^i_{a,d}$ are both used as input to an energy tank whose state is used to compute a modulation gain α^i modulating $\dot{\boldsymbol{x}}^i_{t,d}$ (see section 6.3.4.8). The goal is to give priority to $\dot{\boldsymbol{x}}^i_{a,d}$ when persistent physical human interaction is measured. Regarding $\dot{\boldsymbol{x}}^i_{t,ref}$ and ω^i_d, they are both obtained from a task planner taking as input the state/pose $\boldsymbol{\xi}^m$ of the foot platform (see sections 6.3.4.3, 6.3.4.4, and 6.3.4.5). Finally, the framework includes the design of haptic cues $\boldsymbol{\tau}^m_{M,hc}$ sent to the foot to increase the situational awareness of the surgeon, for instance about the measured tool tip contact forces $\boldsymbol{F}^i_m$ (see section 6.3.5). In the following sections, we present in detail all the components of this architecture.

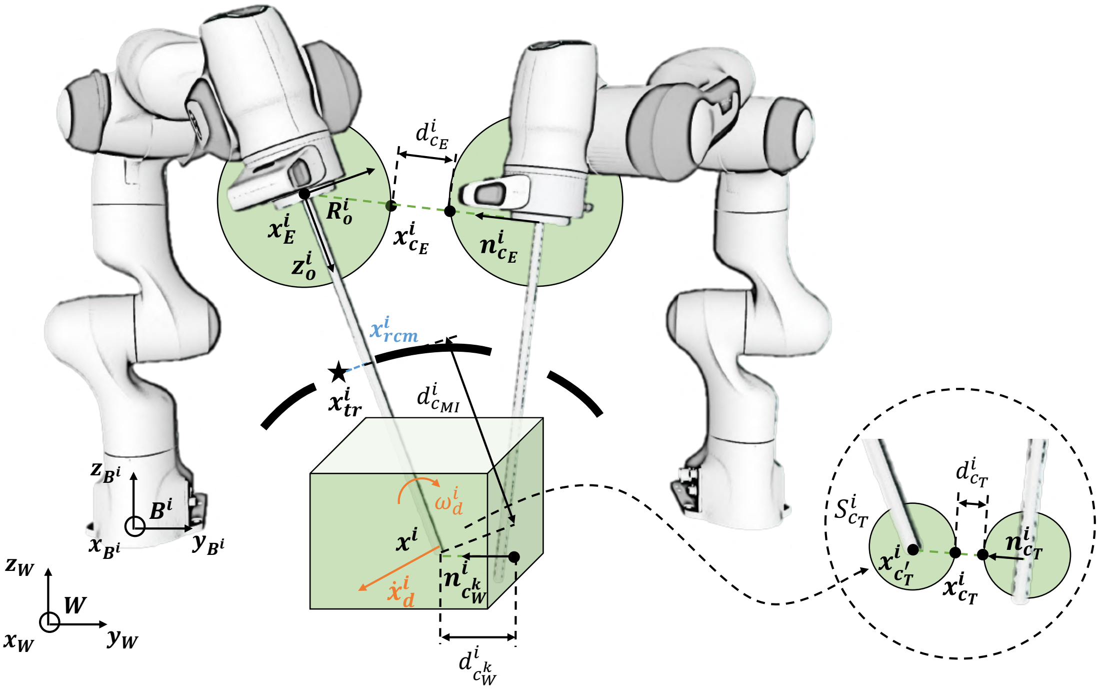


Figure 6.4 – Representation of the two assisting robotic arms with the main task variables and constraints imposed by the laparoscopic tasks.

6.3.2 Robotic Arms' Control Law

In this chapter, the robotic arms' control torque $\boldsymbol{\tau}^i_c$ (see Eq. 2.5) is computed from a joint impedance law (assuming that the robots are torque-controlled) such that:

$$\boldsymbol{\tau}^i_c = \boldsymbol{K}^i_R(\boldsymbol{q}^i_d - \boldsymbol{q}^i) - \boldsymbol{D}^i_R\dot{\boldsymbol{q}}^i + \boldsymbol{g}^i(\boldsymbol{q}^i) + \boldsymbol{C}^i(\boldsymbol{q}^i, \dot{\boldsymbol{q}}^i)\boldsymbol{q}^i \tag{6.1}$$

where $\boldsymbol{K}^i_R \in \mathbb{R}^{N\times N}$ and $\boldsymbol{D}^i_R \in \mathbb{R}^{N\times N}$ are stiffness and damping matrices, while $\boldsymbol{q}^i_d \in \mathbb{R}^N$ denotes the desired joint positions. Note that Eq. 6.1 also includes gravity and Coriolis cancellation terms. The main reason for controlling the robots in joint space is to provide high task tracking accuracy, especially when targeting low velocities. Although the robotic arms are commanded in joint space through $\boldsymbol{q}^i_d$, the task is formulated in the operational space (at the tool tip level). To resolve the Inverse Kinematics (IK) of the task, we exploit a Quadratic Programming (QP) method presented in the next section.

6.3.3 QP-based Inverse Kinematics

For each robotic arm, we propose to solve the following inverse kinematics problem (please refer to Figure 6.4 highlighting the main task variables and constraints imposed by the task):

$$\begin{aligned}\min_{\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}},\boldsymbol{\delta}_{\boldsymbol{rcm}}^{\boldsymbol{i}},\boldsymbol{\delta}_{\boldsymbol{task}}^{\boldsymbol{i}}} \quad & \frac{1}{2}\left(\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}\,T}\boldsymbol{H}_1\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} + \boldsymbol{\delta}_{\boldsymbol{rcm}}^{\boldsymbol{i}}{}^{T}\boldsymbol{H}_2\boldsymbol{\delta}_{\boldsymbol{rcm}}^{\boldsymbol{i}} + \boldsymbol{\delta}_{\boldsymbol{task}}^{\boldsymbol{i}}{}^{T}\boldsymbol{H}_3\boldsymbol{\delta}_{\boldsymbol{task}}^{\boldsymbol{i}}\right) + \\ & \frac{1}{2}\left(\boldsymbol{q}_{\boldsymbol{c}}^{\boldsymbol{i}} - (\boldsymbol{q}_{\boldsymbol{0}}^{\boldsymbol{i}} + dt\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}})\right)^T\boldsymbol{H}_4(\boldsymbol{q}_{\boldsymbol{c}}^{\boldsymbol{i}} - (\boldsymbol{q}_{\boldsymbol{0}}^{\boldsymbol{i}} + dt\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}}))\end{aligned} \tag{6.2}$$

subject to the RCM constraint:

$$\boldsymbol{J}_{\boldsymbol{x}_{\boldsymbol{rcm}}^{\boldsymbol{i}}}^{\boldsymbol{i}}\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} = a_{rcm}(\boldsymbol{x}_{\boldsymbol{tr}} - \boldsymbol{x}_{\boldsymbol{rcm}}) + \boldsymbol{\delta}_{\boldsymbol{rcm}}^{\boldsymbol{i}} \tag{6.3}$$

the tool tip task constraint:

$$\begin{pmatrix} \boldsymbol{J}_{\boldsymbol{x}^{\boldsymbol{i}}}^{\boldsymbol{i}} \\ [\boldsymbol{0}_{\boldsymbol{1}\times(\boldsymbol{N}-\boldsymbol{1})} \;\; 1] \end{pmatrix}\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} = \begin{pmatrix} a_{task,1}\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} \\ a_{task,2}\omega_d^i \end{pmatrix} + \boldsymbol{\delta}_{\boldsymbol{task}}^{\boldsymbol{i}} \tag{6.4}$$

the joint position limit constraint:

$$\boldsymbol{q}_{-}^{\boldsymbol{i}} - \boldsymbol{q}_{\boldsymbol{0}}^{\boldsymbol{i}} \leq dt\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} \leq \boldsymbol{q}_{+}^{\boldsymbol{i}} - \boldsymbol{q}_{\boldsymbol{0}}^{\boldsymbol{i}} \tag{6.5}$$

the end-effector twist limit constraint:

$$\begin{pmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{E},-} \\ \boldsymbol{\omega}_{\boldsymbol{E},-} \end{pmatrix} \leq \boldsymbol{J}^{\boldsymbol{i}}(\boldsymbol{q}^{\boldsymbol{i}})\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} \leq \begin{pmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{E},+} \\ \boldsymbol{\omega}_{\boldsymbol{E},+} \end{pmatrix} \tag{6.6}$$

the end-effector collision constraint:

$$\boldsymbol{n}_{\boldsymbol{c_E}}^{\boldsymbol{i}}{}^{\boldsymbol{T}}\boldsymbol{J}_{\boldsymbol{x}_{\boldsymbol{c_E}}^{\boldsymbol{i}}}^{\boldsymbol{i}}\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} \geq \text{-}\lambda_{c_E}\frac{d_{c_E}^i - d_{c_E,s}}{d_{c_E,inf} - d_{c_E,s}} \tag{6.7}$$

the tool tip collision constraint:

$$\boldsymbol{n}_{\boldsymbol{c_T}}^{\boldsymbol{i}}{}^{\boldsymbol{T}}\boldsymbol{J}_{\boldsymbol{x}_{\boldsymbol{c_T}}^{\boldsymbol{i}}}^{\boldsymbol{i}}\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} \geq \text{-}\lambda_{c_T}\frac{d_{c_T}^i - d_{c_T,s}}{d_{c_T,inf} - d_{c_T,s}} \tag{6.8}$$

the workspace collision constraints:

$$\boldsymbol{n}_{\boldsymbol{c_W^k}}^{\boldsymbol{i}}{}^{\boldsymbol{T}}\boldsymbol{J}_{\boldsymbol{x}^{\boldsymbol{i}}}^{\boldsymbol{i}}\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} \geq \text{-}\lambda_{c_W^k}\frac{d_{c_W^k}^i - d_{c_W^k,s}}{d_{c_W^k,inf} - d_{c_W^k,s}} \quad k = 1,\ldots,5 \tag{6.9}$$

the minimum insertion constraint:

$$\boldsymbol{z}_{\boldsymbol{o}}^{\boldsymbol{i}\,\boldsymbol{T}}\boldsymbol{J}_{\boldsymbol{x}^{\boldsymbol{i}}}^{\boldsymbol{i}}\dot{\boldsymbol{q}}_{\boldsymbol{d}}^{\boldsymbol{i}} \geq \text{-}\lambda_{c_{MI}}\frac{d_{c_{MI}}^i - d_{c_{MI},s}}{d_{c_{MI},inf} - d_{c_{MI},s}} \tag{6.10}$$

the joint velocity limits:

$$\boldsymbol{\dot{q}^i_-} \le \boldsymbol{\dot{q}^i_d} \le \boldsymbol{\dot{q}^i_+} \tag{6.11}$$

the RCM slack limits:

$$\boldsymbol{\delta_{rcm,-}} \le \boldsymbol{\delta^i_{rcm}} \le \boldsymbol{\delta_{rcm,+}} \tag{6.12}$$

and the task slack limits:

$$\boldsymbol{\delta_{task,-}} \le \boldsymbol{\delta^i_{task}} \le \boldsymbol{\delta_{task,+}} \tag{6.13}$$

The objective function in Eq. 6.2 is composed of four quadratic terms where $\boldsymbol{H_1} \in \mathbb{R}^{N\times N}$, $\boldsymbol{H_2} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{H_3} \in \mathbb{R}^{4\times 4}$ and $\boldsymbol{H_4} \in \mathbb{R}^{N\times N}$ are positive definite diagonal matrices. The first three terms in the objective function minimize the norm of the optimization variables, i.e., they aim for minimum desired joint state velocities $\boldsymbol{\dot{q}^i_d} \in \mathbb{R}^N$ and tolerance margins for the tool tip task and RCM equality constraints (i.e., Eq. 6.3 and 6.4), where $\boldsymbol{\delta^i_{rcm}} \in \mathbb{R}^3$ and $\boldsymbol{\delta^i_{task}} \in \mathbb{R}^4$ are slack variables. To derive the desired joint state position $\boldsymbol{q}^i_d$ in Eq. 6.1, the optimized joint state velocity $\boldsymbol{\dot{q}^i_d}$ are integrated forward such that:

$$\boldsymbol{q^i_d} = \boldsymbol{q^i_0} + dt\boldsymbol{\dot{q}^i_d} \tag{6.14}$$

where $\boldsymbol{q^i_0} \in \mathbb{R}^N$ is the previous desired joint state position while $dt > 0$ is the time step. The last term in Eq. 6.2 finally tries to keep $\boldsymbol{q^i_d}$ close to the center of the joint state position range $\boldsymbol{q^i_c}$ (and thereby away from the joint limits), defined by:

$$\boldsymbol{q^i_c} = \frac{\boldsymbol{q^i_-} + \boldsymbol{q^i_+}}{2} \tag{6.15}$$

with $\boldsymbol{q^i_-}, \boldsymbol{q^i_+} \in \mathbb{R}^N$ respectively denoting the minimum and maximum joint position angles of the robotic manipulator.

The first equality constraint in Eq. 6.3 implements the well known Remote Center of Motion (RCM) constraint imposed by the trocar. The trocar, whose position is here denoted by $\boldsymbol{x^i_{tr}} \in \mathbb{R}^3$, is the fixed incision point through which the surgical tool must always move. To this end, we define the RCM point $\boldsymbol{x^i_{rcm}} \in \mathbb{R}^3$ as the instantaneous orthogonal projection of the trocar position along the surgical tool:

$$\boldsymbol{x^i_{rcm}} = \boldsymbol{x^i_E} + \boldsymbol{z^i_o}^{\boldsymbol{T}}(\boldsymbol{x^i_{tr}} - \boldsymbol{x^i_E})\boldsymbol{z^i_o} \tag{6.16}$$

with $\boldsymbol{x^i_E} \in \mathbb{R}^3$ denoting the end-effector position, while $\boldsymbol{z^i_o} \in \mathbb{R}^3$ is the axis of the end-effector orientation matrix $\boldsymbol{R^i_o}$ normal to the end-effector plane. The constraint in Eq. 6.3 forces the trocar and RCM points to match thanks to the error term, where $a_{rcm} > 0$, while $\boldsymbol{J^i_{x^i_{rcm}}} \in \mathbb{R}^{3\times N}$ is the jacobian matrix mapping the joint state velocities to the linear velocity of the RCM point. In general, we define the jacobian matrix $\boldsymbol{J^i_y} \in \mathbb{R}^{3\times N}$, where $\boldsymbol{y} \in \mathbb{R}^3$ is the position of a point physically/virtually attached to the robotic arm's end-effector, as:

$$\boldsymbol{J}_{\boldsymbol{y}}^{\boldsymbol{i}} = \boldsymbol{J}_{\boldsymbol{x}_E^i}^{\boldsymbol{i}} - [\boldsymbol{y} - \boldsymbol{x}_{\boldsymbol{E}}^{\boldsymbol{i}}]_{\times} \boldsymbol{J}_{\boldsymbol{\omega}_E^i}^{\boldsymbol{i}} \tag{6.17}$$

with $\boldsymbol{J}_{\boldsymbol{x}_E^i}^{\boldsymbol{i}}, \boldsymbol{J}_{\boldsymbol{\omega}_E^i}^{\boldsymbol{i}} \in \mathbb{R}^{3\times N}$ corresponding to the linear and angular parts of the robotic arm's geometric Jacobian $\boldsymbol{J}^{\boldsymbol{i}}(\boldsymbol{q}^{\boldsymbol{i}}) \in \mathbb{R}^{6\times N}$.

The second equality constraint in Eq. 6.4 ensures that the tool tip achieves a desired task. Due to the trocar constraint, this task is limited to four degrees of freedom (DoFs) and is represented by a desired linear velocity $\dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\boldsymbol{i}} \in \mathbb{R}^3$ and angular velocity $\omega_d^i \in \mathbb{R}$ defining the desired tool self-rotation dynamics. Please, refer to section 6.3.4.1 for the derivation of these two terms. Regarding the task jacobian, note that it is decomposed into $\boldsymbol{J}_{\boldsymbol{x}^i}^{\boldsymbol{i}}$ (obtained from Eq. 6.17) mapping the joint state velocities to the tool tip velocity ($\boldsymbol{x}^{\boldsymbol{i}} \in \mathbb{R}^3$ being the tool tip position), and a row vector relating the tool self-rotation speed to the velocity of the last joint. Lastly, $a_{task,1}, a_{task,2} \geq 0$ in Eq.6.4 are scalar gains used to modulate the desired tool tip dynamics.

Inequations 6.5 and 6.6 respectively implement end-effector velocity and joint position limits. The former is derived from Eq. 6.14 and ensures that $\boldsymbol{q}_{\boldsymbol{d}}^{\boldsymbol{i}}$ never exceeds the joint limits, preserving the robotic arm's physical integrity. The latter is used as a safety measure as the end-effector is the closest part of the robotic arms' body to the human. It is therefore desired that the end-effector velocity complies to the safety standards defined for collaborative robotic arms (for Standardization (2012)).

Moreover, to guarantee safety of the robotic setup, four additional collision avoidance constraints (Eq. 6.7, 6.8, 6.9, and 6.10) are implemented as linear inequality constraints based on Faverjon and Tournassoud (1987). Each collision constraint c requires to define a collision point $\boldsymbol{x}_{\boldsymbol{c}}^{\boldsymbol{i}} \in \mathbb{R}^3$ (i.e., the closest point of the robotic arm body to collide with the obstacle), a unit vector aligned with the collision direction pointing away from the obstacle $\boldsymbol{n}_{\boldsymbol{c}}^{\boldsymbol{i}} \in \mathbb{R}^3$, a measure of distance to the collision $d_c^i \in \mathbb{R}$, as well as some positive security $d_{c,s} \in \mathbb{R}^+$, and influence $d_{c,inf} \in \mathbb{R}^+$ distance values. Whenever d_c^i becomes lower than $d_{c,s}$, the constraint allows to push the collision point away from the obstacle thanks to the jacobian $\boldsymbol{J}_{\boldsymbol{x}_c^i}^{\boldsymbol{i}}$ (obtained from Eq. 6.17), with a velocity gain $\lambda_c > 0$. Note that in this work, the collision constraints are always applied between two strictly convex objects (two points, two spheres, or one point and a plane). Thereby, the closest points between the objects always move continuously, resulting in continuous robot velocity when applying the constraint. For non-convex objects however, the robot velocity might change discontinuously since the closest points move discontinuously. To deal with that issue, one can follow the method proposed in Kanehiro et al. (2008), which decomposes the interaction between non-strictly convex polyhedra into a set of interactions between polygonal faces which are assumed to be decomposed into triangles.

The first of these four constraints (Eq. 6.7) concerns end-effector collision avoidance c_E. Because of the proximity of the robotic arms in the experimental setup, collision between the end-effectors might occur. To avoid them, the collision points $\boldsymbol{x}^{i}_{\boldsymbol{c_E}}$ are virtually positioned on spheres $S^{i}_{c_E}$ of radius $r^{i}_{c_E} \in \mathbb{R}^{+}$ centered at the robots end-effectors (to take into account the size and shape) such that:

$$\boldsymbol{x}^{i}_{\boldsymbol{c_E}} = \boldsymbol{x}^{i}_{\boldsymbol{E}} - r^{i}_{c_E}\boldsymbol{n}^{i}_{\boldsymbol{c_E}} \tag{6.18}$$

where $\boldsymbol{n}^{i}_{\boldsymbol{c_E}}$ is defined based on the end-effectors position:

$$\boldsymbol{n}^{i}_{\boldsymbol{c_E}} = \frac{\boldsymbol{x}^{i}_{\boldsymbol{E}} - \boldsymbol{x}^{j}_{\boldsymbol{E}}}{\|\boldsymbol{x}^{i}_{\boldsymbol{E}} - \boldsymbol{x}^{j}_{\boldsymbol{E}}\|} \quad \text{with } j = \begin{cases} G & \text{if } i = C \\ C & \text{otherwise} \end{cases} \tag{6.19}$$

The collision distance $d^{i}_{c_E}$ can be then simply computed as follows:

$$d^{i}_{c_E} = (\boldsymbol{x}^{i}_{\boldsymbol{c_E}} - \boldsymbol{x}^{j}_{\boldsymbol{c_E}})^{\boldsymbol{T}}\boldsymbol{n}^{i}_{\boldsymbol{c_E}} \tag{6.20}$$

with j defined as in Eq. 6.19.

The second collision constraint (Eq. 6.8) deals with tool collision avoidance c_T. Such collision may occur between any part of the tools as illustrated in Figure 6.4, where the closest point on the right tool to the left one is not at the tip. The computation of the tool collision point $\boldsymbol{x}^{i}_{\boldsymbol{c_T}}$ is done in two steps. The first one consists to find the closest collision point along the tool $\boldsymbol{x}^{i}_{\boldsymbol{c_T}'} \in \mathbb{R}^{3}$ such that:

$$\boldsymbol{x}^{i}_{\boldsymbol{c_T}'} = \boldsymbol{x}^{i}_{\boldsymbol{E}} + l^{i}_{c_T}\boldsymbol{z}^{i}_{\boldsymbol{o}} \tag{6.21}$$

where $l^{i}_{c_T} \in [0, L^{i}]$ is the collision offset along the tool with $L^{i} > 0$ the tool length. The computation of $l^{i}_{c_T}$ is obtained from Algorithm 1 derived from the work in Allen et al. (1993) showing how to find the closest distance between two finite line segments. Once $\boldsymbol{x}^{i}_{\boldsymbol{c_T}'}$ is known, the collision direction $\boldsymbol{n}^{i}_{\boldsymbol{c_T}}$ can be computed similarly to Eq. 6.19:

$$\boldsymbol{n}^{i}_{\boldsymbol{c_T}} = \frac{\boldsymbol{x}^{i}_{\boldsymbol{c_T}'} - \boldsymbol{x}^{j}_{\boldsymbol{c_T}'}}{\|\boldsymbol{x}^{i}_{\boldsymbol{c_T}'} - \boldsymbol{x}^{j}_{\boldsymbol{c_T}'}\|} \quad \text{with } j = \begin{cases} G & \text{if } i = C \\ C & \text{otherwise} \end{cases} \tag{6.22}$$

The tool collision point $\boldsymbol{x}^{i}_{\boldsymbol{c_T}}$ is then calculated such that:

$$\boldsymbol{x}^{i}_{\boldsymbol{c_T}} = \boldsymbol{x}^{i}_{\boldsymbol{c_T}'} - r^{i}_{c_T}\boldsymbol{n}^{i}_{\boldsymbol{c_T}} \tag{6.23}$$

where $r^{i}_{c_T} \in \mathbb{R}^{+}$ is the radius of the collision sphere $S^{i}_{c_T}$ attached to $\boldsymbol{x}^{i}_{\boldsymbol{c_T}'}$ to account for the tool dimensions. Regarding the tool collision distance $d^{i}_{c_T}$, it is computed as:

$$d^{i}_{c_T} = (\boldsymbol{x}^{i}_{\boldsymbol{c_T}'} - \boldsymbol{x}^{j}_{\boldsymbol{c_T}'})^{\boldsymbol{T}}\boldsymbol{n}^{i}_{\boldsymbol{c_T}} \tag{6.24}$$

with j defined as in Eq.6.19.

Algorithm 1: Computation of $l^i_{c_T}$ and $l^j_{c_T}$

```
// Compute the distance vector $\boldsymbol{r^i_E}$ between the robotic arm end effectors:
$\boldsymbol{r^i_E} = \boldsymbol{x^i_E} - \boldsymbol{x^j_E}$ with j as defined in Eq. 6.19

// Express the shortest distance vector $\boldsymbol{s^i}$ between the collision points on the tools as
   function of $l^i_{c_T}$ and $l^j_{c_T}$:
$\boldsymbol{s^i} = \boldsymbol{x^i_{c'_T}} - \boldsymbol{x^j_{c'_T}} = \boldsymbol{r^i_E} + l^i_{c_T}\boldsymbol{z}^i_{\boldsymbol{o}} - l^j_{c_T}\boldsymbol{z}^j_{\boldsymbol{o}}$

// $\boldsymbol{s^i}$ should satisfy (see Allen et al. (1993)):
$\begin{cases} \boldsymbol{s^i}^{\boldsymbol{T}}\boldsymbol{z^i_o} = 0 \\ \boldsymbol{s^i}^{\boldsymbol{T}}\boldsymbol{z^j_o} = 0 \end{cases}$

// Solving for $l^i_{c_T}$ and $l^j_{c_T}$ leads to:
$\begin{cases} l^i_{c_T} = \dfrac{-\boldsymbol{r^i_E}^{\boldsymbol{T}}\boldsymbol{z^i_o} - (\boldsymbol{z^i_o}^{\boldsymbol{T}}\boldsymbol{z^j_o})(\boldsymbol{r^i_E}^{\boldsymbol{T}}\boldsymbol{z^j_o})}{1 - (\boldsymbol{z^i_o}^{\boldsymbol{T}}\boldsymbol{z^j_o})^2} \\ l^j_{c_T} = \dfrac{\boldsymbol{r^i_E}^{\boldsymbol{T}}\boldsymbol{z^j_o} - (\boldsymbol{z^i_o}^{\boldsymbol{T}}\boldsymbol{z^j_o})(\boldsymbol{r^i_E}^{\boldsymbol{T}}\boldsymbol{z^i_o})}{1 - (\boldsymbol{z^i_o}^{\boldsymbol{T}}\boldsymbol{z^j_o})^2} \end{cases}$

// Ensure that $l^i_{c_T} \in [0, L^i]$ and $l^j_{c_T} \in [0, L^j]$:
if $l^i_{c_T} > L^i$ then
    $l^i_{c_T} = L^i$;
    $l^j_{c_T} = \boldsymbol{r^i_E}^{\boldsymbol{T}}\boldsymbol{z^j_o} + l^i_{c_T}\boldsymbol{z^i_o}^{\boldsymbol{T}}\boldsymbol{z^j_o}$;
end

if $l^j_{c_T} > L^j$ then
    $l^j_{c_T} = L^j$;
    $l^i_{c_T} = -\boldsymbol{r^i_E}^{\boldsymbol{T}}\boldsymbol{z^i_o} + l^j_{c_T}\boldsymbol{z^i_o}^{\boldsymbol{T}}\boldsymbol{z^j_o}$;
end

$l^i_{c_T} = \max\{0, \min\{l^i_{c_T}, L^i\}\}$
$l^j_{c_T} = \max\{0, \min\{l^j_{c_T}, L^j\}\}$
```

The next collision constraints (Eq. 6.9) concern workspace collision avoidance c_W^k. They are used to limit the tool tip motion inside the trocar space. In this work, the boundaries are represented by five collision planes constraining the tool tip motion along the -$\boldsymbol{x_W}$, +$\boldsymbol{x_W}$, -$\boldsymbol{y_W}$, +$\boldsymbol{y_W}$, and -$\boldsymbol{z_W}$ world frame directions. The collision plane are defined by specifying offsets from the insertion point $\boldsymbol{x^i_{d,0}}$ (see section 6.3.4.3 for the definition of $\boldsymbol{x^i_{d,0}}$ and Appendix E.2 for more details about the offsets). From there, the distance collision $d^i_{c_W^k}$ to each collision plane can be easily derived.

The minimum insertion constraint c_{MI} (Eq. 6.10) ensures that the tool does not come out from the trocar space after insertion. It acts as a collision constraint between the tool tip $\boldsymbol{x^i}$ and the RCM position $\boldsymbol{x^i_{rcm}}$ where the collision direction is specified by $\boldsymbol{z^i_o}$ while the collision distance $d_{c_{MI}}$ is in fact the insertion distance from the RCM point such that:

$$d_{c_{MI}} = (\boldsymbol{x^i} - \boldsymbol{x^i_{rcm}})^{\boldsymbol{T}} \boldsymbol{z^i_o} \tag{6.25}$$

Finally, the last three inequations (Eq. 6.11, 6.12 and 6.13) of the inverse kinematics problem impose limits on the optimization variables.

All the constraints presented above are linear expressions of the optimization variables. Given that the objective function is composed of quadratic terms, the inverse kinematics problem above can be formulated as a Quadratic Programming (QP) problem of the form:

$$\min_{\boldsymbol{X_{qp}}} \frac{1}{2} \boldsymbol{X_{qp}}^{\boldsymbol{T}} \boldsymbol{H_{qp}} \boldsymbol{X_{qp}} + \boldsymbol{g_{qp}}^{\boldsymbol{T}} \boldsymbol{X_{qp}} \tag{6.26}$$

subject to:

$$\begin{aligned} \boldsymbol{lbA_{qp}} \leq & \boldsymbol{A_{qp}} \boldsymbol{X_{qp}} \leq \boldsymbol{ubA_{qp}} \\ \boldsymbol{lb_{qp}} \leq & \boldsymbol{X_{qp}} \leq \boldsymbol{ub_{qp}} \end{aligned} \tag{6.27}$$

where $\boldsymbol{X_{qp}} \in \mathbb{R}^{N+7}$ is the optimization state:

$$\boldsymbol{X_{qp}} = \begin{pmatrix} \boldsymbol{\dot{q}^i_d} \\ \boldsymbol{\delta^i_{rcm}} \\ \boldsymbol{\delta^i_{task}} \end{pmatrix} \tag{6.28}$$

$\boldsymbol{H_{qp}} \in \mathbb{R}^{(N+7)\times(N+7)}$ is the quadratic cost hessian matrix:

$$\boldsymbol{H_{qp}} = \begin{pmatrix} \boldsymbol{H_1} + dt^2 \boldsymbol{H_4} & \boldsymbol{0_{N\times 3}} & \boldsymbol{0_{N\times 4}} \\ \boldsymbol{0_{3\times N}} & \boldsymbol{H_2} & \boldsymbol{0_{3\times 4}} \\ \boldsymbol{0_{4\times N}} & \boldsymbol{0_{4\times 3}} & \boldsymbol{H_3} \end{pmatrix} \tag{6.29}$$

$\boldsymbol{g_{qp}} \in \mathbb{R}^{N+7}$ is the gradient vector:

$$\boldsymbol{g_{qp}} = \begin{pmatrix} -dt \boldsymbol{H_4} (\boldsymbol{q^i_c} - \boldsymbol{q^i_0}) \\ \boldsymbol{0_{7\times 1}} \end{pmatrix} \tag{6.30}$$

$\boldsymbol{A_{qp}} \in \mathbb{R}^{(N+22)\times(N+7)}$ is the constraint matrix:

$$\boldsymbol{A_{qp}} = \begin{pmatrix} \boldsymbol{J^i_{x^i_{rcm}}} & \text{-}\boldsymbol{I}_{\boldsymbol{3\times3}} & \boldsymbol{0}_{\boldsymbol{3\times4}} \\ \begin{pmatrix} \boldsymbol{J^i_{x^i}} \\ [\boldsymbol{0}_{\boldsymbol{1\times(N-1)}}\ 1] \end{pmatrix} & \boldsymbol{0}_{\boldsymbol{4\times3}} & \text{-}\boldsymbol{I}_{\boldsymbol{4\times4}} \\ dt\boldsymbol{I}_{\boldsymbol{N\times N}} & \boldsymbol{0}_{\boldsymbol{N\times3}} & \boldsymbol{0}_{\boldsymbol{N\times4}} \\ \boldsymbol{J^i(q^i)} & \boldsymbol{0}_{\boldsymbol{6\times3}} & \boldsymbol{0}_{\boldsymbol{6\times4}} \\ \boldsymbol{n^i_{c_E}}^{\boldsymbol{T}} \boldsymbol{J^i_{x^i_{c_E}}} & \boldsymbol{0}_{\boldsymbol{1\times3}} & \boldsymbol{0}_{\boldsymbol{1\times4}} \\ \boldsymbol{n^i_{c_T}}^{\boldsymbol{T}} \boldsymbol{J^i_{x^i_{c_T}}} & \boldsymbol{0}_{\boldsymbol{1\times3}} & \boldsymbol{0}_{\boldsymbol{1\times4}} \\ \boldsymbol{n^i_{c_{W^1}}}^{\boldsymbol{T}} \boldsymbol{J^i_{x^i}} & \boldsymbol{0}_{\boldsymbol{1\times3}} & \boldsymbol{0}_{\boldsymbol{1\times4}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{n^i_{c_{W^5}}}^{\boldsymbol{T}} \boldsymbol{J^i_{x^i}} & \boldsymbol{0}_{\boldsymbol{1\times3}} & \boldsymbol{0}_{\boldsymbol{1\times4}} \\ \boldsymbol{z^i_o}^{\boldsymbol{T}} \boldsymbol{J^i_{x^i}} & \boldsymbol{0}_{\boldsymbol{1\times3}} & \boldsymbol{0}_{\boldsymbol{1\times4}} \end{pmatrix} \tag{6.31}$$

$\boldsymbol{lbA_{qp}} \in \mathbb{R}^{N+22}$ and $\boldsymbol{ubA_{qp}} \in \mathbb{R}^{N+22}$ are the lower and upper bounds on the constraints:

$$\boldsymbol{lbA_{qp}} = \begin{pmatrix} a_{rcm}(\boldsymbol{x_{tr}} - \boldsymbol{x_{rcm}}) \\ \begin{pmatrix} a_{task,1}\boldsymbol{\dot{x}^i_d} \\ a_{task,2}\omega^i_d \end{pmatrix} \\ \boldsymbol{q^i_-} - \boldsymbol{q^i_0} \\ \begin{pmatrix} \boldsymbol{\dot{x}_{E,-}} \\ \boldsymbol{\omega_{E,-}} \end{pmatrix} \\ \text{-}\lambda_{c_E}\dfrac{d^i_{c_E} - d_{c_E,s}}{d_{c_E,inf} - d_{c_E,s}} \\ \text{-}\lambda_{c_T}\dfrac{d^i_{c_T} - d_{c_T,s}}{d_{c_T,inf} - d_{c_T,s}} \\ \text{-}\lambda_{c^1_W}\dfrac{d^i_{c^1_W} - d_{c^1_W,s}}{d_{c^1_W,inf} - d_{c^1_W,s}} \\ \vdots \\ \text{-}\lambda_{c^5_W}\dfrac{d^i_{c^5_W} - d_{c^5_W,s}}{d_{c^5_W,inf} - d_{c^5_W,s}} \\ \text{-}\lambda_{c_{MI}}\dfrac{d^i_{c_{MI}} - d_{c_{MI},s}}{d_{c_{MI},inf} - d_{c_{MI},s}} \end{pmatrix} \qquad \boldsymbol{ubA_{qp}} = \begin{pmatrix} a_{rcm}(\boldsymbol{x_{tr}} - \boldsymbol{x_{rcm}}) \\ \begin{pmatrix} a_{task,1}\boldsymbol{\dot{x}^i_d} \\ a_{task,2}\omega^i_d \end{pmatrix} \\ \boldsymbol{q^i_+} - \boldsymbol{q^i_0} \\ \begin{pmatrix} \boldsymbol{\dot{x}_{E,+}} \\ \boldsymbol{\omega_{E,+}} \end{pmatrix} \\ c_+ \\ c_+ \\ c_+ \\ \vdots \\ c_+ \end{pmatrix} \tag{6.32}$$

with $c_+ > 0$ an upper bound for the collision constraints, while $\boldsymbol{lb_{qp}} \in \mathbb{R}^{N+7}$ and $\boldsymbol{ub_{qp}} \in \mathbb{R}^{N+7}$

are the lower and upper bounds on the optimization state:

$$\boldsymbol{lb_{qp}} = \begin{pmatrix} \dot{\boldsymbol{q}}^{\boldsymbol{i}}_{-} \\ \boldsymbol{\delta_{rcm,-}} \\ \boldsymbol{\delta_{task,-}} \end{pmatrix} \qquad \boldsymbol{ub_{qp}} = \begin{pmatrix} \dot{\boldsymbol{q}}^{\boldsymbol{i}}_{+} \\ \boldsymbol{\delta_{rcm,+}} \\ \boldsymbol{\delta_{task,+}} \end{pmatrix} \tag{6.33}$$

In practice, we solve this QP in real-time thanks to the open source library qpOASES (Ferreau et al. (2014)).

