UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

INGEGNERIA MECCANICA E INDUSTRIALE
XXXVI CICLO DI DOTTORATO DI RICERCA
ING-INF/04 AUTOMATICA

Human modeling, Game Theory, and Role Arbitration for physical Human-Robot Interaction

Paolo Franceschi
Matricola N. 734981

Advisor: Prof, Manuel Beschi
Correlator: Dr. Nicola Pedrocchi

Academic Year 2022-2023

To Chiara, Giovanna, and
Alberto

Al mondo ci sono soltanto gli uomini che non funzionano. Per il resto ogni cosa funziona perfettamente.

G. Guareschi

ABSTRACT

This thesis addresses some of the aspects related to the physical Human–Robot Interaction (pHRI), with a particular focus on industrial applications. The main motivation is that humans and robots have complementary skills, that, combined, can improve working conditions, safety, and productivity. Indeed, robots are extremely precise, they can possibly carry heavy objects, and, if properly programmed, have exceptional proprioceptive capabilities. Instead, humans can face and solve new problems easily, can adapt quickly to fast-changing scenarios, can reason, and have natural capabilities of understanding and interacting with the environment. Therefore, the objective of this thesis is to make robots trusty assistants for humans, allowing smooth and natural interaction. On the one hand, humans naturally understand partners' goals, and it is easy for two humans to perform a task together, also with limited interaction channels. On the other hand, those capabilities are not intrinsic to robots and must be addressed to allow smooth interaction.

This thesis, without claiming to be exhaustive, wants to address three fundamental aspects of the interaction between partners.

First, this thesis investigates the modeling of human behavior. This is an extremely vast field of study. Therefore, this work limits its focus on modeling human behavior in physical interaction with a robot. In particular, this work addresses human behavior modeling as (i) a feedback controller employing an Extended Kalman Filter (EKF) to recover online the time-variant human's gain matrix; (i) as an Optimal Control, using Inverse Optimal Control (IOC) to recover human's cost function; (iii) and uses information of the interaction to make predictions of future intention of motion of the human, using a Machine Learning model. Finally, various experiments show the applicability and the limitations of the three aspects concerned with human modeling. Then, this thesis investigates the modeling of the physical interaction between humans and robots. The pHRI is described within a Game-Theoretic Framework. In this, the human and the robot are two agents acting on a low-level Cartesian Impedance system. Both Cooperative and Non-Cooperative models are addressed in the continuous (differential game theory) and discrete versions with the formulation of a distributed Model Predictive Control (GT-dMPC). The system's behavior is analyzed for different situations to understand the limitations and applicability of the two models. Finally, the Role Arbitration between the human and the robot is addressed. Role arbitration is the mechanism that assigns the role of leader either to the human or the robot. Since humans and robots have complementary skills, it is fundamental that such skills are used at the right moment. Experiments on lab mockups and in a real industrial scenario are carried out to show that Role Arbitration improves performances on various aspects related to industrial application, such as precision, a limited effort of humans, flexibility, and reduction of cycle time. Not related to modeling, this thesis also evaluates subjective preferences during interaction. Indeed, humans are various and each has specific preferences on everything. The same happens when dealing with the pHRI scenario. To address this aspect, this thesis also proposes Preference Based Optimization (PBO) to tune the robot behavior (*i.e.*, its control parameters) based on users' preferences.

Keywords: physical Human–Robot Interaction, Human modeling, differential Game Theory, Role Arbitration, Preference-Based Optimization.

SOMMARIO

Questa tesi affronta alcuni aspetti legati all'interazione fisica uomo-robot (pHRI) con particolare attenzione alle applicazioni industrial. La motivazione principale è che gli esseri umani e i robot hanno competenze complementari che, combinate, possono migliorare le condizioni di lavoro, la sicurezza e la produttività. Infatti, i robot sono estremamente precisi, possono trasportare oggetti pesanti e, se adeguatamente programmati, hanno eccezionali capacità proprioceptive. Gli esseri umani, invece, sono in grado di affrontare e risolvere problemi, si adattano rapidamente a scenari in rapida evoluzione, sanno ragionare e hanno capacità naturali di comprendere e interagire con i robot e con l'ambiente. Pertanto, l'obiettivo di questa tesi è quello di rendere i robot degli assistenti fidati per l'uomo, consentendo un'interazione fluida e naturale. Da un lato, gli esseri umani capiscono naturalmente gli obiettivi dei partner e per due esseri umani è facile eseguire un compito insieme, anche con canali di interazione limitati. D'altra parte, queste capacità non sono intrinseche ai robot e devono essere affrontate per consentire un'interazione fluida.

Questa tesi, senza pretendere di essere esaustiva, vuole affrontare tre aspetti fondamentali dell'interazione tra partner.

In primo luogo, questa tesi studia la modellazione del comportamento umano. Si tratta di un campo di studio estremamente vasto. Pertanto, questo lavoro si limita a modellare il comportamento umano nell'interazione fisica con un robot. In particolare, questo lavoro affronta la modellazione del comportamento umano come (i) un controllore di feedback attraverso l'utilizzo di un filtro di Kalman esteso (EKF) per recuperare online la matrice del guadagno umano variabile nel tempo; (ii) come un controllore ottimo, utilizzando un controllo ottimo inverso (IOC) per recuperare la funzione di costo dell'uomo; (iii) e utilizza le informazioni dell'interazione per fare previsioni sull'intenzione futura di movimento dell'uomo, utilizzando un modello di apprendimento automatico. Infine, vari esperimenti mostrano l'applicabilità e i limiti dei tre aspetti relativi alla modellazione umana. In seguito, questa tesi studia la modellazione dell'interazione fisica tra uomo e robot. La pHRI è descritta mediante teoria dei giochi. In questo contesto, l'uomo e il robot sono due agenti che agiscono su un sistema descritto come impedenza Cartesiana.Vengono affrontati sia modelli cooperativi che non-cooperativi, sia nella versione continua (teoria dei giochi differenziali) che in quella discreta, con la formulazione di un controllo predittivo distribuito (GT-dMPC). Il comportamento del sistema viene analizzato in diverse situazioni per comprendere i limiti e l'applicabilità dei due modelli. Infine, viene affrontato il tema dell'arbitrato dei ruoli tra l'uomo e il robot. L'arbitrato dei ruoli è il meccanismo che assegna il ruolo di leader all'uomo o al robot. Poiché gli esseri umani e i robot hanno competenze complementari, è fondamentale che tali competenze vengano utilizzate al momento giusto. Gli esperimenti condotti su simulazioni, in laboratorio e in uno scenario industriale reale dimostrano che l'arbitrato di ruolo migliora le prestazioni di vari aspetti legati all'applicazione industriale, come ad esempio precisione, sforzo limitato dell'uomo, flessibilità e riduzione del tempo di ciclo. Come ulteriore aspetto non legato alla modellazione, questa tesi valuta anche le preferenze soggettive durante l'interazione. Infatti, gli esseri umani sono diversi e ognuno ha preferenze specifiche su tutto. Lo stesso accade quando si ha a che fare con uno scenario pHRI. Per affrontare questo aspetto, questa tesi propone anche l'ottimizzazione basata sulle preferenze (PBO) per regolare il comportamento del robot (cioè i suoi parametri di controllo) in base alle preferenze degli utenti.

CONTENTS

LIST OF FIGURES

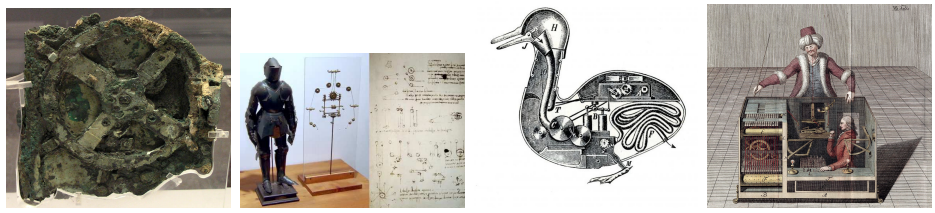
LIST OF TABLES

CHAPTER 1

INTRODUCTION

The idea of automating tasks and creating mechanical beings can be traced back to ancient civilizations. Early examples are the Mechanical Theater of Heron and the Antikythera Mechanism. In the Renaissance, Leonardo da Vinci, the renowned Italian artist, scientist, and inventor, conceptualized and designed various machines and automata, including a humanoid robot. One of his most famous robotic designs is referred to as the "Leonardo da Vinci Robot" or the "Mechanical Knight". He never built a fully functional version of this robot during his lifetime, but he left behind detailed sketches and notes describing its design and mechanisms. The robot was intended to be powered by a combination of clockwork mechanisms and hidden weights. With Jacques de Vaucanson, a French inventor, in the 18th century, the Digesting Duck is considered one of the early examples of automata. Vaucanson's robotic duck was showcased in 1739 and gained considerable attention during its time. It was designed to imitate the motions and behaviors of a real duck, including flapping its wings, drinking water, eating grain, and even digesting food. It was known for its ability to defecate, seemingly adding to its realistic appearance. Wolfgang von Kempelen, an Austrian inventor, created "the Turk".

It was designed to play chess against human opponents and appeared to possess extraordinary skills. The Turk consisted of a large wooden cabinet with a chessboard on top. Inside the cabinet, there was a mechanical figure dressed as a Turk, sitting behind the chessboard. The automaton appeared to be capable of moving the chess pieces and playing the game with great skill. Inside the cabinet, a human chess master concealed himself and controlled the movements of the Turk through a series of levers and pulleys. The Turk left a lasting impact on the public's imagination and became an influential symbol of the intersection between machinery and human-like capabilities.



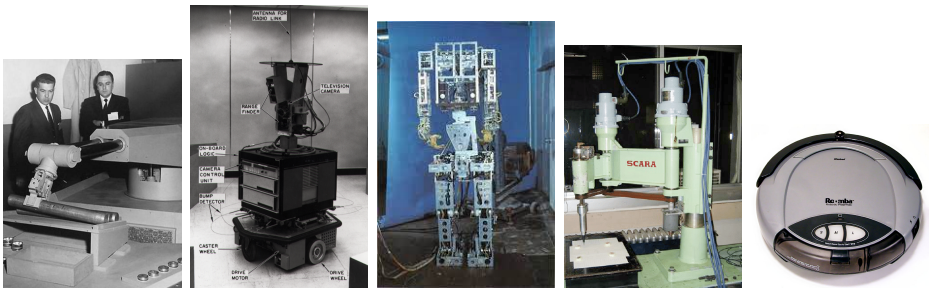
(a) The antikythera mechanism **(b)** The Da Vinci's mechanical knight **(c)** The de Vaucanson's digesting duck **(d)** The von Kempelen's Turk

Figure 1.1: Ancient famous automata examples

The term "robot" first appeared in a play called R.U.R. (Rossum's Universal Robots), published by the Czech author Karel Čapek in 1921, where robots were manufactured biological beings that performed all unpleasant manual labor. Another author, Isaac Asimov, in 1942 formulated the Three Laws of Robotics and, in the process, coined the word "robotics".

The history of robotics, in the last century, faced exponential growth. The foundation of the modern robotics industry dates back to 1954, when George Devol invented the Unimate. Devol sold the first Unimate to General Motors in 1960, and it was installed in 1961 in a plant in Ewing Township, New Jersey, to lift hot pieces of metal from a die-casting

machine and place them in cooling liquid. The first mobile robot capable of reasoning about its surroundings, Shakey, was built in 1970 by the Stanford Research Institute. Waseda University initiated the WABOT project in 1967, and in 1972, it completed the WABOT-1, the world's first full-scale humanoid intelligent robot. The SCARA, Selective Compliance Assembly Robot Arm, was created in 1978 as an efficient, 4-axis robotic arm. Best used for picking up parts and placing them in another location, the SCARA was introduced to assembly lines in 1981. The popular Roomba, a robotic vacuum cleaner, was first released in 2002 by the company iRobot.



(a) The Unimate **(b)** The Shakey **(c)** The WABOT-1 **(d)** The SCARA **(e)** The Roomba robot

Figure 1.2: Modern robots examples

In recent decades, robotic platform design and production have continuously grown, with specific designs for various applications. Autonomous Mobile Robots (AMR) [3, 128] are spreading for indoor and outdoor applications, from warehouse logistics to agriculture [12]. Domestic robots are nowadays a reality to help with tedious housework. Consider, for example, lawn mowing [22], vacuum cleaning [72], or window cleaning robots [96, 157]. Social robots are also used in and for education [13]. They belong to all the categories of robots interacting with humans, such as robot receptionists, waiters, etc. Robots are also found

in operating theaters, where they do not replace the surgeon but are used as accurate assistants for minimally invasive procedures [7, 133]. Some robots take inspiration and try to mimic natural behaviors and principles. They are the so-called biologically inspired [54, 106] robots. Among them, particular attention is given to Humanoid robots [152], robots with the aspects and capabilities typical of humans. Along with the spread of robots in the most disparate fields, traditional robots, and their first applications are related to the industry [131]. Historically, industrial robots have always been confined in cages for safety reasons to avoid any possible -dangerous- interaction with humans. This involves production plants designed on purpose to allow robots and humans to have different, separate workspaces. In today's industries, the desire for a safe and flexible manufacturing environment has pushed researchers to study solutions where the robots can be easily moved, possibly removing cages, still ensuring safety. To this purpose, Collaborative robots or cobots are becoming increasingly important [161, 162]. Indeed, such robots are designed on purpose to work alongside humans, simplifying the certification of an application from a safety point of view.