6.3.4 Task Planning

6.3.4.1 Desired Tools' Tip Dynamics

In this section, we describe the computation of the desired tool tip dynamics $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{d}}$ and ω^i_d. $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{d}}$ is composed of a desired task velocity $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,d}} \in \mathbb{R}^3$ with a velocity generated from an admittance controller $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,d}} \in \mathbb{R}^3$ to allow hand-guiding motion through physical human interaction, such that:

$$\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{d}} = (1-\alpha^i)\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,d}} + \dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,d}} \tag{6.34}$$

$\alpha^i \in [0,1]$ is a modulation gain giving priority to $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,d}}$ when persistent physical human interactions are sensed (see section 6.3.4.8). To account for the fulcrum effect, the computation of $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,d}}$ and $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,d}}$ follows a two-step process where reference velocities $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,ref}}, \dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,ref}} \in \mathbb{R}^3$ are first computed before being modified to produce $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,d}}$ and $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,d}}$ (see section 6.3.4.7). The calculation of $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,ref}}$ is explained in section 6.3.4.6. The computation of $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,ref}}$ and ω^i_d uses the human foot input (i.e., $\boldsymbol{\xi^m}$, see Appendix C) and changes depending on the control phase (see section 6.3.4.2) and the type of tool, whether it is the camera (see section 6.3.4.3 and 6.3.4.4) or the gripper (see section 6.3.4.5).

6.3.4.2 Description of the Control Phases of the Robotic Tools

Let us assume that, before starting the task, the tip of the tools are positioned at the desired trocar locations by the human. This can be easily done through physical interaction by putting the robots in full compliance mode (i.e, in gravity compensation mode). Once the trocars' position is registered, the task can start with both robots going through two control phases:

- An *insertion* phase where the robots perform an automatic insertion inside the phantom.
- An *operation* phase where the human starts controlling the tools with the feet.

This decomposition of the task into two phases is mainly used as safety to provide a stable insertion close to the trocar position until reaching a safe distance where the human starts controlling the tools. Moreover, we ensure that once the tools reach the *operation* phase, reverse movements are prevented, i.e., the robotic arms cannot retract the tools outside of the trocar. In the following sections, we describe the derivation of $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,ref}}$ and ω^i_d in both phases for each tool.

6.3.4.3 Reference Camera Motion Dynamics

During *insertion*, the camera is controlled such that it automatically reaches an insertion point $\boldsymbol{x}^{\boldsymbol{C}}_{\boldsymbol{d,0}} \in \mathbb{R}^3$ defined by specifying a 3D offset $\boldsymbol{\delta}^{\boldsymbol{C}}_{\boldsymbol{d,0}} \in \mathbb{R}^3$ from the trocar position such that:

$$\boldsymbol{x}^{\boldsymbol{C}}_{\boldsymbol{d,0}} = \boldsymbol{x}^{\boldsymbol{C}}_{\boldsymbol{tr}} + \boldsymbol{\delta}^{\boldsymbol{C}}_{\boldsymbol{d,0}} \tag{6.35}$$

To this end, $\dot{\boldsymbol{x}}^{\boldsymbol{C}}_{\boldsymbol{t,ref}}$ is computed as a linear DS with attractor $\boldsymbol{x}^{\boldsymbol{C}}_{\boldsymbol{d,0}}$, while ω^C_d is set to zero to prevent any tool self-rotation, resulting into:

$$\begin{cases} \dot{\boldsymbol{x}}^{\boldsymbol{C}}_{\boldsymbol{t,ref}} = a_0(\boldsymbol{x}^{\boldsymbol{C}}_{\boldsymbol{d,0}} - \boldsymbol{x}^{\boldsymbol{C}}) \\ \omega^C_d = 0 \end{cases} \tag{6.36}$$

where $\boldsymbol{x}^{\boldsymbol{C}}$ is the camera tip position, while $a_0 \in \mathbb{R}^+$ is a gain. Once $\boldsymbol{x}^{\boldsymbol{C}}$ reaches $\boldsymbol{x}^{\boldsymbol{C}}_{\boldsymbol{d,0}}$ (within a distance threshold), the camera moves to the *operation* phase.

In the *operation* phase, without assistance, the camera is fully controlled by the human. The foot is used as a joystick to move the tool tip in the camera frame $C = \{\boldsymbol{x_C}, \boldsymbol{y_C}, \boldsymbol{z_C}\}$, where $\boldsymbol{x_C}, \boldsymbol{y_C}, \boldsymbol{z_C} \in \mathbb{R}^3$ are the camera axis respectively aligned with the down/up, left/right, and zoom-out/zoom-in camera directions, expressed in the world frame. To achieve that behaviour, a subset of the foot interface DoFs are first mapped to a desired reference velocity ${}^{\boldsymbol{C}}\dot{\boldsymbol{x}}_{\boldsymbol{ref}} \in \mathbb{R}^3$ in the camera frame through:

$$\begin{pmatrix} {}^{\boldsymbol{C}}\dot{\boldsymbol{x}}_{\boldsymbol{ref}} \\ \omega^C_d \\ \sigma_a \end{pmatrix} = \boldsymbol{A_C}\, {}^{\boldsymbol{C}}\boldsymbol{\xi}^{\boldsymbol{C}} \quad \text{with: } {}^{\boldsymbol{C}}\boldsymbol{\xi}^{\boldsymbol{C}} = \boldsymbol{\Upsilon_C}\boldsymbol{\xi}^{\boldsymbol{C}}_{\boldsymbol{dz}} \tag{6.37}$$

where $\boldsymbol{\xi}^{\boldsymbol{C}}_{\boldsymbol{dz}} \in \mathbb{R}^5$ is the foot pose measured in the platform joint space with dead-zones specified by the lower and upper limits $\boldsymbol{\xi}^{\boldsymbol{C}}_{\boldsymbol{dz,-}}, \boldsymbol{\xi}^{\boldsymbol{C}}_{\boldsymbol{dz,+}} \in \mathbb{R}^5$. $\boldsymbol{\Upsilon_C} \in \mathbb{R}^{5\times5}$ is a tele-functioning matrix mapping $\boldsymbol{\xi}^{\boldsymbol{C}}_{\boldsymbol{dz}}$ to a scaled input ${}^{\boldsymbol{C}}\boldsymbol{\xi}^{\boldsymbol{C}} \in \mathbb{R}^5$ in the desired degrees of freedom of the camera's task space, such that each entry of ${}^{\boldsymbol{C}}\boldsymbol{\xi}^{\boldsymbol{C}}$ belongs to [-1, 1], while $\boldsymbol{A_C} \in \mathbb{R}^{5\times5}$ is a positive gain diagonal matrix. Note that all the degrees of freedom of the foot interface are exploited, the fifth one being used to activate the camera assistance mode through the variable $\sigma_a \in [-1, 1]$ (see section 6.3.4.4 for more details). Once ${}^{\boldsymbol{C}}\dot{\boldsymbol{x}}_{\boldsymbol{ref}}$ is computed, $\dot{\boldsymbol{x}}^{\boldsymbol{C}}_{\boldsymbol{t,ref}}$ can be simply derived by converting ${}^{\boldsymbol{C}}\dot{\boldsymbol{x}}_{\boldsymbol{ref}}$ into the world frame:

$$\dot{\boldsymbol{x}}^{\boldsymbol{C}}_{\boldsymbol{t,ref}} = \boldsymbol{R}^{\boldsymbol{C}}_{\boldsymbol{o}}\, {}^{\boldsymbol{E}}\boldsymbol{R_C}\, {}^{\boldsymbol{C}}\dot{\boldsymbol{x}}_{\boldsymbol{ref}} \tag{6.38}$$

where ${}^{\boldsymbol{E}}\boldsymbol{R_C} = [{}^{\boldsymbol{E}}\boldsymbol{x_C}\ {}^{\boldsymbol{E}}\boldsymbol{y_C}\ {}^{\boldsymbol{E}}\boldsymbol{z_C}]$, is the rotation matrix from the camera frame C to the end-effector frame E, with ${}^{\boldsymbol{E}}\boldsymbol{x_C}, {}^{\boldsymbol{E}}\boldsymbol{y_C}, {}^{\boldsymbol{E}}\boldsymbol{z_C} \in \mathbb{R}^3$ the coordinates of the camera axis expressed in E.

6.3.4.4 Camera Assistance: Adaptive Autonomous Tool Tip Following

As mentioned in section 6.3.4.3, the human can use one of the foot platform DoF to activate camera assistance through the variable σ_a (see Eq. 6.37) that is compared to a predefined threshold $\sigma_{a,th} \in]\text{-}1,1[\backslash\{0\}$. Each time $\mathrm{sgn}(\sigma_{a,th})\sigma_a > |\sigma_{a,th}|$, the human activates or deactivates the assistance.

The camera assistance only affects the calculation of ${}^{C}\dot{\boldsymbol{x}}_{\boldsymbol{ref}}$, the computation of ω_d^C remaining the same (see Eq. 6.37). It is designed to:

- Allow tool tip following by automatically centering the camera field of view on a specific tool among the three existing ones, namely the two surgical tools held by the human and the gripper held by the other robotic assistant.
- Allow the human to decide which tool to follow through simple foot gestures.

To provide such assistance, a tool tip tracking strategy is needed. In this work, we decided to take advantage of the available camera/endoscope to track color markers attached at the tip of the surgical tools, based on simple image processing algorithms (see Appendix E.1 for more details about the tracking strategy).

Thus, let us define ${}^{C}\boldsymbol{r_t} \in \mathbb{R}^2$ as the relative position of tool tip $t = \{1,..,T\}$ with respect to the image center and expressed in normalized units in the camera image/frame, where T is the number of tools (here equal to 3). Taking inspiration from the work in Khoramshahi and Billard (2019), to align the center of the image on the desired tool t, we compute a desired relative velocity in the camera frame ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{d}} \in \mathbb{R}^2$ as a linear combination of all the ${}^{C}\boldsymbol{r_t}$:

$$
{}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{d}} = \sum_{t=1}^{T} b_t a_C {}^{C}\boldsymbol{r_t} \qquad \text{with: } \sum_{t=1}^{T} b_t = 1 \tag{6.39}
$$

where $a_C > 0$ is a gain, while $b_t \in [0,1]$ is the belief associated to tool t. The formulation in Eq. 6.39 allows a smooth transition to the desired tool to follow by updating the b_t's according to the human input.

Indeed, the human input is used to compute a relative velocity ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{h}} \in \mathbb{R}^2$ in the camera frame to inform the robot about the human intent to follow a specific tool. To this end, ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{h}}$ is computed by extracting the first two components of ${}^{C}\boldsymbol{\xi}^{\boldsymbol{C}}$ (obtained from Eq. 6.37), representing the scaled human input mapped to the $\{\boldsymbol{x_C}, \boldsymbol{y_C}\}$ axis of the camera frame:

$$
{}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{h}} = {}^{C}\boldsymbol{\xi}^{\boldsymbol{C}}_{\boldsymbol{(1:2)}} \tag{6.40}
$$

From there, similarly to Khoramshahi and Billard (2019), we construct a belief adaptation rule based on similarity measures between the human input and the tools' relative position, to ensure a smooth transition (through ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{d}}$) to the tool with the highest similarity, such that:

$$\dot{b}_t = \dot{b}_{t,1} + \dot{b}_{t,2} + \dot{b}_{t,3} \tag{6.41}$$

with:

$$\begin{cases} \dot{b}_{t,1} = a_{b,1}\Lambda^{+}_{r_-,r_+}(\|{}^{C}\boldsymbol{r_t}\|){}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{h}}{}^{T}\dfrac{{}^{C}\boldsymbol{r_t}}{\|{}^{C}\boldsymbol{r_t}\|} \\ \dot{b}_{t,2} = a_{b,2}\exp\big(-\gamma_{b,2}\|{}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{h}}\|\|{}^{C}\boldsymbol{r_t}\|\big) \\ \dot{b}_{t,3} = a_{b,3}(b_t - 0.5) \end{cases} \tag{6.42}$$

where $a_{b,1}, r_-, r_+\sigma_{b,1}, a_{b,2}, \gamma_{b,2}, a_{b,3} \in \mathbb{R}^+$, while the function $\Lambda^{+}_{a,b}(u)$ provides a smooth transition from 0 to 1 as u transits from a to b:

$$\Lambda^{+}_{a,b}(u) = \begin{cases} 0 & u < a \\ \dfrac{1}{2}\Big(1 - \cos\Big(\pi\dfrac{u-a}{b-a}\Big)\Big) & a \le u \le b \\ 1 & u > b \end{cases} \tag{6.43}$$

The first term in Eq. 6.42 (i.e., $\dot{b}_{t,1}$) considers alignment of the human input with the direction vector of the tools, hence the dot product. The more the input is aligned with tool t, the higher will be the corresponding $\dot{b}_{t,1}$. Moreover, the smooth step function is used in the expression of $\dot{b}_{t,1}$ to make it vanishing if tool t gets very close to the image center. It avoids ambiguities related to the normalization of small vectors. While $\dot{b}_{t,1}$ accounts for direction, $\dot{b}_{t,2}$ considers the distance of the tools to the image center. If several tools have the same directions (i.e, their $\dot{b}_{t,1}$ are the same), the closest one to the image center will be favoured thanks to $\dot{b}_{t,2}$. Finally, $\dot{b}_{t,3}$ guarantees that b_t converges to zero or one when the uncertainty about the tool is low (i.e., b_t very close to zero or one). Once all the $\dot{b}_t$ are computed, they go through a Winner Takes All (WTA) process (as in Khoramshahi and Billard (2019)) ensuring that only one tool has a positive belief update while all the others have a negative one.

While ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{d}}$ allows to follow a specific tool in the 2D camera frame, the control of the zoom-in/zoom-out is still provided to the human exactly like section 6.3.4.3. Therefore, we express the 3D desired reference velocity in the camera frame as ${}^{C}\dot{\boldsymbol{x}}_{\boldsymbol{ref}}$:

$${}^{C}\dot{\boldsymbol{x}}_{\boldsymbol{ref}} = \begin{pmatrix} {}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{d}} \\ \boldsymbol{A}_{\boldsymbol{C(3,3)}}\,{}^{C}\boldsymbol{\xi}^{\boldsymbol{C}}_{\boldsymbol{(3)}} \end{pmatrix} \tag{6.44}$$

From there, $\dot{\boldsymbol{x}}^{\boldsymbol{C}}_{\boldsymbol{t,ref}}$ is finally obtained using Eq. 6.38.

6.3.4.5 Reference Gripper Motion Dynamics

Similarly to the camera, during *insertion*, the gripper autonomously reaches an insertion point $\boldsymbol{x}^{\boldsymbol{G}}_{\boldsymbol{d,0}}$ from a specified offset $\boldsymbol{\delta}^{\boldsymbol{G}}_{\boldsymbol{d,0}}$ using Eq. 6.36. This 3D offset allows to position as desired a cuboid workspace restricting the gripper motion during the *operation* phase.

Indeed, during *operation*, the human fully controls the position of the gripper by mapping a subset of the foot interface DoFs to a desired tip position $\boldsymbol{x_d^G} \in \mathbb{R}^3$ inside the cuboid workspace through the following equation:

$$\begin{pmatrix} \boldsymbol{x_d^G} \\ \omega_d^G \\ \sigma_g \end{pmatrix} = \boldsymbol{A_G}\, {}^{G}\boldsymbol{\xi}^{G} + \begin{pmatrix} \boldsymbol{x_{d,0}^G} \\ 0 \\ 0 \end{pmatrix} \text{with: } {}^{G}\boldsymbol{\xi}^{G} = \boldsymbol{\Upsilon_G \xi_{dz}^G} \tag{6.45}$$

$\boldsymbol{\Upsilon_G} \in \mathbb{R}^{5\times5}$ plays the same role than $\boldsymbol{\Upsilon_C}$ in Eq. 6.37 by mapping the foot's pose to a scaled input ${}^{G}\boldsymbol{\xi}^{G} \in \mathbb{R}^5$ in the desired degrees of freedom of the gripper's task space (i.e., in the world frame). Alike Eq. 6.37, ω_d^G is computed through a position-velocity mapping with $\boldsymbol{A_G} \in \mathbb{R}^{5\times5}$ being a positive gain diagonal matrix, while $\sigma_g \in [-1,1]$ is the grasping assistance signal.

To reach $\boldsymbol{x_d^G}$, the desired tool tip velocity $\boldsymbol{\dot{x}_d^G}$ is finally computed as a linear dynamical system:

$$\boldsymbol{\dot{x}_{t,ref}^G} = a_G(\boldsymbol{x_d^G} - \boldsymbol{x^G}) \tag{6.46}$$

where $a_G \in \mathbb{R}^+$ is a gain.

6.3.4.6 Reference Admittance Velocity Computation

In this work, the robotic assistants not only accept commands from the foot (i.e, through tele-operation) but also from the physical interactions with humans thanks to an admittance controller producing the reference admittance velocity $\boldsymbol{\dot{x}_{a,ref}^i}$. Thereby, the robotic tools can be hand-guided and safety increases as the robots become compliant when persistent interactions are measured.

To derive $\boldsymbol{\dot{x}_{a,ref}^i}$, let us first introduce $\boldsymbol{F_{ext}^i} \in \mathbb{R}^3$ as the external forces measured at the level of the robot's i end-effector and expressed in the World frame [1]. For simplicity, a dead-zone with upper limit $F_{ext,dz,+}^i \in \mathbb{R}^+$ is applied to the norm of $\boldsymbol{F_{ext}^i}$ to distinguish physical human interactions from other disturbances. Let us denote $\boldsymbol{F_h^i}$ as the resulting force vector. In order to produce $\boldsymbol{\dot{x}_{a,ref}^i}$, $\boldsymbol{F_h^i}$ should be transformed into a meaningful force at the tool tip level. This is achieved by Algorithm 2 that transforms the direction of $\boldsymbol{F_h^i}$ to a corresponding force admittance direction $\boldsymbol{n_a^i} \in \mathbb{R}^3$ at the tool tip, respecting the RCM constraint. Once $\boldsymbol{n_a^i}$ is calculated, $\boldsymbol{\dot{x}_{a,ref}^i}$ is generated from the admittance control law below:

$$\boldsymbol{M_a^i \ddot{x}_{a,ref}^i} = \text{-}\boldsymbol{D_a^i \dot{x}_{a,ref}^i} + \|\boldsymbol{F_h^i}\|\boldsymbol{n_a^i} \tag{6.47}$$

where $\boldsymbol{M_a^i}$, $\boldsymbol{D_a^i} \in \mathbb{R}^{3\times3}$ denote the positive definite diagonal mass and damping matrices.

---

[1] In practice $\boldsymbol{F_{ext}^i}$ is obtained from the robots' manufacturer.

Algorithm 2: Computation of $\boldsymbol{n_a^i}$

```
// We first use the direction of $\boldsymbol{F_h^i}$ to compute a desired end-effector velocity:
$\dot{\boldsymbol{x}}_{\boldsymbol{E,d}} = \frac{\boldsymbol{F_h^i}}{\|\boldsymbol{F_h^i}\|}$
// From there, we compute the ideal desired end-effector angular velocity $\boldsymbol{\omega_{E,d}}$ satisfying
   the RCM constraint: $\boldsymbol{P}(\boldsymbol{z_o^i})\big(\dot{\boldsymbol{x}}_{\boldsymbol{E,d}} + \boldsymbol{\omega_{E,d}} \times (\boldsymbol{x_{rcm}^i} - \boldsymbol{x_E^i})\big) = \boldsymbol{0}$
// with $\boldsymbol{P}(\boldsymbol{z_o^i}) = (\boldsymbol{I}_{3\times3} - \boldsymbol{z_o^i}\boldsymbol{z_o^i}^{\boldsymbol{T}})$ the orthogonal projector of $\boldsymbol{z_o^i}$. This equation ensures that
   the velocity components of the RCM point orthogonal to $\boldsymbol{z_o^i}$ (the end-effector axis
   aligned with the tool) are zero, i.e., only insertion/retraction motion are allowed at the
   RCM point. Putting the system in matrix form leads to:
$\boldsymbol{P}(\boldsymbol{z_o^i})\dot{\boldsymbol{x}}_{\boldsymbol{E,d}} = \boldsymbol{P}(\boldsymbol{z_o^i})[\boldsymbol{x_{rcm}^i} - \boldsymbol{x_E^i}]_{\times}\boldsymbol{\omega_{E,d}}$
// After solving for $\boldsymbol{\omega_{E,d}}$, $\boldsymbol{n_a^i}$ is simply obtained by normalizing the resulting velocity at
   the tool tip:
$\boldsymbol{n_a^i} = \frac{\dot{\boldsymbol{x}}_{\boldsymbol{E,d}} + \boldsymbol{\omega_{E,d}} \times (\boldsymbol{x^i} - \boldsymbol{x_E^i})}{\|\dot{\boldsymbol{x}}_{\boldsymbol{E,d}} + \boldsymbol{\omega_{E,d}} \times (\boldsymbol{x^i} - \boldsymbol{x_E^i})\|}$
```

6.3.4.7 Fulcrum Effect Consideration

An important phenomena to consider in order to compute $\dot{\boldsymbol{x}}_{\boldsymbol{t,d}}^{\boldsymbol{i}}$ and $\dot{\boldsymbol{x}}_{\boldsymbol{a,d}}^{\boldsymbol{i}}$ is the fulcrum effect. It results in the fact that the tool tip and end-effector motions are reversed with an amplitude ratio for the rotational motions depending on the insertion depth. Indeed, close to the trocar, small tool tip rotational motions require large motions at the end-effector, while far from the trocar the opposite happens. To avoid large rotational end-effector motion close to the trocar and increase safety, the components of the reference velocity $\dot{\boldsymbol{x}}_{\boldsymbol{k,ref}}^{\boldsymbol{i}}$ (with $k = \{t, a\}$) orthogonal to $\boldsymbol{z_o^i}$ (i.e., the ones induced by rotational motions) are scaled down based on the insertion depth $\delta_{x^i} = (\boldsymbol{x^i} - \boldsymbol{x_{tr}^i})^{\boldsymbol{T}}\boldsymbol{z_o^i}$ such that:

$$\dot{\boldsymbol{x}}_{\boldsymbol{k,d}}^{\boldsymbol{i}} = \boldsymbol{A_F^i}\dot{\boldsymbol{x}}_{\boldsymbol{k,ref}}^{\boldsymbol{i}} \tag{6.48}$$

with:

$$A_F^i = \boldsymbol{R_o^i}\begin{pmatrix} \alpha(\delta_{x^i}) & 0 & 0 \\ 0 & \alpha(\delta_{x^i}) & 0 \\ 0 & 0 & 1 \end{pmatrix}\boldsymbol{R_o^i}^{\boldsymbol{T}}\dot{\boldsymbol{x}}_{\boldsymbol{k,ref}}^{\boldsymbol{i}} \tag{6.49}$$

where $\alpha(\delta_{x^i}) \in [0, 1]$ is calculated as:

$$\alpha(\delta_{x^i}) = \min\left\{\max\{0, a_\delta \delta_{x^i}\}, 1\right\} \tag{6.50}$$

with $a_\delta \in \mathbb{R}^+$ a positive gain.

6.3.4.8 Task Modulation Gain Computation

After deriving the expressions of $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,d}}$ and $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,d}}$, we can finally explain the computation of α^i modulating the desired task dynamics introduced in Eq. 6.34. α^i is designed to allow smooth transitions between a stiff behaviour ($\alpha^i = 0$), where the robot follows the desired task dynamics (i.e, $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{t,d}}$), and a compliant behaviour ($\alpha^i = 1$) where the robot follows the desired admittance velocity (i.e., $\dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,d}}$).

To this end, let us introduce the energy tank $s^i \in [0,1]$ governed by the following dynamics:

$$\dot{s}^i = a_{s,1}P_a - (a_{s,2} - \alpha^i)P_d \tag{6.51}$$

where $a_{s,1} > 0$ and $a_{s,2} > 1$ are positive gains, while $P_a = \|\boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{h}}\| \dot{\boldsymbol{x}}^{\boldsymbol{i}}_{\boldsymbol{a,d}}{}^{\boldsymbol{T}} \boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{a}}$ and $P_d > 0$ respectively denote the input power generated from the admittance controller and the dissipative power. P_a is used to fill the tank when persistent interaction forces are measured (i.e., a persistent reference admittance velocity is generated resulting in positive power values) while P_d constantly removes energy from the tank with a rate modulating by $a_{s,2} - \alpha^i$. From there, α^i is determined based on the energy stored in the tank such that:

$$\alpha^i = \Lambda^{+}_{0,s_+}(s^i) \tag{6.52}$$

where $s_+ \in [0,1]$ is the tank value above which the robot becomes fully compliant to physical human interactions (i.e., $\alpha^i = 1$, see Eq. 6.43).

6.3.5 Haptic Cues Design

Event-related haptic cues are provided at different moments during the task execution, essentially to increase the situational awareness of the surgeon regarding safety. The goal is to convey haptic messages related to important no-go situations such as collisions between the robotic tools and the robots' end-effectors. Also, to guide the foot towards a required part of the workspace at the beginning of the task execution. The computation of the reference robot-to-human guidance haptic cues $\boldsymbol{\tau}^{\boldsymbol{m}}_{\boldsymbol{M,ref}}$ to the master foot interface follows a two-phase process:

1. A wrench $\boldsymbol{W}^{\boldsymbol{i}}_{\boldsymbol{d}} \in \mathbb{R}^4$ is computed in the tool tip task space of the robotic assistant i controlled by the foot interface m (see section 6.3.5.1).

2. The desired wrench $\boldsymbol{W}^{\boldsymbol{i}}_{\boldsymbol{d}}$ is converted to haptic guidance torques $\boldsymbol{\tau}^{\boldsymbol{m}}_{\boldsymbol{M,hc}} \in \mathbb{R}^5$ in the platform's joint space (see section 6.3.5.2).

We describe both phases in the following sections.

6.3.5.1 Guidance Wrench Computation

The guidance/feedback wrench is composed by a force $\boldsymbol{F_d^i} \in \mathbb{R}^3$ and a torque $\tau_d^i \in \mathbb{R}$ such that:

$$\boldsymbol{W_d^i} = \begin{pmatrix} \boldsymbol{F_d^i} \\ \tau_d^i \end{pmatrix} \tag{6.53}$$

$\boldsymbol{F_d^i}$ is made up of different guidance forces. A force is firstly sent by the robotic assistant holding the gripper at the beginning of the *operation* phase to align the foot pose with the center of the platform. Indeed, during *operation*, a position-position mapping is used to compute the desired gripper tip position from the foot pose (see section 6.3.4.5). To prevent fast robot motion when transitioning from the *insertion* to the *operation* phase, $\boldsymbol{x_d^G}$ in Eq. 6.45 should be close to $\boldsymbol{x_{d,0}^i}$, i.e., the foot should be centered on the platform. For that purpose, at the beginning of the *operation* phase, a spring-effect is used to reduce the error between $\boldsymbol{x_d^G}$ and $\boldsymbol{x_{d,0}^G}$:

$$\boldsymbol{F_{d,0}^i} = \begin{cases} a_{F_{d,0}}(\boldsymbol{x_{d,0}^G} - \boldsymbol{x_d^G}) & \text{if } i = G \\ \boldsymbol{0_{3\times 1}} & \text{otherwise} \end{cases} \tag{6.54}$$

where $\boldsymbol{F_{d,0}^i} \in \mathbb{R}^3$ is the guidance force and $a_{F_{d,0}} > 0$ is a positive stiffness gain. As long as the error $\|\boldsymbol{x_{d,0}^G} - \boldsymbol{x_d^G}\|$ is higher than a predefined threshold, the virtual stiffness is present while the gripper is disconnected from the platform.

Guidance forces are also used for both robotic assistants to inform the surgeon about collision and prevent them. As described in the inverse kinematics problem (see section 6.3.3), in this work we consider four types of collision: end-effector, tool, workspace collision, and minimum insertion. For each collision $c = \{c_E, c_T, c_W^k, c_{MI}\}$, we provide a force feedback $\boldsymbol{F_{d,c}^i} \in \mathbb{R}^3$ based on the collision distance d_c^i to help the surgeon moving away from the collision. To this end, we express $\boldsymbol{F_{d,c}^i}$ as:

$$\boldsymbol{F_{d,c}^i} = F_T \Lambda_{d_-,d_+}^-(d_c^i) \boldsymbol{R_{F_c}^i} \boldsymbol{n_{F_c}^i} \tag{6.55}$$

where $F_T, d_-, d_+ \in \mathbb{R}^+$ (see Eq. 3.10), while $\boldsymbol{n_{F_c}^i} \in \mathbb{R}^3$ is a unit vector specifying the direction of the guidance force in the world frame. $\boldsymbol{n_{F_c}^i}$ is rotated in the tool tip task space through the matrix $\boldsymbol{R_{F_c}^i}$ defined by:

$$\boldsymbol{R_{F_c}^i} = \begin{cases} (\boldsymbol{R_o^C}\, {}^E\boldsymbol{R_C})^{\boldsymbol{T}} & \text{if } i = C \\ \boldsymbol{I_{3\times 3}} & \text{otherwise} \end{cases} \tag{6.56}$$

Indeed, the force should be expressed in the task space where the foot pose is mapped, i.e., the camera frame C for the foot controlling the camera and the world frame W for the foot controlling the gripper (hence the identity matrix in Eq. 6.56).

The computation of $\boldsymbol{n_{F_c}^i}$ depends on the collision point position $\boldsymbol{x_c^i}$ and the collision direction $\boldsymbol{n_c^i}$. When applied to the foot, the guidance force directly affects the tip motion and indirectly the motion of the collision point. Therefore, we compute $\boldsymbol{n_{F_c}^i}$ to ensure that the collision point $\boldsymbol{x_c^i}$ moves away from the obstacle along the collision direction $\boldsymbol{n_c^i}$. For workspace collision c_W^k

and minimum insertion c_{MI}, $\boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{F_c}}$ is aligned with the collision direction (i.e., $\boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{F}_{\boldsymbol{c}^k_W}} = \boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{c}^k_W}$ and $\boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{F}_{\boldsymbol{c}^k_W}} = \boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{c}^k_W}$) because the collision point is exactly at the tool tip. This is not the case for tool c_T (the collision point lies along the tool) and end-effector c_E collision. For them, we use the method described in Algorithm 3, that transforms a desired motion direction at the collision point (i.e, $\boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{c}}$) into a desired motion direction at the tip (i.e, $\boldsymbol{n}^{\boldsymbol{i}}_{\boldsymbol{F_c}}$), while taking into account the RCM constraint.

Finally, the estimated measured tool tip contact forces $\boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{m}} \in \mathbb{R}^3$ are also fed back to the human.

```
Algorithm 3: Computation of $\boldsymbol{n}^i_{\boldsymbol{F_c}}$ for end-effector $c_E$ and tool collision $c_T$
  // The first step consists to compute the ideal linear $\dot{\boldsymbol{x}}_{\boldsymbol{E,d}}$ and angular $\boldsymbol{\omega}_{\boldsymbol{E,d}}$ end-effector
     velocity such that the collision point $\boldsymbol{x}^i_{\boldsymbol{c}}$ moves along the collision direction $\boldsymbol{n}^i_{\boldsymbol{c}}$. For that
     purpose, $\dot{\boldsymbol{x}}_{\boldsymbol{E,d}}$ and $\boldsymbol{\omega}_{\boldsymbol{E,d}}$ should satisfy the system of equations below:
  $\begin{cases} \boldsymbol{n}^i_{\boldsymbol{c}} = \dot{\boldsymbol{x}}_{\boldsymbol{E,d}} + \boldsymbol{\omega}_{\boldsymbol{E,d}} \times (\boldsymbol{x}^i_{\boldsymbol{c}} - \boldsymbol{x}^i_{\boldsymbol{E}}) \\ \boldsymbol{0} = \boldsymbol{P}(\boldsymbol{z}^i_{\boldsymbol{o}})\big(\dot{\boldsymbol{x}}_{\boldsymbol{E,d}} + \boldsymbol{\omega}_{\boldsymbol{E,d}} \times (\boldsymbol{x}^i_{\boldsymbol{rcm}} - \boldsymbol{x}^i_{\boldsymbol{E}})\big) \end{cases}$
  // with $\boldsymbol{P}(\boldsymbol{z}^i_{\boldsymbol{o}}) = (\boldsymbol{I}_{3\times 3} - \boldsymbol{z}^i_{\boldsymbol{o}} \boldsymbol{z}^{i\,T}_{\boldsymbol{o}})$, the orthogonal projector of $\boldsymbol{z}^i_{\boldsymbol{o}}$. The first equation ensures
     that the collision point moves with a desired velocity $\boldsymbol{n}^i_{\boldsymbol{c}}$. The second equation is used to
     constraint the velocity of the RCM point to allow only insertion/retraction motion at
     that point, i.e., the velocity components orthogonal to $\boldsymbol{z}^i_{\boldsymbol{o}}$ (the end-effector axis aligned
     with the tool) should be set to 0. Putting the system in matrix form leads to:
  $\begin{pmatrix} \boldsymbol{n}^i_{\boldsymbol{c}} \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_{3\times 3} & -[\boldsymbol{x}^i_{\boldsymbol{c}} - \boldsymbol{x}^i_{\boldsymbol{E}}]_\times \\ \boldsymbol{P}(\boldsymbol{z}^i_{\boldsymbol{o}}) & -\boldsymbol{P}(\boldsymbol{z}^i_{\boldsymbol{o}})[\boldsymbol{x}^i_{\boldsymbol{rcm}} - \boldsymbol{x}^i_{\boldsymbol{E}}]_\times \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{E,d}} \\ \boldsymbol{\omega}_{\boldsymbol{E,d}} \end{pmatrix}$
  // After solving for $\dot{\boldsymbol{x}}_{\boldsymbol{E,d}}$ and $\boldsymbol{\omega}_{\boldsymbol{E,d}}$, $\boldsymbol{n}^i_{\boldsymbol{F_c}}$ is simply obtained by normalizing the resulting
     velocity at the tool tip:
  $\boldsymbol{n}^i_{\boldsymbol{F_c}} = \dfrac{\dot{\boldsymbol{x}}_{\boldsymbol{E,d}} + \boldsymbol{\omega}_{\boldsymbol{E,d}} \times (\boldsymbol{x}^i - \boldsymbol{x}^i_{\boldsymbol{E}})}{\|\dot{\boldsymbol{x}}_{\boldsymbol{E,d}} + \boldsymbol{\omega}_{\boldsymbol{E,d}} \times (\boldsymbol{x}^i - \boldsymbol{x}^i_{\boldsymbol{E}})\|}$
```

To this end, a haptic force feedback $\boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d,m}} \in \mathbb{R}^3$ is computed by scaling $\boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{m}}$ through:

$$\boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d,m}} = \boldsymbol{A}_{\boldsymbol{F_{d,m}}} \boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{m}} \tag{6.57}$$

where $\boldsymbol{A}_{\boldsymbol{F_{d,m}}} \in \mathbb{R}^{3\times 3}$ is a positive gain diagonal matrix.

At the end, $\boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d}}$ is obtained by summing up all the guidance force components:

$$\boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d}} = \boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d,0}} + \boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d},\boldsymbol{c_E}} + \boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d},\boldsymbol{c_T}} + \sum_{k=1}^{5} \boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d},\boldsymbol{c_{Wk}}} + \boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d},\boldsymbol{c_{MI}}} + \boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d,m}} \tag{6.58}$$

Note that in practice, all these components are never active at the same time. $\boldsymbol{F}^{\boldsymbol{i}}_{\boldsymbol{d,0}}$ is only active before entering the operation phase, while the other terms get only activated in case of potential collision ($\boldsymbol{F}^{i}_{d,c_E}$, $\boldsymbol{F}^{i}_{d,c_T}$, $\boldsymbol{F}^{i}_{d,c_{Wk}}$, and $\boldsymbol{F}^{i}_{d,c_{MI}}$) or contact with the environment ($\boldsymbol{F}^{i}_{d,m}$). Furthermore, when guidance forces related to collision are fed-back to the human, a visual cue is also displayed on the screen to indicate the type of collision and increase the interpretability of the haptic cues received at the feet.

Regarding τ^{i}_{d}, it is computed to avoid joint limit during tool self-rotation motion such that:

$$\tau^{i}_{d} = \tau_T\big(\Lambda^{-}_{0,\delta_{q(7)}}(\boldsymbol{q}^{i}_{(7)} - \boldsymbol{q}^{i}_{(7),-}) - \Lambda^{-}_{0,\delta_{q(7)}}(\boldsymbol{q}^{i}_{(7),+} - \boldsymbol{q}^{i}_{(7)})\big) \tag{6.59}$$

where $\tau_T, \delta_{q(7)} \in \mathbb{R}^{+}$ (see Eq. 3.10 and 6.43).

6.3.5.2 Haptic Guidance Torques Computation

The haptic guidance torques $\boldsymbol{\tau}^{m}_{M,hc}$ at the foot platform level are obtained by projecting back the task space guidance wrench $\boldsymbol{W}^{i}_{d}$ in the platform joint space. This is achieved through the tele-functionning matrices $\boldsymbol{\Omega}_{C}, \boldsymbol{\Omega}_{G} \in \mathbb{R}^{5\times 4}$ such that:

$$\boldsymbol{\tau}^{m}_{M,hc} = \begin{cases} \boldsymbol{A}_{\tau}\boldsymbol{\Omega}_{C}\boldsymbol{W}^{C}_{d} & \text{if } m = C \\ \boldsymbol{A}_{\tau}\boldsymbol{\Omega}_{G}\boldsymbol{W}^{G}_{d} & \text{otherwise} \end{cases} \tag{6.60}$$

where $\boldsymbol{A}_{\tau} \in \mathbb{R}^{5\times 5}$ is a positive gain diagonal matrix.

These haptic cues require a fast action response by the user. In practice, we adapt them to increase the perception of the feedback by combining vibrotactile and kinesthetic cues, thereby producing the final haptic guidance torques. This part was developed by Jacob Hernandez Sanchez and is therefore not presented in this thesis.

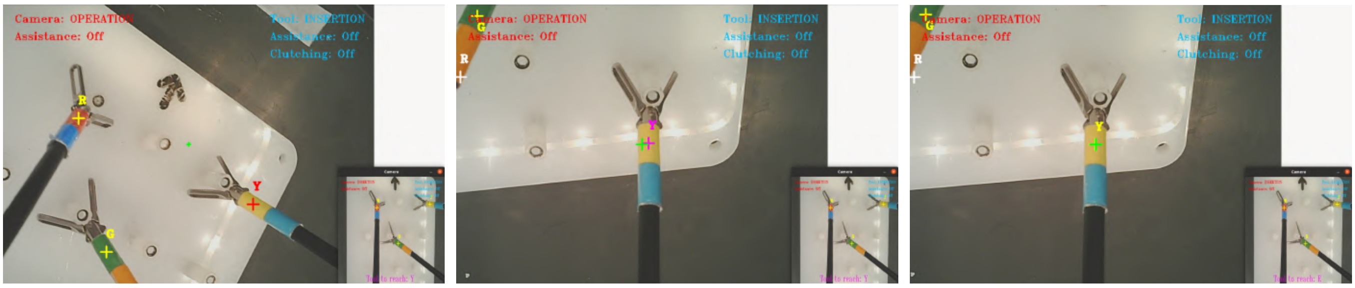
6.4 Experimental Evaluations

In this section, we first evaluate the shared control strategies described earlier in uni-pedal, bi-pedal, and four-handed tasks through a user study presented in section 6.4.1. Our objective is to assess the benefits of the camera and gripper assistance modalities in terms of subjective and objective metrics. We then highlight and validate the different control features developed through a series of small experiments described in section 6.4.2. Technical details of the implementations can be found in Appendix E.2.

6.4.1 User Study

6.4.1.1 Tasks Description

Uni-pedal Speed Control of a Camera ($T1$): The first task is meant to learn how to control the camera in speed/joystick mode. The subject uses the left foot to control the orientation of



(a) (b) (c)

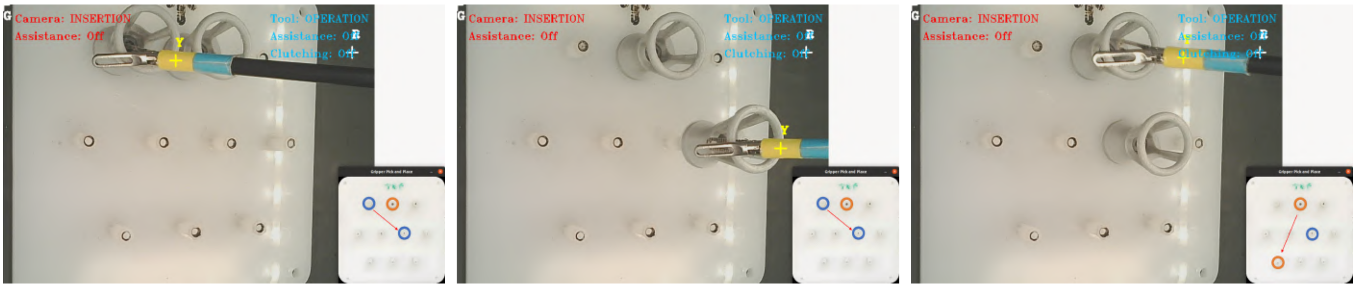
Figure 6.5 – Snapshots of the camera task ($T1$) performed by participant 9. On the bottom right corner of the images, a small figure shows the disposition of the laparoscopic tools and indicates the current target to the subject. The associated cue is colored in red on the real-time image (see Figure 6.5a). Once the 2D position and orientation error become lower than pre-defined thresholds, the color of the target cue becomes purple (see Figure 6.5b). From that moment, the participant has 10 seconds to fine-tune the accuracy. After this time, the target changes to another one (see Figure 6.5c).

the image as well as the 2D position and size (through the insertion depth) of a green visual cue at the center of the image. The task consists to successively reach and match the size of three target cues positioned at the center of color markers attached on laparoscopic tools disposed inside the training phantom. For each target, the participant is also tasked to rotate the camera in order to align the body of the tool with the vertical direction of the image frame (see Figure 6.5).

Uni-pedal Gripper Peg Transfer ($T2$): After learning how to move the camera, the second task consists to learn how to control the robotic gripper. The subject uses the right foot to control the 3D tip position of the gripper, the self-rotation of the tool as well as the opening/-closing of the gripper. The camera tip position is fixed and selected to provide the widest view over the task. The task consists to successively pick and place two small towers one after the other in order to achieve eight peg transfer in total. It is performed twice to evaluate two conditions:

- *Without grasping assistance* $T2\_WO$: the subject directly controls the gripper aperture by mapping the foot roll angle to the desired gripper angle $q_{G,d}$ (see Figure 6.6).
- *With grasping assistance* $T2\_W$: the participant uses the roll angle to trigger autonomous opening/closing of the gripper through a small gesture. When triggered, haptic cues are sent to the foot to keep it in place and avoid unintentional gripper motions.

Bipedal Gripper + Camera Control ($T3$): In the third task, the subject uses the two feet to control both the camera and the robotic gripper at the same time. The task is similar to task



(a) (b) (c)

Figure 6.6 – Snapshots of the gripper task without grasping assistance ($T2\_WO$) performed by participant 3. On the bottom right corner of the images, a small figure indicates which tower to pick and where to place it. In Figure 6.6a the participant moves the gripper to reach and grasp the tower indicated. Once grasped, the subject moves the tower to the target position (see Figure 6.6b). After releasing the tower, the participant moves the gripper to pick and place the other tower (see Figure 6.6c). Note that one pick and place is considered as completed if the subject did not drop the tower during the operation. At any fall, the experiment is paused to replace the tower at its last position and a counter is incremented. The experiment is then re-started and the participant can try the same peg transfer again.

$T2$ where the participant should control the gripper to successively achieve eight peg transfer. In addition, the subject has to control the camera to always keep the tip of the gripper (middle of the clamps) inside a rectangle cue of size 256×192 px centered on the image. Similarly to before, the task is realized twice to evaluate two conditions:

- *Without camera assistance* $T3\_WO$: the participant is in charged to keep the tip of the robotic gripper inside the rectangle cue by controlling all the degrees of freedom of the camera (as in $T1$) except the insertion/retraction which is disabled (see Figure 6.7).
- *With camera assistance* $T3\_W$: the robot autonomously moves the camera to keep the gripper at the center of the image, i.e, inside the rectangle. The subject can still control the orientation of the image but the insertion/retraction is also deactivated.

For both conditions, grasping assistance is provided to control the gripper as in $T2\_W$.

Four Handed Tri-manipulation + Camera ($T4$): In this final task, the participant uses the two feet to control the camera and the robotic gripper and the two hands to manipulate two laparoscopic grippers. The subject is tasked to perform three successive tri-manual peg transfer by moving the three grippers in coordination (as much as possible), while keeping the foot controlled gripper inside the rectangle (as in $T3$). Two conditions are evaluated for this task:

- *Without grasping and camera assistance* $T4\_WO$: the participant directly controls the

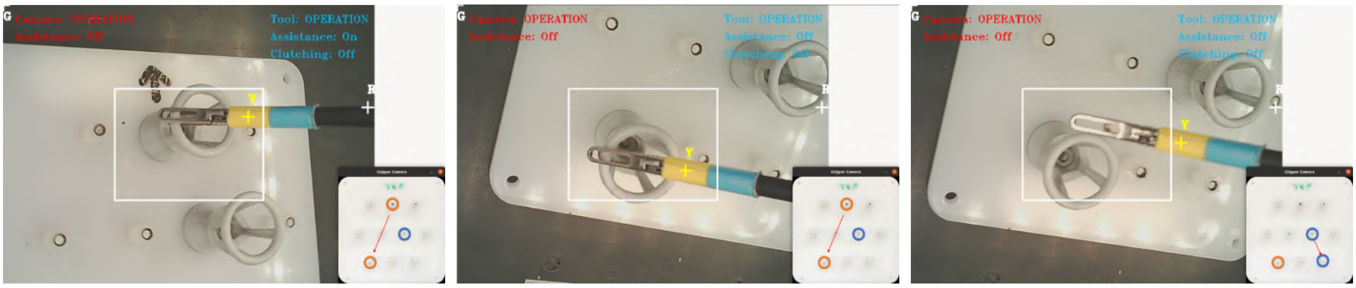

(a) (b) (c)

Figure 6.7 – Snapshots of the camera-gripper task without camera assistance ($T3\_WO$) performed by participant 7. The subject controls both the camera and the gripper to reach and grasp the tower indicated while keeping the tip of tool inside the white rectangle cue (see Figure 6.7a). The participant has to move both the camera and gripper in coordination to place the tower to the target position while satisfying the camera task (see Figure 6.7b). After releasing the tower, the subject finally moves the gripper to perform the next pick and place indicated (see Figure 6.7c).

gripper aperture as in $T2\_WO$ and the camera as in $T3\_WO$.

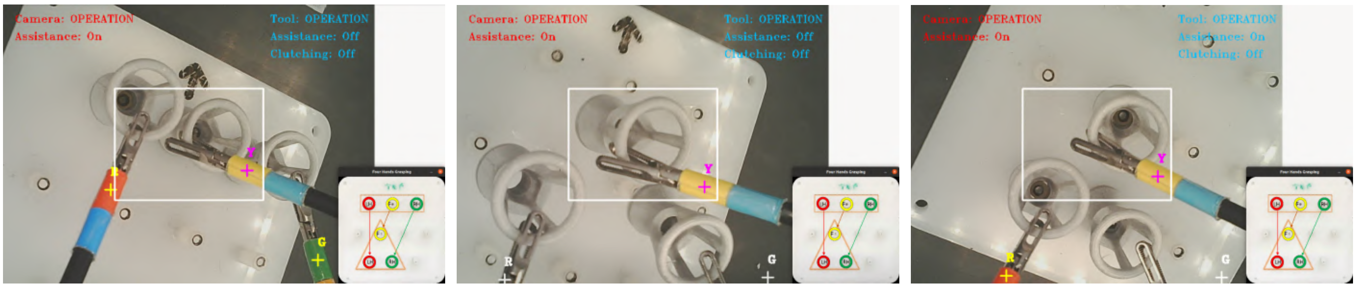
- *With grasping and camera assistance* $T4\_W$: the subject is provided grasping assistance and camera assistance as in $T3\_W$ (see Figure 6.8). Regarding camera assistance, the participants are encouraged to follow the foot-controlled gripper and change the followed tool only if needed (e.g., to have a better camera view during tower placement).

6.4.1.2 Protocol

The user study was conducted with twelve participants (5 females and 7 males) of 27.67 ± 4.40 years old. All subjects were initially naive in the use of foot interfaces to control robotic arms for laparoscopic surgical tasks. Before starting the experiments, the impedance gains of the foot platforms (stiffness and damping gains) were adjusted according to the participant's preference. Each subject was asked to perform the four tasks in the same order starting from the uni-pedal camera task to the four-handed tri-manipulation task (i.e., $T1$ to $T4$). The conditions for each task were however randomized for all participants. For each task condition, the subjects first went through a training session (5 min for $T1$ and $T3$ and 5-10 min for $T2$ and $T4$ depending on the participant's confidence). They were then asked to perform the task within a 5 min time interval. As in section 5.3.1, the experimental protocol was also approved by the Human Research Ethics Committee of EPFL.

6.4.1.3 Evaluation Metrics and Statistical Studies

Similarly to section 5.3.2, to study the effect of the assistance modalities across the different tasks, subjective and objective metrics are defined. The subjective metrics include the fluency



(a) (b) (c)

Figure 6.8 – Snapshots of the anded task with grasping and camera ($T4\_W$) performed by participant 6. The subject manipulates two laparoscopic grippers with the hands (red and green color markers), the robotic gripper with the right foot (yellow color marker), and the camera with the left foot. The camera is autonomously following the tool marked with a purple cue. The participant controls the three grippers to reach and grasp the towers as indicated by the small figure in the bottom right corner (see Figure 6.8a). The subject is allowed to grasp the towers one after the other but constrained to lift and move them in coordination (as much as possible) (see Figure 6.8b). After reaching the target positions, the participant can also place the towers separately (see Figure 6.8c) but is asked to release them at the same time.

and NASA-TLX questionnaires seen previously (see section 5.3.2). Note that in the task-load questionnaire, we dissociate the actions of *grasping* and *positioning*. Here, the questionnaires were filled for each task condition and the answers assessed using a 7 Point Likert Scale from 1 = *strong disagreement* to 7 = strong agreement.

The objective metrics are defined for all tasks except $T1$ and include:

- Completeness c, defined as the ratio between the number of target tower successfully picked and placed $N_t \in [0, N_{max}]$, and N_{max} the maximum number possible (equals to 8 for $T2$ and $T3$, 3 for $T4$).
- The average time spent per target tower $t_{avg} = \frac{\sum_{k=1}^{N_t} t_k}{N_t}$, where t_k is the time needed to pick and place target k.
- The average target efficiency $\sigma_{avg} \in [0, 1]$, defined as:

$$\sigma_{avg} = \frac{\sum_{k=1}^{N_t} \sigma_k}{N_t} \tag{6.61}$$

  with $\sigma_k \in [0, 1]$ the individual target efficiency calculated similarly to the sub-task efficiency in section 5.3.2:

$$\sigma_k = \frac{t_{max} - t_k}{t_{max}} \tag{6.62}$$

  where $t_{max} = 5\ min$ is the maximum time allowed for the task. If a target was not accomplished, t_k is considered to be equal to t_{max} and hence the efficiency (σ_k) is zero.

- The number of object falls N_f that counts the number of times the towers unintentionally fall during the task.
- The visual task success $\beta_{T,C} \in [0,1]$ calculated as the fraction of time the gripper tool tip stays inside the rectangle cue, used only for $T3$ and $T4$.
- The peg-transfer success metric $\beta_{T,G} \in [0,1]$ which is the overall success for the gripper tool tip encompassing the total number of pick and place achieved, the targets' efficiency and penalizing the number of falls:

$$\beta_{T,G} = \frac{\sum_{k}^{N_t} \sigma_k}{N_{max} \exp\left(\frac{N_f}{N_{max}}\right)} \tag{6.63}$$

The evaluation metrics above are used in two statistical studies. Study I analyzes the effect of the robotic assistance, namely gripper and camera assistance, on the metrics responses for tasks $T2$, $T3$, and $T4$ separately. The second study compares the responses between the different task-condition (e.g, $T1$, $T2\_WO$, $T2\_W$ etc.). In the next section we summarize the main results obtained from the user study (see Hernandez Sanchez et al. (2021) for more details).

6.4.1.4 Results

Results from study I can be observed in Appendix E.3 through the Tables E.1, E.2, and E.3. For the uni-pedal peg-transfer task (i.e., $T2$), among all the objective metrics, the use of grasping assistance only allowed a significant decrease in the number of falls (N_f) from an average of 2.75 ± 2.05 to 1.25 ± 1.53. Regarding task load, several differences were obtained. Grasping assistance provoked a significant decrease in physical demand for the action of *grasping* as well as in physical demand, effort and frustration for *positioning*. Furthermore, the participants expressed significantly higher ease of use, fluency, satisfaction and robot intelligence.

In the bi-pedal task (i.e., $T3$), the use of camera assistance significantly reduced the average time spent per target (t_{avg}) from 56.36 ± 15.79 to 45.66 ± 11.55 seconds. It also significantly increased the visual task success ($\beta_{T,C}$) from $82 \pm 11\%$ to $99 \pm 1\%$. By delegating the control of the camera to the autonomous system to follow the gripper tool tip, the participants only used one foot to control the gripper in $T3\_W$. This explains the significant improvements observed in all task load metrics for the action of *positioning*, except performance where the increase is not significant. For *grasping* no significant differences were observed (except for physical demand), as expected, because grasping assistance was used in both conditions (i.e, $T3\_W$ and $T3\_WO$). Moreover, with camera assistance, the subjects perceived a significant increase in all categories of fluency, except trust where the improvement is not significant.

Finally, in the four-arm task (i.e., $T4$), the combined use of grasping and camera assistance significantly improved almost all the objective metrics; i.e., the average target efficiency (σ_{avg}) from 52 ± 31% to 65 ± 18%, completeness (c) from 61.11 ± 40.45% to 75.00 ± 27.64%, number of falls (N_t) from 2.08 ± 1.75 to 1.00 ± 1.41, as well as the visual task success ($\beta_{T,C}$) from 71.52 ± 10.76% to 99.95 ± 0.05%, and peg-transfer success ($\beta_{T,G}$) from 29.86 ± 26.44% to 42.06 ± 27.79%. A similar impact was obtained for task load where the use of both assistance modalities significantly improved mental demand, effort, frustration, and performance for both the *positioning* and *grasping* actions, plus physical demand for *grasping*. Moreover, it led to a significant increase in all aspects of fluency, except trust.

The role of the assistance modalities can be further studied by comparing the responses across the different task conditions. This is performed by Study II whose results can be observed in Appendix E.3 through Figures E.1, E.2, and E.3. The tasks were presented to the participants with an increasing level of complexity, where the complexity here relates to the number of limbs to control (1 for $T1$ and $T2$, 2 for $T3$, and 4 for $T4$). Study II first confirmed the increase in mental load when increasing the complexity of the task, where $T3\_WO$ and $T4\_WO$ were highly more mentally demanding than $T1$ for the action of *positioning*. However, the drop in mental demand is mitigated by the use of robotic assistance as shown by the absence of significant differences between $T1$ and $T3\_W$, $T4\_W$. The same phenomena can be observed when assessing fluency. For instance, $T2\_W$ was perceived as significantly easier to use than $T3\_WO$ and $T4\_WO$ while no significant differences were obtained between $T2\_W$ and $T3\_W$, $T4\_W$. Furthermore, study II showed that the participants perceived significantly higher robot intelligence and fluency of the human-robot collaboration in the average of tasks with robotic assistance (i.e., $T2\_W$, $T3\_W$, and $T4\_W$) with respect to the average of tasks without it (i.e., $T2\_WO$, $T3\_WO$, and $T4\_WO$).

6.4.2 Control Framework Validation

6.4.2.1 Trocar and Tool Tip Task Constraints Achievement

The robotic assistants should achieve the desired tool tip task while ensuring that the tools always move through the trocar position (i.e., achieving the RCM constraint). These two tasks are implemented as constraints in the QP-based inverse kinematics proposed in section 6.3.3 (see Eq. 6.3 and 6.4). To evaluate the accuracy of the robot control to keep these two constraints, we used the data recorded during the user study. The robotic assistants were particularly accurate into keeping the trocar position. Over all the experiments conducted, the camera and gripper assistants respectively had an average trocar error of $6.22 \times 10^{-4} \pm 1.23 \times 10^{-4}$ and $1.52 \times 10^{-3} \pm 8.92 \times 10^{-4}$ m. Similarly, the gripper assistant could precisely follow the desired position commanded through the foot (i.e., $\boldsymbol{x}_d^G$), with an average tool tip position error (i.e., $\|\boldsymbol{x}_d^G - \boldsymbol{x}^G\|$) of $3.81 \times 10^{-3} \pm 1.03 \times 10^{-3}$ m over all the experiments. Regarding the camera assistance modality, it could successfully keep the tool tip of the gripper at the center of the image with an average error of 16.78 ± 5.29 px over all the experiments where the assistance was active (i.e., in $T3\_W$ and $T4\_W$). The above overall performance measures are illustrated

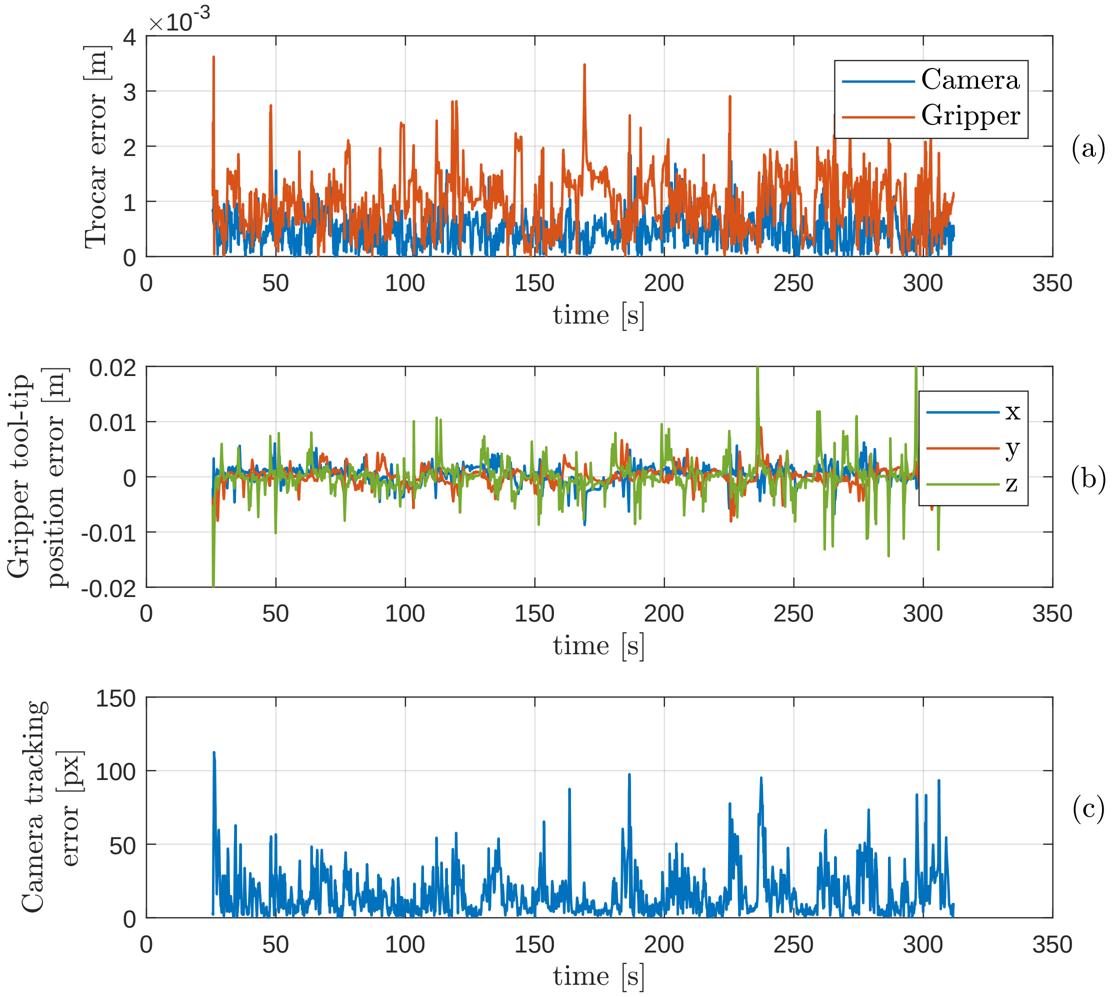


Figure 6.9 – Measured trocar error (a) gripper tool tip position error (b) and camera tracking error (c) signals obtained in the bi-pedal task with assistance (i.e., $T3\_W$) performed by participant 2.

in Figure 6.9 showing the measured trocar error, gripper tool tip position error, and camera tracking error signals recorded during one of the experiments.

6.4.2.2 Compliance to Physical Human Interaction

In this section, an experiment is conducted to evaluate the ability of the robotic assistants to comply to physical human interaction. Snapshots of the experiment are provided in Figure 6.10 while Figure 6.11 shows the norm of the estimated human forces $\|\boldsymbol{F}_{\boldsymbol{h}}^{\boldsymbol{i}}\|$ (Figure 6.11a) as well as the energy tanks s^i (Figure 6.11b) and modulation gains α^i (Figure 6.11c) signals.