Among the various, this thesis focuses on industrial robots and applications of robotics in strict collaboration with human operators, where the Human-Robot Interaction field begins.

1.1 HUMAN-ROBOT INTERACTION

Human-robot interaction (HRI) is the discipline that studies and allows safe and natural interaction between humans and robots [139]. Interaction can be seen from various points of view, which can be briefly summarized in *physical* and *social* interaction [9]. The social interacting robots belong to all those that mainly give information to humans, and today, are spreading across hospitals, hotel receptions [19], and airports [70]. The Physically interacting robots belong to all kinds of robots capable of

manipulating objects and interacting with the physical world and people. On the one hand, social robots can also interact with humans physically and require capabilities to deal with the surrounding environment, such as localization and navigation, obstacle avoidance, etc. On the other hand, physically interacting robots must be aware of humans and must have capabilities of more "social" interactions such as intention understanding [43], various communication channels such as verbal (NLP) and nonverbal [153], etc. Therefore, the distinction between physical and social interacting robots is usually fuzzy, and both aspects should be considered when dealing with robots.

(a) A social robot **(b)** A physical interacting robot from [132]

Figure 1.3: Examples of social and physical interacting robots

One major issue that arises with the HRI relates to human safety. In order to develop safe and precise HRI, robots should be designed and developed so that they do not provoke accidents (*i.e.* deaths, injuries, and property damage). According to [178], there are several types of robots (differentiated based on their functionality, degrees of freedom, workspace, etc.), possibly categorized with respect to their interaction with humans. The first category consists of the *Robots in the wild*, where humans and robots are not collocated. They are spatially or temporally separated, have remote and limited HRI, and require high autonomy. The

second category comprises the *Professional Service Robots*, which interact proximately with co-workers. The third category consists of the *Personal service robots*, which posse the complex, rich, and proximately HRI and intermediate level of autonomy as humans and robots are collocated, *e.g.*, robots assistants to elderly and disabled people.

This thesis focuses on the Professional Service Robots category, with a bit of personalization. In particular, this thesis addresses robots and their spread in manufacturing and industrial scenarios. The current state-of-the-art in collaborative robotics for industrial applications is the result of a long legacy of research and development in actuation principles and mechanisms from the 2000s onward, together with abundant literature about improving control performances, intuitive interaction modes, and perception with sensors [169].

Safety, also in these scenarios, is a crucial aspect that requires clear definitions [170] and appropriate metrics and control strategies [179]. Since safety is a crucial node to be addressed for any robotics application involving humans, normatives have appeared in recent years. According to the standard rules defined by the ISO in [59–61], safety depends on and is evaluated considering the entire application and not the single modules such as the robotic platform, its control, and the other modules alone. According to the standards, the robot at the end-effector should not move faster than 250 mm/s to certify the application. It should also stop if the external forces and power exceed a threshold according to the so-called Power Force Limitation (PFL). To ease the deployment of collaborative applications, robots also evolved to comply with the safety requirements. Such kinds of robots are often referred to as *Cobots*. The Cobots are robotic manipulators that implicitly implement and ensure some safety, such as safety stops when detecting external forces, limited end-effector velocities, etc. Some examples are the Kuka Iiwa, the Universal Robots, the Franka-Emika Panda, and many others.

(a) The Kuka iiwa robot

(b) The UR5 robot

(c) The panda robot

Figure 1.4: Examples of *Cobots*

Human–robot collaboration has found several applications [47, 80]. Despite this, real-world applications in industrial scenarios still face some issues [78] such as employee-centered factors like the fear of job loss and ensuring an appropriate level of trust in the robot are considered essential. Also, acceptability makes actual deployment challenging [113]. One relevant challenge of Industry 5.0 is the design of human-centered smart environments (i.e., prioritizing human well-being while maintaining production performance). In these environments, robots and humans will share the same space and collaborate to reach common objectives. This makes the requirement of measures and metrics to assess the quality of HRC [20].

1.2 GAME THEORY OVERVIEW

Consider now a shared task between a human and a robot, such as an assembly or carrying an object together. Both are agents that can do something on a specific object. Both know the presence of the other. They possibly also know what the other should do, and they can help each other or not. All these elements are typical of the mathematical branch called Game Theory (GT). Game theory is the study of mathematical models of

strategic interactions among rational agents. Therefore, with the word *game*, we refer to a set of rules, agents, information, etc. To characterize a game, one needs to specify several items: (i) the players are the agents that make decisions, (ii) the rules define the actions allowed by the players and their effects, (iii) the information structure specifies what each player knows before making each decision [1]. (iv) The objective specifies the goal of each player. For a mathematical solution to a game, one further needs to make assumptions about the player's rationality regarding questions such as: Will the players always pursue their best interests to fulfill their objectives? Will the players form coalitions? Will the players trust each other? Moreover, games can be zero-sum or non-zero-sum. A zero-sum game is a mathematical representation in game theory and economic theory of a situation that involves two sides, where the result is an advantage for one side and an equivalent loss for the other. In other words, player one's gain is equivalent to player two's loss, resulting in zero net improvement in the game's benefit. In contrast, non-zero-sum describes a situation where the interacting parties' aggregate gains and losses can be less than or more than zero. A situation where one's win does not necessarily mean another's loss, and one's loss does not necessarily mean the other party wins. Finally, the mode of play can be mainly Cooperative and Non-Cooperative.

In this thesis, the system the players will interact with is defined by a Cartesian impedance model (see chapter 3). The human and the robot are considered as two players. The forces they can apply to the Cartesian impedance system are their actions, and a specific cost function will define their goals. Moreover, in this thesis, we consider a particular type of system, called Differential/Dynamic, that describes the evolution of a system that depends on time. Throughout this thesis, both cooperative

[1] For example, chess is a full-information game because the game's current state is fully known to both players as they make their decisions. In contrast, Poker is a partial information game.

and non-cooperative interactive models are addressed and implemented for pHRI applications. It will be shown that, in some situations, cooperation is more desirable since it allows the robot to assist the human. In some others, the non-cooperative situation is adopted to let the robot pursue its own goal, which can be in contrast to the human goal. In figure 1.5, a schema of the possible situation found in Differential/Dynamic is visible.

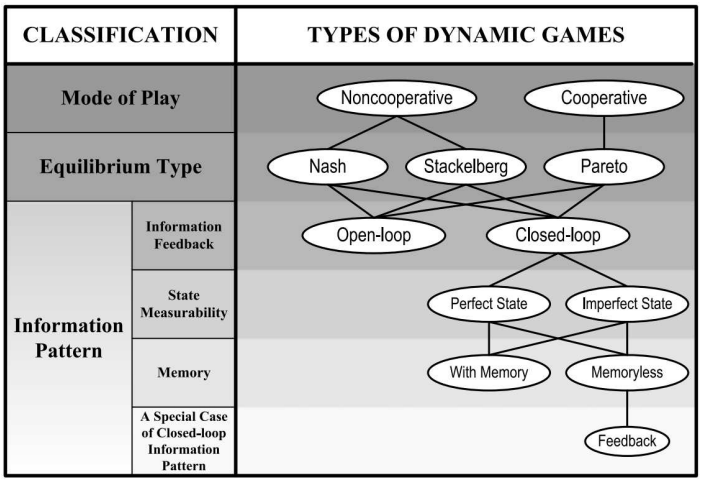


Figure 1.5: Classification of Differential/Dynamic games according to [120]

1.3 MOTIVATION

The motivation behind this work is that, despite being widely investigated, the HRI, particularly the pHRI, still requires many advancements and studies before real industrial, easy-to-use applications can be deployed in real-world scenarios.

Without any claim of solving once and for all the problems mentioned above, this thesis wants to present some advancements in the pHRI fields with potentialities for future industrial applications.

The motivation for Game-Theoretic modeling is that it models interactions among different decision-makers. Therefore, the pHRI modeling of the interaction within the GT framework is worth investigating.

In particular, a significant focus is given to the human being. Some aspects are required for the robot to assist and adequately interact with the human. Therefore, modeling human behavior and understanding its intentions are investigated. Instead, some other aspects are crucial for the human acceptance of a robotic teammate. Therefore, this thesis also investigates humans' subjective perception of the interaction, proposing questionnaires to evaluate the various aspects and also an optimization of the robot's parameters based on personal preferences.

1.4 CONTRIBUTION

The main contributions of this thesis in the human modeling field are: (i) modeling human behavior as a full-state linear feedback control, (ii) Inverse Optimal Control (IOC) to understand the fundamental behavior of humans interacting with a variable, passive system, (iii) modeling and prediction of human intention in pHRI tasks.

In particular, modeling human behavior as a full-state linear feedback control neither investigates nor considers the complex dynamics of the human arm, as other works do. Despite this simplification, this model can explain human behavior, particularly in interaction with a machine or a robot, and is relevant and useful from a control point of view, particularly for direct and inverse optimal control problems and game-theoretic modeling. Note that such a model does not consider any additional external contact on purpose. It is meant to solve applications of human-robot co-transportation of large or heavy objects. Therefore, it models only the free motion in the space at the current state of the study. The contribution is to propose an implementation of the EKF as an identification method for human control gains in pHRI tasks. Concerning previ-

ous works, We investigate the possibility that such gains are time-variant, possibly quickly. We present simulation results to support our choice and real experiments to show feasibility in real-world applications. The main result of modeling humans is to prove that linear feedback control can opportunely describe human behavior in pHRI tasks. This result opens the way to control design and inverse optimal control to identify humans' objectives.

This thesis also aims to implement Inverse Optimal Control (IOC) to understand the fundamental behavior of humans interacting with a variable, passive system. It is possible to obtain basic yet valuable information to develop natural pHRI controllers. The simple LQR model is studied. Indeed, despite its simplicity, as shown, LQR cost functions can capture essential human behaviors during the interaction. Moreover, modeling the human cost function as the LQR cost function allows easy integration in the Linear Quadratic Game-Theoretic (LQGT) framework, which provides useful tools for pHRI controller design.

Finally, defining human intention as the desired motion intention of a human over a finite rolling prediction horizon, this thesis proposes a learning model with a novel training procedure based on iterative training and Transfer Learning to make such a prediction. First, an iterative training procedure allows adapting the model to improve the prediction error and provide proper assistance. After that, transfer learning is proposed to address the time issue related to handling new situations. Such a model is integrated into the GT-dMPC framework proposed by this thesis.

This thesis also presents contributions to Game-theoretic modeling of the pHRI. Previous works in the literature deal with implementing game-theoretical frameworks to describe and manage pHRI, focusing on the Non-Cooperative models. As described in the previous section, it turns out that cooperation improves the outcome for the players. Therefore,

this thesis wants to investigate Cooperative models based on Cooperative Game Theory.

Specifically, in such an application field, the contributions of this thesis are: (i) to study the solutions of the LQ-CGT and the LQ-NCGT games to see the game's different behaviors according to different tuning parameters; (ii) to give insights into the game-theoretic description of pHRI tasks and prepare the way to develop adaptive controllers based on the GT formulation, capable of switching from one model to the other according to role arbitration logic; (iii) to analyze such games according to parameter tuning; (iv) to propose a Game-Theoretic distributed Model Predictive Control (GT-dMPC) to describe the pHRI; (v) to solve the infinite horizon LQ-CGT case (as a minor contribution).

Moreover, in general, pHRI studies consider the robot to be a passive assistant of the human, with follower capabilities. Still, no initiative is given to the robot to lead a task. This thesis wants to investigate all these situations where the human is leading, and the robot must assist, or conversely, the robot drives the system far from unwanted situations and all the possible situations in between to have a smooth transition. Motivated that humans and robots should cooperate and take advantage of cooperation, this thesis aims to realize a control framework for natural and mutual collaboration. In this sense, this thesis defines Role Arbitration frameworks based on differential game theory to allow pHRI.

Finally, since each person has different preferences during interaction, this thesis also proposes a tool for tuning the relevant parameters to make such interaction smooth and pleasant to users. The main contribution in this field is presenting a method for tuning a pHRI controller based on the preferences of different subjects. Such a procedure must be fast and easy. Moreover, interesting results can be observed as general human preferences. We tested our methodology on two different tasks, one requiring precise path following and the other requiring a fast and large motion

toward a target position. The data are analyzed to see general human behavior and preferences. Finally, a questionnaire is proposed to the users to check this method's applicability in real environments, involving time required for tuning and satisfaction.

1.5 THESIS OUTLINE

The content of the thesis is divided into eight chapters and an appendix.