After inserting the camera with the left foot, the human physically interacts with the camera assistant between 72.9 and 75.9 s to rapidly re-position the camera and find the gripper in the

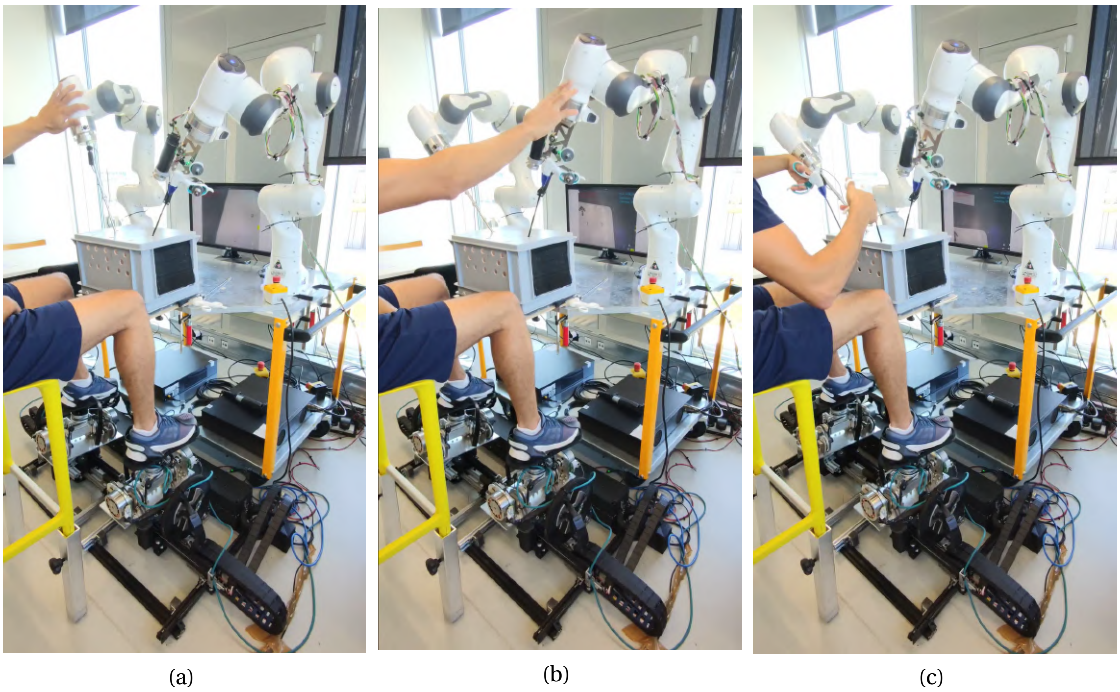

(a) (b) (c)

Figure 6.10 – Snapshots of the experiment evaluating compliance towards physical human interaction. In Figure 6.10a, the human physically interacts with the camera to find the gripper in the image (see Figure 6.10a). In Figure 6.10b, the human disturbs the gripper through physical interaction. The gripper complies to the interaction and comes back to the position commanded with the foot once the disturbance disappears. Finally, in Figure 6.10c, the human uses both hands and feet to perform a four-handed task. While moving the camera with the foot, an unexpected collision occurs between the camera assistant and the human left hand. The robot rapidly complies to the interaction and stops moving the camera.

image (see Figure 6.10a). During that period, the measured interaction forces produce the admittance velocity $\dot{\boldsymbol{x}}^{C}_{\boldsymbol{a,d}}$ (see Figure 6.11a). The power generated is used to fill the tank s^C which gets full very rapidly in 100 ms approximately. Once the interaction stops, the camera keeps the new position (as it is velocity controlled) and the tank begins to empty through the dissipation term, reaching 0 in about 1.3 s (see Figure 6.11b). Note that the modulation gain α^C follows the same behaviour than s^C (see Figure 6.11c). From 75.9 and 84.2 s, the human uses the right foot to move the gripper around. It is followed by two interaction periods where the human intentionally disturbs the gripper through physical interaction (from 84.2 to 85.2 s and from 87.9 to 90.6 s, see Figure 6.10b). Unlike the camera, the gripper does not keep the new position once the disturbance disappears but smoothly comes back to the position commanded through the foot (i.e. $\boldsymbol{x}^{\boldsymbol{G}}_{\boldsymbol{d}}$) as the modulation gain α^G decreases to 0. Finally, in the last part of the experiment we simulate a four-arm task where both human hands and feet are used (see Figure 6.10c). While moving the camera with the foot, an unexpected collision occurs between the camera assistant and the left human hand. The sudden peak in the measured interaction forces at around 104.7 s is enough to fill the camera tank to the maximum level, i.e.,

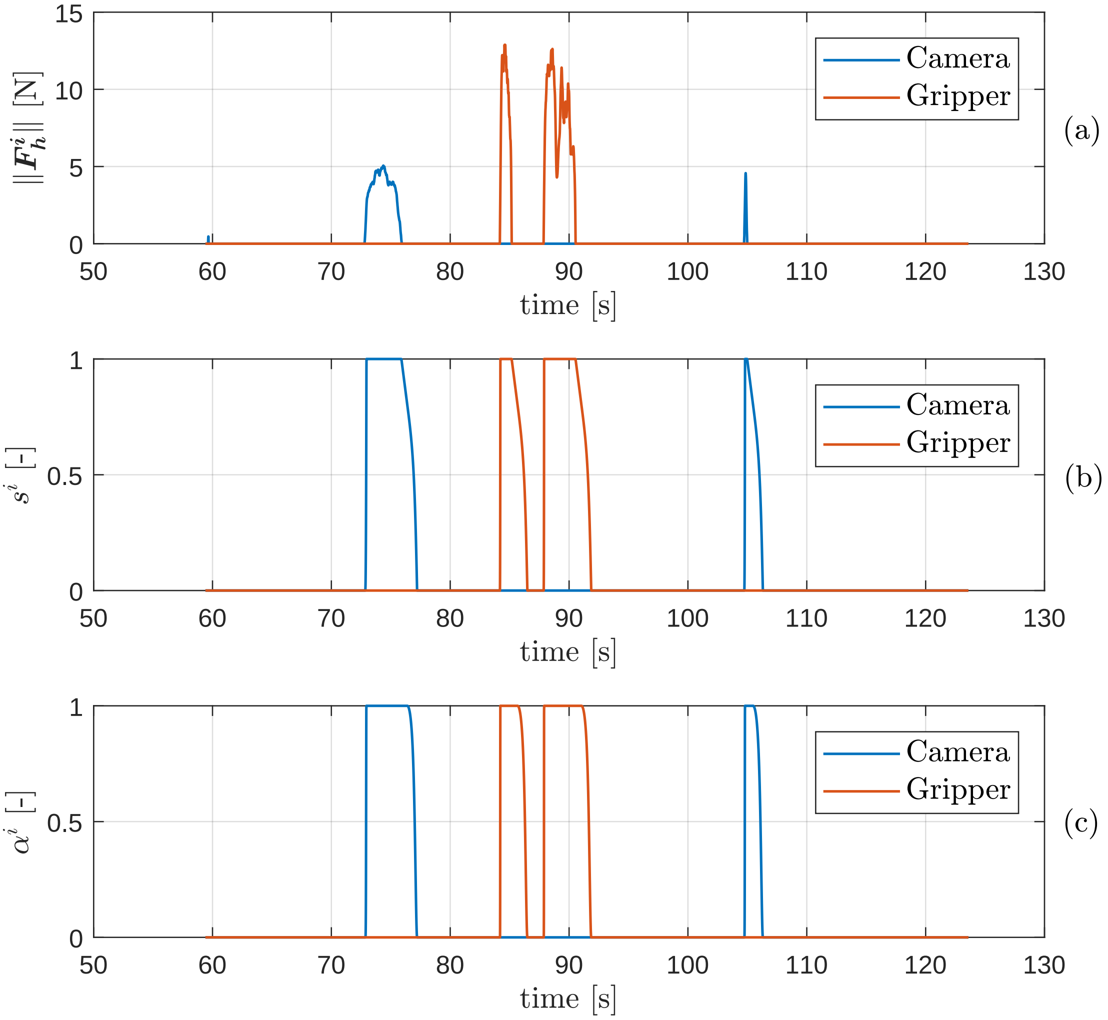


Figure 6.11 – Compliance towards physical human interaction: The measured human interaction forces are shown in Figure 6.11a while Figures 6.11b and 6.11c respectively show the energy tanks level and modulation gains signals.

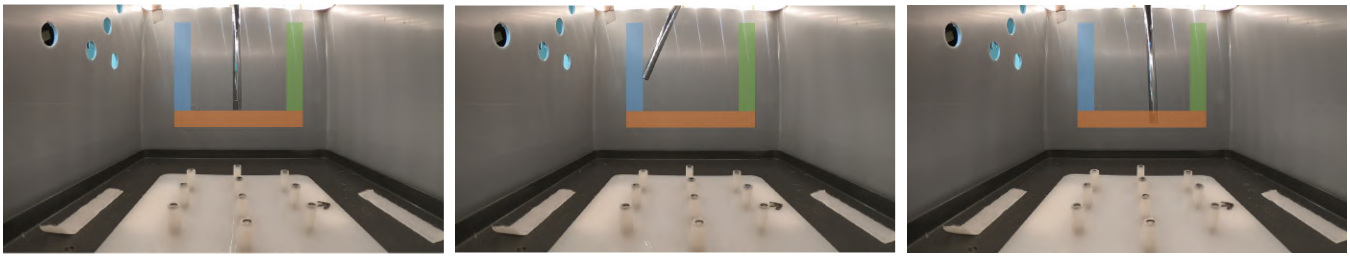


Figure 6.12 – Snapshots of the experiment evaluating workspace collision avoidance. The snapshots are taken inside the phantom. Three color rectangles are overlaid on the images to represent three of the five collision planes/boundaries used to constraint the camera motion. The colors used for the rectangles match the ones of the collision distance signals ($d^{C}_{c^{k}_{W}}$) in Figure 6.13a. In Figure 6.12a and 6.12b, the human controls the camera with the left foot to bring the tip to the workspace boundaries. The robot control prevents the tip to pass the boundaries in both cases. In Figure 6.12c, the human physically interacts with the camera assistant to intentionally bring the camera tip in collision with the bottom boundary. Because the robot controller still has some compliance inside and as the result of the human forces, the boundary is slightly violated in that case.

the camera assistant reacts fast to the disturbance and stops moving, ignoring the command from the foot. As the robot stops, the interaction forces disappear leading the energy tank to empty. However, the human acknowledges the collision and brings the foot back to the center of the platform to stop moving the camera with the foot. Such reactivity towards external disturbances is particularly desired when the surgeon is focusing on a task. It helps to preserve the physical integrity of the surrounding environment and contributes to increase safety.

6.4.2.3 Workspace Collision Avoidance

Constraining the motion of the tools tip is particularly important in surgery to guarantee safety of the patient. In this section, we evaluate the ability of the camera assistant to restrict the motion of its tool tip inside the workspace delimited by the five collision planes. Snapshots of the task taken inside the phantom are presented in Figure 6.12. While Figure 6.13 shows the measured distance signals $d^{C}_{c^{k}_{w}}$ to the collision planes (Figure 6.13a) and the haptic guidance forces $\boldsymbol{F}^{\boldsymbol{C}}_{\boldsymbol{d}}$ (Figure 6.13b).

Between 20.4 and 63.3 s, the human controls the camera with the left foot to bring the camera tool tip close to the workspace boundaries. The human first inserts the camera until reaching the lower boundary (red/orange boundary in Figure 6.10a) at around 30.3 s (see $d^{C}_{c^{1}_{W}}$ in Figure 6.13a). Before reaching the limit, haptic feedback starts to be generated to inform the human about the proximity to the boundary and to move away from it. In that particular case, a positive force is generated in the z component of $\boldsymbol{F}^{\boldsymbol{G}}_{\boldsymbol{d}}$ from 28.8 to 34 s so that the human moves

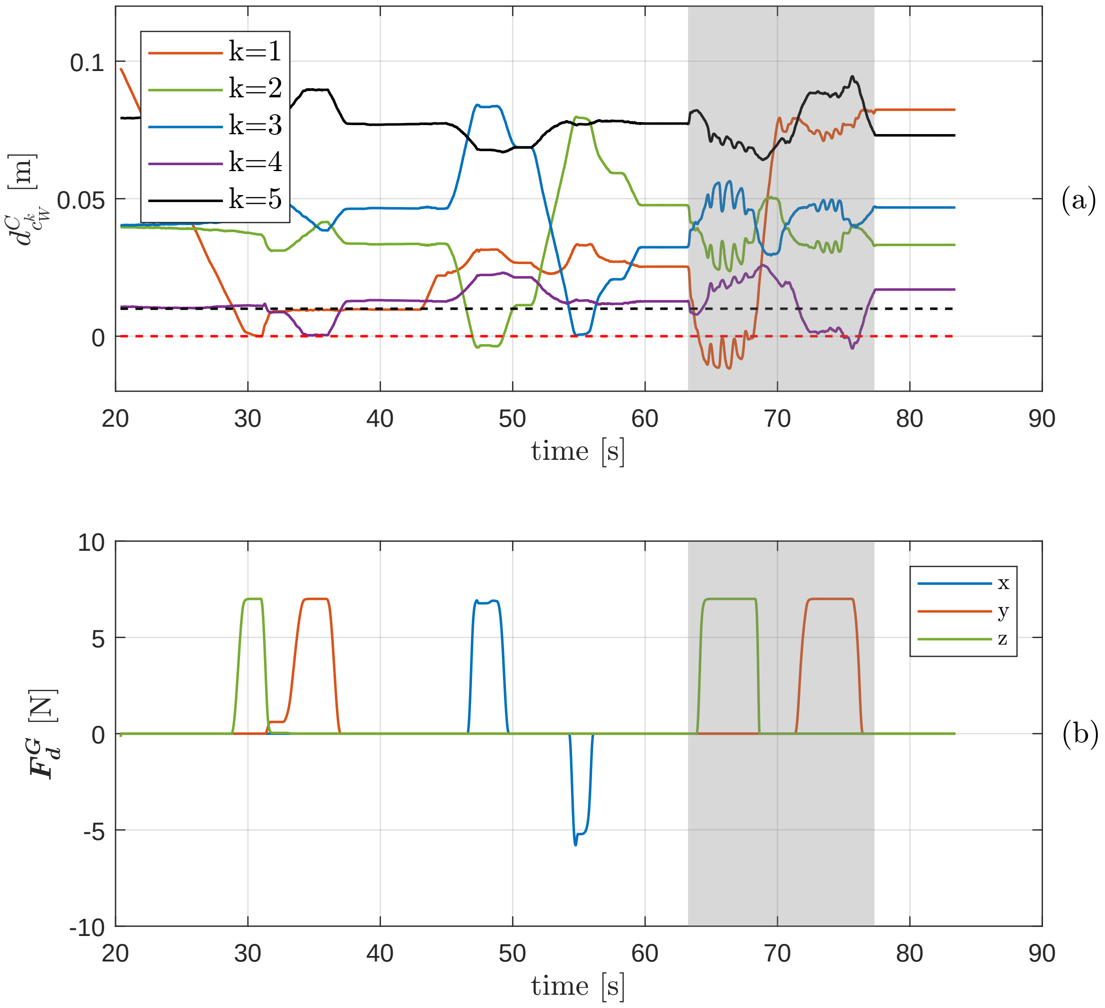


Figure 6.13 – Workspace collision avoidance: In Figure 6.13a, the measured distance signals between the camera tool tip and the five collision planes/boundaries are depicted. The dashed black line is the distance from which the haptic cues start to be generated (d_+ in Eq. 6.55) while the dashed red line is the security distance ($d_{c^k_W,s}$ in Eq. 6.9) used in the QP. The generated haptic guidance forces are provided in Figure 6.13b. The gray shaded areas finally correspond to the period where the human physically interacts with the camera assistant.

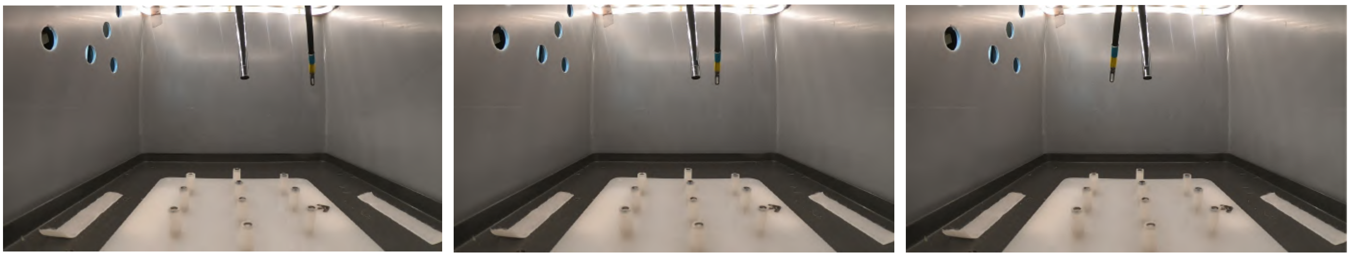

(a) (b) (c)

Figure 6.14 – Snapshots of the experiment evaluating tool collision avoidance. The human uses both feet to position the camera and gripper as in Figure 6.14a. Then, the human decides to move the gripper at the left of the camera by moving the foot in the corresponding direction. The robot control prevents the gripper to collide with the camera (Figure 6.14b). Through the haptic feedback the human understands how to move the gripper around the camera to reach the position in Figure 6.14c.

the camera tool tip up. The same behaviour can be observed when the human moves the camera to the blue boundary in Figure 6.12b at 54.6 s, resulting in a negative force generated in the x component of $\boldsymbol{F_d^G}$. Overall, when the human uses the foot to control the camera, the tip never exceeds the boundaries except at 47.1 s, where a very small violation of 0.0036 m can be measured for the green boundary. Finally between 63.3 and 72.3 s, the human physically interacts with the camera assistant to bring the tip in collision with the boundaries. Because the robot controller still has some compliance inside, the human can violate the boundaries through the interaction. However, despite the high human forces, up to 20 N when pushing down to the orange/red boundary from 64.2 to 68.2 s, the boundary violation remains small (at most 0.012 m, see Figure 6.12c).

6.4.2.4 Tool Collision Avoidance

The camera and gripper tool tip share parts of their workspace inside the trocar space. Therefore, it is important to avoid collision between the bodies of the tools while they are moving close to each other. The ability of the control framework to prevent such collisions is illustrated here through a small experiment. Snapshots of the experiment taken inside the phantom are presented in Figure 6.14. The evolution of the measured closest distance between the tools bodies $d_{c_T}^G$ is shown in Figure 6.15a, while the haptic guidance forces of the gripper $\boldsymbol{F_d^G}$ are visible in Figure 6.13b.

From 28.5 to 49.3 s, the human uses both feet to position the camera and gripper as indicated in Figure 6.14a. From there, the human decides to move the gripper at the left side of the camera which corresponds to a motion along the $\boldsymbol{+x_W}$ axis of the robot world frame. As the human moves the gripper towards the camera (in the positive x direction), $d_{c_T}^G$ decreases. Before reaching the security distance, negative haptic forces are generated in the x compo-

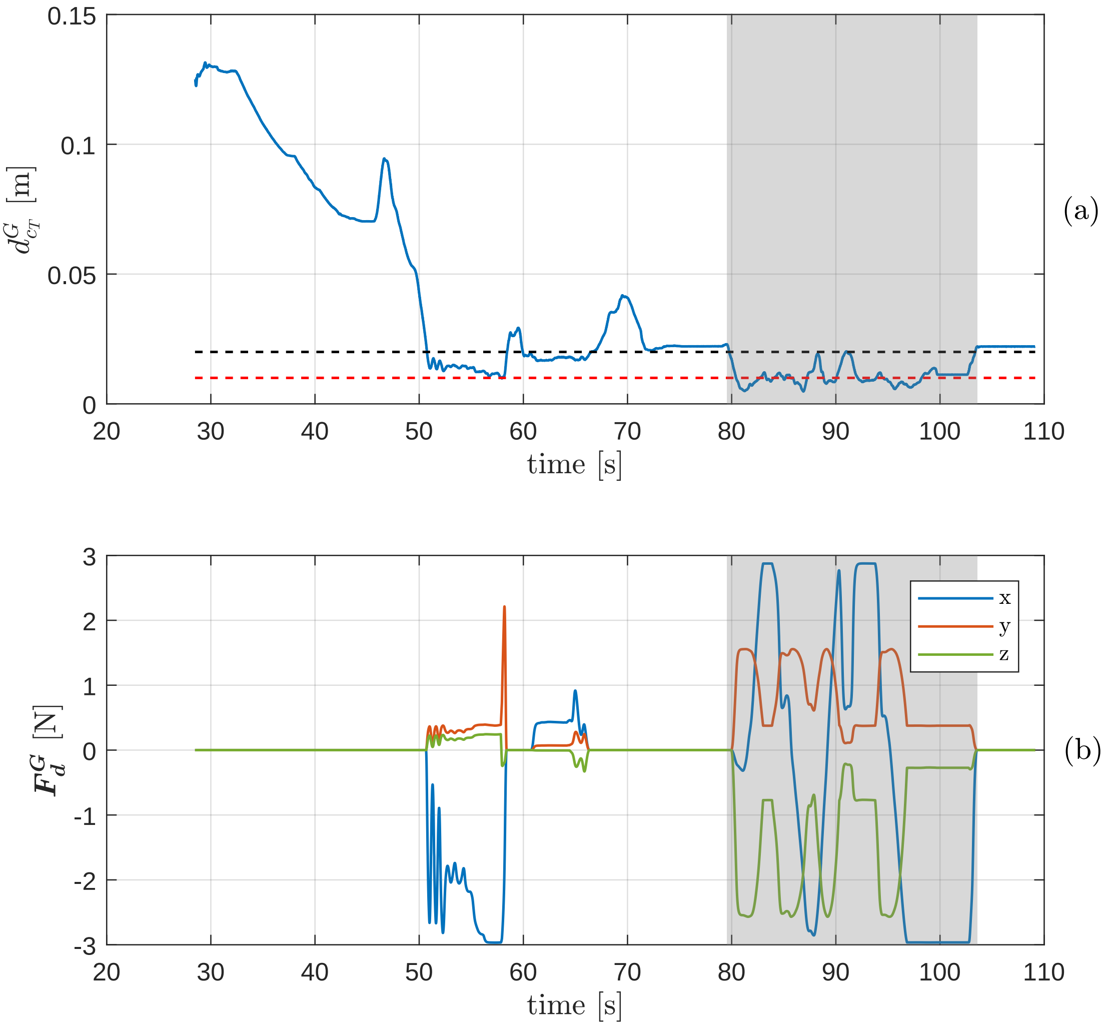


Figure 6.15 – Tool collision avoidance: The measured tool collision distance signal is shown in Figure 6.15a, while Figure 6.15b provides the the generated haptic guidance forces signal for the gripper. Similarly to Figure 6.13a, the dashed black line in Figure 6.15a is the distance from which the haptic cues start to be generated (d_+ in Eq. 6.55) while the dashed red line is the security distance ($d_{c_T,s}$ in Eq. 6.8) used in the QP. The gray shaded areas finally highlight the physical human interaction period.

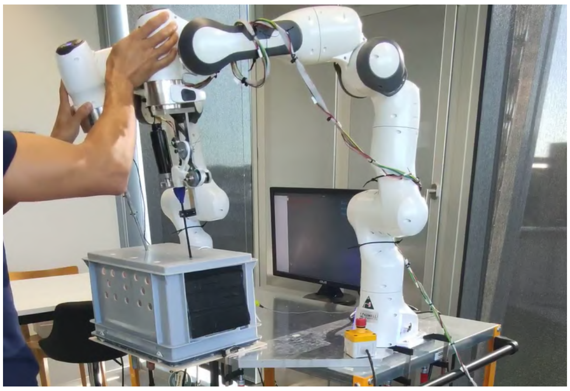


Figure 6.16 – Snapshot of the experiment evaluating end-effector collision avoidance. The human is physically interacting with both robotic assistants to push their end-effectors against each other.

nent of $\boldsymbol{F_d^G}$ to prevent collision with the camera (see Figure 6.14b). The conflict between the haptic cues and the human intended motion first results in some oscillations in the generated feedback. However, the human quickly acknowledges the proximity to collision and uses the haptic forces direction to move the gripper around the camera from 58 to 60 s. Indeed, the haptic forces direction changes with the collision direction as illustrated by the simultaneous increase in the x and y component of $\boldsymbol{F_d^G}$ at around 58 s. The human can use the gripper haptic cues to understand where is the camera with respect to the gripper, and move the gripper accordingly to reach the desired position while avoiding the camera, as achieved in Figure 6.14c. Finally, from 79.6 to 103.6 s, the human physically interacts with the camera to bring it in collision with the gripper from various directions. As can be appreciated from Figure 6.15a, the robot control manages to limit the violation of the security distance ($d_{c_T,s}$, set to 0.01 m) to 0.005 m despite the high human forces reaching here up to 25 N.

6.4.2.5 End-effector Collision Avoidance

The two robotic assistants being positioned close to each other, the robot control should prevent any collisions between their bodies. In particular, the most likely body parts to collide are the robot end-effectors on which the camera or gripper tools are mounted. In this section, we present an experiment where the human is physically interacting with both robotic assistants to bring their end-effectors in collision. A snapshot of the experiment is provided in Figure 6.16, while Figure 6.17 shows the measured collision distance between both end-effectors $d_{c_E}^G$ (Figure 6.17a) and the norm of the human interaction forces $\|\boldsymbol{F_h^i}\|$ (Figure 6.17b).

Three main interaction phases can be observed in Figure 6.17b. A first one from 31.5 to 37.5 s where the human is only pushing the gripper assistant against the camera. The second and third one from 41.1 to 59.2 s and from 68.5 to 82 s respectively, where the human is pushing both robot end-effectors against each other. As can been seen in Figure 6.17a, the

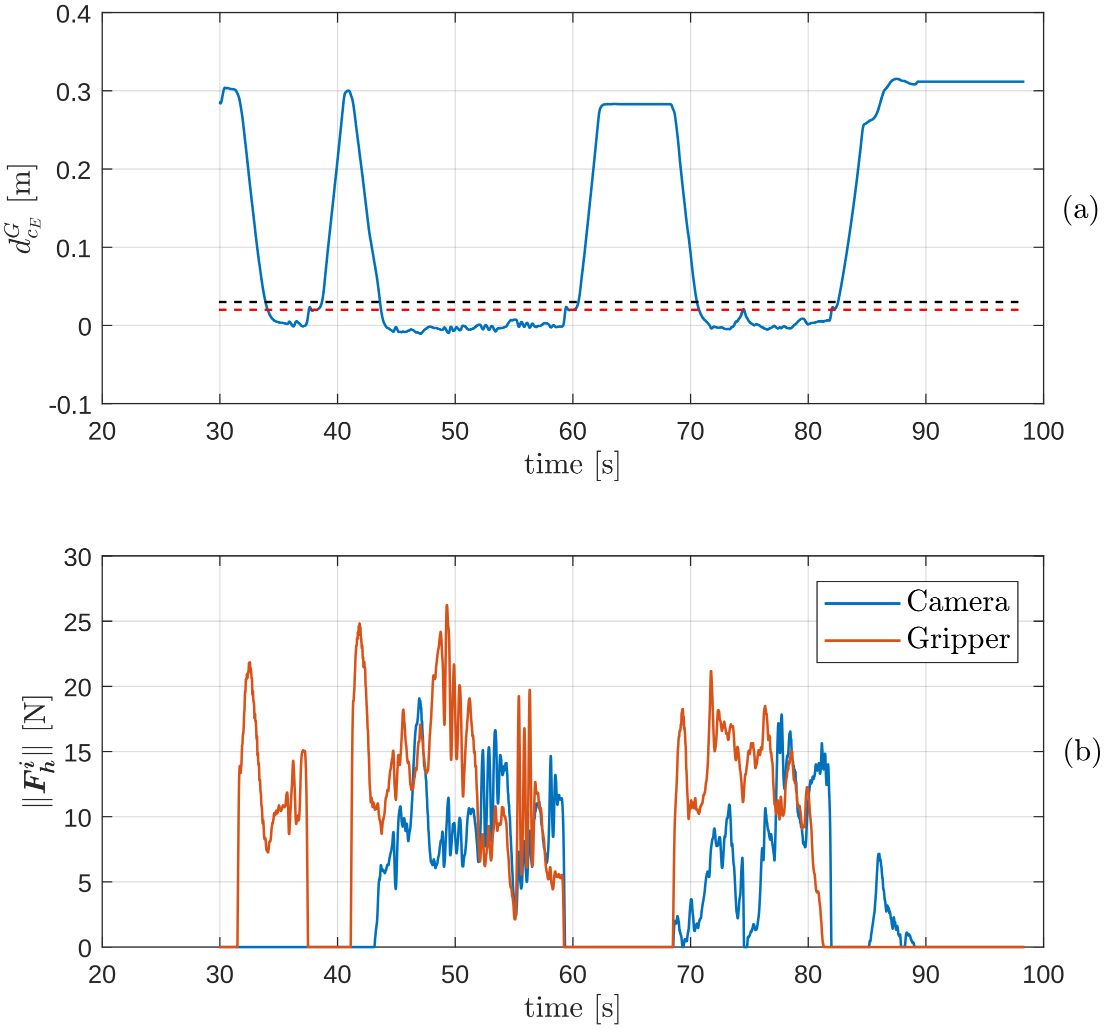
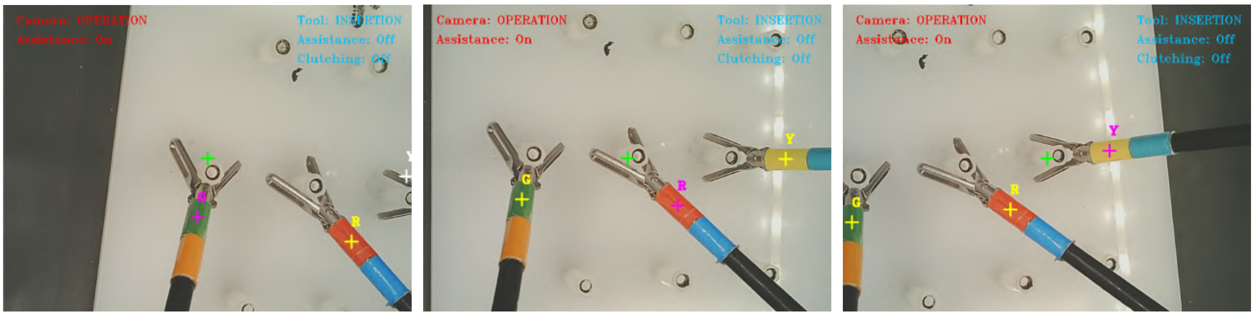


Figure 6.17 – End-effector collision avoidance: The measured end-effector collision distance signal is depicted in Figure 6.15a, while the measured human interaction forces are given in Figure 6.17b. Similarly to Figure 6.13a, the dashed black line in Figure 6.15a is the distance from which the haptic cues start to be generated (d_+ in Eq. 6.55) while the dashed red line is the security distance ($d_{c_E,s}$ in Eq. 6.7) used in the QP.



(a) (b) (c)

Figure 6.18 – Snapshots of the experiment validating the camera's adaptive tool tip following assistance. Three surgical tools are positioned inside the phantom. In Figure 6.18a, the camera is following the tool tip with the green marker. Through a small gesture to the right with the left foot, the human switches the following assistance to the tool tip with the red marker (Figure 6.18b). The human repeats the same operation to switch to the tool tip with the yellow marker in Figure 6.18c. Note that when the color markers at the tip of the tools are visible, a yellow cross is depicted at the center of the marker. When the camera assistance is active, the color of the cross changes to purple for the followed tool tip. Finally, the green cross indicates the image center.

strong human forces (up to 25 N) cause a violation of the security distance ($d_{c_E,s}$ set to 0.02 m here) of, at most, 0.03 m. Although the violation is higher than what we saw previously with workspace and tool collision, in practice, no collisions happened thanks to the safety radius ($r^i_{c_E}$ set to 0.1 m for both robots) providing some security room for the collision. Finally, note that when the interactions stop (at 37.5 or 59.5 s for instance), $d^G_{c_E}$ first comes back around the security value for 1 s approximately before increasing until about 0.3 m. This is because the gripper assistant smoothly comes back to the position commanded with the foot as its energy tank gets drained with the dissipation term.

6.4.2.6 Camera's Adaptive Tool Tip Following Assistance

We finally conclude this series of mini-experiments by demonstrating the adaptive tool tip following assistance of the camera. In this last experiment, three surgical tools are positioned inside the phantom and the human uses the camera assistance to follow the tools and switch from one to another. Snapshots of the experiment taken from the camera image are provided in Figure 6.18. Figure 6.19 shows the evolution of the tools' beliefs b_t (Figure 6.19a), the human input ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{h}}$ (Figure 6.19b), and the relative tools distance from the image center ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{t}}$ (Figure 6.19c).

From 27 to 49.9 s, the camera assistant is following the tool tip with the green marker ($b_2 = 1$), which is manipulated by the human. During that period, despite the fast human motions, when the marker is visible, the camera assistant can keep the tool tip close to the image center

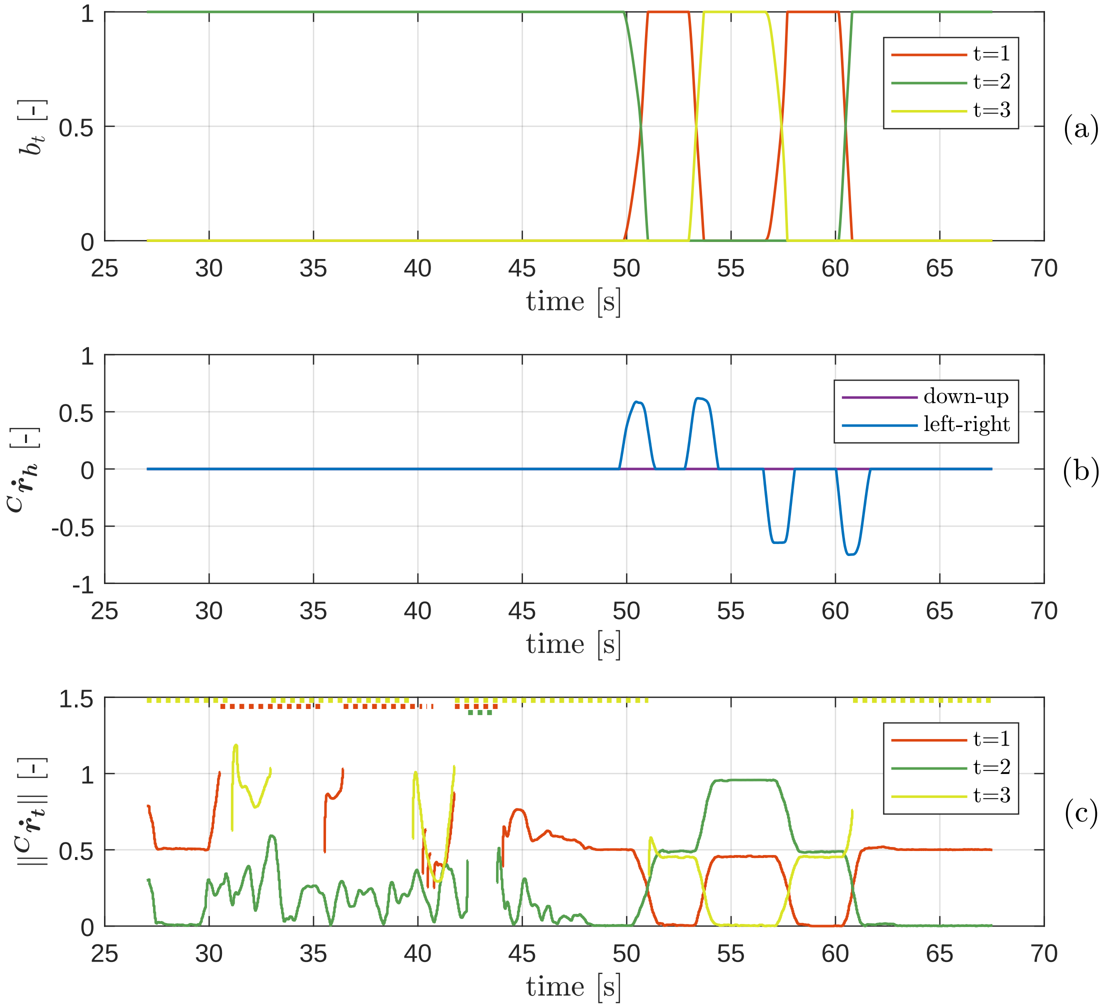


Figure 6.19 – Camera's adaptive tool tip following assistance: Figure 6.19a shows the evolution of the tools' beliefs during the experiment. The measured human input expressed in the camera frame is provided in Figure 6.19b. Figure 6.19c finally shows the norm of the relative tools' tip position (expressed in normalized units). Note that in both Figures 6.19a and 6.19c, the curves color match the tool tip color markers. Furthermore, when a color marker is not visible, its relative position is considered to be zero in the beliefs' adaptation algorithm. However, in Figure 6.19c, for better clarity, we set the relative distance to some predefined value at the top of the graph when the markers are not visible and put the corresponding parts of the signals in dashed.

with an average error of 88.9±78.2 px. At 42.38 s, the human quickly removes the tool from the image field of view. As the camera assistant cannot cope with the human speed the tool is lost. In that case, camera assistance remains active (b_2 still equals 1) but ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{t}}$ is set to $\mathbf{0}_{\mathbf{2}\times\mathbf{1}}$ leading the camera to stop. At 43.8 s, the human brings the tool back in the camera field of view. The robot starts to follow the tool tip again, and the human finally positions the tip as indicated in Figure 6.18a. At 49.9 s, the human decides to follow the tool tip with the red marker. Because the latter is at the right of the image center in Figure 6.18a, the human simply moves the foot to generate a human input velocity in the right direction of the camera frame (see Figure 6.19b). The camera assistant quickly understands the human intention to switch to the tool tip with the red marker as demonstrated by the beliefs' profiles where b_2 and b_1 smoothly shift to 0 and 1 respectively, resulting in the configuration in Figure 6.18b. Note that the human only needs to provide a short foot gesture (1.5 s approximately) to change the followed tool tip. The same principle can be observed in the rest of the experiment, where the human successively switches to the tool with the yellow (see Figure 6.18c), red, and green markers. In Figure 6.18c, despite the fact that the red and green tools are both in the left direction with respect to the image center, it is the belief of the red tool, the closest one, that increases when the human starts generating an input velocity to the left at 56.7 s. This validates the design of the beliefs' adaptation mechanism presented in section 6.3.4.4, where we consider separately the distance from the tool to the image center and the similarity between the tool and human input directions (see Eq. 6.42). If several tools are aligned in the same direction, the proposed adaptation rule allows to switch to the closest one, which would not be possible if we apply the rule of Khoramshahi and Billard (2019); i.e., replacing $\dot{b}_{t,1} + \dot{b}_{t,2}$ in Eq. 6.41 by a dot product between ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{t}}$ and ${}^{C}\dot{\boldsymbol{r}}_{\boldsymbol{h}}$. The dot product in that case would always favor the furthest tool tip.

6.5 Discussion and Conclusion

In this chapter, we proposed a control framework for four-arm robotic assisted laparoscopic surgery. In the envisioned scenario, a surgeon is performing the surgery with the hands, while being assisted by two robotics assistants, one holding a camera, the other one manipulating a laparoscopic gripper, controlled through foot interfaces. The first goal was to guarantee a safe collaboration between the surgeon and the robotic assistants. To this end, a QP-based inverse kinematics solver was developed to satisfy all the tasks and safety related constraints such as the RCM constraint, workspace or tool collision avoidance constraints. The framework is enriched with haptic feedback strategies to increase the situational awareness of the surgeon, especially towards the collision constraints. The experimental evaluations showed the ability of the framework to accurately follow the desired tool tip tasks while maintaining all the constraints specified, even under drastic human disturbances, and providing meaningful haptic feedback. The framework not only provides accuracy in the tracking of the tasks, but is also reactive and compliant to external disturbances such as physical human interaction and allows hand-guided motions of the robotic tools.

Finally, to ease the control of the camera through the foot, autonomous tool tip following was proposed as an assistance modality for the camera where the surgeon can change the followed tool tip through simple foot gestures. The camera assistance was evaluated jointly with a grasping assistance modality for the gripper (autonomous opening/closing) in a user study where uni-pedal, bi-pedal, and four-handed tasks involving target-seeking/following and peg-transfer were achieved. The results showed the effectiveness of the assistance modalities into reduce task load while increasing the performances and perception of fluency for each task, especially the four-arm one (i.e., $T4$). Furthermore, when increasing the task complexity (i.e., the number of limbs to control), the use of shared control allows to alleviate the degradation of the responses to the subjective metrics in particular, as observed for mental demand or ease of use.

7 Conclusion

In this chapter we summarize the main contributions of the thesis, point out the limitations and indicate potential future work directions.

7.1 Main Contributions

In this thesis, we proposed control strategies to perform (non-)collaborative contact tasks in unstructured environment subject to uncertainties. To this end, we mainly relied on the time-invariant dynamical system framework providing robustness to real-time disturbances and immediate re-planning of the task.

While dynamical systems were extensively used as motion generators, in chapter 3, we proposed a simple modulation strategy to generate contact forces when the robot(s) get(s) close to the contact surface. Our modulated dynamical system combines a nominal DS responsible of motion with a modulation term normal to the surface in charge of the contact forces. We theoretically proved passivity of the system and empirically assessed the proposed strategy in two experimental scenarios: a polishing task on a non-flat surface and a bi-manual manipulation task. We showed the ability of the system to track the desired contact forces while providing stability and robustness towards large disturbances such as a human breaking the contact with the surface.

To compensate for the force tracking error coming from the uncertainties on the robot's dynamics and environment, we proposed in chapter 4 to learn a state-dependent force correction model added to the normal modulation term defined in chapter 3. The model is encoded with gaussian radial basis functions and adapted online using the force tracking error. We showed the effectiveness of the approach in achieving high force tracking accuracy on a polishing task with different levels of robot velocities and target contact forces. We also emphasized the reactivity and adaptability of the approach (and the DS framework in general) through a collaborative cleaning task where we simultaneously achieved force adaptation (through the interaction with the surface) with motion and task adaptation (through physical human interaction).

In chapter 5, we designed (shared) control strategies for four-arm manipulation tasks involving the two human biological arms and two robotic arms controlled through haptic foot-interfaces. To assist the human in supporting and moving the manipulated object, we proposed two assistance modalities initiated upon contact: autonomous contact force generation and auto-coordination of the robotic arms through single foot control. A user study was conducted to evaluate the benefits of the assistance modalities and control strategies. The results showed the effectiveness of force assistance in providing robust manipulations while being perceived as better than coordination assistance in terms of human robot fluency. Combining both assistance modalities resulted however in the most successful and appreciated strategy among the four proposed ones.

Finally, in chapter 6, we presented a control framework to achieve four-arm robotic assisted laparoscopic surgery, where two robotic assistants, holding a camera and a gripper and controlled through haptic foot interfaces, assist a surgeon during the operation. The framework ensures safe collaboration between the surgeon and the robotic assistants through a QP-based inverse kinematics satisfying all the task and safety related restrictions. We showed the efficacy of the framework in achieving the desired tool tip tasks accurately while respecting the constraints and being able to comply to external disturbances such as physical human interaction. To assist the surgeon in controlling the camera, an assistance modality was designed to autonomously follow an instrument tool tip, with the surgeon being able to control the followed instrument through foot gestures. The camera assistance was assessed jointly with a grasping assistance modality for the gripper through a user-study consisting in uni-pedal, bipedal, and four-handed laparoscopic tasks involving target seeking/following and peg-transfer. The results showed the benefits of the assistance modalities in increasing performance, fluency and reducing task load for each task individually especially in the four handed task where the human has to control all the limbs. Finally, if increasing the task complexity (i.e., number of limbs to control) worsened the perceived mental load or ease of use, the use of shared control was able to mitigate this effect.

7.2 Outlook

7.2.1 Chapter 3

In chapter 3 we used the DS-impedance controller to track the modulated DS and generate both motion and contact forces. The DS-impedance controller provides selective compliance/impedance on the directions orthogonal and tangent to the desired velocity. For contact tasks, this selective compliance would be more advantageous in the directions normal and tangent to the surface. Thereby, the robot could provide a stiffer behaviour in the tangential directions to reduce the sensitivity to friction and follow more accurately the desired motion dynamics on the surface. While a more compliant behaviour in the normal direction could be desirable to ensure a safe interaction. Such a damping matrix could be constructed by selecting the nominal DS direction and the normal vector to the surface $\boldsymbol{n}(\boldsymbol{x})$ as first and second eigenvector, the third one being the cross-product of the first two.

Finally, the absence of force feedback in the normal modulation term also limits the achievable force tracking accuracy which remains still affected by the uncertainties in the robot dynamics and the environment. In chapter 4, we tried to tackle this limitation.

7.2.2 Chapter 4

The state-dependent force correction model proposed in chapter 4 constraints the user to manually distribute the RBF kernels on the contact surface. We opted for a simple uniform distribution to cover the desired robot motion on the surface. Such distribution is particularly effective with smooth surfaces. With more complex surfaces, a uniform distribution would not be able to capture all the surface non-linearities, and increasing globally the number of RBFs is not a proper solution regarding computational cost. One could consider Gaussian Process Regression (GPR) (Rasmussen (2003)) as an alternative to our method which is similar but more generic. Indeed, GPR can model complex non-linear functions and has been applied to learning surfaces (Martens et al. (2016)). Furthermore, our model only depends on the robot's end-effector position. Such models perform satisfactorily as long as the robot re-visits the same position with similar velocities. Mathematically speaking, non-stationary behaviors with relatively lower dynamics than the adaptation can be compensated. To improve the performance further, one can include other variables such as velocity in the model. However, one has to deal with the curse of dimensionality or find efficient ways to distribute the RBFs in the input space.

7.2.3 Chapter 5

The force assistance strategy was efficient in supporting the object during the manipulation. The robustness of the grasp can be further improved by auto-aligning the robots' end-effectors before generating the desired contact forces. Indeed, an excessive misalignment of the end-effectors at the contact can result in unintentional object rotation due to the forces. This behaviour was especially observed for strategies without coordination assistance (i.e., *U* and *U+F*) where the human controls the end-effectors position independently. Helping the human in aligning the end-effectors in these cases, could be an additional useful assistance level. Furthermore, given the difficulties encountered by the participants when only having coordination assistance (i.e., strategy *C*), the workspace and force mapping gains between the foot platform and robot arms might need a special adjustment for this control strategy.

Finally, an important step to consider in the future is stability or passivity of the shared bilateral tele-operation. In standard bilateral tele-operation, passivity has been extensively studied (Lee and Spong (2006), Franken et al. (2011)). Franken et al. (2011) introduced a general two-layer control architecture combining passivity and transparency for a bilateral tele-operation system. The idea is to explicitly separate the achievement of transparency and stability in two different layers. A top transparency layer is proposed to compute the desired control commands/torques for both the master and the slave devices. A bottom passivity layer is used to adjust the desired commands by monitoring and enforcing the energy balance of the whole system via the use of energy tanks, and by ensuring passivity and stability of the tele-operation. The intuitiveness and flexibility provided by this two-layer architecture has been recently exploited in Ferraguti et al. (2015b); Heck et al. (2018). Similarly to the works above, we could include passivity assurance with energy tanks, to monitor the energy flow in the system and prevent the generation of extra energy coming from the control actions. As the total system is composed of two master interfaces and slave robots in our case, we could use one energy tank for each of them (four in total) and adapt their control laws to make sure that the total system stays passive.

7.2.4 Chapter 6

The control framework proposed in chapter 6 differs from the previous chapters. Indeed, we used a QP-based inverse kinematics to output desired joint angles tracked through a joint impedance controller. The main motivation behind the QP was to centralize all the task and safety related constraints in one place and to generate a solution that conciliates as best as possible all of them. While the joint impedance controller was used to provide the accuracy needed to keep the RCM constraint and follow the desired tool tip tasks, especially when targeting small tool tip velocities. From there, to recreate the lack of compliance due to the use of high gains in the joint impedance controller, we used an admittance law in the expression of the desired tool tip dynamics. It allowed the robotic assistants to react and comply fast to external disturbances such as physical human interaction, and contributed to increase safety of the overall system and the surrounding environment. Nonetheless, the "feeling" of interaction is different than traditional impedance controller. This is due to the time delay introduced by the integration of the tank dynamics from which the task modulation gain is derived. It requires some tuning to find the best trade-off between smoothness of the interaction and reactivity. Moreover, the fluency of the interaction also depends on the estimation of the human interaction forces. In practice they were estimated based on the external forces obtained from the robot manufacturer. The latter were processed by first removing the forces due to the interaction of the instruments with the trocar and the environment inside the phantom (that we estimated through a F/T sensor), and second by applying a dead-zone to the resulted forces. For the gripper in particular, the dynamic uncertainties constrained us to choose a big dead-zone, leading the human to apply high forces to physically interact with the robot. Despite of that, we could demonstrate robustness to real-time disturbances outside the phantom. However, regarding robustness inside the phantom, we are still limited. This is due to the integration over time of the task-specific DS through the QP, in contrast to using the DS in closed-loop as done in the previous chapters, which is a limitation of the current approach. Nonetheless, the use of haptic feedback to inform the human/surgeon about potential collisions between the tools or contact with the environment, helps to partially alleviate this limitation, as the human can be aware of potential disturbances inside and react accordingly. Estimating the tool tip contact forces is however not an easy task, especially when the F/T sensor is not placed at the tip but at the base of the instruments as done in our case. Finally, the QP proposed is purely kinematics and can be enhanced by including the dynamics of the robot to generate joint velocities/accelerations that are physically consistent. Similarly, limiting torques and forces along specific directions are additional safety constraints that should be integrated into the QP in the future to preserve the physical integrity of the environment (patient, surgeon, etc.).

Regarding the camera assistance modality, it was particularly precise into following the tool-tip of the surgical tools. The possibility to decide which tool to follow through simple foot gestures was appreciated by the participants although only a few of them exploited this feature during task $T4$ (as we advise them to mainly follow the foot-controlled gripper and switch tools only when needed). Following one tool is not optimal when multiple ones need to move in coordination. Some can be lost while it is important to keep all of them in the camera field of view for safety reasons. Therefore, future work could consider additional camera assistance modalities such as centre of geometry following to deal with this limitation.

A Appendix Chapter 3

A.1 Common Experimental Details

For all experiments:

- The control strategy runs at $200Hz$.
- The DS impedance controller presented in section 2.2 is used with $d_1 = d_2 = d_3 = 150$.
- The orientation controller presented in section 2.3 is used with $\boldsymbol{K_\zeta} = \text{diag}([15, 15, 15])$ and $\boldsymbol{D_\omega} = \text{diag}([2, 2, 2])$.
- The desired orientation profile of the robots' end-effector is defined based on the distance to the target surface. The $\boldsymbol{z_o}$ axis of the end-effector (aligned with the tool) is controlled to progressively align itself with the normal vector of the surface $\boldsymbol{n}(\boldsymbol{x})$ as the robot becomes closer to the surface. This is achieved in several steps:
    - We first compute the rotation matrix $\tilde{\boldsymbol{R}}(\boldsymbol{x})$ needed to align $\boldsymbol{z_o}$ with $\boldsymbol{n}(\boldsymbol{x})$ through the Rodrigues' rotation formula:

$$\begin{cases} c = \boldsymbol{n}(\boldsymbol{x})^{\boldsymbol{T}} \boldsymbol{z}^{\boldsymbol{E}} \\ \boldsymbol{k} = \boldsymbol{z}^{\boldsymbol{E}} \times \boldsymbol{n}(\boldsymbol{x}) \\ s = \|\boldsymbol{k}\| \\ \boldsymbol{K} = [\boldsymbol{k}]_\times \\ \tilde{\boldsymbol{R}}(\boldsymbol{x}) = \boldsymbol{I_{3\times 3}} + s\boldsymbol{K} + (1-c)\boldsymbol{K}^2 \end{cases} \tag{A.1}$$

    - $\tilde{\boldsymbol{R}}(\boldsymbol{x})$ is then converted into a unit rotation quaternion $\tilde{\boldsymbol{Q}}(\boldsymbol{x})$ which is applied to the measured one $\boldsymbol{Q}$ in order to obtain a target quaternion in contact $\boldsymbol{Q_t}(\boldsymbol{x}) = \tilde{\boldsymbol{Q}}(\boldsymbol{x}) \star \boldsymbol{Q}$ where $\star$ denotes the quaternion product.
    - Spherical Linear Interpolation (SLERP) is finally used between the measured and target

quaternion in contact to provide a smooth desired orientation profile $\boldsymbol{Q_d}(\boldsymbol{x})$:

$$\boldsymbol{Q_d}(\boldsymbol{x}) = \frac{\sin\Big(\Omega\big(1 - w(\boldsymbol{x})\big)\Big)\boldsymbol{Q} + \sin\left(\Omega w(\boldsymbol{x})\right)\boldsymbol{Q_t}(\boldsymbol{x})}{\sin(\Omega)} \tag{A.2}$$

where $\Omega = \arccos\big(\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{Q_t}(\boldsymbol{x})\big)$ and $w(\boldsymbol{x}) \in [0,1]$ is an interpolation parameter that can be designed as a function of the distance to the contact surface $\Gamma(\boldsymbol{x})$. For $w(\boldsymbol{x})$, we choose:

$$w(\boldsymbol{x}) = 1 - \tanh(\kappa\Gamma(\boldsymbol{x})) \tag{A.3}$$

where $\kappa > 0$ controls how fast the end-effector orientation should be aligned with the normal vector to the surface based on the normal distance.

- A desired angular velocity is computed based on the measured and desired quaternion:

$$\boldsymbol{Q_\omega}(\boldsymbol{x}) = k_\omega \boldsymbol{Q}^{-1} \star \big(\boldsymbol{Q_d}(\boldsymbol{x}) - \boldsymbol{Q}\big) \tag{A.4}$$

where $\boldsymbol{Q_\omega}(\boldsymbol{x}) = [0\ \boldsymbol{\omega_d^{E^T}}]^T$ with $\boldsymbol{\omega_d^E}$ the desired angular velocity expressed in the end-effector frame. To express it in the world frame we simply use the end-effector orientation matrix $\boldsymbol{R_o}$ such that $\boldsymbol{\omega_d} = \boldsymbol{R_o}\boldsymbol{\omega_d^E}$. $k_\omega > 0$ is a positive gain that we set to 5 for all experiments.

A.2 Polishing Task on a Non-Flat Surface

- The shape of the non-linear surface is learned with C Support Vector Regression (C-SVR). The process of building the train dataset and learning the surface model offline before evaluating it online to obtain the normal distance $\Gamma(\boldsymbol{x})$ and vector $\boldsymbol{n}(\boldsymbol{x})$ to the surface is described thereafter:
  - The robot is first used to collect points on the surface. By putting the robot in gravity compensation, we bring the tool-tip in contact with the surface and sweep the surface manually. The force-torque sensor is used to make sure that the recorded points are contact points (using a threshold of $3N$). Each point recorded is expressed in a local frame attached to the surface:

$$\boldsymbol{x_s} = \boldsymbol{R_s^T}(\boldsymbol{x} - \boldsymbol{p_s}) \tag{A.5}$$

    where $\boldsymbol{R_s} \in \mathbb{R}^{3\times 3}$ is the wooden plate orientation matrix expressed in the world frame, constructed from the three optitrack markers attached to the plate, while p_s is the position of one of the marker. $\boldsymbol{x} \in \mathbb{R}^3$ and $\boldsymbol{x_s} \in \mathbb{R}^3$ denote the tool-tip position in the world and surface frame respectively. In practice a good coverage of the surface was obtained with around 70000 ~ 100000 points recorded at a frequency of $200Hz$.
  - After collecting points on the surface, 30000 points are randomly generated above the surface in a space delimited by the collected points, over a height of $0.4m$. For each of these generated points, the normal distance to the surface is approximated by the

distance to the closest point on the surface (among the ones collected). Thereby, we build our dataset whose input are 3D points above the surface expressed in the surface frame and whose output/target is the approximated normal distances. Note that this dataset can be also enriched by the points collected on the surface by setting their approximated normal distances to 0. In practice however, no significant differences were observed in the learning result when adding or not the collected points to the dataset.

  - LIBSVM is used to learn the C-SVR model $f_{svr} : \mathbb{R}^3 \to \mathbb{R}$ such that $\Gamma(\boldsymbol{x}) = f_{svr}(\boldsymbol{x_s})$. The hyperparameters to be selected for training are the penalty term C, the kernel type, and the epsilon tube ϵ. In our case, we used a gaussian kernel with a width $\sigma = 0.2$, and chose $C = 100$ and $\epsilon = 0.01$. The gradient of $f_{svr}(\boldsymbol{x_s})$ with respect to $\boldsymbol{x_s}$ gives us the normal vector $\boldsymbol{n_s}(\boldsymbol{x})$ expressed in the surface frame: $\boldsymbol{n_s}(\boldsymbol{x}) = \nabla_{\boldsymbol{x_s}} f_{svr}(\boldsymbol{x_s})$. In practice, the gradient computation is achieved thanks to the SVMGrad package available at https://github.com/nbfigueroa/SVMGrad.
  - When evaluating the model online the tool tip position $\boldsymbol{x}$ is converted in the surface frame using Eq. A.5 to obtain $\Gamma(\boldsymbol{x})$ and $\boldsymbol{n_s}(\boldsymbol{x})$. The normal vector is then converted back to the world frame through the orientation matrix $\boldsymbol{R_s}$: $\boldsymbol{n}(\boldsymbol{x}) = \boldsymbol{R_s}\boldsymbol{n_s}(\boldsymbol{x})$.

- The nominal DS $\boldsymbol{f}(\boldsymbol{x})$ is defined by:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x})\, v_0 \tag{A.6}$$

where:

  - $v_0 = 0.2\ m/s$ is the target velocity norm.
  - $\boldsymbol{R}(\boldsymbol{x})$ is a rotation matrix designed to progressively align $\boldsymbol{n}(\boldsymbol{x})$ with the normalized desired dynamics in contact with the surface $\hat{\boldsymbol{f}}_{\boldsymbol{contact}}(\boldsymbol{x}) \in \mathbb{R}^3$, as the robot becomes close to contact. It is obtained through the Rodrigues' rotation formula:

$$\begin{cases} \phi = \arccos\left(\boldsymbol{n}(\boldsymbol{x})^T \hat{\boldsymbol{f}}_{\boldsymbol{contact}}(\boldsymbol{x})\right) \\ \theta = \phi\Big(1 - \tanh\left(40\Gamma(\boldsymbol{x})\right)\Big) \\ \boldsymbol{k} = \boldsymbol{n}(\boldsymbol{x}) \times \hat{\boldsymbol{f}}_{\boldsymbol{contact}}(\boldsymbol{x}) \\ \boldsymbol{K} = [\boldsymbol{k}]_\times \\ \boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{I}_{3\times 3} + \sin(\theta)\boldsymbol{K} + (1 - \cos(\theta))\boldsymbol{K}^2 \end{cases} \tag{A.7}$$

  - $\hat{\boldsymbol{f}}_{\boldsymbol{contact}}(\boldsymbol{x})$ is the unit vector in the direction of $\boldsymbol{f}_{\boldsymbol{contact}}(\boldsymbol{x})$ obtained from the projection of the desired circular dynamics $\boldsymbol{f}_{\boldsymbol{circular}}(\boldsymbol{x})$ on the surface:

$$\boldsymbol{f}_{\boldsymbol{contact}}(\boldsymbol{x}) = \left(\boldsymbol{I}_{3\times 3} - \boldsymbol{n}(\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x})^T\right)\boldsymbol{f}_{\boldsymbol{circular}}(\boldsymbol{x}) \tag{A.8}$$

   - The desired circular dynamics are first expressed in cylindrical coordinates:

$$\begin{cases} \dot{r}_d = r_d - r \\ \dot{\theta}_d = \pi \\ \dot{z}_d = z_c - z \end{cases} \tag{A.9}$$

  with:

$$\begin{cases} r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \\ \theta = \arctan\left(\frac{y - y_c}{x - x_c}\right) \end{cases} \tag{A.10}$$

  where $\boldsymbol{x} = [x, y, z]^T$ and $\boldsymbol{x_c} = [x_c, y_c, z_c]^T$ denote the robot tool-tip position and the center of the desired circular motion respectively, while $\dot{r}_d$, $\dot{\theta}_d$, and $\dot{z}_d$, represent the desired radius, angle, and height dynamics. For the experiment, we set the desired radius r_d to 0.05 m while $\boldsymbol{x_c}$ is approximately positioned at the center of the wooden plate thanks to the optitrack markers attached to it.

  The desired circular dynamics are then converted back to Cartesian coordinates such that:

$$\boldsymbol{f_{circular}}(\boldsymbol{x}) = \begin{pmatrix} \dot{r}_d \cos(\theta) - r\dot{\theta}_d \sin(\theta) \\ \dot{r}_d \sin(\theta) + r\dot{\theta}_d \cos(\theta) \\ z_c - z \end{pmatrix} \tag{A.11}$$

- The desired force profile in contact $F_d(\boldsymbol{x})$ is implemented such that:

$$F_d(\boldsymbol{x}) = \begin{cases} F_T & \mu_F \geq \epsilon_F \ \wedge \ \Gamma(\boldsymbol{x}) \leq \epsilon_\Gamma \\ F_{T,min} & \mu_F < \epsilon_F \ \wedge \ \Gamma(\boldsymbol{x}) \leq \epsilon_\Gamma \\ 0 & \text{otherwise} \end{cases} \tag{A.12}$$

  where:

   - μ_F is the mean value of the measured normal force over a sliding window of n samples (set to 10) while $\epsilon_F \geq 0$ is a force threshold (set to 3 N).

   - $\epsilon_\Gamma \geq 0$ is a tolerance margin on the surface location (set to 0.05 m)

   - $F_{T,min}$ is the target force close to contact (set to 3 N). It ensures that the contact with the surface happens and contributes to reduce the impact at the contact.

   - F_T is the target force in contact. It takes a different value among 10, 15 and 20 N, every time the robot comes in contact with the surface (see Figure 3.6a).

- The desired quaternion is computed as explained in appendix A.1 with $\kappa = 5$ in Eq. A.3.

- The energy tank-based passivity correction is used with $s_+ = 60$ J and $\delta_s = 0.1 s_+$.

A.3 Bi-manual Reaching and Grasping Task

- A grasping state gs is defined based on the distance to the object and the measured forces:

$$gs = \begin{cases} \text{grasped} & \mu_F^L \geq \varepsilon_F \wedge \mu_F^R \geq \varepsilon_F \wedge e_C \leq \varepsilon_C \wedge e_D \leq \varepsilon_D \\ \text{almost grasped} & \neg(\mu_F^L \geq \varepsilon_F \wedge \mu_F^R \geq \varepsilon_F) \wedge e_C \leq \varepsilon_C \wedge e_D \leq \varepsilon_D \\ \text{not grasped} & \text{otherwise} \end{cases} \tag{A.13}$$

  with:

$$e_C = \|\boldsymbol{x^C} - \boldsymbol{x_o^C}\| \qquad e_D = \left(\boldsymbol{x^D} - \boldsymbol{x_o^D}\right)^{\boldsymbol{T}} \frac{\boldsymbol{x_o^D}}{\|\boldsymbol{x_o^D}\|} \tag{A.14}$$

  where:

  - μ_F^i is the mean value of the measured normal force over a sliding window of n samples (set to 10) for robot i with i=L, R while $\varepsilon_F \geq 0$ is a force threshold (set to 3 N).
  - e_C and e_D respectively denote the error to the object center position and distance vector while ε_C and ε_D are positive threshold values set to 0.1 m and 0.05 m respectively.

- The robots' nominal DS are implemented by setting the gain matrices $\boldsymbol{A_C} = 4\boldsymbol{I_{3\times3}}$ and $\boldsymbol{A_D} = 2\boldsymbol{I_{3\times3}}$ where $\boldsymbol{I_{3\times3}}$ is the identity matrix while the attractors are set to:

$$\boldsymbol{x_d^C} = \begin{cases} \boldsymbol{x^C} & gs = \text{grasped} \\ \boldsymbol{x_o^C} & \text{otherwise} \end{cases} \qquad \boldsymbol{x_d^D} = \boldsymbol{x_o^D} \tag{A.15}$$

- The desired force profile in contact $F_d(\boldsymbol{x^L}, \boldsymbol{x^R})$ is implemented such that:

$$F_d(\boldsymbol{x^L}, \boldsymbol{x^R}) = \begin{cases} F_T & gs = \text{grasped} \\ F_{T,min} & gs = \text{almost grasped} \\ 0 & gs = \text{not grasped} \end{cases} \tag{A.16}$$

  where $F_{T,min}$ is the target contact force when the object is about to be grasped (set to 3 N), while F_T is the target contact force when the object is grasped (set to 15 N).