Chapter 1 introduces the topics of the thesis, presenting motivations and contributions and a list of publications related to the topics presented in the present thesis.

Chapter 2 reviews the most relevant work on which this thesis takes inspiration, on the various topics addressed. First, the physical Human–Robot Interaction is reviewed, focusing on Game-theoretical modeling of the pHRI problems. Then, the human behavior modeling is presented, addressing how it can be modeled as a feedback controller by an Inverse Optimal Control approach and by intention of motion. Moreover, the Game-theoretic modeling of human interaction is reviewed, as it represents the basis for most of the work presented in this thesis. The main aspects related to shared control and role arbitration are presented. Finally, strategies for Preference Based Optimization are reviewed.

Chapter 3 presents the Cartesian Impedance Control, which represents the basic system of the entire thesis in its continuous and discrete formulations.

Chapter 4 presents the modeling of human behavior addressed in this thesis in all its formulations. It presents an Extended Kalman Filter to recover control gains, Inverse Optimal Control to recover human cost function, and deployment and training of a Learning model to predict desired human intentions. All the methods are introduced with a problem formulation, the methodology is detailed and simulated, and experimental results are finally presented.

Chapter 5 presents the modeling of the pHRI in a Game-Theoretic framework. Both Cooperative and Non-cooperative models are addressed with continuous and discrete formulations. The system performances are analyzed with simulations to understand the behavior.

Chapter 6 presents the implementation of the methodologies proposed in this thesis to solve different pHRI tasks. It presents applications for human and robot co-handling large/heavy components, soft components, and Role arbitration methodologies.

Chapter 7 presents a method to tune the controller parameters according to subjective preferences.

Finally, chapter 8 presents the conclusions that can be drawn from this thesis and possible future developments of the methodologies presented in this work.

The thesis concludes with a bibliography and an Appendix with additional material and useful mathematical formulations.

1.6 PUBLICATION LIST

This section presents a list of the work published in international conferences and journals related to the topics presented in this thesis.

Journal articles:

1. **P. Franceschi**, N. Pedrocchi and M. Beschi, "Human–Robot Role Arbitration via Differential Game Theory," in IEEE Transactions on Automation Science and Engineering, doi: 10.1109/TASE.2023.3320708.

2. **P. Franceschi**, N. Pedrocchi, M. Beschi, "Identification of human control law during physical Human–Robot Interaction", Mechatronics, 2023,
   doi: https://doi.org/10.1016/j.mechatronics.2023.102986.

Conference proceedings:

1. **P. Franceschi**, F. Bertini, F. Braghin, L. Roveda, N. Pedrocchi and M. Beschi, "Learning Human Motion Intention for pHRI Assistive Control," 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 2023, pp. 7870-7877, doi: 10.1109/IROS55552.2023.10342014.

2. **P. Franceschi**, M. Beschi, N. Pedrocchi and A. Valente, "Modeling and Analysis of pHRI with Differential Game Theory," 2023 21st International Conference on Advanced Robotics (ICAR), Abu Dhabi, United Arab Emirates, 2023, pp. 277-284, doi: 10.1109/ICAR58858.2023.10406758.

3. **P. Franceschi**, M. Maccarini, D. Piga, M. Beschi and L. Roveda, "Human Preferences' Optimization in pHRI Collaborative Tasks," 2023 20th International Conference on Ubiquitous Robots (UR), Honolulu, HI, USA, 2023, pp. 693-699, doi: 10.1109/UR57808.2023.10202313.

4. **P. Franceschi**, N. Pedrocchi and M. Beschi, "Inverse Optimal Control for the identification of human objective: a preparatory study for physical Human-Robot Interaction," 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), Stuttgart, Germany, 2022, pp. 1-6, doi: 10.1109/ETFA52439.2022.9921553.

5. **P. Franceschi**, N. Pedrocchi and M. Beschi, "Adaptive Impedance Controller for Human-Robot Arbitration based on Cooperative Differential Game Theory," 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 2022, pp. 7881-7887, doi: 10.1109/ICRA46639.2022.9811853.

CHAPTER 2

RELATED WORKS

This chapter reviews the relevant works in the fields related to the work proposed in this thesis. First, it presents an overview of the most relevant and interesting works in the physical Human-Robot Interaction field. Then, it analyzes different methodologies to infer human intentions and modeling. Finally, it investigates how shared autonomy and role arbitration are addressed in the literature.

2.1 PHYSICAL HUMAN–ROBOT INTERACTION

HRI has been growing as a research field in recent years, considering the need for collaborative manufacturing tasks shared between humans and robots within modern factories[62, 138]. In particular, in the field of HRI, Human-Robot Collaboration (HRC) deals with a collaborative approach that allows the robot and the human operator to perform complex tasks together, with direct interaction and coordination [171]. When the interaction and collaboration between humans and robots become physical, We are dealing with the physical Human-Robot Interaction (pHRI) [24].

A widely used technique to handle pHRI is Impedance/Admittance

Control [45, 51]. This type of controller, proposed initially by Hogan [51] lets the robot manipulator behave as an equivalent mass-spring-damper system when subject to external forces. It spread in the pHRI field because it is intrinsically compliant, making the interaction smooth and safe. Many studies aimed at making the Impedance Control adaptive to improve interaction performances. Two main methods exist to make Impedance Control adaptive: (i) modify the impedance set-point, and (ii) modify the impedance parameters, *i.e.,* the mass, spring, and damper values. Examples of the adaptation of the impedance set-point according to interaction with a human can be found in [101], where optimal trajectory deformation is studied, in [44] that uses a Neural Network to identify the set-point, in [158] with a nonlinear model reference adaptation, in [146, 147], where a fuzzy logic control updates online the set-point, in [46] the reference trajectory is shaped to ensure it is within the constrained task space, and in [103] with application to teleoperation. Adaptation of the mass-spring-damper parameters is exploited as in [26, 34, 117] and [177]. Recent works aim to modify both impedance parameters and set-point simultaneously. In [145], a hybrid controller allows manual guidance for a robot to assemble an aircraft panel. In [149, 150], Reinforcement Learning updates the parameters online, while [91] exploits a neural network (NN) to update the desired position and the impedance parameters to maintain stability. In [86], a controller that adapts impedance parameters and velocity is proposed, allowing interaction between a human, a robot, and the environment.

The main drawback of the abovementioned approaches is that the robot always represents a passive helper to the human. Many situations require Shared Autonomy (SA) and Shared Control (SC) of the task. [156] presents a brief survey on SA in pHRI. In this situation, the robot and the human can interact with the controlled system differently, possibly switching roles. Consider that many tasks may require the robot to

lead the action away from unwanted situations, taking control over the human in circumstances such as singularity proximity, joint limits, and robot workspace boundaries that may not be visible to a human. In this case, SA relates to Role Arbitration (RA), which can be defined as the mechanism that assigns the control of a task to either the human or the robot [100].

Target applications can be co-manipulation of large objects, such as aeronautical components, as in [145]. In the case of large object manipulation, the precise positioning at the target pose is imposed by assembly tolerances, but the connecting trajectory from the picking pose to the target pose can be adjusted [37, 38]. It may also happen that large objects can occlude the human operator sight, and the robot should prevent collisions with the environment [10]. Figure 2.1 shows a typical large component co-manipulation.

Another application can involve flexible material co-transportation as in [124, 164]. Similarly to the previous case, a precise position with respect to the target pose is sometimes needed to match exactly the component's design, such as in composite material draping. Otherwise, structural and aesthetical properties drop. Still, the connecting trajectory can be modified during transportation, with the constraints imposed by the robot limitations and target precision requirement. Other target applications involve the teleoperation of robots, such as in [71], where the human remotely operates a robotic arm. In that scenario, it is possible that the human cannot see the scene clearly because of occlusions, and the robot should take control [126], or the human guides at a high level and the robot provides trajectory correction[176].

This work wants to investigate Role Arbitration for pHRI to solve such applications. In particular, we rely on Impedance Control. Since making Impedance Control can be risky and complex and may lead to instability [79], We want to keep constant the low-level impedance con-

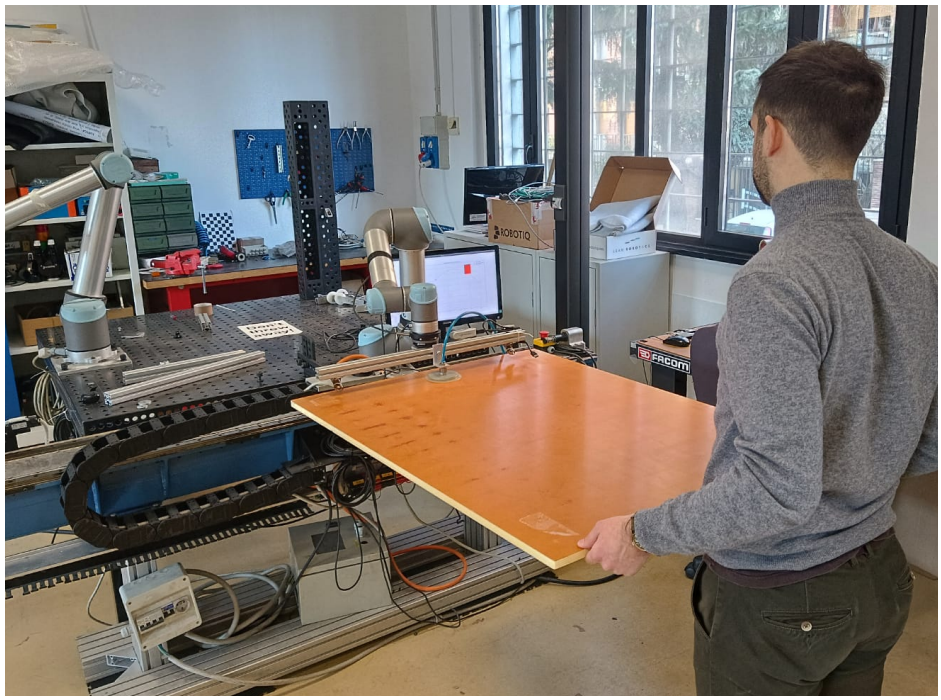


Figure 2.1: The typical large object co-manipulation scenario. On one side, a lightweight robot is grasping a large object. The robot cannot hold the object alone, as it has a limited payload and workspace. On the other side, a human is grasping the same object, with possibly the same problems. In this situation, the robot can guide the human to approach precisely a target pose without exceeding its payload.

trol parameters and consider it as a given system on which the human and the robot can interact with. Because Game Theory provides mathematical models of strategic interaction among players, We also want to investigate the Game-Theoretical formulation of the human and the robot interacting with the mass-spring-damper system given by the Impedance Control. Therefore, the following subsection is dedicated to the review of the related works regarding (i) Game-Theoretical formulation of human behavior, (ii) Game-Theoretical formulation related to the specific pHRI

applications, and (iii) Shared Autonomy and Role Arbitration applications and methods.

2.1.1 GAME-THEORY IN PHRI

Since GT represents a powerful framework to describe interactions and is capable of describing human interaction, applications of GT modeling to pHRI control are arising in the literature. The concept of Nash Equilibrium is exploited in [93], and similarly, [16] to update the robot cost function with unknown information about the human based on the interaction force. The same differential non-cooperative game-theoretic modeling is also proposed in [95] and [94]. These works exploit policy iteration to update the robot's cost function similarly compared to the previous two works presented. An observer to estimate the opponent control law is presented in [89] for the two-player game and in [184] for the multiple agents game. These works propose a universal game-theoretical framework that addresses various game-theoretical behaviors under certain control parameter tuning care. Extension to these works is in [114], where the trajectory tracking problem is addressed in the non-cooperative scenario. Finally, the authors also investigated the cooperative scenario in [40], where the weighting factor is made variable to allow the adaptive impedance behavior of the robot.

2.2 HUMAN BEHAVIOR MODELING

This section reviews the main works related to modeling human behavior. In particular, this thesis relates to modeling the human as (i) a feedback controller, (ii) an optimal controller, (iii) modeling its intention as a prediction of desired motion, and (iv) modeling human interaction with Game Theory.

2.2.1 HUMAN MODELING AS FEEDBACK CONTROLLER

It was shown that humans understand each other's intentions while physically interacting to perform a task [167]. This suggests that such a capability can also be beneficial while humans and robots interact. On the one hand, the robot's intentions are known and pre-programmed, and the human has the natural capabilities to understand them. On the other hand, this is not true for the robot, which requires some method to identify the human control law. Hence, methodologies can be defined to make the robot capable of understanding human behavior. Human intentions identification is relevant in many different fields of robotics, spanning from mobile robots [76, 130] to industrial manipulators [127, 134]. On the one hand, artificial intelligence methods are promising [23]. On the other hand, they typically require huge amounts of data. Therefore, model-based techniques are still widely used and investigated.