- The desired quaternion for both robots is computed similarly to Appendix A.1 with $\kappa = 3$ and replacing $\Gamma(\boldsymbol{x})$ by e_D in Eq. A.3.

- The energy tank-based passivity correction is used with $s_+ = 4$ J and $\delta_s = 0.1 s_+$ for both robots.

A.4 Bi-manual Reaching, Grasping, and Manipulation Task

- The grasping state is computed alike Appendix A.3.

- The gain matrices of the robots' nominal DS $\boldsymbol{A_C}$ and $\boldsymbol{A_D}$ are the same than in Appendix A.3.

The attractors change to:

$$\boldsymbol{x_d^C} = \begin{cases} \boldsymbol{x_a^C} & gs = \text{grasped} \\ \boldsymbol{x_o^C} & \text{otherwise} \end{cases} \qquad \boldsymbol{x_d^D} = \begin{cases} \boldsymbol{x_a^D} & gs = \text{grasped} \\ \boldsymbol{x_o^D} & \text{otherwise} \end{cases} \tag{A.17}$$

where $\boldsymbol{x_a^C}$ and $\boldsymbol{x_a^D}$ are the desired center location and dimension vector at the attractor point. $\boldsymbol{x_a^C}$ is composed of a fixed component (set to $[-0.4, 0.45, 0.45]^T m$ corresponding to the desired location after lifting the object) and a variable offset modulated through a graphical interface. $\boldsymbol{x_a^D}$ is fixed to $l[0, -1, 0]^T$ where $l = 0.2m$ is the object's dimension along the desired grasping direction.

- The desired contact force profile $F_d(\boldsymbol{x^L}, \boldsymbol{x^R})$, quaternions, and energy tank-based passivity corrections are computed as in Appendix A.3.

B Appendix Chapter 4

For both experiments:

- The control strategy runs at $200Hz$.
- The DS impedance controller presented in section 2.2 is used with $d_1 = d_2 = d_3 = 150$.
- The orientation controller presented in section 2.3 is used with $\boldsymbol{K_\zeta} = \text{diag}([15, 15, 15])$ and $\boldsymbol{D_\omega} = \text{diag}([2, 2, 2])$.
- The desired robot's end-effector orientation $\boldsymbol{Q_d}(\boldsymbol{x})$ is computed as in Appendix A.1 with $\kappa = 5$ and $k_\omega = 5$.

For the first experimental task evaluating the force adaptation on the non-flat surface under various conditions:

- The nominal DS $\boldsymbol{f}(\boldsymbol{x})$ is computed alike Eq. A.6 in Appendix A.2.
- The desired contact profile $F_d(\boldsymbol{x})$ is obtained similarly to Eq. A.12.

For the collaborative cleaning task:

- The attractor $\boldsymbol{x_{h,a}}$ of the homing task $\boldsymbol{f^1}(\boldsymbol{x})$ is set to $[-0.5, 0.0, 0.4]^T$.
- The cleaning task $\boldsymbol{f^2}(\boldsymbol{x})$ is computed as in Eq. A.6 in Appendix A.2. The difference concerns the attractor of the circular DS $\boldsymbol{x_c}$ which is decomposed in a fixed term approximately located at the center of the wooden plate and a variable offset updated based on the human interaction using the work in Khoramshahi et al. (2018).
- The beliefs of both tasks are updated based on Khoramshahi and Billard (2019).
- Task adaptation is activated when there is no contact with the surface. The contact condition is defined by $\mu_F \geq \varepsilon_F \ \wedge \ \Gamma(\boldsymbol{x}) \leq \varepsilon_\Gamma$ where $\varepsilon_F = 3N$ and $\varepsilon_\Gamma = 0.05m$.
- Force adaptation is enabled during the cleaning task when the tangential forces measured by the force-torque sensor remain below $\varepsilon_{F,t} = 15N$.

C Master Haptic Foot Interface

In the four-arm collaborative scenarios studied in chapter 5 and 6, the two supernumerary robotic arms are controlled through the feet of the human thanks to haptic foot interfaces developed by Jacob Hernandez Sanchez. The kinematic description of a foot platform is presented in Figure C.1, while the list of specifications can be found in Table C.1.

Similarly to section 2.2, we can define the dynamics of a master foot interface in its five-dimensional joint space:

$$\boldsymbol{M_M(\xi)\ddot{\xi} + C_M(\xi,\dot{\xi})\dot{\xi} + g_M(\xi) = \tau_{M,c} + \tau_{M,ext}} \tag{C.1}$$

where $\boldsymbol{\xi} = [x, y, \theta, \phi, \psi]^T \in \mathbb{R}^5$ denotes the platform's state with x, y, θ, ϕ, ψ representing respectively the x, y, pitch, roll and yaw coordinates along their corresponding linear $(x,\ y)$ and rotation axes (θ, ϕ, ψ), see Figure C.1. $\boldsymbol{M_M(\xi)} \in \mathbb{R}^{5\times5}$ is the mass matrix, $\boldsymbol{C_M(\xi,\dot{\xi})\dot{\xi}} \in \mathbb{R}^5$ the centrifugal and coriolis forces, and $\boldsymbol{g_M(\xi)} \in \mathbb{R}^5$ represents the gravity torques. Furthermore, $\boldsymbol{\tau_{M,c}} \in \mathbb{R}^5$ and $\boldsymbol{\tau_{M,ext}} \in \mathbb{R}^5$ represent the control and measured external torques respectively. In chapter 5, $\boldsymbol{\tau_{M,c}}$ is composed of a spring and damping terms with an open-loop rendering of desired reflected torques fed-back from the environment:

$$\boldsymbol{\tau_{M,c} = \tau_d - K_M\xi - D_M\dot{\xi}} \tag{C.2}$$

Feature	y	x	θ	ϕ	ψ
Nominal F/T $[N\ Nm]$	22	19	11.5	22.2	22.2
Range of Motion $[m\ deg]$	0.195	0.180	55	100	240
Resolution F/T $[mN\ mNm]$	150	150	5	5	2
Res. Motion $[um\ deg]$	28.5	14.03	0.0018	0.0018	0.0017
Shoe Size	36-44 European				

Table C.1 – Specifications of the master foot interface (©Jacob Hernandez Sanchez)

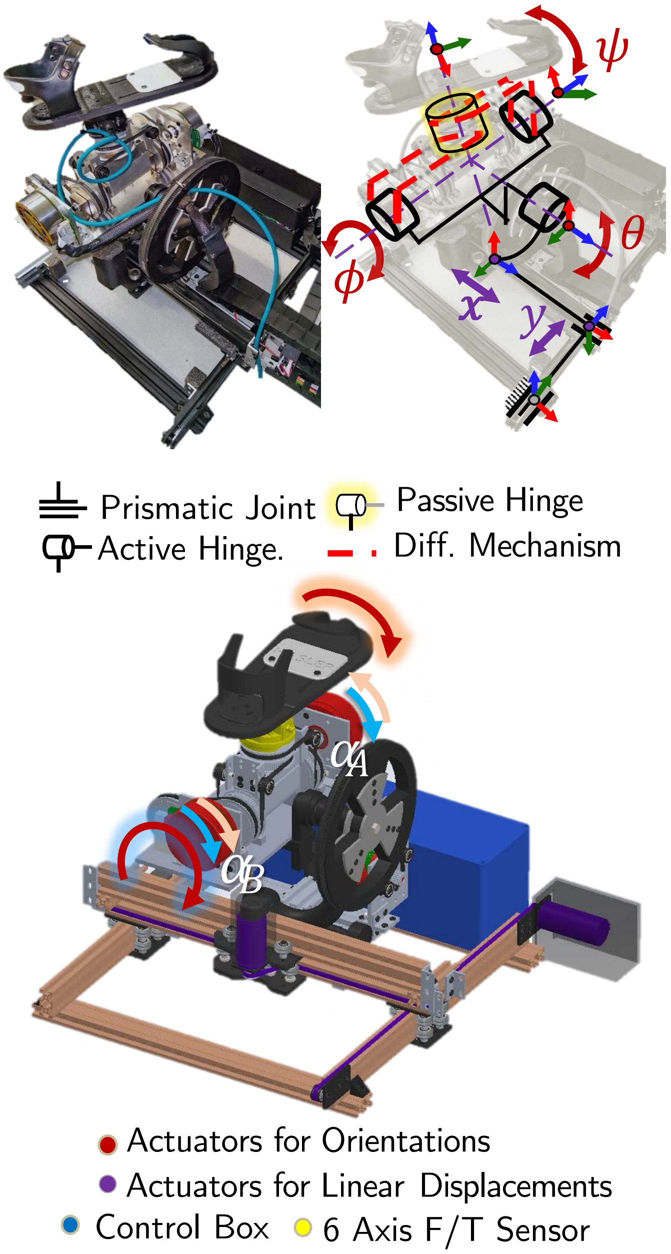


Figure C.1 – Kinematic model of the 5-DoF haptic foot interface (©Jacob Hernandez Sanchez).

where $\boldsymbol{\tau_d} \in \mathbb{R}^5$ are the desired reflected torques, while $\boldsymbol{K_M} \in \mathbb{R}^{5\times 5}$ and $\boldsymbol{D_M} \in \mathbb{R}^{5\times 5}$ are the stiffness and damping (positive definite diagonal) matrices pushing the foot towards $\boldsymbol{\xi} = \mathbf{0}_{5\times 1}$. In chapter 6, the calculation of the control torques $\boldsymbol{\tau_{M,c}}$ is more sophisticated and includes several components such as transparency torques to compensate the dynamics of the platform (i.e., coriolis, inertia, and gravity), as well as haptic torques combining grasping feedback (for the gripper), variable virtual impedance, and guidance haptic cues.

D Appendix Chapter 5

D.1 Experimental Details

- For all participants, each control strategy runs at a frequency of 200 Hz (i.e., $dt = 0.005$ s).
- For both robotic arms, the control gains were set to
  - $d_1^i = d_2^i = d_3^i = 150$.
  - $\boldsymbol{K_\zeta^i} = \text{diag}([15, 15, 15])$.
  - $\boldsymbol{D_\omega^i} = \text{diag}([2, 2, 2])$.
- For both foot interfaces, we set the control gains to:
  - $\boldsymbol{K_M^i} = \text{diag}([0, 0, 0, 3500, 1000])$.
  - $\boldsymbol{D_M^i} = \text{diag}([3, 5, 20, 60, 30])$.
- To compute the axis-angle orientation error $\tilde{\boldsymbol{\zeta}}^{\boldsymbol{i}}$ between $\boldsymbol{z_o^i}$ and $\boldsymbol{z_{o,d}^i}$, we proceed in two steps. We first use the Rodrigues' rotation formula to compute the rotation matrix $\tilde{\boldsymbol{R}}$ aligning $\boldsymbol{z_o^i}$ with $\boldsymbol{z_{o,d}^i}$:

$$\begin{cases} c = \boldsymbol{z_o^i}^{\boldsymbol{T}} \boldsymbol{z_{o,d}^i} \\ \boldsymbol{k} = \boldsymbol{z_o^i} \times \boldsymbol{z_{o,d}^i} \\ s = \|\boldsymbol{k}\| \\ \boldsymbol{K} = [\boldsymbol{k}]_\times \\ \tilde{\boldsymbol{R}} = \boldsymbol{I_{3\times 3}} + s\boldsymbol{K} + (1-c)\boldsymbol{K}^2 \end{cases} \tag{D.1}$$

  We then convert $\tilde{\boldsymbol{R}}$ into $\tilde{\boldsymbol{\zeta}}^{\boldsymbol{i}}$.

- The desired angular velocity $\boldsymbol{\omega_d^i}$ is decomposed into two terms:

$$\boldsymbol{\omega_d^i} = \boldsymbol{\omega_{d,\zeta}} + \boldsymbol{\omega_{d,h}} \tag{D.2}$$

- $\boldsymbol{\omega_{d,\zeta}}$ helps aligning $\boldsymbol{z_o^i}$ with $\boldsymbol{z_{o,d}^i}$. To this end, the axis-angle error is converted into a unit rotation quaternion $\boldsymbol{\tilde{Q}^i}$ in order to obtain a desired quaternion $\boldsymbol{Q_d^i} = \boldsymbol{\tilde{Q}^i} \star \boldsymbol{Q^i}$ similarly to section A.1. Using the relationship between angular velocity and quaternion time derivative, we obtain $\boldsymbol{\omega_{d,\zeta}}$ through:

  $$\boldsymbol{Q_\omega} = k_\omega \boldsymbol{Q^{i}}^{-1} \star \left(\boldsymbol{Q_d^i} - \boldsymbol{Q^i}\right) \tag{D.3}$$

  where k_ω is a positive gain set to 5, and $\boldsymbol{Q_\omega} = [0\ \boldsymbol{\omega_{d,\zeta}^{E^T}}]^T$. $\boldsymbol{\omega_{d,\zeta}^E}$ is $\boldsymbol{\omega_{d,\zeta}}$ expressed in the end-effector frame. To express it back in the world frame, we simply use the end-effector orientation matrix as in section A.1.

- $\boldsymbol{\omega_{d,h}}$ is the component commanded by the human to rotate the robots' end-effector:

  $$\boldsymbol{\omega_{d,h}} = \begin{cases} \omega_h^i \boldsymbol{z_o^i} & \text{strategy } U/U{+}F \\ \omega_h \boldsymbol{n^j} & \text{strategy } C/C{+}F \end{cases} \tag{D.4}$$

  where j is the id of the foot platform used to control both robots in the coordinated strategies.

- For control strategy U:
  - $\boldsymbol{A^i} = \text{diag}([2,2,2])$.
  - $\boldsymbol{\Upsilon_p^U} = \begin{bmatrix} 0 & \frac{2R_x}{r_y} & 0 & 0 & 0 \\ -\frac{2R_y}{r_x} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2R_z}{r_\theta} & 0 & 0 \\ 0 & 0 & 0 & \frac{2R_\omega}{r_\phi} & 0 \end{bmatrix}$, where $R_x = R_y = 0.4\ m$, $R_z = 0.3\ m$, $R_\omega = 1.5\ rad$ are desired ranges of motion on the robots' side, and $r_x = 0.195\ m$, $r_y = 0.18\ m$, $r_\theta = 48°$, and $r_\phi = 40°$ are ranges of motion on the foot interface.
  - $\boldsymbol{x_0^i} = [-0.5, 0.45, 0.2]^T$.
  - $\boldsymbol{z_{o,d}^L} = [0,-1,0]^T$ and $\boldsymbol{z_{o,d}^R} = [0,1,0]^T$.
  - $\boldsymbol{\Upsilon_F^U} = \alpha \begin{bmatrix} 0 & -s_x & 0 \\ s_y & 0 & 0 \\ 0 & 0 & s_\theta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, where $\alpha = 1$, $s_x = 1$, $s_y = 1$, and $s_\theta = 0.2$ are scaling factors.

- For control strategy $U{+}F$:
  - $F_T^{U,i} = 30\ N$.
  - $\varepsilon_F = 5\ N$ and $n = 10$.
  - $\boldsymbol{\Upsilon_p^U}$ is equal to above except $\alpha = 0.7$.

  - All the other variables as in control strategy U.

- For control strategy *C+F*:

  - $\boldsymbol{A_C} = \text{diag}([3,3,3])$ and $\boldsymbol{A_D} = \text{diag}([2,2,2])$.
  - $P = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.
  - $\boldsymbol{\Upsilon_F^C}$ is similar to above except $\alpha = 0.7$.
  - $F_T^{C,L} = F_T\Big(2 - \Lambda^+_{a,b}\big(d_1^R \boldsymbol{\dot{x}_d^C}^{\boldsymbol{T}} \boldsymbol{n^R} + F_T\big)\Big)$ and $F_T^{C,R} = F_T\Big(2 - \Lambda^+_{a,b}\big(d_1^L \boldsymbol{\dot{x}_d^C}^{\boldsymbol{T}} \boldsymbol{n^L} + F_T\big)\Big)$ with $a = 10\ N$, b $= 15\ N$, $F_T = 30\ N$ (see Eq. 6.43). The target force for each robot is modulated to account for the balance between the motion force in the grasping direction (i.e., $d_1^i \boldsymbol{\dot{x}_d^C}^{\boldsymbol{T}} \boldsymbol{n^i}$) and the nominal contact force F_T of the other robot. When moving along the grasping direction, one robot would have its motion and target-contact forces opposite to each other. As a result, the applied contact force would reduce and the grasp might be broken. To limit this effect, we modulate the target force of the other robot to apply more force (up to twice the nominal one). This keeps the object within the robots' grasp, without reducing the robots' velocity.
  - $v_{h,+} = 0.3\ m/s$ and $\omega_{h,+} = 1.5\ rad/s$.
  - The right foot is used for all participants to control both robots in coordination: $j = R$.
  - $\boldsymbol{\Upsilon_p^C} = \begin{bmatrix} 0 & \frac{2R_x}{r_y} & 0 & 0 & 0 \\ -\frac{2R_y}{r_x} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2R_z}{r_\theta} & 0 & 0 \\ 0 & 0 & 0 & \frac{2R_\omega}{r_\phi} & 0 \\ 0 & 0 & 0 & 0 & \frac{2R_{\dot{x}}}{r_\psi} \end{bmatrix}$, where $R_{\dot{x}} = 0.3\ m/s$, $r_\psi = 40°$, while the other variables are equal to before.
  - $\boldsymbol{x_o^C} = \boldsymbol{x_0^R} - \frac{\boldsymbol{x_0^D}}{2}$.
  - $\boldsymbol{\Upsilon_F^C} = \alpha \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & s_\psi & 0 \end{bmatrix}$, where $\alpha = 0.7$ and $s_\psi = 1$.
  - $\varepsilon_D = \|\boldsymbol{x_0^D}\| + 0.2\ m$ and $\varepsilon_x = 0.07\ m$.

- $\boldsymbol{\Omega}_{\boldsymbol{F}}^{\boldsymbol{C}} = k \begin{bmatrix} 0 & \text{-}s_x & 0 \\ s_y & 0 & 0 \\ 0 & 0 & s_\theta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, where $k = 200$ while $s_x = 1$, $s_y = 1$, and $s_\theta = 0.2$.

- For control strategy C:
  - $\boldsymbol{\Upsilon}_{\boldsymbol{F}}^{\boldsymbol{C}}$ is equal to above except $\alpha = 1$.
  - All the other variables as in control strategy C+F.

D.2 Results of the Statistical Studies

Ease of Use (Cronbach $\alpha = 0.9481$)

1. *"Overall, it was easier to execute the 4Armed task (w.r.t the other modes)."*
2. *"Overall, it was more tiresome to execute the 4Armed task (w.r.t the other modes)"* **R**
3. *"It was easier to move the robotic arms."*
4. *"It was easier to manipulate the object."*
5. *"It was easier to move the object."*
6. *"It was easier to rotate the object."*
7. *"Overall, I felt I was improving the performance as the trials progressed."*
8. *"This system was easy to use."*
9. *"This system was simple to use."*
10. *"This system was user friendly (easy to understand)."*
11. *"This system was flexible."*
12. *"This system was effortless."*
13. *"I could use this system without written instructions."*
14. *"I did not notice any inconsistencies as I used it."*
15. *"Both occasional and regular users would like it."*
16. *"I could recover from mistakes quickly and easily."*
17. *"I could use it successfully every time."*

Usefulness

1. *"Overall, this strategy was useful."*

Predictability

1. *"The robots responded to my motion inputs in a predictable way."*

Fluency (Cronbach $\alpha = 0.8889$)

1. *"The robots and I worked fluently together as a team."*
2. *"The robots contributed to the fluency of the interaction."*

Trust (Cronbach $\alpha = 0.8462$)

1. *"I trusted the robot to do the right thing at the right time."*
2. *"The robots were trustworthy."*

Goal Perception

1. *"The robots perceived accurately what my goals were."*

Robot Intelligence (Cronbach $\alpha = 0.9003$)

1. *"The robots were intelligent."*
2. *"The robots were able to independently make decisions throughout the task."*
3. *"The robots had an understanding of the task."*
4. *"The robots had an understanding of my goal during the task."*

Satisfaction (Cronbach $\alpha = 0.9094$)

1. *"Overall, I preferred this strategy."*
2. *"I am satisfied with the 4A system."*
3. *"This system was fun to use."*
4. *"This system worked the way I wanted it to work."*

Figure D.1 – Subjective questionnaire to evaluate the fluency of shared Control (Hoffman (2019)) (©Jacob Hernandez Sanchez). R means reverse scale. The Cronbach's Alpha tests the internal consistency in the categories featuring multiple questions. One can notice the high internal consistency as $\alpha > 0.84$.

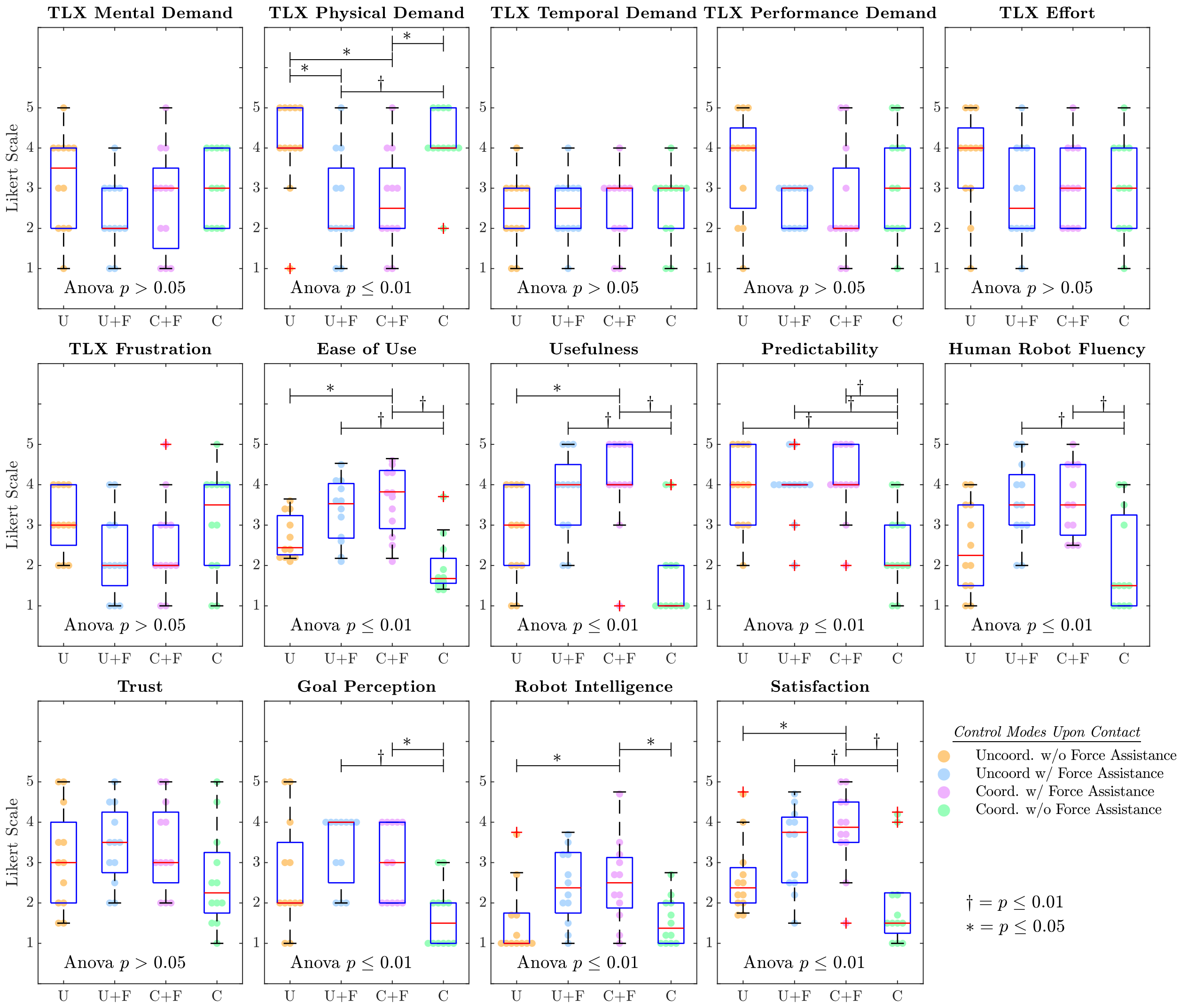


Figure D.2 – Study I. NASA Task Load Index (**the lower the better**) and human-robot fluency (**the higher the better**) and outcomes of the shared control for a four-arm manipulation (©Jacob Hernandez Sanchez). Twelve participants were involved in this study. The Tukey plots are overlaid on the data points for clarity. The type of statistical test (one-Way Anova with repeated measures or Kruskal-Wallis, depending on the normality of the data) is indicated in each subplot with its p-value. Furthermore, the pair-wise significant differences under 1% and 5% are indicated with the symbols explained in the legend of the figure.

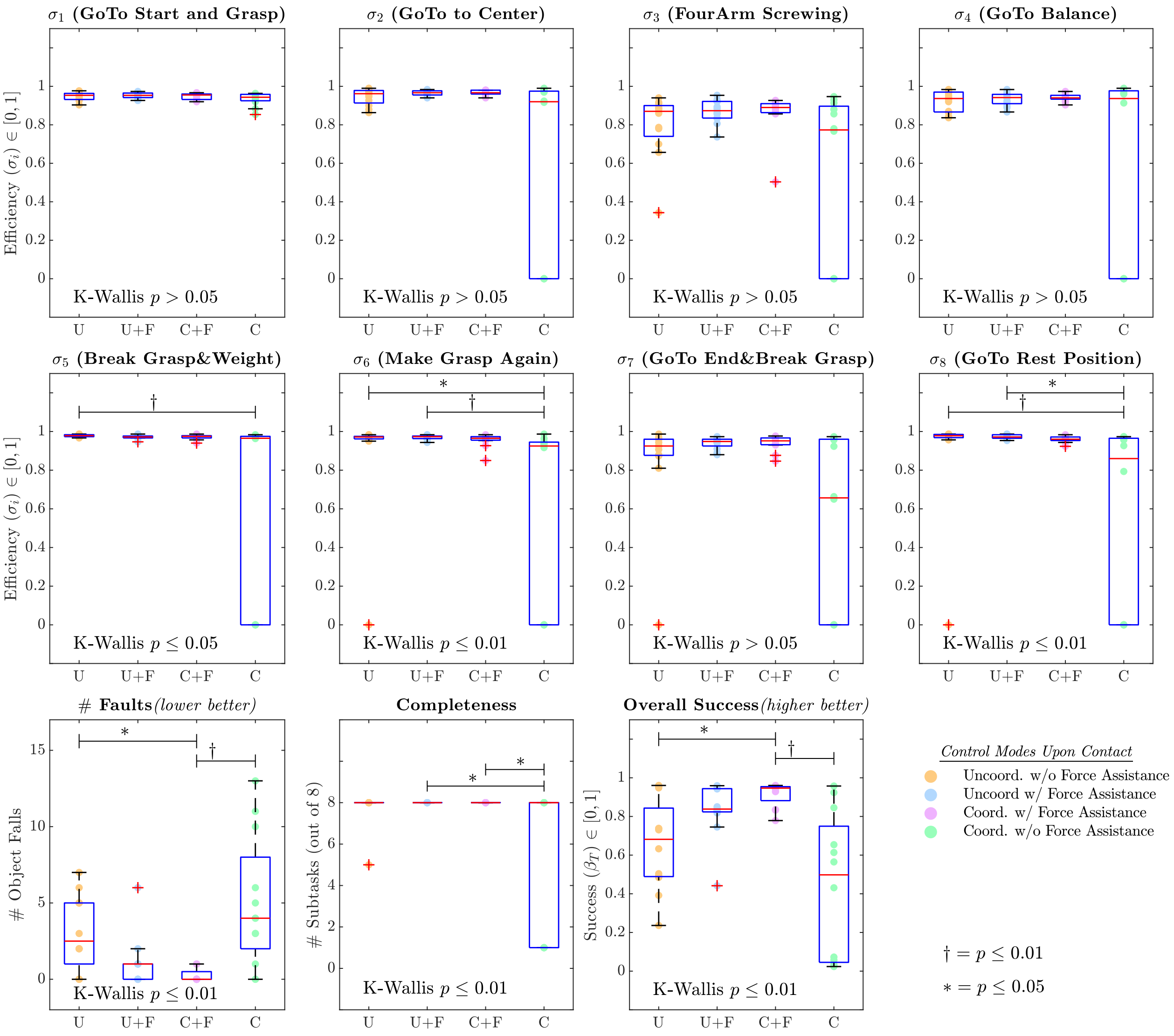


Figure D.3 – Study I. Objective performance metrics (©Jacob Hernandez Sanchez). Relative efficiency of each sub-task (**the higher the better**), number of object falls (**the lower the better**), completeness and overall success of the shared four-arm manipulation (**the higher the better**). The Tukey plots are overlaid on the data points for clarity. The type of statistical test (one-Way Anova with repeated measures or Kruskal-Wallis, depending on the normality of the data) is indicated in each subplot with its p-value. Furthermore, the pair-wise significant differences under 1% and 5% are indicated with the symbols explained in the legend of the figure.

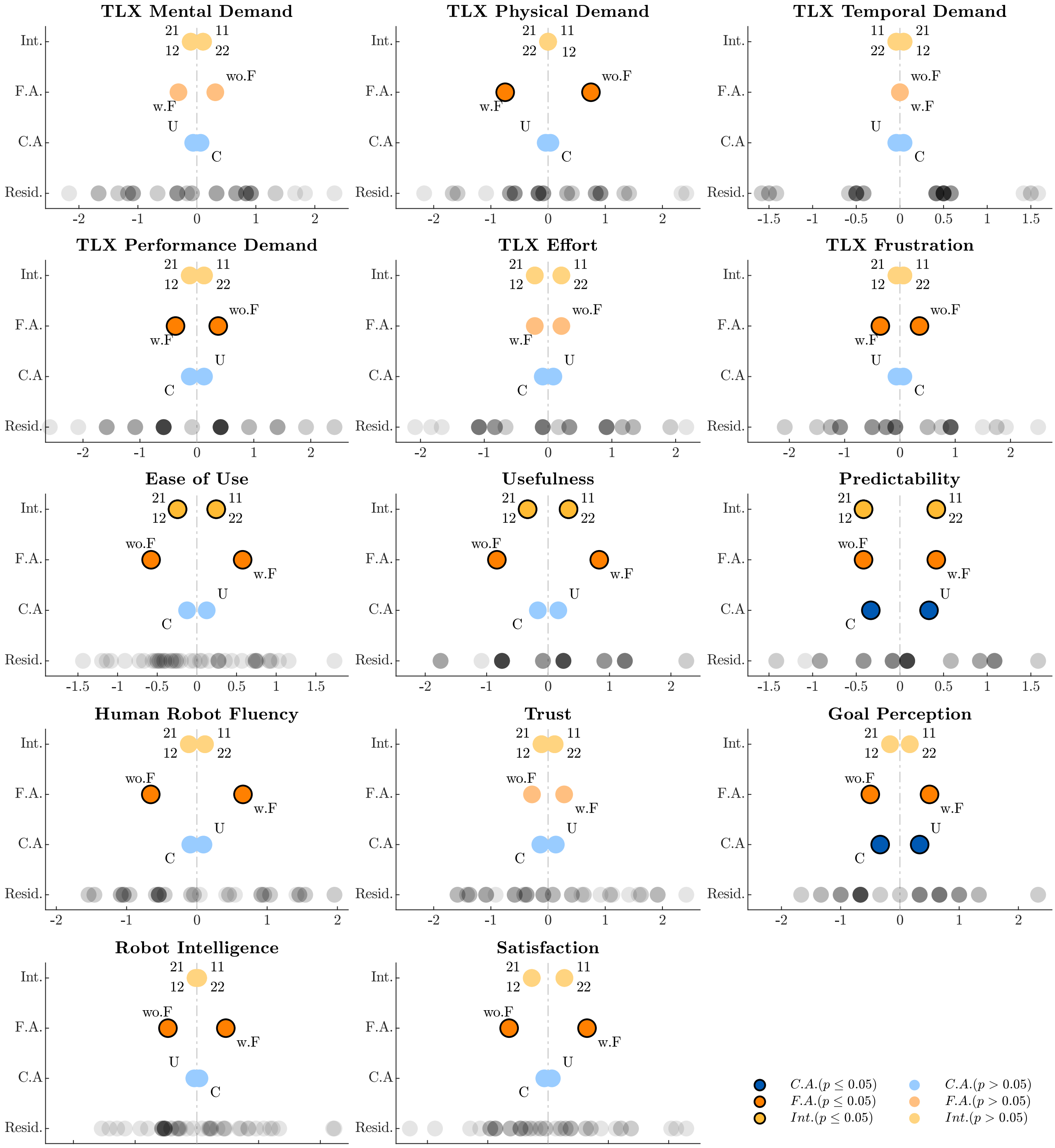


Figure D.4 – Study II. Results of the n-Way ANOVA used to test statistical significance among the assistance modalities ($F.A$, $C.A$) and to consider their interactions ($F.A * C.A$) (©Jacob Hernandez Sanchez). We model the main individual and interaction effects as constant coefficients models for normally distributed data. The plot shows the dot-plots of the models for the subjective metrics of task load index and shared control fluency and outcomes. One can observe the relative effect of the presence/absence force assistance ($F.A$) and coordination ($C.A$), as well as their interaction. Statistical significance is highlighted in bold color with black edge. Residuals are faded and depicted in the first line of the dot-plots.

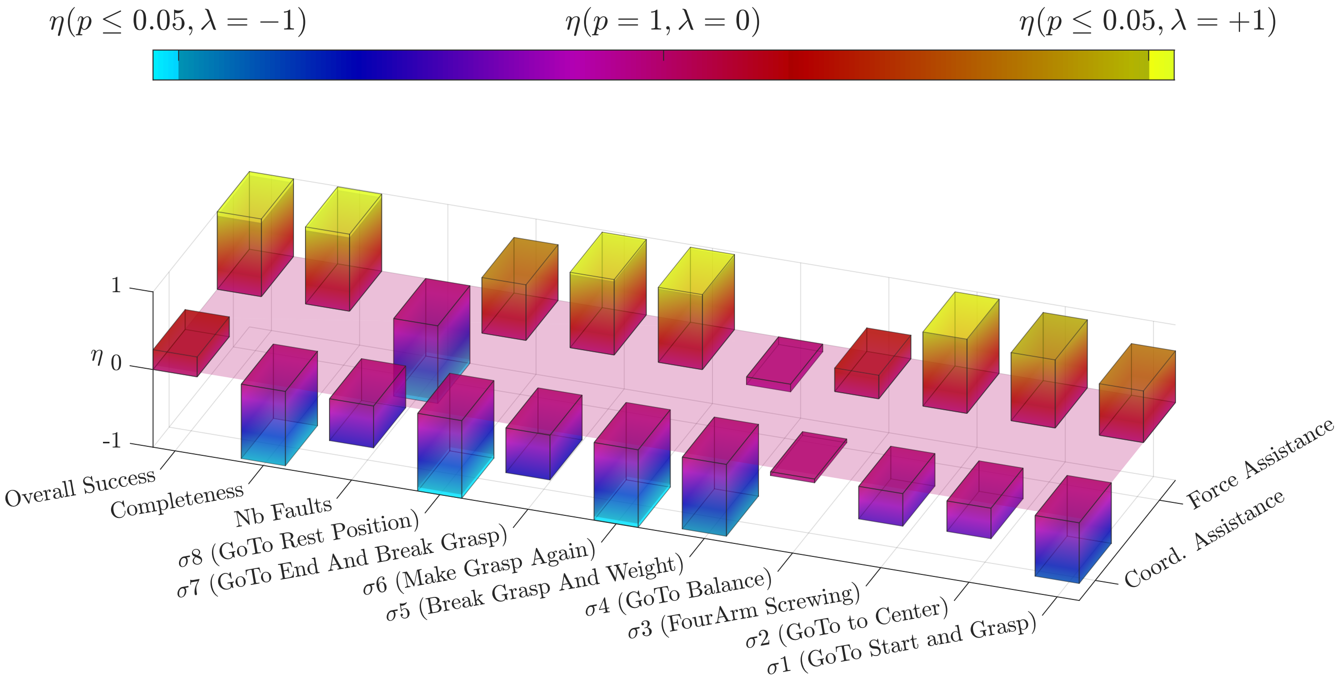


Figure D.5 – Study II. 3D Bar-plot to show the importance of each assistance modality in the objective metrics of efficiency per sub-task, number of object falls, completeness and overall success (©Jacob Hernandez Sanchez). The results are based on a Friedman non-parametric analysis made twice, using each time either force or coordination assistance as a column effect (factor). The interaction cannot be tested with the Friedman test. *Note that the bars' height does not correspond to the magnitude of the effect, as it is not modelled in the non-parametric analysis*. Indeed, the bars' height represents the function $\eta = \lambda(1-p)$, where p is the p-value of the χ^2 Test. Moreover, $\lambda \in [-1, 1]$ is an integer describing the effect of the factor. Specifically, $\lambda = 1$ when the *presence* of the factor has a positive impact in the result, $\lambda = -1$ when the effect is negative, while $\lambda = 0$ when neutral. This sign comes from the comparison between the mean ranks (from the Friedman Test) associated with each factor's group. If the mean rank of *(absence of assistance)* < *(presence of assistance)*, then the effect is positive, and vice versa. Finally, when the p-value is less or equal to 0.05, the extreme of the color bars is highlighted.

E Appendix Chapter 6

E.1 Camera-Based Tool Tip Tracking Strategy

As mentioned in section 6.3.4.4, a camera-based tracking strategy is used to derive the tools' tip position. The endoscope/camera image is indeed processed to detect color markers attached at the tips' base. Two markers are used for each tool, to figure out the tool tip position and direction. The implementation of the strategy is done with the well known OpenCV library, through simple image processing techniques.

Let us first refer as marker 1 and 2 the color markers attached to the tool, marker 1 being the closest to the tip. For each tool t, the following strategy is applied to marker $j = \{1, 2\}$:

1. Take the original image and apply a first color filter in the Hue-Saturation-Value (HSV) colorspace, by specifying min/max values for each attribute of the representation. The output of this step is a binary image where all pixels are either white or black based on whether or not the pixel is within the color range.

2. Remove the noise of the binary image using standard filtering techniques. Here we use a morphological opening filter with a 5×5 kernel size for the structuring element.

3. Find all contours on the filtered binary image.

4. Get the biggest contour (if they exist) and evaluate the area $a_{t,j}$ of the object it delimits.

5. Get the object's centroid position $\boldsymbol{p_{t,j}} \in \mathbb{R}^2$ (if the size is within a predefined range) with respect to the image center.

Once the centroid position of both markers is obtained, the tool tip position is computed through:

$$^{c}\boldsymbol{r_t} = \boldsymbol{A_r}\left(\boldsymbol{p_{t,1}} + \beta l_{t,1} \frac{\boldsymbol{p_{t,1}} - \boldsymbol{p_{t,2}}}{\|\boldsymbol{p_{t,1}} - \boldsymbol{p_{t,2}}\|}\right) \tag{E.1}$$

where $l_{t,1}$ is the estimated length of marker 1 in pixels calculated as:

$$l_{t,1} = \sqrt{\frac{a_{t,1}}{A_{t,1}}} L_{t,1} \tag{E.2}$$

with $a_{t,1}$ and $A_{t,1}$ the 2D areas of marker 1 expressed in pixels (in the image) and mm^2 (in reality) respectively. In Eq. E.1, β (set to 1.5) is a positive gain scaling $l_{t,1}$ to have the estimated tool tip position at the middle of the tool clamps, while $\boldsymbol{A_r} \in \mathbb{R}^{2\times 2}$ maps the relative tool tip position in pixels to a relative position expressed in normalized units in the camera frame C:

$$\begin{bmatrix} 0 & -\frac{2}{h} \\ \frac{2}{w} & 0 \end{bmatrix} \tag{E.3}$$

with w and h respectively denoting the image width and height.

E.2 Technical Details

- For all participants, each control strategy runs at a frequency of 200 Hz (i.e., $dt = 0.005$ s).
- For both robotic arms, the control gains of the joint impedance controller (section 6.3.2) were set to:
  - $\boldsymbol{K^i_R} = \text{diag}([700, 700, 700, 700, 500, 500, 300])$.
  - $\boldsymbol{D^i_R} = \text{diag}([30, 30, 30, 30, 20, 20, 10])$.
- The parameters of the QP-based inverse kinematics (section 6.3.3) were set to:
  - $\boldsymbol{H_1} = \boldsymbol{I_{7\times7}}$, $\boldsymbol{H_2} = 1000\boldsymbol{I_{3\times3}}$, $\boldsymbol{H_3} = 1000\boldsymbol{I_{4\times4}}$, $\boldsymbol{H_4} = 50\boldsymbol{I_{7\times7}}$.
  - $a_{rcm} = 40$, $a_{task,1} = a_{task,2} = 1$.
  - $\boldsymbol{q^i_-} = \begin{bmatrix} -2.8973 \\ -1.7628 \\ -2.8973 \\ -3.0718 \\ -2.8973 \\ -0.0175 \\ -2.8973 \end{bmatrix} + 6\pi/180 \qquad \boldsymbol{q^i_+} = \begin{bmatrix} 2.8973 \\ 1.7628 \\ 2.8973 \\ -0.0698 \\ 2.8973 \\ 3.7525 \\ 2.8973 \end{bmatrix} - 6\pi/180$
  - $\boldsymbol{\dot{x}_{E,+}} = -\boldsymbol{\dot{x}_{E,-}} = \sqrt{0.25^2/3} \times \boldsymbol{1_{3\times1}}$, $\boldsymbol{\dot{\omega}_{E,+}} = -\boldsymbol{\dot{\omega}_{E,-}} = 1.5 \times \boldsymbol{1_{3\times1}}$
  - $\lambda_{c_E} = 0.05$, $d_{c_E,s} = 0.03$, $d_{c_E,inf} = 0.06$, $r^i_{c_E} = 0.1$ for both robots.
  - $\lambda_{c_T} = 0.05$, $d_{c_T,s} = 0.03$, $d_{c_T,inf} = 0.06$, $r^i_{c_T} = 0.005$ for both robots.
  - $L^i = \begin{cases} 0.407m & \text{if } i = C \\ 0.599m & \text{otherwise} \end{cases}$

  - $\lambda_{c_W^k} = 0.05$, $d_{c_W^k,s} = 0$, $d_{c_W^k,inf} = 0.1 \forall k \in [1..5]$. $\boldsymbol{n}_{\boldsymbol{c}_W^1} = [0,0,1]^T, \boldsymbol{n}_{\boldsymbol{c}_W^2} = [1,0,0]^T, \boldsymbol{n}_{\boldsymbol{c}_W^3} = [\text{-}1,0,0]^T, \boldsymbol{n}_{\boldsymbol{c}_W^4} = [0,1,0]^T, \boldsymbol{n}_{\boldsymbol{c}_W^5} = [0,\text{-}1,0]^T$. In practice, workspace collision is only used for the robot holding the camera during the *operation* phase as the workspace of the gripper is already constrained by the position mapping. The camera tip workspace is delimited by specifying offset $o_{c_W^k}$ from the insertion point $\boldsymbol{x}_{\boldsymbol{d,0}}^{\boldsymbol{i}}$ along the $\boldsymbol{n}_{\boldsymbol{c}_W^k}$ directions: $o_{c_W^1} = \text{-}0.1m$, $o_{c_W^2} = o_{c_W^3} = \text{-}0.04m$, $o_{c_W^4} = \text{-}0.01m$, $o_{c_W^5} = \text{-}0.08m$.
  - $\lambda_{c_{MI}} = 0.05$, $d_{c_{MI},s} = 0.02$, $d_{c_{MI},inf} = 0.04$ for both robots. This constraint is only enabled during the *operation* phase.
  - $\dot{\boldsymbol{q}}_+^{\boldsymbol{i}} = \text{-}\dot{\boldsymbol{q}}_-^{\boldsymbol{i}} = 0.2 \times \begin{bmatrix} 2.175 \\ 2.175 \\ 2.175 \\ 2.175 \\ 2.61 \\ 2.61 \\ 2.61 \end{bmatrix}$.
  - $\delta_{rcm,+} = \text{-}\delta_{rcm,-} = \mathbf{1_{3\times1}}$.
  - $\delta_{task,+} = \text{-}\delta_{task,-} = \mathbf{1_{3\times1}}$.
  - $c_+ = 1$.

- The parameters used to compute the desired camera motion dynamics (section 6.3.4.3) are set to:

  - $\boldsymbol{\delta}_{\boldsymbol{d,0}}^{\boldsymbol{C}} = [0,0,\text{-}0.03]^T\ m$
  - $a_0 = 1$.
  - $\boldsymbol{\Upsilon_C} = \begin{bmatrix} 0 & \frac{2}{r_y} & 0 & 0 & 0 \\ \frac{2}{r_x} & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{-}\frac{2}{r_\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{-}\frac{2}{r_\psi} \\ 0 & 0 & 0 & \frac{2}{r_\phi} & 0 \end{bmatrix}$ where $r_x = 0.195m$, $r_y = 0.18m$, $r_\theta = 20°$, $r_\phi = 30°$, $r_\psi = 35°$ are the range of motions of the foot platforms.
  - $\boldsymbol{A_C} = \begin{cases} \text{diag}([0.03, 0.015, 0.03, 1, 1]) & \text{for task } T1 \\ \text{diag}([0.03, 0.015, 0, 1, 1]) & \text{otherwise} \end{cases}$
  - $\boldsymbol{\xi}_{\boldsymbol{dz},-}^{\boldsymbol{C}} = [\text{-}0.02, \text{-}0.03, \text{-}7, 0, \text{-}5]^T$.
  - $\boldsymbol{\xi}_{\boldsymbol{dz},+}^{\boldsymbol{C}} = [0.02, 0.03, 3, 0, 15]^T$.
  - ${}^{\boldsymbol{E}}\boldsymbol{R_C} = \begin{bmatrix} 0 & \text{-}1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

- The parameters used for camera assistance (section 6.3.4.4) were set to:
  - $\sigma_{a,th} = 0.5$.
  - $a_C = 0.07$.
  - $a_{b,1} = 2$, $r_- = 0.05$, $r_+ = 0.2$.
  - $a_{b,2} = 0.5, \gamma_{b,2} = 4$.
  - $a_{b,3} = 0.25$.
- The parameters used to compute the desired gripper motion dynamics are set to:
  - $\boldsymbol{\delta^G_{d,0}} = [0, \text{-}0.1, 0.12]^T\ m$
  - $\boldsymbol{\Upsilon_G} = \begin{bmatrix} 0 & \text{-}\frac{2}{r_y} & 0 & 0 & 0 \\ \frac{2}{r_x} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{r_\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{-}\frac{2}{r_\psi} \\ 0 & 0 & 0 & \frac{2}{r_\phi} & 0 \end{bmatrix}.$
  - $\boldsymbol{A_G} = \text{diag}([0.09, 0.09, 0.1, 1, 1])$.
  - $\boldsymbol{\xi^G_{dz,-}} = [0, 0, 0, 0, \text{-}15]^T$.
  - $\boldsymbol{\xi^G_{dz,+}} = [0, 0, 0, 0, 5]^T$.
  - $a_G = a_{G,0} + a_{G,1} \exp\left(\frac{\text{-}(\|\boldsymbol{x^G_d} - \boldsymbol{x^G}\|}{2a^2_{G,2}}\right)$ with $a_{G,0} = a_{G,1} = 4$, $a_{G,2} = 0.07$
- The parameters for the reference admittance velocity computation (section 6.3.4.6) are set to:
  - $F^i_{ext,dz} = \begin{cases} 5N & \text{if } i = C \\ 12N & \text{otherwise} \end{cases}$
  - $\boldsymbol{M^i_a} = \boldsymbol{I_{3\times3}}$, $\boldsymbol{D^i_a} = 200\boldsymbol{I_{3\times3}}$ for both robots.
- The fulcrum scaling gain a_δ is set to $4/L^i$.
- The parameters used to compute the task modulation gain α^i (section 6.3.4.8) are set to:
  - $a_{s,1} = 1000, a_{s,2} = 1.2, P_d = 2W$.
  - $s_+ = 0.8$.
- The parameters for the guidance wrench computation (section 6.3.5.1) are set to:
  - $a_{F_{d,0}} = 200$.
  - $F_T = 7N$, $d_- = 0m$, $d_+ = 0.01m$.

  - $\tau_T = 0.4Nm, \delta_{q(7)} = 0.2rad$.
  - $\boldsymbol{A_{F_{d,m}}} = \text{diag}([0,0,0.4])$. The measured contact forces are only reflected along the $\boldsymbol{z_W}$ direction for the gripper. The measured force torque sensor is mounted between the end-effector and gripper plates.

- The parameters for the reference guidance torques computation are set to:
  - $\boldsymbol{\Omega_C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$.
  - $\boldsymbol{\Omega_G} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$.
  - $\boldsymbol{A_\tau} = \text{diag}([1,1,0.2,0,1])$.

E.3 Results of the Statistical Studies

Task	Action	Metrics(m)	$\bar{m}_1$	σ_1	$\tilde{m}_1$	$\bar{m}_2$	σ_2	$\tilde{m}_2$	TestType	TStat	Df	W	PValue
T1	Positioning	TLX Physical	2.75	1.36	2								
T1	Positioning	TLX Mental	3.25	1.66	3								
T1	Positioning	TLX Temporal	1.83	1.47	1								
T1	Positioning	TLX Performance	5.08	1.51	5.5								
T1	Positioning	TLX Effort	3.50	1.57	3								
T1	Positioning	TLX Frustration	2.33	1.37	2								
T2	**Positioning**	**TLX Physical**	**4.08**	**1.61**		**2.92**	**1.38**		**r-pttest**	**2.88**	**11**		**0.01**
T2	Positioning	TLX Mental	3.75	1.42		3.50	1.04		r-pttest	2.88	11		0.17
T2	Positioning	TLX Temporal	1.75	1.01		1.67	0.94		r-pttest	2.88	11		0.17
T2	Positioning	TLX Performance	4.50	1.66		4.92	1.32		l-pttest	2.88	11		0.17
T2	**Positioning**	**TLX Effort**	**4.08**	**1.93**		**3.42**	**1.61**		**r-pttest**	**2.88**	**11**		**0.04**
T2	**Positioning**	**TLX Frustration**	**3.08**	**1.71**		**2.42**	**1.11**		**r-pttest**	**2.88**	**11**		**0.02**
T2	**Grasping**	**TLX Physical**	**4.50**	**1.66**		**2.08**	**1.44**		**r-pttest**	**3.30**	**11**		**0.00**
T2	Grasping	TLX Mental	4.00	2.31		3.08	1.66		r-pttest	3.30	11		0.09
T2	Grasping	TLX Temporal	2.08	1.38		2.50	1.55		r-pttest	3.30	11		0.95
T2	Grasping	TLX Performance	4.00	2.08		4.50	1.44		l-pttest	3.30	11		0.22
T2	Grasping	TLX Effort	4.17	2.15		3.08	1.26		r-pttest	3.30	11		0.07
T2	Grasping	TLX Frustration	3.17	1.99		3.00	1.53		r-pttest	3.30	11		0.39
T3	**Positioning**	**TLX Physical**	**4.75**	**1.79**		**3.00**	**1.63**		**r-pttest**	**3.54**	**11**		**0.00**
T3	**Positioning**	**TLX Mental**	**5.50**	**1.44**		**3.50**	**1.85**		**r-pttest**	**3.54**	**11**		**0.00**
T3	**Positioning**	**TLX Temporal**	**3.08**	**1.80**		**1.67**	**0.75**		**r-pttest**	**3.54**	**11**		**0.00**
T3	Positioning	TLX Performance	4.17	1.72		4.83	0.80		l-pttest	3.54	11		0.10
T3	**Positioning**	**TLX Effort**	**4.92**	**1.71**		**3.67**	**1.70**		**r-pttest**	**3.54**	**11**		**0.01**
T3	**Positioning**	**TLX Frustration**	**4.00**	**1.87**		**2.75**	**1.36**		**r-pttest**	**3.54**	**11**		**0.01**
T3	**Grasping**	**TLX Physical**	**3.42**	**2.22**		**2.50**	**1.76**		**r-pttest**	**1.84**	**11**		**0.05**
T3	Grasping	TLX Mental	3.17	1.86		2.92	1.55		r-pttest	1.84	11		0.10
T3	Grasping	TLX Temporal	2.42	1.44		2.58	1.50		r-pttest	1.84	11		0.83
T3	Grasping	TLX Performance	4.67	1.49		4.83	1.14		l-pttest	1.84	11		0.17
T3	Grasping	TLX Effort	3.08	1.71		2.83	1.67		r-pttest	1.84	11		0.14
T3	Grasping	TLX Frustration	3.00	1.47		3.00	1.35		r-pttest	1.84	11		0.50
T4	Positioning	TLX Physical	5.00	1.41		4.08	1.75		r-pttest	1.61	11		0.07
T4	**Positioning**	**TLX Mental**	**5.42**	**1.50**		**3.75**	**1.48**		**r-pttest**	**1.61**	**11**		**0.00**
T4	Positioning	TLX Temporal			2.00			1.50	r-wil.sgn.rank			10	0.06
T4	**Positioning**	**TLX Performance**	**4.08**	**1.89**		**5.08**	**1.11**		**l-pttest**	**1.61**	**11**		**0.01**
T4	**Positioning**	**TLX Effort**	**4.83**	**1.77**		**4.00**	**1.53**		**r-pttest**	**1.61**	**11**		**0.04**
T4	**Positioning**	**TLX Frustration**	**4.08**	**1.89**		**2.92**	**1.50**		**r-pttest**	**1.61**	**11**		**0.03**
T4	**Grasping**	**TLX Physical**	**4.58**	**1.61**		**2.08**	**1.61**		**r-pttest**	**3.95**	**11**		**0.00**
T4	**Grasping**	**TLX Mental**	**3.92**	**1.89**		**2.83**	**1.62**		**r-pttest**	**3.95**	**11**		**0.04**
T4	Grasping	TLX Temporal	1.67	0.75		2.42	1.66		r-pttest	3.95	11		0.93
T4	**Grasping**	**TLX Performance**	**4.58**	**1.55**		**5.42**	**0.95**		**l-pttest**	**3.95**	**11**		**0.04**
T4	**Grasping**	**TLX Effort**	**4.33**	**2.05**		**3.25**	**2.05**		**r-pttest**	**3.95**	**11**		**0.01**
T4	**Grasping**	**TLX Frustration**	**3.25**	**1.36**		**2.33**	**1.18**		**r-pttest**	**3.95**	**11**		**0.02**

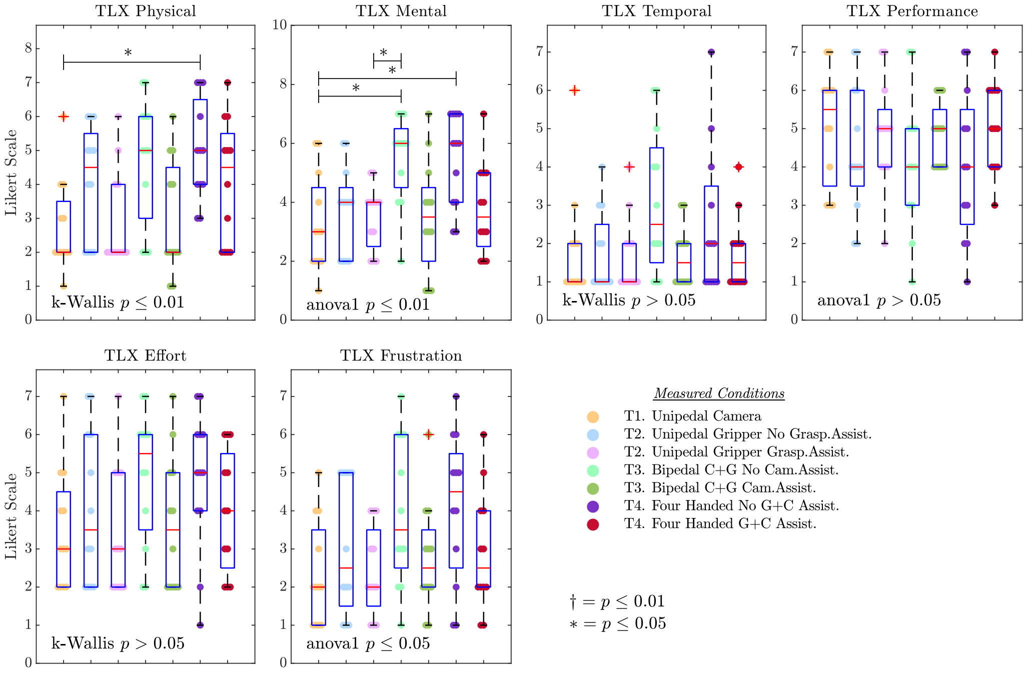
Table E.1 – Study I. Statistics for Perceived Task Load (©Jacob Hernandez Sanchez). The metrics for task $T1$ (baseline) are highlighted in yellow. The statistically-significant differences with a 95% confidence interval are highlighted in light-green, while the ones with 99% confidence in bold-green. The notation $\bar{m}_1 \pm \sigma_1$ and $\bar{m}_2 \pm \sigma_2$, refer to the mean and standard deviation of the responses for the condition 1 (i.e. without assistance WO) and condition 2 (i.e. with assistance W), respectively, while $\tilde{m}_1$ and $\tilde{m}_2$ represent the medians. *TStat* represents the statistic of the t-student distribution of the paired-t-test (*pttest*), *Df* are the degrees of freedom and W is the signed-rank statistic of the wilcoxon tests (*wil.sgn.rank*), which was performed using an exact method. Moreover, the prefixes *r-* and *l-* within the field *TestType*, represent the tail of the distribution for which the one-sided paired test was performed, respectively as a right and left tail.

Task	Metrics(m)	$\bar{m}_1$	σ_1	$\tilde{m}_1$	$\bar{m}_2$	σ_2	$\tilde{m}_2$	TestType	TStat	Df	W	PValue
T2	**Ease of Use**	**4.31**	**0.80**		**4.86**	**0.69**		**l-pttest**	**-2.47**	**11**		**0.02**
T2	Predictability	5.67	1.37		5.50	1.26		l-pttest	-2.47	11		0.67
T2	**Fluency**	**4.63**	**1.45**		**5.25**	**1.39**		**l-pttest**	**-2.47**	**11**		**0.03**
T2	Trust	5.67	1.39		5.67	1.28		l-pttest	-2.47	11		0.50
T2	Goal Perception	4.67	1.93		5.75	1.48		l-pttest	-2.47	11		0.07
T2	**Robot Intelligence**	**2.81**	**1.89**		**3.98**	**2.11**		**l-pttest**	**-2.47**	**11**		**0.01**
T2	**Satisfaction**	**4.90**	**0.92**		**5.58**	**1.02**		**l-pttest**	**-2.47**	**11**		**0.04**
T3	**Ease of Use**	**3.80**	**0.89**		**4.88**	**0.74**		**l-pttest**	**-4.18**	**11**		**0.00**
T3	**Predictability**	**4.83**	**1.62**		**5.67**	**1.11**		**l-pttest**	**-4.18**	**11**		**0.01**
T3	**Fluency**	**3.79**	**1.39**		**5.25**	**1.23**		**l-pttest**	**-4.18**	**11**		**0.00**
T3	Trust	5.33	1.52		5.75	1.03		l-pttest	-4.18	11		0.15
T3	**Goal Perception**	**4.33**	**1.97**		**5.83**	**0.90**		**l-pttest**	**-4.18**	**11**		**0.01**
T3	**Robot Intelligence**	**2.94**	**1.85**		**5.00**	**1.76**		**l-pttest**	**-4.18**	**11**		**0.00**
T3	**Satisfaction**	**4.29**	**1.56**		**5.73**	**0.75**		**l-pttest**	**-4.18**	**11**		**0.01**
T4	**Ease of Use**	**3.62**	**0.76**		**4.67**	**0.78**		**l-pttest**	**-3.92**	**11**		**0.00**
T4	**Predictability**	**5.33**	**1.25**		**6.00**	**0.82**		**l-pttest**	**-3.92**	**11**		**0.01**
T4	**Fluency**	**4.08**	**1.77**		**5.46**	**1.30**		**l-pttest**	**-3.92**	**11**		**0.00**
T4	Trust	5.21	1.61		5.71	1.07		l-pttest	-3.92	11		0.12
T4	**Goal Perception**	**4.58**	**1.98**		**5.50**	**1.19**		**l-pttest**	**-3.92**	**11**		**0.04**
T4	**Robot Intelligence**	**2.79**	**1.83**		**4.79**	**1.84**		**l-pttest**	**-3.92**	**11**		**0.00**
T4	**Satisfaction**	**4.08**	**1.80**		**5.67**	**0.81**		**l-pttest**	**-3.92**	**11**		**0.01**

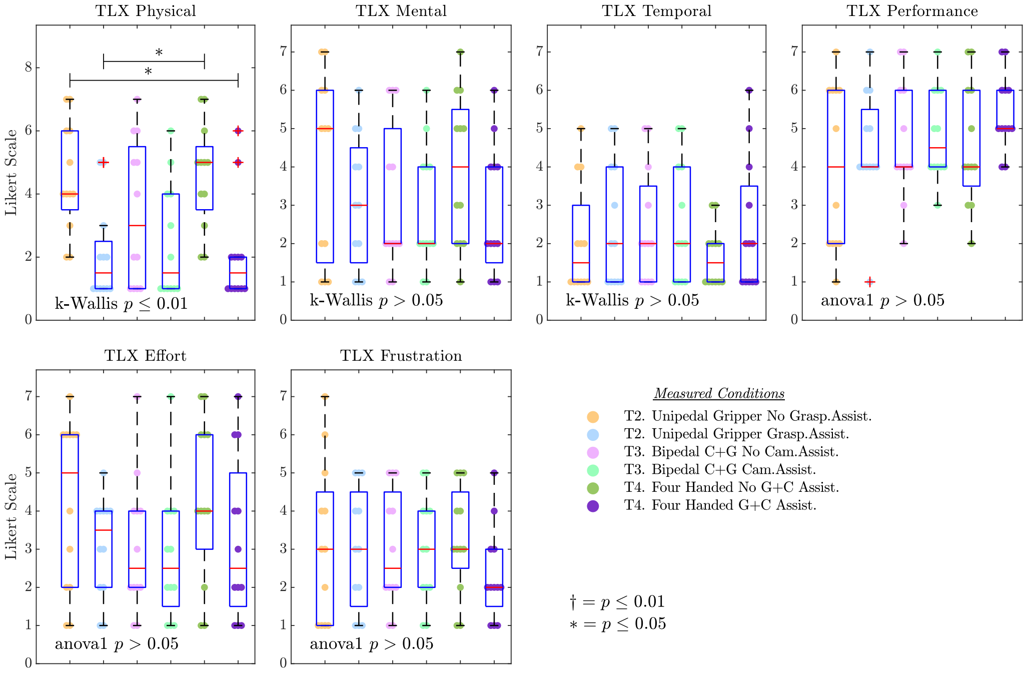
Table E.2 – Study I. Statistics for perceived fluency (©Jacob Hernandez Sanchez).

Task	Metrics(m)	Units	$\bar{m}_1$	σ_1	$\tilde{m}_1$	$\bar{m}_2$	σ_2	$\tilde{m}_2$	TestType	TStat	Df	W	PValue
T1	Average Position Error	mm	17.19	10.51	12.68								
T1	Average Angle Error	deg	4.33	1.90	4.93								
T1	Average Camera Cue Size	px	21.17	2.10	20.76								
T1	Average Time Per Target	**s**	22.72	6.17	22.61								
T1	Average Target Efficiency	%	92.43	2.06	92.46								
T2	Average Time Per Target	s			40.05			40.20	r-wil.sgn.rank			34	0.48
T2	Average Target Efficiency	%	48.90	30.24		54.83	22.51		l-pttest	-0.88	11		0.20
T2	Completeness	%	56.25	32.48		63.54	25.24		l-pttest	-0.88	11		0.18
T2	**Nb Falls**	**#**	**2.75**	**2.05**		**1.25**	**1.53**		**r-pttest**	**-0.88**	**11**		**0.03**
T2	Peg Transfer Success	%	39.54	29.09		49.82	24.21		l-pttest	-0.88	11		0.10
T3	**Average Time Per Target**	**s**	**60.27**	**19.92**		**47.01**	**11.93**		**r-pttest**	**2.74**	**11**		**0.01**
T3	Average Target Efficiency	%	44.81	25.94		56.82	19.02		l-pttest	2.74	11		0.09
T3	Completeness	%	54.17	28.11		66.67	19.98		l-pttest	2.74	11		0.10
T3	Nb Falls	#	1.00	0.71		1.08	1.44		r-pttest	2.74	11		0.57
T3	**Visual Task Success**	**%**	**82.21**	**10.71**		**99.25**	**0.64**		**l-pttest**	**2.74**	**11**		**0.00**
T3	Peg Transfer Success	%	41.14	25.90		51.62	22.42		l-pttest	2.74	11		0.16
T4	Average Time Per Target	s			75.71			68.63	r-wil.sgn.rank			35	0.25
T4	**Average Target Efficiency**	**%**	**47.55**	**32.06**		**60.29**	**23.08**		**l-pttest**	**-1.93**	**11**		**0.04**
T4	**Completeness**	**%**	**61.11**	**40.45**		**75.00**	**27.64**		**l-pttest**	**-1.93**	**11**		**0.05**
T4	**Nb Falls**	**#**	**2.08**	**1.75**		**1.00**	**1.41**		**r-pttest**	**-1.93**	**11**		**0.01**
T4	**Visual Task Success**	**%**	**69.36**	**11.43**		**99.79**	**0.37**		**l-pttest**	**-1.93**	**11**		**0.00**
T4	Percentage Time All Visible	%	33.99	8.52		24.72	8.92		l-pttest	-1.93	11		0.99
T4	**Peg Transfer Success**	**%**	**29.86**	**26.44**		**42.06**	**27.79**		**l-pttest**	**-1.93**	**11**		**0.02**

Table E.3 – Study I. Summary of statistics for objective metrics (©Jacob Hernandez Sanchez).