A widespread field of research focuses on identifying human models and human control laws. Many studies investigate the identification of the human model and its control law coupled with a controlled dynamical system subject to fast changes, with applications mainly focused on manual control [111]. These works model the human with a transfer function composed of equalization parameters (position and rate gains) of the human controller model and time delay and neuromuscular parameters and study how it varies when a fast change modifies the controlled dynamical system, focusing on the identification of the transfer function parameters.

Among these works, in [136], the authors use a linear model for the human dynamics and a dual Extended Kalman Filter (EKF)[166] to estimate both state (gain and delay) and dynamics (frequencies and damping) parameters. A similar approach with an Unscented Kalman Filter (UKF) is used in [143] to understand how gains vary according to the varying dynamics of the controlled system dynamics.

These works show the interesting fact that linear control can describe human actions. Despite this, involving a model identification of the human dynamics makes the problem complex and applicable only to specific applications (e.g., manual control of an aircraft), restricting the usage to a small range of applications or frequencies. On the contrary, for shared tasks during pHRI, identifying the human control law, described as a full state feedback controller as in [89, 90], might be sufficient and more general, avoiding the introduction of complex dynamical models that require more complex model identification. In this case, the human control action is described by a gain matrix, which multiplies the system's state.

Techniques to identify the gain matrix of linear system models appear in the literature. These works focus on identifying only the linear feedback control law that produces an observed output rather than a complete system identification. In [140], the control law described as full state feedback gain matrix is estimated by a maximum-likelihood estimator (MLE) via minimizing the least-squares error for a linear state-space model, while in [33], the gain matrix is computed by minimizing the normalized residuals for the same linear state-space formulation. In [108], the Least Squares (LS) estimation method is suggested for the identification. Such methods have the drawback of requiring complete control and state histories to compute a constant gain matrix, making such an approach not applicable to online identification.

Moreover, following game-theoretical nomenclature, identifying with *player* an agent which performs control actions on the system, these works consider only the one-player case while performing the identification in pHRI applications involves two (or more) players, namely, the human and the robot.

When more than one player is considered, for linear systems, their control action can still be described by linear feedback gain matrices, one

for each player.

The LS formulation appears in [55] for the two-player game. Again, it displays limitations for online applicability and considers constant gain matrices for both players.

Recursive Least Squares (RLS) are used in [184] to identify the human control law online for human-robot-human co-handling of an object. [89] proposes an observer with an updated law based on a game-theoretical formulation of the interaction for online identification. Such approaches [89, 184] show good results and convergence in simulation but assume constant gain matrices, both for the human and the robot, which result from a Nash Equilibrium in a game-theoretical description of the interaction. This comes from the assumption that a human behaves as a rational player in a non-cooperative game and keeps his control law constant when an equilibrium is reached. Similarly, in [90], a UKF is used for online gain matrix identification, still considering constant gains, computed as minimization of a constant cost function.

2.2.2 INVERSE OPTIMAL CONTROL

Many techniques and models are studied and presented in the literature to describe human and human-machine behavior. Such models span from elementary models (humans as a spring) to complex neuromuscular models and Optimal Control models, depending on the need. Please see [155] for a complete and up-to-date review.

Provided that Game Theory (GT) represents a robust framework to describe an individual's behavior during interaction [63], this work focuses on the identification of optimal behaviors of humans and the recovery of the human cost function, as this modeling allows direct integration into GT frameworks. Optimal Control (OC) aims to find a control action for a dynamic system over a time window to minimize a cost function. Conversely, Inverse Optimal Control (IOC) techniques are adopted to re-

cover the cost function that produced observed control actions and state histories.

The most common strategy used in IOC is to parametrize an unknown objective function as a weighted sum of relevant features (or basis functions) with unknown weights [1]. This model is also adopted to describe human behaviors in performing different tasks. Human jumping is studied in [173], highlighting the possibility that the cost function varies during the task. In [67] and [97] study human movement by detecting changes to the optimization criterion during a squat task. Human arm reaching is studied in [129], considering free-space reaching motions. The results show a trade-off between kinematics and dynamics-based controllers depending on the reaching task. Interestingly, this trade-off depends on the initial and final arm configurations. [104] also studies human arm motions with applications to human-robot collaboration in a shared workspace. These works involve cost functions with multiple features but consider only motions in the free space without accounting for interactions.

If the only cost function's features considered are the state and control action, the problem is reduced to a typical Linear Quadratic Regulator (LQR). In this case, more specific IOC techniques exist. Such techniques compute the weight matrices starting from the feedback matrix and not the complete control and state histories[108] [64] [55]. Interestingly, such methods do not require the entire trajectory to identify the cost function, but only the knowledge of the gain matrix is sufficient. Finally, such approaches appear suitable for studying humans in interaction with autonomous agents, as described by Game Theory, as in [58, 77, 144].

Motivated by the previous studies, this work aims to implement IOC to understand the fundamental behavior of humans interacting with a variable, passive system. It is possible to obtain basic yet valuable information to develop natural pHRI controllers. The simple LQR model

is studied. Indeed, despite its simplicity, as shown, LQR cost functions can capture essential human behaviors during the interaction. Moreover, modeling the human's cost function as the LQR cost function allows easy integration in the Linear Quadratic Game-Theoretic (LQGT) framework, which provides useful tools for pHRI controllers design [89, 93, 94].

2.2.3 HUMAN INTENTION DETECTION

The two main research branches that aim at understanding the desired human intention of motion rely on (i) model-based and (ii) data-driven approaches.

The model-based approach builds on modeling the human arm with the standard impedance model [32, 159]. This approach requires estimating the human arm impedance parameters. The main drawback of this approach is that such parameters are subjective, time-variant, and depend on the specific task considered. For example, lifting an arm on the vertical plane has different parameters with respect to opening the same arm on the horizontal plane, and so on. Therefore, such a physical human arm model presents low flexibility and generalization. A different approach is to implement a more control-oriented modeling of the human rather than a physical model, considering it as a feedback controller [42, 137], with control gains as unknown parameters to be recovered. This approach still requires a control model that might introduce modeling errors and parameter estimation techniques. Finally, in [141], the human intent is obtained by double integration of the estimated acceleration imposed on the robot admittance control.

The data-driven approach aims at training Machine Learning algorithms based on data (either real or synthetic) collected and builds a model that transforms inputs into outputs. Among the various, Neural Networks (NN) achieve excellent results in approximating complex non-linear systems with high uncertainties. In particular, when dealing

with sequences (either logical or temporal), Recurrent Neural Networks (RNN) are a very effective implementation of NN. Indeed, RNNs account for previous events, allowing information to persist over a certain horizon. There are various types of RNN [160], vanilla RNN, Gated recurrent unit (GRU), and Long-Short Term Memory (LSTM) are the most common types. In particular, the LSTM [49] architecture outperforms the classical RNN and is now widely adopted to solve multiple problems where the sequential pattern of the data may store important information, such as speech recognition. The same reasoning applies to human intent estimation since the previous motion state may be a manifestation of the intent.

Indeed, various works adopt RNN architectures for the prediction of human motion. Vision-based data are usually exploited [109, 180, 181], as cameras provide a clear understanding of the scene and are able to detect the human skeleton without requiring any contact to feed both LSTMs and GRUs models. [172] presents the AutoRegressive Integrated Moving Average (ARIMA) model for the visual identification of the elbow motion of a human. The main drawback of using vision-based information is that specific hardware is required, and image data are quite complex to be processed, increasing the time required for algorithms training.

To overcome camera-related issues, other works exploit different data information. In [6], the input data to the LSTM model are read via Force MyoGraphy (FMG). Multiple subjects are used to train the model, but the approach is not general. A different sensor based on electromyography (EMG) signals acquired from human arm muscles is used in [165]. In this work, the authors propose using a NN to classify the intended direction of human movement. This work limits to the classification of the desired direction of motion to allow the robot assists. Despite this, no prediction of future motion intentions in terms of desired trajectory is addressed. Also, Gaussian processes can be used such as in [99] and [107]. The first proposes identifying human motion intention interacting

with an exoskeleton via a sparse Gaussian process. The second models the human arm as 7 degrees of freedom (DoFs) impedance and estimates the intention as the human force by Gaussian Process.

In [52], an Adaptive Neural Network estimates the joint coordinates of the human lower limb interacting with an exoskeleton for rehabilitation. In contrast, [21] predict human motion intention by online learning without training. In [91], the model is based on Radial Basis Function Neural Networks (RBFNN). In this work, an updating law adjusts the NN weights online to guarantee estimation accuracy even when human motion intention changes. Still, the prediction horizon is one step. Interestingly, [105] and [102], propose an LSTM to predict the reference setpoint at the next step but do not address any adaptation to new users or objects. In addition, [102] proposes training and using LSTM to predict the reference one time instant but does not adapt the model to new users, so each new human has to record the full set. Finally, one of the author's previous works addresses a model composed of LSTM cascaded with Fully Connected (FC) standard NN layers [35]. Such a model is iteratively trained to adapt the model to understand the interaction better, and Transfer Learning (TL) is proposed to adapt the same model to new users, trajectories, or co-manipulated objects, showing the capacity of generalization and long-term prediction.

2.2.4 GAME-THEORY FOR MODELING HUMAN BEHAVIOR

Let's now review how human behavior can be modeled using GT formulations and which modes have been studied to understand how they can be included in the pHRI framework. Game-theoretic modeling of humans interacting with machines has been increasingly exploited recently. Typical applications of GT models are used for humans interacting with a programmed active front steering (AFS) in the shared driving of vehicles (Non-Cooperative [119], Cooperative [121], and others [65, 120]).

Such works are limited to modeling interaction and lack experimental data to confirm that GT models describe interaction with real humans. Experimental evaluations are presented in [122] and in [118]. The first work presents results on the behavior of six individuals interacting with the AFS. Compared with the standard optimal control, it is shown that their interacting behavior is better described by the Non-Cooperative formulation. Similarly, the second shows that GT-based formulation better describes the driver's behavior than the driver's classic steering control strategy.

Studies on a human-human dyad, rather than human-machine, allow an understanding of how their behavior changes according to the interacting situations. In [58], a human-human dyad performs a shared reference tracking without the possibility of communicating. It shows that the non-cooperative model is more descriptive than a model that considers the partner's action as a system disturbance. Human-human interaction is also studied in [17] and [98]. These works compare the dyad with a single individual behavior. Individual players tended towards cooperative behavior to find the solution, whereas two players tended towards non-cooperation on average. Reasonably, the cooperative solution is best if the players can communicate and trust each other. If no agreement exists because there is no communication and trust, human behavior can be modeled as non-cooperative. Note that these works put the participants in non-cooperative situations on purpose (human dyads are not allowed to communicate, and the machines the humans are interacting with are programmed in a non-cooperative way).

2.3 SHARED AUTONOMY AND ROLE ARBITRATION

Finally, we want to present the most common and recent advancements in the RA and SA fields to give a background to the proposed work. The most common way to describe Shared Autonomy (SA) and Shared Con-

trol (SC) is a linear combination of human and robot control inputs as $h(u_h, u_r) = \alpha\ u_h + (1 - \alpha)\ u_r$. This formulation is used, possibly with different definitions of the u_i, in many of the following works. In [156], Shared Autonomy and Shared Control in pHRI are reviewed, highlighting the differences, pros, and cons. Ultimately, SC is more specific to the application domain and task. At the same time, SA approaches provide a favorable autonomy level adaptation feature that leverages inference of human intentions. In [116], authors investigate the objective and subjective effects of dynamic role allocation for a physical robotic assistant for a cooperative load transport task. Two dynamic Arbitration laws are proposed for effort sharing. A continuous first-order dynamical system governs one with the arbitration parameter bounded within the interval [-1,1], and the other is discrete. Results show that the dynamic role allocation is objectively better than its fixed counterpart. [102] proposes variable impedance control along with an assistive controller that gradually decreases to zero when the human user applies forces to pull the robot away from the predicted goal. In [88], to help a human, a strategy based on a multi-modal intent inference of human intention is developed. By looking at the natural eye-hand cooperation during a natural human manipulation, it is possible to understand the human intention of motion. The environment and the estimation introduce uncertainties, so a confidence index on the identification is defined. The arbitration weight of the robotic agent is defined as a combination of confidence in the intent inference and robotic autonomy. Role arbitration is presented in [93] and [94] as a Variable Impedance Control, where human intentions are detected from interaction force. In [18], the robot assumes two roles, with a control scheme that switches between a teaching phase (the robot is a passive follower) and an active phase (the robot is in adaptive admittance control). In [175], a controller capable of learning human behavior and providing assistive or resistive force is proposed, but no

dynamic role allocation is proposed. Similarly, [110] propose a switching controller based on assistive adaptive impedance control. The stiffness parameter goes from high values to zero. This allows trajectory tracking and autonomous task execution in the first case and manual guidance in the second. In [87], a fuzzy controller introduces a cooperative coefficient according to different driving intentions, safety, and performance parameters in a cooperative driving scenario. In [66], the shared autonomy problem for the human-robot collaboration is introduced into a Partially Observable Markov Decision Problem. In [73], the Cooperative problem is addressed for the human-driver assistant problem. Finally, [74] propose a GT formulation of the problem, allowing switching between the Cooperative and the Non-Cooperative interaction models for collision avoidance in a shared human-robot driving scenario. Despite the interesting problem formulation and arbitration solution, only simulation results are presented.

2.4 PBO

Given the control framework, different parameters have to be tuned. In particular, the control algorithm typically has parameters that depend on the user, which can only be estimated, and parameters that can be set according to the required task, performance, and user experience level. For example, according to previous research, [28, 53, 84], operators prefer low values of the impedance parameters for large movements at high velocities, while high values when performing fine movements at low velocities. In [34], these insights are used to make a variable admittance control. The damping parameter is made variable according to the velocity of the robot within a stable region.

Moreover, human subjective preferences may influence the tuning of the controllers. Different works analyze this crucial aspect of the HRI, even in domains not directly related to pHRI. The works in [4, 5] propose

tuning robotic prosthesis control parameters to meet the user's preference while ensuring their safety. [27] evaluates the participants' perception of the collaboration, and [50] evaluates human-robot collaboration's fluency based on subjective and objective evaluations. [154] defines a questionnaire to try to remove the subjective part from evaluations. Finally, [25] defines subjective metrics to assess an HRI motion planner. Therefore, it is clear that a method for control parameter tuning should be defined according to the various requirements, not least human preferences.

A novel method to optimize parameters based on human preferences is the so-called Preference Based Optimization (PBO). PBO aims at optimizing a set of parameters where a cost function cannot be directly evaluated but only recovered from preferences expressed by a user. By iteratively proposing to the user a new comparison to make, the algorithm learns a surrogate of the cost function preferences expressed by the user. Among the various, in this work, We selected the GLISp algorithm [15] to tune the robot controller's parameters. Such an algorithm has proven great results in tuning control algorithms based on human preferences as MPC controller calibration for line keeping, obstacle avoidance, continuous stirring tank reactor [182, 183], and path-based velocity planner for tuning material deposition in [148, 151].

CHAPTER 3

IMPEDANCE CONTROL

This chapter introduces the basic Cartesian Impedance control for the robotic manipulators, representing the base interactive system. By implementing Cartesian impedance, the robot motion at the end effector behaves following a mass-spring-damper system dynamics, which is generally considered to be compliant and safe for interaction with humans and the environment. Therefore, in this thesis, the basic, low-level system is considered to be an equivalent mass-spring-damper system, and the human and the robot are considered two external agents acting on it. On top of it, a Game-Theoretical framework can be constructed to let the robot understand human intention and act consequently, as explained in sections 5.2 and 5.3.

3.1 IMPEDANCE CONTROL

Denoting with $q(t)$ the vector of joints coordinates, the standard manipulator dynamics in the joint space is given by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{3.1}$$

where $M(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis and centrifugal force, $G(q) \in \mathbb{R}^n$ is the gravitational force, and $\tau \in \mathbb{R}^n$ is the torque control input. The forward kinematics gives the end-effector Cartesian pose

$$x(t) = f(q(t)) \tag{3.2}$$

where $x(t)$ denotes the Cartesian position vector and f a function that maps joint coordinates into Cartesian pose at the end-effector, and the Cartesian velocity and accelerations are given by differentiating it as

$$\dot{x}(t) = J\,\dot{q}(t) \tag{3.3}$$

where $J(q) \in \mathbb{R}^{n \times n}$ is the Jacobian matrix. It is then possible to rewrite (3.1) in the Cartesian space as

$$M_x(q)\ddot{x} + C_x(q, \dot{q})\dot{x} + G_x(q) = J(q)^{-T}\tau \tag{3.4}$$

The feedback linearization is realized by imposing the control input τ as

$$\tau = J(q)^T[M_x(q)\ddot{x}_{ref} + C_x(q, \dot{q})\dot{x} + G_x(q)] \tag{3.5}$$

leading to $\ddot{x} = \ddot{x}_{ref}$, with $\ddot{x}_{ref}$ defines the desired acceleration.

Cartesian Impedance control can be implemented to make the robot's behavior responsive and compliant with human interaction. It represents a virtual mass-spring-damper system, and the human can impose wrenches to move it. The impedance model in the Cartesian space is described as follows:

$$M_i\ (\ddot{x} - \ddot{x_0}) + C_i\ (\dot{x} - \dot{x}_0) + K_i\,(x - x_0) = u_h(t) \tag{3.6}$$

where M_i, C_i and $K_i \in \mathbb{R}^{6 \times 6}$ are the desired inertia, damping, and stiffness matrices, respectively, $\ddot{x}(t)$, $\dot{x}(t)$ and $x(t) \in \mathbb{R}^6$ are the Cartesian

accelerations, velocities and positions at the end-effector, x_0 is the equilibrium position of the virtual spring, and $u_h(t) \in \mathbb{R}^6$ represents the human wrench applied to the system. The Cartesian coordinates in x are defined according to [163], with the vector $x = [p^T \ \theta^T]^T$ where p^T are the position coordinates and θ^T the set of Euler angles. This choice assumes that the angular rotation maintains limited values in the target applications.

Since the purpose of this thesis is to design assistive controllers, it is convenient to add a virtual wrench $u_r(t) \in \mathbb{R}^6$ that represents an additional external contribution that the robot can provide, leading to

$$M_i \ (\ddot{x} - \ddot{x_0}) + C_i \ (\dot{x} - \dot{x}_0) + K_i \ (x - x_0) = u_h(t) + u_r(t) \tag{3.7}$$

The definition and computation of the additional control input $u_r(t)$ is the main objective of the Game-Theoretical framework presented in sections 5.2 and 5.3. A schema of the Impedance Control with the human and the robot acting on it is visible in figure 3.1.

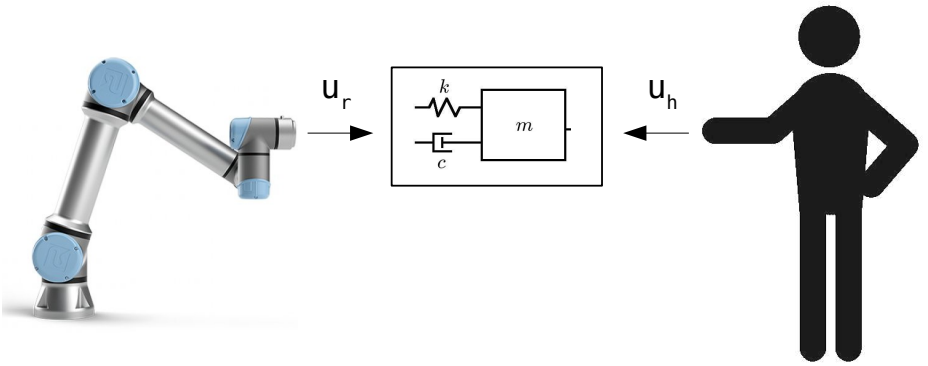


Figure 3.1: The schema of the impedance system with the two external contributions.

3.2 STATE-SPACE FORMULATION

The equation (3.7) can be rewritten in a linearized state-space formulation around the working point as

$$\dot{z} = Az + B_h u_h + B_r u_r \tag{3.8}$$

where $z = [x\ \dot{x}]^T \in \mathbb{R}^{12}$ is the state space vector, the matrix

$$A = \begin{bmatrix} 0^{6\times 6} & J_a \\ -M_i^{-1}K_i & -M_i^{-1}C_i \end{bmatrix} \tag{3.9}$$

is the state, or system matrix,

$$B_h^{12\times 6} = B_r^{12\times 6} = \begin{bmatrix} 0^{6\times 6} \\ M_i^{-1} \end{bmatrix} \tag{3.10}$$

is the control matrix, with $0^{6\times 6}$ denoting a 6×6 zero matrix and J_a the analytical Jacobian matrix, with the dimensions of the considered Cartesian components. To provide the robot with reference in the joint space, kinematic inversion is computed at the velocity level as

$$\dot{q}_{ref}(t) = J(q)^{+}\dot{x}(t) \tag{3.11}$$

where $\dot{q}_{ref}(t) \in \mathbb{R}^n$, are the reference velocities in the joint space, $J(q)^+$ is the pseudoinverse of the analytical Jacobian matrix. Joint positions are then computed via a simple integration. Assume $\dot{q} \simeq \dot{q}_{ref}$, considering that today's robots have excellent tracking performance in the frequency range excitable by the operator.

The low-level Impedance Control loop described in (3.7) can be seen at a higher level as a given system, with two agents acting on it, the human and the robot, namely.

3.2.1 DISCRETE-TIME FORMULATION

Finally, since the robot controllers accept commands in discrete time, and data are typically collected in discrete time, it is also convenient to rewrite the system described in 3.8 in discrete time.

$$\begin{aligned} z(k+1) &= A_d\, z(k) + B_{d,h} u_h(k) + B_{d,r} u_r(k) \\ y(k) &= C_d\, z(k) \end{aligned} \tag{3.12}$$

with A_d, $B_{d,h}$ and $B_{d,r}$ indicating the discrete versions of the matrices A, B_h and B_r, and k indicating the current time instant, $z(k+1)$ the evolution of the system at the next step, and C_d the output matrix that converts $z(k)$ to $y(k)$.

The method used in this work to convert from continuous to discrete systems is briefly explained in appendix A.1.

CHAPTER 4

MODELING HUMAN BEHAVIOR

This chapter investigates various aspects of human modeling, which must be addressed for a proper Game-Theoretic description of the pHRI. First, it is convenient to consider human wrenches as the output of a linear feedback controller. Therefore, the first section is dedicated to recovering the online human feedback control gain matrix by adopting an Extended Kalman Filter (EKF) for parameter identification. Then, provided that the human control inputs can be modeled as the output of a linear state feedback control, it is interesting to model how such feedback gains are generated. Taking inspiration from the fact that, in general, humans tend to optimize something when they engage in activities, Inverse Optimal Control (IOC) is adopted to approximate the cost function that humans minimize during a reaching task with the robot. Finally, predicting the intention of motion of humans is a necessary piece of information for the robot to assist appropriately. This thesis proposes the training and using a Recurrent Neural Network (RNN) capable of predicting the desired human motion within a defined rolling horizon.

4.1 IDENTIFICATION OF HUMAN CONTROL GAINS VIA EXTENDED KALMAN FILTER

This section is based on the method and results presented in [42].

This work extends the previous literature by addressing the online identification of a human's possibly time-varying control gains while physically interacting with a robot. Indeed, it is reasonable to assume that a human modifies his control law during a task. Therefore, the objective is to define an online method capable of identifying the time-varying control gains of the human, which can explain the coupled system behavior while keeping the human model simple.

A modification of the LS technique proposed in [55] and the RLS method from [184] are implemented for comparison with the proposed EKF. The usage of EKF is less computationally expensive than a UKF estimator, as the Jacobians can be computed analytically offline. The quality of the model tracking is good since the system is close to linear, as experiments demonstrated (see Discussion) EKF and UKF were experimentally tested, and no visible improvements were found using the UKF. Moreover, the tuning of the UKF was found to be more complex. Simulation results show that the EKF approach outperforms the LS and RLS methods in the accuracy of the identification of the control law, mainly when time-varying control laws are considered. Experimental results show that the identified laws can explain and reproduce human behavior.

This section, as others [56, 92], wants to investigate human behavior modeling as a full-state linear feedback control. Such a model is limited to the human's control behavior and does not investigate nor consider the complex dynamics of the human arm, as other works do. Despite this simplification, this model can explain human behavior, particularly in interaction with a machine or a robot, and is relevant and useful from a control point of view, particularly for direct and inverse optimal control problems and game-theoretic modeling. Note that such a model does not

consider any additional external contact on purpose. It is meant to solve applications of human-robot co-transportation of large or heavy objects. Therefore, it models only the free motion in the space at the current state of the study.

The main contribution of this section is to propose an implementation of the EKF as an identification method for human control gains identification in pHRI tasks. Compared to previous works, it investigates the possibility that such gains are time-variant, possibly quickly. It presents simulation results to support the EKF choice and real experiments to show feasibility in real-world applications. The main result of this section is to prove that linear feedback control can opportunely describe human behavior in pHRI tasks. This result opens the way to control design and inverse optimal control to identify humans' objectives.

4.1.1 PROBLEM FORMULATION

Consider the state-space Cartesian impedance control presented in (3.8). Due to the impedance behavior of the robot end-effector, the objective is to design a controller for the robot control input u_r to allow end-effector motion according to the desired task, as set-point reaching or trajectory following.

The robot control input is formulated as a linear state feedback control law as

$$u_r(t) = -K_r(t)z(t) \tag{4.1}$$

where $K_r(t)$ is the time-varying feedback gain matrix. Assuming that the human also implements a linear feedback control law, as in [89, 90], his/her control input results in

$$u_h(t) = -K_h(t)z(t) \tag{4.2}$$

where $K_h(t)$ is the time-varying feedback gain matrix. Note that the hu-

man control input u_h is the force the human applies to the system, which can be measured. From now on, human control effort and force are used as synonyms.

In control design, the tuning of the gain matrix is a broad research area, and techniques such as pole placement [135], which allows specifying the closed-loop poles of the controlled system, or LQR [81], which allows designing optimal controllers for desired performances, are typically used. These classical approaches can be adopted to define the robot gain matrix $K_r(t)$. Conversely, the human gain matrix must be identified.

Given the robot control law (the LQR method is used in this work), the problem is to recover the human control law parameters described in (4.2) to provide the robot with helpful information about human control. The following problem can then be formulated.

Problem: *given the current system state $z(t)$, the robot control input $u_r(t)$ computed as (4.1), and the measured human control input $u_h(t)$ assumed to be results of the linear feedback control law 4.2, given the system dynamics 3.8, identify the control gain matrix $K_h(t)$ that produced such control input $u_h(t)$.*

4.1.2 EXTENDED KALMAN FILTER DESIGN

To ease the reading, the rest of this section focuses on the one-DoF formulation of the problem. In doing this, with a bit of abuse of notation, the state vector $z = [\Delta x \ \dot{x}]^T \in \mathbb{R}^2$ reduces to two elements and Δx and $\dot{x}$ identify the position and linear velocity relative to the robot base frame on one single axis, and not the 3×1 position and velocity vectors presented in the system modeling subsection. Moreover, the time dependency of the gain matrices will be hidden despite being considered in the modeling and the experiments.

Substituting 4.2 and 4.1 in 3.8, results in

$$\dot{z} = (A - BK_h - BK_r)\, z. \tag{4.3}$$

Assuming the robot gain matrix is known, considering then z, the gain matrix K_h will be a 1×2 matrix with elements $K_h = [k_{1,h}\ k_{2,h}]$, with components $k_{1,h}$ and $k_{2,h}$ to be identified. An augmented state is defined as

$$\zeta = \begin{bmatrix} x \\ \dot{x} \\ k_{1,h} \\ k_{2,h} \end{bmatrix} \tag{4.4}$$

The filter dynamics can then be defined as

$$f(\zeta, \nu) = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{k}_{1,h} \\ \dot{k}_{2,h} \end{bmatrix} = \begin{bmatrix} \nu_{\dot{x}} \\ \frac{-(k_{1,h}+k)\Delta x - (k_{2,h}+d)\dot{x} - K_r z}{m} + \nu_{\ddot{x}} \\ \nu_{\dot{k}_{1,h}} \\ \nu_{\dot{k}_{g_{2,h}}} \end{bmatrix} \tag{4.5}$$

where the vector $\nu = [\nu_{\dot{x}}, \nu_{\ddot{x}}, \nu_{k_{1,h}}, \nu_{k_{2,h}}]^T$ accounts for uncertainties in model parameters/estimates and k and d denote the single element of the matrices K_i and D_i of (3.7). The observer for the augmented system can be defined as

$$\begin{cases} \dot{\hat{\zeta}} = f(\hat{\zeta}, \nu) + K_{EKF}(y - C\hat{\zeta}) \\ y = h(\zeta, w) \end{cases} \tag{4.6}$$

with $\dot{\hat{\zeta}}$ the augmented state estimate derivative, C the observation matrix for the robot position and velocity, and K_{EKF} the gain matrix, defined as:

$$K_{EKF} = PCR^{-1} \tag{4.7}$$

with measurements noise matrix R defined as $R = H\mathbb{E}(ww^T)H^T$. h is the observation function that maps sample inaccuracies, due to measurement noise, through the matrix $H = \frac{\partial h}{\partial w}\Big|_{\hat{\zeta}}$. The covariance matrix P is

continuously updated as

$$\dot{P} = A_a P + P A_a^T - P C^T R^{-1} C P + Q \tag{4.8}$$

with matrix $A_a = \frac{\partial f}{\partial \zeta}\Big|_{\hat{\zeta}}$, and matrix Q used for parameter estimation defined as $Q = G_a \mathbb{E}(\nu\nu^T) G_a^T = G_a V G_a^T$, with $G_a = \frac{\partial \nu}{\partial \zeta}\Big|_{\hat{\zeta}}$.

The estimated feedback control gain matrix $\hat{K}_h$ can then be reconstructed from the augmented state as $\hat{K}_h = [\hat{k}_{1,h}\ \hat{k}_{2,h}]$.

Remark 1

A very similar approach can be applied with the discrete-time system in (3.12), with the difference that, in that case, two different phases (predict and update the filter) are required.

4.1.3 VALIDATION

The presented approach is validated with simulations and real experiments. The results are presented and discussed in this section. The proposed approach is compared with two previously implemented methods: Least Squares Method (LSM) and Recursive Least Squares (RLS).

The procedure in figure 4.1 is applied to evaluate the proposed approach for both simulation and experimental results. State and control data are offline simulated/collected and recorded. Subsequently, such data are replayed, involving the "online" estimation of the time-varying gain matrix. Finally, the estimated time-varying matrix simulates the system and reproduces the original behavior.

Remark 2

The data simulation/collection and gain matrix identification steps can be performed online together. This work chooses to separate them to compare the same data sets for the three methods, avoiding real-time related issues. Moreover, in this way, it is possible to use the same data for comparison.

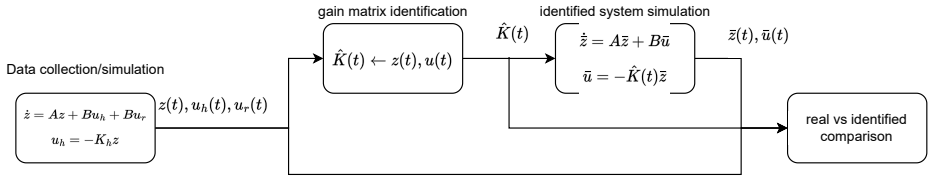


Figure 4.1: The evaluation procedure.

4.1.3.1 PERFORMANCE INDEXES

Some numerical performance indexes are defined and introduced below to compare the proposed method with others.

The human feedback gain matrix is known and available in the simulated scenario so that a direct comparison can be made between the nominal and the identified one. The mean of the percentage identification error is defined as

$$\mathcal{E}_{K_h} = \frac{1}{t_f} \int_0^{t_f} \frac{||\hat{K}_h(t) - K_h(t)||}{||K_h(t)||} \, dt \tag{4.9}$$

Conversely, the human gain matrix is unknown in real experiments. Therefore, the values obtained for the identified gain matrix are used to compute the same trajectory in a simulated environment to measure the estimation's reliability. Two additional indexes are then defined. The mean

percentage state error is measured as

$$\mathcal{E}_z = \frac{1}{t_f} \int_0^{t_f} \frac{||\hat{z}(t) - z(t)||}{||z(t)||} \, dt \tag{4.10}$$

where $z(t)$ is the measured Cartesian state and $\hat{z}(t)$ is the Cartesian state computed simulating the system (3.8) with $K_h = \hat{K}_h(t)$, estimated by 4.6.

Finally, the Root Mean Squared Error (RMSE) of the control input $\bar{u}$ is considered to measure the control input u. This is because if the system is not adequately excited and the control input is close to zero, a high percentage error may arise due to division by $\|u_h\|$, close to zero, even if the error is not significant, making the percent error comparison not reliable. The RMSE is computed as the square root of the Mean Squared Error (MSE), computed as

$$RMSE_u = \sqrt{\frac{1}{t_f} \int_0^{t_f} (u_h(t) - \hat{u}_h(t))^2 \, dt} \tag{4.11}$$

The results obtained are compared with the other two common identification techniques. The first one is the Least Square Method (LSM) similar to the one presented in [55], with the difference that [55] requires the complete history of both $x(t)$ and $u_h(t)$. In this work, due to the online purpose, the integral is computed between a rolling interval $T = [t_i \leq t \leq t_c]$ with t_c denoting the current time, $t_i = t_c - \Delta t$ the initial time and Δt indicates a fixed time interval. The matrix $\hat{K}_h$ is computed as

$$\hat{K}_h(t_c) = \min_{K_h} \int_{t_c - \Delta t}^{t_c} ||K_h \, z(t) + u_h(t)||^2 dt \tag{4.12}$$

In the ideal case, $u_h(t) = -K_h z(t)$, minimizing (4.12) allows to find a good estimate of the matrix $\hat{K}_h$.

The second method used for comparison is the Recursive Least Squares (RLS), as in [184]. The authors define $\theta = -K_h^T$, $y = u_h^T$, and $W = z^T$, with

$y = W\theta$, and estimate the values of the gain matrix with the following update rules.

$$\begin{aligned} \dot{\hat{\theta}} &= -PW^T e \\ \dot{P} &= \lambda P - PW^T WP \\ e &= \hat{y} - y \end{aligned} \tag{4.13}$$

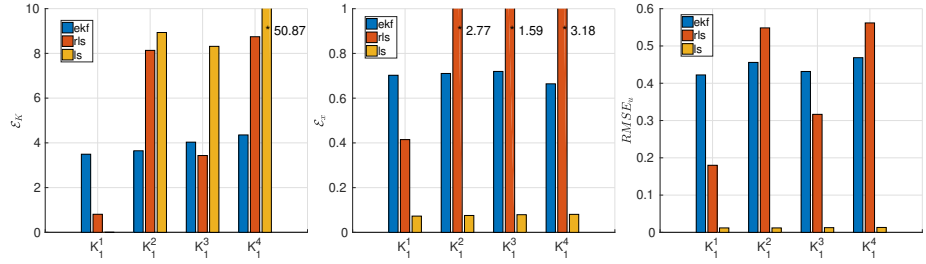
The proposed method and the others are tested first by making null robot contribution (*i.e.* $u_r(t) = 0,\ \forall t$), then adding the robot control input both in simulations and real experiments.

4.1.3.2 SIMULATION RESULTS

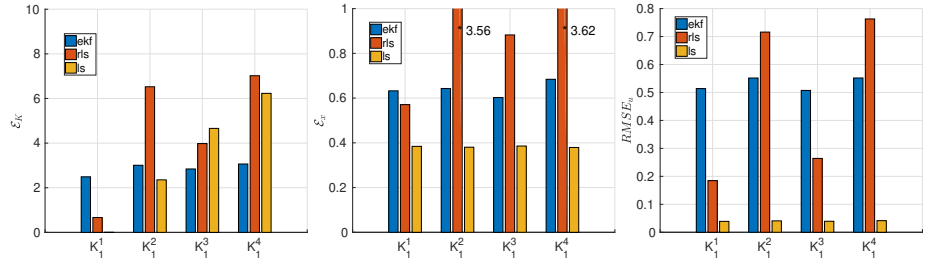
First, simulations are performed to assess the feasibility of such an approach for control gain matrix estimation.

The system presented in (3.8) is simulated along one direction only, identified as x, with $m = 10\,kg$, $d = 25\,Ns/m$ and $k = 0\,N/m$, leading to matrix $A = \left[\begin{smallmatrix} 0 & 1 \\ 0 & -2.5 \end{smallmatrix}\right]$, $B = \left[\begin{smallmatrix} 0 \\ 0.1 \end{smallmatrix}\right]$. Note that we use lowercase letters for the simulation with one DoF to denote mass, damping, and stiffness parameters because they are scalar values and not matrices. The target position is variable with time, and is $x_{ref}(t) = 0.5\,m$ until $x = x_{ref}$, then $x_{ref}(t) = -0.5\,m$ and so on. The simulated human control gain matrix is set to $K_h(t) = [50, 10]$, while, if active, the robot control gain matrix is set to $K_r(t) = [20, 1]$. Moreover, to have a time-varying behavior, the first element of the gain matrix ($k^1_{1,h} = 50$) is modified, adding a slow-changing term as $k^2_{1,h}(t) = k^1_{1,h} + 20sin(0.2t)$, a fast-changing term as $k^3_{1,h}(t) = k^1_{1,h} + 2sin(5t)$, and both $k^4_{1,h}(t) = k^1_{1,h} + 20sin(0.2t) + 2sin(5t)$. The gain parameters for the EKF are $Q_{EKF} = diag([.1,\ .1,\ 10e4,\ 10e4])$ and $R_{EKF} = diag([.01,\ .01]))$, for the RLS $P_{RLS}(0) = diag([10e2, 10e2])$ and $\lambda = 0.95$ and for the LS the selected time interval is $\Delta t = 0.05s$. Such parameters are manually tuned to minimize the error of the identified

control input. The initial guess of the human control gains is the same for the three methods and is set to $\hat{K}_h(0) = 0.3K_h(0)$. The simulation is run for 35 seconds, with a time step of 0.005 seconds.



(a) mean percentage error for gain identification (4.9) **(b)** mean percentage error for state identification (4.10) **(c)** RMS error for control input identification (4.11)



(d) mean percentage error for gain identification (4.9) **(e)** mean percentage error for state identification (4.10) **(f)** RMS error for control input identification (4.11)

Figure 4.2: Results from the evaluation of the proposed indexes. The figures are cut on the y-axis to make the most relevant results visible. The values above the y limits are indicated on the corresponding bars. The first row presents results for the $u_r(t) = 0$ case, while the second row presents results for the case in which the robot is active.

Eight simulations are run, one for each value of $k_{1,h}(t)$, without and with the robot active, and the results of the proposed indexes are summarized in figure 4.2.

The first row shows the simulation results with $u_r(t) = 0$, while the second row shows the results of the simulations with the robot contribu-

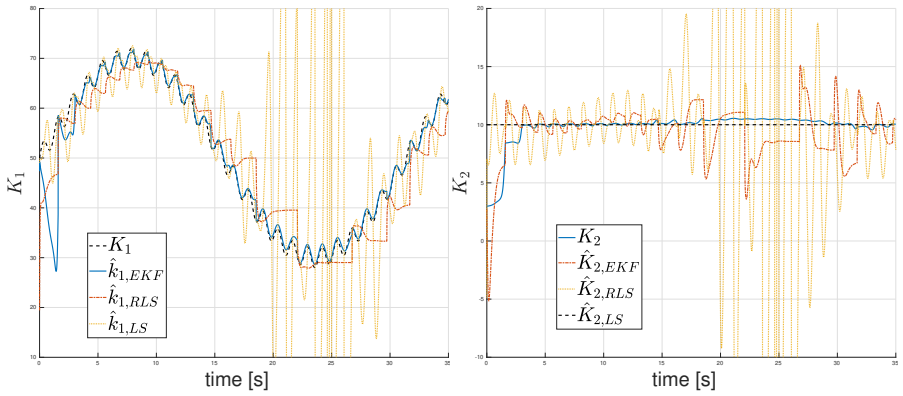
tion. In particular, from figures 4.2a and 4.2d, it appears clear that, when the gains are non-constant, the EKF represents the best solution, even if for constant values, the LS method can perfectly recover the real values. Indeed, the real values of the gains are identified with an error between 2 and 4 % with the EKF, while other methods are more variable. Moreover, a significant error in the EKF identification is due to the time needed for convergence; after that, it reduces dramatically.

The system's state is well reconstructed by the EKF and LS methods, with a mean error lower than 1% in all the cases, while RLS shows worse performance.

Finally, looking at the reconstructed control input, the LS method is the best. Despite this, the well-identified control input comes from a very unstable identification of the control gains. This may lead to unstable behavior when the identified values are used for control purposes. Finally, the RMSE relative to the EKF identification appears small if compared to the maximum control input magnitude (about $30N$), making its estimation acceptable.

As an explanatory case, in figure 4.3, the history of the estimation of the control gains is shown for $k^4_{1,h}$, with $u_r(t) = 0$. It can be seen that the EKF way better identifies the gains, and LS presents a very unstable and oscillating behavior.

Simulations show that the EKF results as the most suitable method for gain identification, with a mean percentage identification error on the control gains almost constant around 4% in the $u_r(t) = 0$ case around 3% when the robot is active. The LS method can reproduce the control input. Still, it can also be very unstable in understanding the human control strategy, making its usage possibly dangerous for developing cooperative control strategies. It must be noted that the LS method performances strictly depend on the choice of the width of the considered window. Indeed, the LS method can perform very well if the window is enlarged and



(a) identified values of $k_{1,h}$ **(b)** identified values of $k_{2,h}$

Figure 4.3: identified control gains for the simulated case

the instability does not arise, but the capability to recover fast changes of the control gains, such as in the $k^4_{1,h}(t)$, decreases. Therefore it is worth considering its use for the cases in which the parameters sought do not change rapidly or if the general human behavior is the core of the study and not the particular, possibly online, time-variant behavior sought, as in [41, 55]. On the other hand, the use of the EKF guarantees better performances in cases where the objective is to precisely recover the gains of the human at each time instant.

The simulations focus on the one DoF case to ease the presentation of the results. To show that the presented method can also deal with multi-DoF systems, figure 4.4 shows the gains identified by the EKF during a simulated task involving three DoFs on the horizontal x–y plane. The DoFs considered are the translations on the x and y-axis and the rotation θ around the z-axis. The task considered is the same as the one DoF case, but the set-points are different along the various axes and are switched together as soon as the target along the x-axis is reached. Note

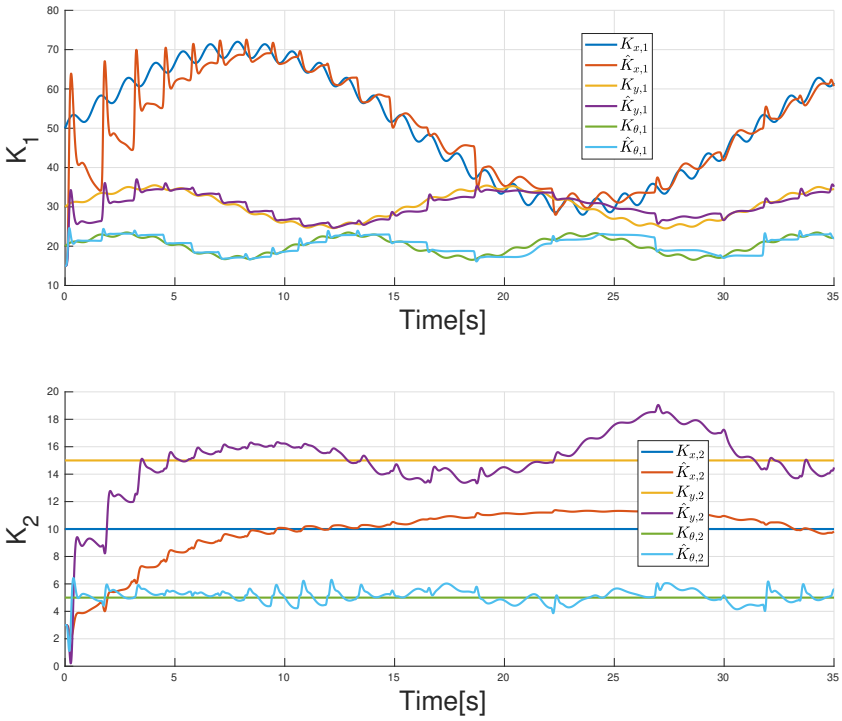


Figure 4.4: The values of the gains in the 3 DoFs simulated case, with results comparable to what was observed in one of the cases.

that adding Cartesian rotations to the problem introduces nonlinearities. Despite this, the problem of rotations can be simplified by linearizing around a working point. In particular, considering relevant human-robot interactive applications such as large object co-manipulation, no relevant rotations are allowed, and typically they happen along one axis at a time. For example, consider a large panel co-transport. The main rotations will likely take place around the vertical axis only. Therefore, the proposed simulation wants to mimic the scenario where a human and a robot co-

manipulate a large object with large motions on the x–y plane, and the rotations along the vertical axis are partially allowed (no full turns can be performed). The simulated parameters are $M = diag([10, 10, 5])$, $D = diag([25, 25, 10])$ and the stiffness matrix is kept null. The reference gains are $K_1 = [50 + 20sin(0.2t) + 2sin(5t), 30 + 5sin(0.4t) + 0.5sin(5t), 20 + 3sin(0.6t) + 0.5sin(5t)]$ and $K_2 = [10, 15, 5]$.

4.1.3.3 EXPERIMENTAL RESULTS

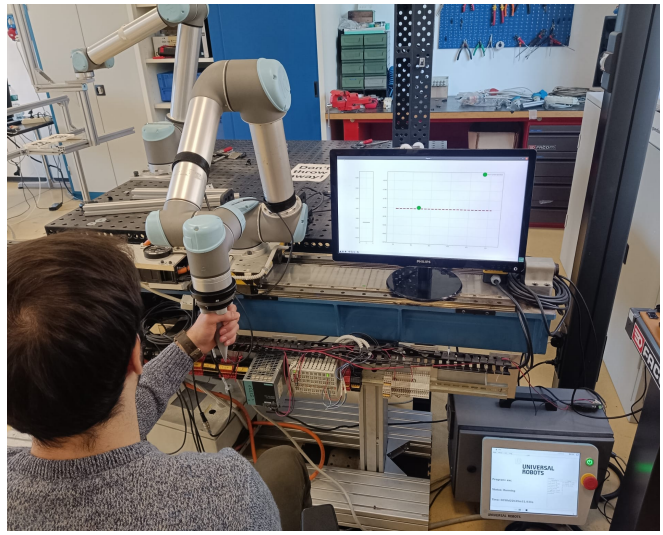


Figure 4.5: The subject performing experiments with the real setup. On the left is the UR5 with the Robotiq FT300 sensor mounted at the end-effector and a handle to allow the human to grasp it. On the right is visible the monitor showing in green dot the current tip position, in dotted red the reference trajectory. The human is asked to move the robot tip (green dot) along the red trajectory, and as long as it reaches one end, switch goals to reach the other.

Simulations show that the EKF approach is suitable and performs well in gain estimation. Hence, experiments with a real robot and a human are performed to identify the feedback gain parameters for the human. Note that this work only shows results relative to one human

subject. The question of whether the approach is generalizable might arise and is worth a brief explanation. This work does not want to model general humans' behavior, general humans' precision in reaching a target, or anything else related to general humans. This work aims to recover online the time-variant gains of a human modeled as linear state feedback. This identification is independent of any subject performing the proposed task and performs as well as with any subject. Three subjects were tested during the design, and no meaningful differences were observed in the identification. Of course, the gains' values vary from subject to subject, but this is irrelevant to prove the proposed approach. The selected robotic platform is a UR5 equipped with a Robotiq FT300 sensor for force/torque sensing at the end effector. The designed experiment involves a reaching task similar to the simulation. A switching reference is set, and the human has to move the robot tip to the target position in Cartesian impedance control. The target and the current robot tip positions are shown on a monitor as a reference for the human. The target set-point is set to $\pm 0.3\,m$, the impedance parameters chosen are $m = 10\,kg$, $k = 0\,N/m$, and three different values of damping are considered for testing the methods on various system dynamics, as $d_1 = 25\,Ns/m$, $d_2 = 50\,Ns/m$ and $d_3 = 100\,Ns/m$. The real setup can be seen in figure 4.5, showing the robot, the sensor, and the monitor used for reference.

The Robot control gain matrix K_r is computed as a solution of the LQR problem for system dynamics described by (3.8), without human contribution (*i.e.* $u_h = 0$), with matrices $Q = diag([10,\ 1])$ and $R = 0.1$, resulting in $K_{r,1} = diag([10,\ 3.89]).K_{r,2} = diag([10,\ 2.04])$ and $K_{r,3} = diag([10,\ 1.04])$

The parameters for the EKF are unchanged concerning the simulated cases. At the same time, for the RLS, the matrix $P_{RLS}(0)$ was found to perform better if set to $P_{rls}(0) = diag([10e3,\ 10e3])$, with a forgetting factor

Table 4.1: The errors computed for the human as (4.10) and (4.11)

		EKF		RLS		LS	
		$u_r = 0$	$u_r \neq 0$	$u_r = 0$	$u_r \neq 0$	$u_r = 0$	$u_r \neq 0$
$\mathcal{E}_z$	d_1	2.26	1.99	25.42	16.22	10.96	5.29
	d_2	2.85	2.33	34.30	16.60	13.49	8.89
	d_3	2.22	2.34	29.23	19.02	unst	unst
RMS_u	d_1	0.51	0.57	3.98	3.44	5.57	1.64
	d_2	1.19	1.10	9.75	7.02	12.16	7.94
	d_3	1.75	1.60	12.56	11.41	unst	unst

$\lambda = 0.8$, and the time interval for the LS is set to 0.08 seconds.

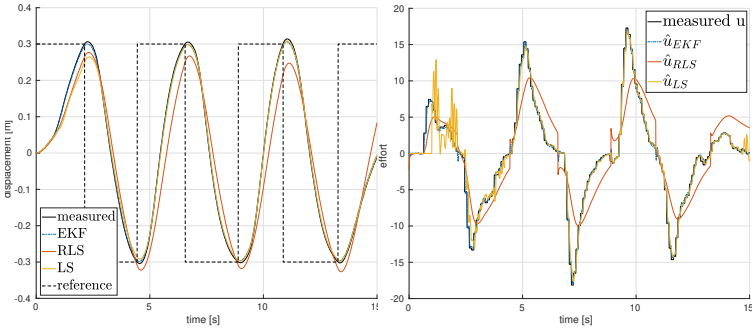
Table 4.1 summarizes the evaluation indexes in the real experiments computed for humans. Only the mean percentage error on the state (4.10) and the RMSE on the control input (4.11) can be computed, as the real control gains are not available in the real experiments.

The mean percentage error on the identified state is kept small for the EKF (between 2 and 3 %) for all the cases, while it is huge for the RLS and LS. Moreover, the LS method shows highly unstable behavior for cases where the damping increases, proving its impossibility to be used for online gain identification. The EKF shows the lowest RMSE compared to RLS and LS.

Figure 4.6 shows the measured and identified state and control histories using the three methods for $d = 25$ and robot active case.

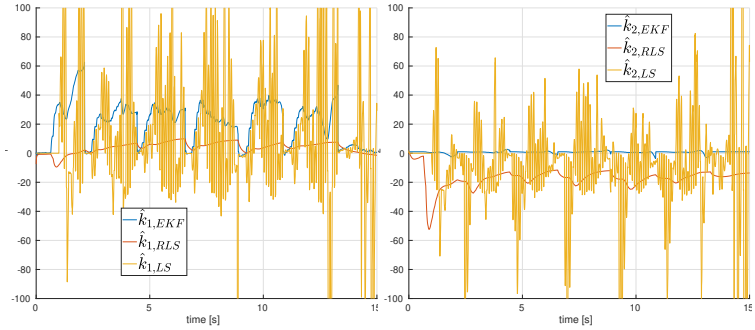
In particular, as shown in the simulation, the LS method is capable of reproducing the control input (figure 4.6b) but is very unstable in identifying the control gains. In contrast, the EKF shows a more consistent behavior and faithfully reproduces the control input and state history.

Since in the experiments with the robot active (*i.e.*, $u_r \neq 0$), its gains are known, and the same indexes can be computed for the robot. The



(a) identified position

(b) identified control input/human force



(c) identified values of $k_{1,h}$

(d) identified values of $k_{2,h}$

Figure 4.6: identified human's state, effort and control gains for the real case

same procedure described in figure 4.1 is applied, computing $K_r(t)$ instead of $K_h(t)$. This allows a direct comparison between the identified and the real gains. The above indexes are presented in table 4.2.

Table 4.2 shows good results for the RLS and EKF methods, while the LS performs poorly in gains identification. This is due to some measurement noise, which makes more unstable the LS computation. The RLS performs better than the EKF, which can also be seen from the simulation results where the control gains are kept constant. Despite this, identifi-

Table 4.2: The errors computed for the robot as (4.9), (4.10) and (4.11)

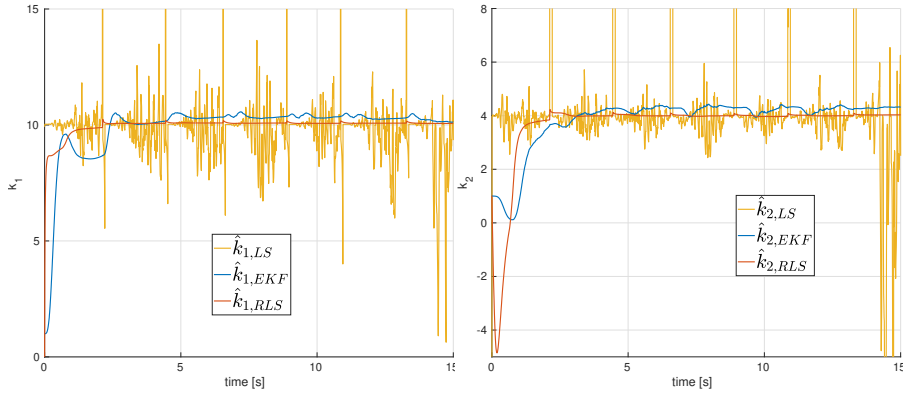
		EKF	RLS	LS
$\mathcal{E}_K$	d_1	6.94	2.05	22.20
	d_2	6.14	1.21	40.16
	d_3	6.67	0.79	10.03
$\mathcal{E}_z$	d_1	3.85	1.30	0.45
	d_2	4.29	1.28	0.54
	d_3	4.27	1.29	0.66
RMS_u	d_1	0.51	0.35	0.26
	d_2	0.53	0.35	0.26
	d_3	0.47	0.11	0.03

cation errors and RMSE are small, showing values comparable to what was observed in the simulation for the EKF. This comparison represents a useful tool for assessing the EKF behavior when used in a real scenario, showing its applicability to this identification problem.

Figure 4.7 shows the identified gains for the $d = 25$ case. Once again, the figure shows that the LS has an unstable behavior while the EKF approaches the real value, even though with a small offset.

4.1.3.4 DISCUSSION

The presented results show that the usage of the EKF provides better results than the other methods used for comparison. A direct comparison with the UKF is not directly presented, as for systems close to linear as the one considered in this work, no sensible differences were found. As visible from (4.5), the nonlinearities are due to the coupling between state variables and parameters to be estimated, which augment the state. The approximation of the system dynamics is presented in appendix (A.4). As visible, no high nonlinearities are present; hence the approximation of the nonlinear system to its linearized version does not introduce sensible



(a) identified values of $k_{1,r}$ **(b)** identified values of $k_{2,r}$

Figure 4.7: identified robot's control gains for the real case

errors.

Finally, the average computational time for the method is computed for online usage. All the computations are done with Matlab R2021b on a PC with an Intel i7 core processor. The mean computational time for the EKF was about 0.05 ms, while for the UKF was 0.1 ms, about double. The LS method is the one that requires more time, with about 1 ms, while RLS average time is the lowest, with about 0.01 ms.

4.1.3.5 CONCLUSION

This section proposes an EKF to identify the time-variant human control law. A state-space formulation and the control law as linear state feedback describe the problem. Augmenting the state with the unknown control gains is possible to identify them online. A reaching task involving time-varying control laws is simulated, comparing EKF with a modified LS and an RLS. The results show that the EKF approach performs better than the LS and RLS methods, particularly in identifying the gains in the

time-variant cases. Experimental results show that the EKF approach is more stable than LS and performs better than RLS and LS.

4.2 HUMAN COST FUNCTION IDENTIFICATION VIA INVERSE OPTIMAL CONTROL

The following is based on results presented in [41]. This section investigates the problem of recovering the human cost function. The human cost function is responsible for generating the human control gain matrix of the previous section.

This section considers the Cartesian impedance equation in (3.6), without the robot contribution. Following the state space formulation presented in 3.2, it can be written as

$$\dot{z} = Az + B_h u_h \tag{4.14}$$

with A, B and z defined as in 3.2.

4.2.1 OPTIMAL CONTROL PROBLEM

Many previous works describe human's intention as the minimization of a quadratic cost function. In particular, it is possible to assume the cost function as quadratic on the state and control input, leading to the LQR formulation of the problem (as examples see [89, 93, 119]). Therefore, the human objective can be described as the minimization of a quadratic cost function, defined as

$$J_h = \int_0^\infty \left(z\, Q_h\, z + u_h\, R_h\, u_h \right)\, dt \tag{4.15}$$

where J_h is the cost that the human incurs, $Q_h \in R^{2n \times 2n}$ is a matrix containing weights on the state, and $R_h \in R^{6\times 6}$ is a matrix of weights on the control input.

Remark 3

The human model weights Q_h and R_h are unknown and must be identified. A reasonable estimate of the human parameters, $\hat{Q}_h$ and $\hat{R}_h$, can be obtained with the IOC method as presented in the following section.

Given the control objective in (4.15), and the system dynamics in (4.14), the LQR Optimal Control Problem can be summarized as

$$\begin{aligned} \min_u J_h = \int_0^\infty (z\, Q_h\, z + u_h\, R_h\, u_h)\; dt \\ s.t. \dot{z} = A\, z + B_h\, u_h \\ z(t_0) = z_0 \end{aligned} \tag{4.16}$$

The LQR optimal control has the feedback form

$$u_h = -K\, z(t) \tag{4.17}$$

in which the human control action is described as linear feedback proportional to the system's state. In (4.17), the matrix K represents the feedback gain matrix, computed as

$$K = R_h^{-1} B^T P \tag{4.18}$$

in which the matrix P is the unique solution for the feedback linear-quadratic optimal control problem, solution of the infinite horizon Continuous Algebraic Riccati Equation (CARE), given by

$$A^T P + PA - PB_h R_h^{-1} B_h^T P + Q_h = 0. \tag{4.19}$$

4.2.2 HUMAN OBJECTIVE IDENTIFICATION

Given the definition of the direct Optimal Control Problem as in (4.16), this section addresses the Inverse Optimal Control problem. That is, given observed state and control histories, denoted as $\bar{z}$ and $\bar{u}_h$, given the system dynamics in (4.14), recover the cost function (4.15) that produced such control histories.

As discussed, different IOC techniques exist. Among the various, this work implements the one presented in [55], considering the human as the only active player, as briefly described in this section.

4.2.2.1 GAIN IDENTIFICATION

The method relies on the knowledge of the feedback gain matrix. Because the gain matrix of the human is not known, it has to be recovered. If the complete trajectory (or a sufficient part of it) is known, it is possible to apply the Least Square Method (LSM), and the matrix K_h is obtained from

$$\hat{K}_h = \underset{K_h}{\operatorname{argmin}} \int_{t_i}^{t_f} \|K_h\, \bar{z}(t) + \bar{u}_h(t)\|^2\, dt \tag{4.20}$$

where t_i and t_f indicate the initial and final time of the trajectory (or a sufficient portion of it), and the symbol $\hat{(\cdot)}$ indicates an estimate of the real value.

4.2.2.2 INVERSE OPTIMAL CONTROL

This section briefly presents the main steps to recover the cost function of a player. Refer to [55] for the full treatment. As preliminary, denote $\oplus$ as the Kronecker sum, defined as

$$X \oplus Y = (X \otimes I_q) + (I_r \otimes Y) \tag{4.21}$$

with $X \in R^{r\times r}$ and $Y \in R^{q\times q}$.

Given A, B, and $\hat{K}_h$, the Inverse Optimal Control (IOC) problem can be solved as follows. First, denote F, $F_{\oplus}$, $\hat{K}_{\oplus}$ and Z as

$$\begin{aligned} F &= A - B\hat{K}_h, \\ F_{\oplus} &= F^T \oplus F^T, \\ \hat{K}_{\oplus} &= \hat{K}_h^T \oplus \hat{K}_h^T, \\ Z &= (I_n \otimes B^T) F_{\oplus}^{-1} \\ M &= \begin{bmatrix} Z & Z\hat{K}_{\oplus} + \hat{K}^T \otimes I \end{bmatrix} \end{aligned}$$

Then, denote θ as the vector of vectorized weights as

$$\theta = [vec(\hat{Q})\ vec(\hat{R})]^T. \tag{4.22}$$

As demonstrated in [55], if θ satisfies

$$M\theta = 0 \tag{4.23}$$

the $\hat{Q}$ and $\hat{R}$ are the correct unkown parameter for (4.19).

Given that (4.23) is a reformulation of (4.19), the parameter set of the inverse LQ differential system is given by

$$\theta = ker(M) \tag{4.24}$$

with convex boundaries such that $R_h > 0$. As typical in IOC problems, a residual is defined as

$$r = M\theta \tag{4.25}$$

to consider non-optimal behaviors and imperfect modeling of $\hat{K}$. The

following quadratic problem is formulated to minimize the residual

$$\begin{aligned} \min_{\theta} \quad & ||r||_2^2 = \frac{1}{2}\theta^T H \theta \\ \text{s.t.} \quad & I\,\theta \geq 0 \\ & R > 0 \end{aligned} \tag{4.26}$$

with $H = 2M^T M$.

Remark 4

All the $\lambda\,\theta$, $\lambda > 0$ solutions are acceptable, since the solution of problem (4.16) is the same for any $\lambda\,J$, $\lambda > 0$.

Remark 5

The constraints in (4.26) are imposed by Optimal Control necessary conditions, *i.e.* matrix Q semi-positive definite and matrix R strictly positive definite.

4.2.3 EXPERIMENTS

In this section, experimental results are presented. First, the design of experiments is presented, with comments on the evaluation procedure, then results are presented and analyzed.

4.2.3.1 DESIGN OF EXPERIMENTS

Three subjects are involved in this study. Each subject is asked to move the robot end-effector to a fixed, unknown set-point along the z-axis. The impedance parameters of the robot are set to $M = 10$, and three values of damping $D = [25, 50, 75]$ are tested to verify how different systems

modify the control objective. The stiffness is set $K = 0$. Each subject is asked to perform 12 set-point reaching for each damping value for a total of 36 reaching tasks. Each reaching is along the x–axis only, with the set-point at various, random distances. Before each recording session, the subject is allowed to practice for a while to gain confidence with the system.

The measured exchanged forces, the reference position, and the actual positions compute the IOC problem. The forces are measured with a Robotiq FT300 sensor mounted on the robot tip. The robotic system gives the positions directly, and the velocities are computed by differentiating the positions.

The human cost function matrices, as typically happens [58, 77], are assumed to be nonzero only on the diagonal terms, resulting in $Q = diag([q_1, q_2])$, and $R = r$. Moreover, since the solution to the optimal control problem is the same for any λJ_h with λ scalar, as an additional constraint to (4.26) is set $q_1 = 1$. In this way, all the recovered cost functions are comparable.

A first analysis is done on the recovered features (q_1, q_2 and r) from (4.26) to check their variation with respect to the variations of the system. In the ideal scenario, the weights can be different between different subjects. Despite this, each subject should have