(a) Action of Positioning



(b) Action of Grasping

Figure E.1 – Study II. Combined effect of robotic assistance and type of task in the perceived mental load across the tasks (©Jacob Hernandez Sanchez). The type of statistical test (one-Way Anova or Kruskal-Wallis, depending on the normality of the data) is indicated in each subplot with its p-value. *For clarity, we omitted the post-hoc comparisons within-tasks from this analysis, as they were already reported in study I.*

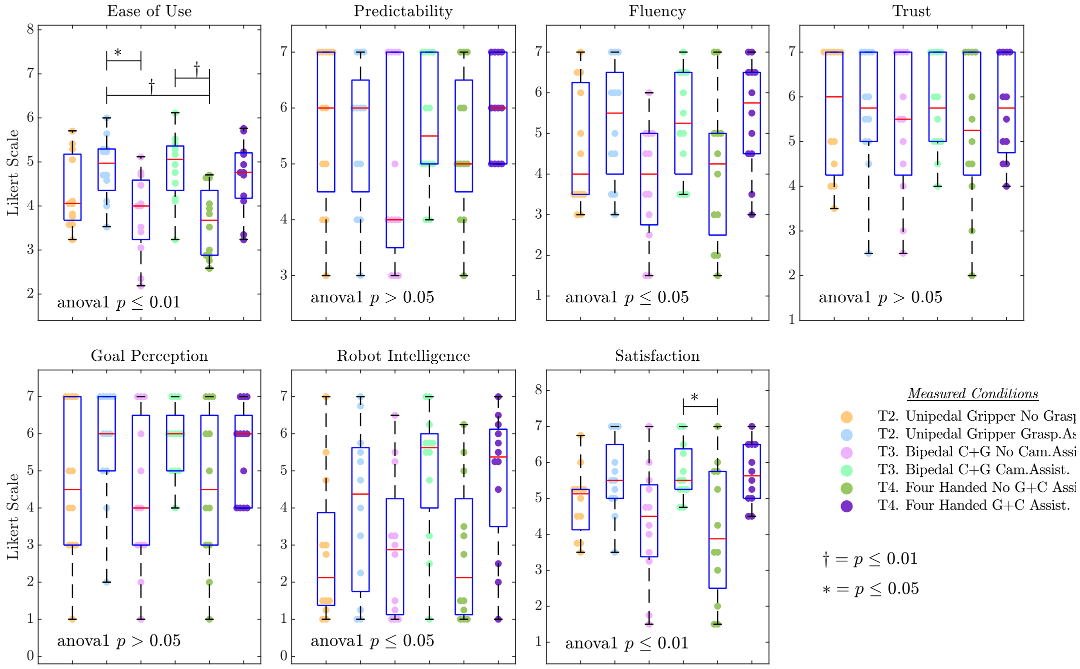
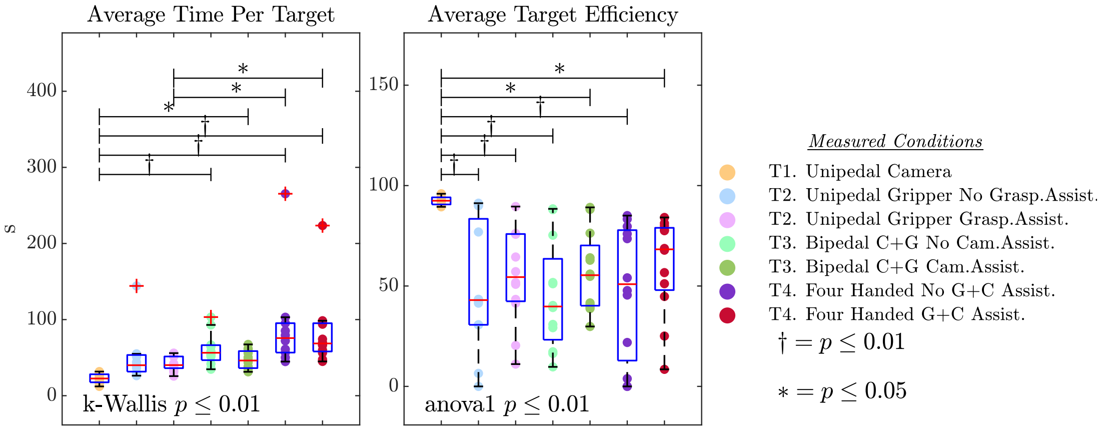
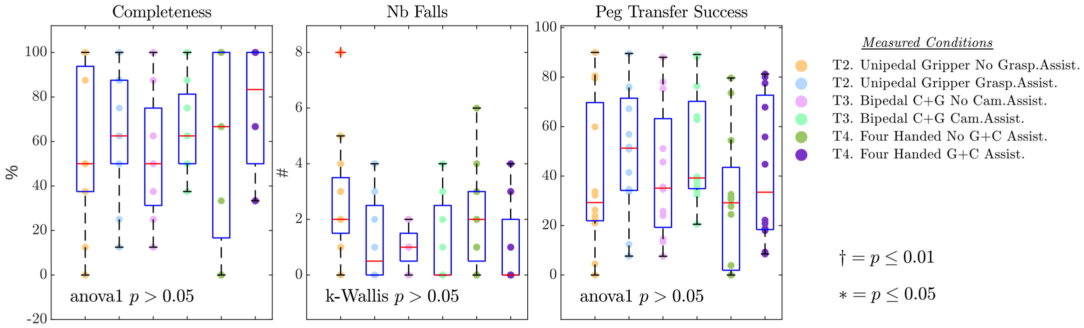


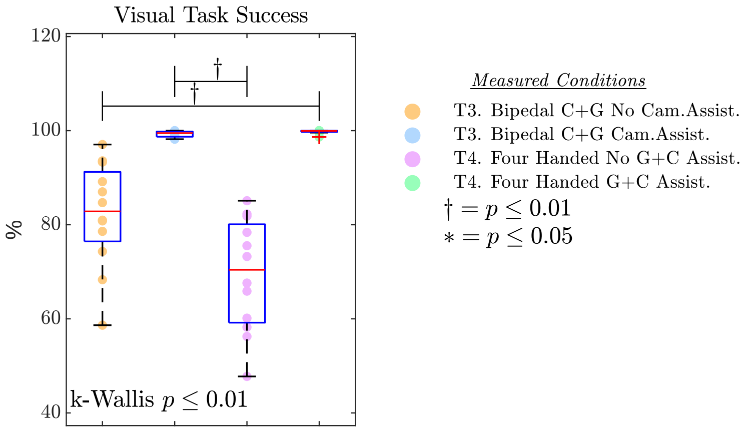
Figure E.2 – Study II. Combined effect of robotic assistance and type of task in the perceived fluency across the tasks (©Jacob Hernandez Sanchez).



(a) Comparison $T1 \sim T4$



(b) Comparison $T2 \sim T4$



(c) Comparison $T3 \sim T4$

Figure E.3 – Study II. Combined effect of robotic assistance and type of task in comparable objective metrics across the tasks (©Jacob Hernandez Sanchez).

Bibliography

Abbink, D. and van der Helm, F. (2004). Force perception measurements at the foot. In *2004 IEEE Int. Conf. Syst. Man Cybern. (IEEE Cat. No.04CH37583)*, volume 3, pages 2525–2529. IEEE.

Abbink, D. A. and Mulder, M. (2010). Neuromuscular analysis as a guideline in designing shared control. In *Advances in haptics*. IntechOpen.

Abdi, E., Burdet, E., Bouri, M., and Bleuler, H. (2015). Control of a supernumerary robotic hand by foot: An experimental study in virtual reality. *PloS one*, 10(7):e0134501.

Abdi, E., Burdet, E., Bouri, M., Himidan, S., and Bleuler, H. (2016). In a demanding task, three-handed manipulation is preferred to two-handed manipulation. *Scientific Reports*.

Abi-Farraj, F., Pacchierotti, C., and Giordano, P. R. (2018). User evaluation of a haptic-enabled shared-control approach for robotic telemanipulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE.

Abi-Farraj, F., Pedemonte, N., and Giordano, P. R. (2016). A visual-based shared control architecture for remote telemanipulation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4266–4273. IEEE.

Aghakhani, N., Geravand, M., Shahriari, N., Vendittelli, M., and Oriolo, G. (2013). Task control with remote center of motion constraint for minimally invasive robotic surgery. In *2013 IEEE international conference on robotics and automation*, pages 5807–5812. IEEE.

Allen, M. P., Evans, G. T., Frenkel, D., and Mulder, B. (1993). Hard convex body fluids. *Advances in chemical physics*, 86:1–166.

Amanhoud, W., Khoramshahi, M., and Billard, A. (2019). A dynamical system approach to motion and force generation in contact tasks. In *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany.

Amanhoud, W., Khoramshahi, M., Bonnesoeur, M., and Billard, A. (2020). Force adaptation in contact tasks with dynamical systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6841–6847. IEEE.

Amanhoud, W., Sanchez, J. H., Bouri, M., and Billard, A. (2021). Contact-initiated shared control strategies for four-arm supernumerary manipulation with foot interfaces. *The International Journal of Robotics Research*, 40(8-9):986–1014.

Åström, K. J. and Wittenmark, B. (2013). *Adaptive control*. Courier Corporation.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bodner, J., Wykypiel, H., Wetscher, G., and Schmid, T. (2004). First experiences with the da vinci™ operating robot in thoracic surgery. *European Journal of Cardio-thoracic surgery*, 25(5):844–851.

Boessenkool, H., Abbink, D. A., Heemskerk, C. J., van der Helm, F. C., and Wildenbeest, J. G. (2012). A task-specific analysis of the benefit of haptic shared control during telemanipulation. *IEEE Transactions on Haptics*, 6(1):2–12.

Brett, P., Taylor, R., Proops, D., Coulson, C., Reid, A., and Griffiths, M. (2007). A surgical robot for cochleostomy. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1229–1232. IEEE.

Buhmann, M. D. (2003). *Radial basis functions: theory and implementations*, volume 12. Cambridge university press.

Dahroug, B., Tamadazte, B., and Andreff, N. (2016). 3d path following with remote center of motion constraints. In *International Conference on Informatics in Control, Automation and Robotics*.

De Rossi, G., Minelli, M., Sozzi, A., Piccinelli, N., Ferraguti, F., Setti, F., Bonfé, M., Secchi, C., and Muradore, R. (2019). Cognitive robotic architecture for semi-autonomous execution of manipulation tasks in a surgical environment. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7833.

Dieterich, S. and Gibbs, I. C. (2011). The cyberknife in clinical use: current roles, future expectations. In *IMRT, IGRT, SBRT*, volume 43, pages 181–194. Karger Publishers.

Dragan, A. D. and Srinivasa, S. S. (2013). A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805.

Duan, J., Gan, Y., Chen, M., and Dai, X. (2018). Adaptive variable impedance control for dynamic contact force tracking in uncertain environment. *Robotics and Autonomous Systems*, 102:54–65.

Elprama, S. A., Kilpi, K., Duysburgh, P., Jacobs, A., Vermeulen, L., and Van Looy, J. (2013). Identifying barriers in telesurgery by studying current team practices in robot-assisted surgery. In *2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, pages 224–231. IEEE.

Erlbacher, E. A. (2000). Force control basics. *Industrial Robot: An International Journal*.

Faverjon, B. and Tournassoud, P. (1987). A local based approach for path planning of manipulators with a high number of degrees of freedom. In *Proceedings. 1987 IEEE international conference on robotics and automation*, volume 4, pages 1152–1159. IEEE.

Feder, H. J. S. and Slotine, J.-J. (1997). Real-time path planning using harmonic potentials in dynamic environments. In *Proceedings of International Conference on Robotics and Automation*, volume 1, pages 874–881. IEEE.

Ferraguti, F., Preda, N., Bonfe, M., and Secchi, C. (2015a). Bilateral teleoperation of a dual arms surgical robot with passive virtual fixtures generation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4223–4228. IEEE.

Ferraguti, F., Preda, N., Manurung, A., Bonfe, M., Lambercy, O., Gassert, R., Muradore, R., Fiorini, P., and Secchi, C. (2015b). An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery. *IEEE Transactions on Robotics*, 31(5):1073–1088.

Ferraguti, F., Secchi, C., and Fantuzzi, C. (2013). A tank-based approach to impedance control with variable stiffness. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4948–4953. IEEE.

Ferreau, H. J., Kirches, C., Potschka, A., Bock, H. G., and Diehl, M. (2014). qpoases: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363.

Figueroa, N. and Billard, A. (2018). A physically-consistent bayesian non-parametric mixture model for dynamical system learning. In *CoRL*, pages 927–946.

for Standardization, I. O. (2012). *Robots and Robotic Devices–Safety Requirements for Industrial Robots: Robotter Og Robotudstyr–Sikkerheds-krav Til Industrirobotter–Del 1: Robotter. Robots.* International Organization for Standardization.

Franken, M., Stramigioli, S., Misra, S., Secchi, C., and Macchelli, A. (2011). Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency. *IEEE transactions on robotics*, 27(4):741–756.

Geitner, C., Birrell, S., Krehl, C., and Jennings, P. (2018). Haptic Foot Pedal: Influence of Shoe Type, Age, and Gender on Subjective Pulse Perception. *Hum. Factors J. Hum. Factors Ergon. Soc.*, 60(4):496–509.

Goil, A., Derry, M., and Argall, B. D. (2013). Using machine learning to blend human and robot controls for assisted wheelchair navigation. In *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6. IEEE.

Gopinath, D., Jain, S., and Argall, B. D. (2016). Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE robotics and automation letters*, 2(1):247–254.

Guthart, G. S. and Salisbury, J. K. (2000). The intuitive/sup tm/telesurgery system: overview and application. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 618–621. IEEE.

Hansson, A. and Servin, M. (2010). Semi-autonomous shared control of large-scale manipulator arms. *Control Engineering Practice*, 18(9):1069–1076.

Hart, S. G. and Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier.

Heck, D., Saccon, A., Beerens, R., and Nijmeijer, H. (2018). Direct force-reflecting two-layer approach for passive bilateral teleoperation with time delays. *IEEE Transactions on Robotics*, 34(1):194–206.

Hernandez Sanchez, J., Amanhoud, W., Billard, A., and Bouri, M. (2021). Enabling hybridrobotic-assisted solo-surgery with four arms via haptic foot interfaces. *The International Journal of Robotics Research (Under Review)*.

Hirche, S., Stanczyk, B., and Buss, M. (2003). Transparent exploration of remote environments by internet telepresence. In *Proceedings of Int. Workshop on High-Fidelity Telepresence and Teleaction jointly with the conference HUMANOIDS.*

Hockstein, N. G., Gourin, C., Faust, R., and Terris, D. J. (2007). A history of robots: from science fiction to surgical robotics. *Journal of robotic surgery*, 1(2):113–118.

Hoffman, G. (2019). Evaluating Fluency in Human-Robot Collaboration. *IEEE Trans. Human-Machine Syst.*, 49(3):209–218.

Hogan, N. (1984). Impedance control: An approach to manipulation. In *1984 American control conference*, pages 304–313. IEEE.

Hogan, N. (1987). Stable execution of contact tasks using impedance control. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1047–1054. IEEE.

Huber, L., Billard, A., and Slotine, J.-J. (2019). Avoidance of convex and concave obstacles with convergence ensured through contraction. *IEEE Robotics and Automation Letters*, 4(2):1462–1469.

Ichinose, A., Gomikawa, Y., and Suzuki, S. (2013). Driving assistance through pedal reaction force control with consideration of JND. In *2013 IEEE RO-MAN*, pages 484–489. IEEE.

Kanehiro, F., Lamiraux, F., Kanoun, O., Yoshida, E., and Laumond, J.-P. (2008). A local collision avoidance method for non-strictly convex polyhedra. *Proceedings of robotics: science and systems IV*, page 33.

Khansari-Zadeh, S. M. and Billard, A. (2011). Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957.

Khansari-Zadeh, S. M. and Billard, A. (2012). A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4):433–454.

Khansari-Zadeh, S. M., Kronander, K., and Billard, A. (2012). Learning to play minigolf: A dynamical system-based approach. *Advanced Robotics*, 26(17):1967–1993.

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53.

Khoramshahi, M. and Billard, A. (2019). A dynamical system approach to task-adaptation in physical human–robot interaction. *Autonomous Robots*, 43(4):927–946.

Khoramshahi, M., Laurens, A., Triquet, T., and Billard, A. (2018). From human physical interaction to online motion adaptation using parameterized dynamical systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1361–1366. IEEE.

Kim, W. S., Tendick, F., Ellis, S. R., and Stark, L. W. (1987). A Comparison of Position and Rate Control for Telemanipulations with Consideration of Manipulator System Dynamics. *IEEE J. Robot. Autom.*, 3(5):426–436.

Koditschek, D. E. (1989). Robot planning and control via potential functions. *The robotics review*, page 349.

Kramberger, A., Shahriari, E., Gams, A., Nemec, B., Ude, A., and Haddadin, S. (2018). Passivity based iterative learning of admittance-coupled dynamic movement primitives for interaction with changing environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6023–6028. IEEE.

Kronander, K. and Billard, A. (2015). Passive interaction control with dynamical systems. *IEEE Robotics and Automation Letters*, 1(1):106–113.

Kronander, K., Khansari, M., and Billard, A. (2015). Incremental motion learning with locally modulated dynamical systems. *Robotics and Autonomous Systems*, 70:52–62.

Kuo, C.-H., Dai, J. S., and Dasgupta, P. (2012). Kinematic design considerations for minimally invasive surgical robots: an overview. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 8(2):127–145.

Laghi, M., Maimeri, M., Marchand, M., Leparoux, C., Catalano, M., Ajoudani, A., and Bicchi, A. (2018). Shared-autonomy control for intuitive bimanual tele-manipulation. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9. IEEE.

Lee, D. and Spong, M. W. (2006). Passive bilateral teleoperation with constant time delay. *IEEE transactions on robotics*, 22(2):269–281.

Lee, K. and Buss, M. (2008). Force tracking impedance control with variable target stiffness. *IFAC Proceedings Volumes*, 41(2):6751–6756.

Lefebvre, T., Xiao, J., Bruyninckx, H., and De Gersem, G. (2005). Active compliant motion: a survey. *Advanced Robotics*, 19(5):479–499.

Li, M., Hang, K., Kragic, D., and Billard, A. (2016). Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75:352–364.

Martens, W., Poffet, Y., Soria, P. R., Fitch, R., and Sukkarieh, S. (2016). Geometric priors for gaussian process implicit surfaces. *IEEE Robotics and Automation Letters*, 2(2):373–380.

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

Meli, L., Pacchierotti, C., and Prattichizzo, D. (2017). Experimental evaluation of magnified haptic feedback for robot-assisted needle insertion and palpation. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 13(4):e1809.

Mirrazavi Salehian, S. S., Khoramshahi, M., and Billard, A. (2016). A dynamical system approach for catching softly a flying object: Theory and experiment. *IEEE Transaction on Robotics*.

Moustris, G. P., Hiridis, S. C., Deliparaschos, K. M., and Konstantinidis, K. M. (2011). Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature. *The international journal of medical robotics and computer assisted surgery*, 7(4):375–392.

Muelling, K., Venkatraman, A., Valois, J.-S., Downey, J. E., Weiss, J., Javdani, S., Hebert, M., Schwartz, A. B., Collinger, J. L., and Bagnell, J. A. (2017). Autonomy infused teleoperation with application to brain computer interface controlled manipulation. *Autonomous Robots*, 41(6):1401–1422.

Neumann, K., Lemme, A., and Steil, J. J. (2013). Neural learning of stable dynamical systems based on data-driven lyapunov candidates. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1216–1222. IEEE.

Niemeyer, G., Preusche, C., Stramigioli, S., and Lee, D. (2016). Telerobotics. In *Springer handbook of robotics*, pages 1085–1108. Springer.

Nurok, M., Sundt, T. M., and Frankel, A. (2011). Teamwork and communication in the operating room: relationship to discrete outcomes and research challenges. *Anesthesiology clinics*, 29(1):1–11.

Okamura, A. M. (2009). Haptic feedback in robot-assisted minimally invasive surgery. *Current opinion in urology*, 19(1):102.

Parietti, F., Chan, K. C., Hunter, B., and Asada, H. H. (2015). Design and control of supernumerary robotic limbs for balance augmentation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5010–5017. IEEE.

Passenberg, C., Peer, A., and Buss, M. (2010). A survey of environment-, operator-, and task-adapted controllers for teleoperation systems. *Mechatronics*, 20(7):787–801.

Perrin, N. and Schlehuber-Caissier, P. (2016). Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems. *Systems & Control Letters*, 96:51–59.

Punt, M. M., Stefels, C. N., Grimbergen, C. A., and Dankelman, J. (2005). Evaluation of voice control, touch panel control and assistant control during steering of an endoscope. *Minimally Invasive Therapy & Allied Technologies*, 14(3):181–187.

Raibert, M. H. and Craig, J. J. (1981). Hybrid position/force control of manipulators.

Rakita, D., Mutlu, B., Gleicher, M., and Hiatt, L. M. (2019). Shared control–based bimanual robot manipulation. *Science Robotics*, 4(30).

Rana, M. A., Li, A., Fox, D., Boots, B., Ramos, F., and Ratliff, N. (2020). Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems. In *Learning for Dynamics and Control*, pages 630–639. PMLR.

Randell, R., Honey, S., Alvarado, N., Pearman, A., Greenhalgh, J., Long, A., Gardner, P., Gill, A., Jayne, D., and Dowding, D. (2016). Embedding robotic surgery into routine practice and impacts on communication and decision making: a review of the experience of surgical teams. *Cognition, Technology & Work*, 18(2):423–437.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer.

Rimon, E. (1990). *Exact robot navigation using artificial potential functions*. PhD thesis, Yale University.

Rosenberg, L. B. (1993). Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Proceedings of IEEE virtual reality annual international symposium*, pages 76–82. IEEE.

Roveda, L., Iannacci, N., Vicentini, F., Pedrocchi, N., Braghin, F., and Tosatti, L. M. (2016). Optimal impedance force-tracking control design with impact formulation for interaction tasks. *IEEE Robotics and Automation Letters*, 1(1):130–136.

Roveda, L., Vicentini, F., and Tosatti, L. M. (2013). Deformation-tracking impedance control in interaction with uncertain environments. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1992–1997. IEEE.

Sadeghian, H., Zokaei, F., and Jazi, S. H. (2019). Constrained kinematic control in minimally invasive robotic surgery subject to remote center of motion constraint. *Journal of Intelligent & Robotic Systems*, 95(3):901–913.

Salisbury, J. K. (1980). Active stiffness control of a manipulator in cartesian coordinates. In *1980 19th IEEE conference on decision and control including the symposium on adaptive processes*, pages 95–100. IEEE.

Sandoval, J., Poisson, G., and Vieyres, P. (2017). A new kinematic formulation of the rcm constraint for redundant torque-controlled robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4576–4581. IEEE.

Sanner, R. M. and Slotine, J.-J. E. (1995). Stable adaptive control of robot manipulators using "neural" networks. *Neural computation*, 7(4):753–790.

Scherzinger, S., Roennau, A., and Dillmann, R. (2017). Forward dynamics compliance control (fdcc): A new approach to cartesian compliance for robotic manipulators. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4568–4575. IEEE.

Schindlbeck, C. and Haddadin, S. (2015). Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 440–447. IEEE.

Secchi, C., Stramigioli, S., and Fantuzzi, C. (2006). Position drift compensation in port-hamiltonian based telemanipulation. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4211–4216. IEEE.

Selvaggio, M., Abi-Farraj, F., Pacchierotti, C., Giordano, P. R., and Siciliano, B. (2018). Haptic-based shared-control methods for a dual-arm system. *IEEE Robotics and Automation Letters*, 3(4):4249–4256.

Selvaggio, M., Chen, F., Gao, B., Notomista, G., Trapani, F., and Caldwell, D. (2016). Vision based virtual fixture generation for teleoperated robotic manipulation. In *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 190–195. IEEE.

Shahriari, E., Kramberger, A., Gams, A., Ude, A., and Haddadin, S. (2017). Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives. In *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference on*, pages 136–142. IEEE.

Sheridan, T. B. (1993). Space teleoperation through time delay: Review and prognosis. *IEEE Transactions on robotics and Automation*, 9(5):592–606.

Shirkhodaie, A. (2002). Supervised control of cooperative multi-agent robotic vehicles. In *Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory (Cat. No. 02EX540)*, pages 386–390. IEEE.

Smisek, J., van Paassen, M., and Schiele, A. (2015). Haptic guidance in bilateral teleoperation: Effects of guidance inaccuracy. In *2015 IEEE World Haptics Conference (WHC)*, pages 500–505. IEEE.

Spaner, S. J. and Warnock, G. L. (1997). A brief history of endoscopy, laparoscopy, and laparoscopic surgery. *Journal of Laparoendoscopic & Advanced Surgical Techniques*, 7(6):369–373.

Steinke, F., Schölkopf, B., and Blanz, V. (2005). Support vector machines for 3d shape processing. In *Computer Graphics Forum*, volume 24, pages 285–294. Wiley Online Library.

Storms, J. G. and Tilbury, D. M. (2014). Blending of human and obstacle avoidance control for a high speed mobile robot. In *2014 American Control Conference*, pages 3488–3493. IEEE.

Stramigioli, S. (2015). Energy-aware robotics. In *Mathematical Control Theory I*, pages 37–50. Springer.

Su, H., Yang, C., Ferrigno, G., and De Momi, E. (2019). Improved human–robot collaborative control of redundant robot for teleoperated minimally invasive surgery. *IEEE Robotics and Automation Letters*, 4(2):1447–1453.

Tobergte, A., Helmer, P., Hagn, U., Rouiller, P., Thielmann, S., Grange, S., Albu-Schäffer, A., Conti, F., and Hirzinger, G. (2011). The sigma. 7 haptic interface for mirosurge: A new bi-manual surgical console. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3023–3030. IEEE.

Van der Schaft, A. J. and Van Der Schaft, A. (2000). *L2-gain and passivity techniques in nonlinear control*, volume 2. Springer.

van Oosterhout, J., Wildenbeest, J. G., Boessenkool, H., Heemskerk, C. J., de Baar, M. R., van der Helm, F. C., and Abbink, D. A. (2015). Haptic shared control in tele-manipulation: Effects of inaccuracies in guidance on task execution. *IEEE transactions on haptics*, 8(2):164–175.

Villani, L. and De Schutter, J. (2016). Force control. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, pages 195–220. Springer International Publishing.

Vukobratovic, M. (2009). Chapter 1 control of robots in contact tasks: A survey. In *Dynamics and robust control of robot-environment interaction*, volume 2. World Scientific.

Wagner, C. R., Howe, R. D., and Stylopoulos, N. (2002). The role of force feedback in surgery: analysis of blunt dissection. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, International Symposium on*, pages 73–73. Citeseer.

Wang, X., Yang, C., Ma, H., and Cheng, L. (2015). Shared control for teleoperation enhanced by autonomous obstacle avoidance of robot manipulator. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4575–4580. IEEE.

Whitney, D. E. (1977). Force feedback control of manipulator fine motions.

Xiong, L., Chng, C. B., Chui, C. K., Yu, P., and Li, Y. (2017). Shared control of a medical robot with haptic guidance. *International journal of computer assisted radiology and surgery*, 12(1):137–147.

Yang, Y.-J., Udatha, S., Kulić, D., and Abdi, E. (2020). A novel foot interface versus voice for controlling a robotic endoscope holder. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 272–279. IEEE.

Yuen, S. G., Kettler, D. T., Novotny, P. M., Plowes, R. D., and Howe, R. D. (2009). Robotic motion compensation for beating heart intracardiac surgery. *The International journal of robotics research*, 28(10):1355–1372.

Walid Amanhoud
20 Route du clos
74200 Anthy sur Léman France 15.02.1993
00 33 6 48 74 55 28 Unmarried
walid.amanhoud@gmail.com French
Linkedin : https://fr.linkedin.com/pub/walid-amanhoud/103/7b9/547

Formation

2015	**Master of Science MSc in Microengineering (Robotics and Autonomous Systems)** *Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland*
2013	**Bachelor of Science BSc in Microengineering** *Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland*
2010	**Baccalauréat scientifique with honors (High school degree)** *Lycée La Versoie, Thonon les Bains, France*

Experience

06.2017-now	**Phd student** *Learning Algorithms and Systems Laboratory (LASA), EPFL, Switzerland* Force Control, Human-Robot Interaction, Human-Robot collaboration, Robot-Assisted Surgery.
03.2016-03.2017	**Robotics software engineer** *Automata Technologies Ltd, London, United Kingdom* Responsible of motion planning and control for a 6/7 DOF robotic arm including direct/inverse kinematics modeling, self-collision handling, trajectory generation in joint/cartesian space, direct dynamics modeling, angular position/velocity control for trajectory tracking, torque control for backdriving/teaching by example.
03.2015-07.2015	**Master thesis: Model-based control of quadcopters** *Automatic Control Laboratory (LA), EPFL, Switzerland* Implementation of a real-time Nonlinear Model Predictive Control (NMPC) strategy on a quadcopter platform to track desired trajectories, including the construction and identification of the quadcopter model and the development of a reliable state estimator.
09.2014-03.2015	**Engineering internship: Research and Development for Civilian Drones** *SenseFly, Cheseaux-sur-Lausanne, Switzerland* Implementation and maintenance of various sensors acquisition and calibration code. Development of utilities applications to support main software development.
02.2014-05.2014	**Semester project: Audio-based Control of Micro Air Vehicles** *Laboratory of Intelligent Systems (LIS), EPFL, Switzerland* Development an on-board audio-based motion control system for a Micro Air Vehicle (MAV) capable of navigating the robot relative to a desired sound source.
09.2013-12.2013	**Semester project: Trajectory following with real quadrotors** *Distributed Intelligent Systems and Algorithms Laboratory (DISAL), EPFL, Switzerland* Modelization and simulation of the quadcopter dynamics, construction and implementation of a trajectory generation algorithm on a real platform and calibration of the simulation with the experimental data.

Languages

French: Native speaker
English: Good command (Lived one year in London, UK)
Moroccan Arabic: Good command

Computer skills

Programming languages: C/C++, Python, Matlab
Development tools: Matlab, Simulink
Office tools: MS office suite (Word, Excel, PowerPoint), Latex
Version control: Git

Hobbies

Sport: football, running, swimming
Cinema, Music

List of publications

- Hernandez Sanchez, J., Amanhoud, W., Billard, A., and Bouri, M. (2021). **Enabling hybrid robotic-assisted solo-surgery with four arms via haptic foot interfaces**. Under review at *The International Journal of Robotics Research*.
- Amanhoud, W., Hernandez Sanchez, J., Bouri, M., and Billard, A. (2021). **Contact-initiated shared control strategies for four-arm supernumerary manipulation with foot interfaces**. *The International Journal of Robotics Research*, 40(8–9), 986–1014.
- Amanhoud, W., Khoramshahi, M., Bonnesoeur, M., and Billard, A. (2020). **Force adaptation in contact tasks with dynamical systems**. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6841–6847. IEEE.
- Amanhoud, W., Khoramshahi, M., and Billard, A. (2019). **A dynamical system approach to motion and force generation in contact tasks**. In *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany.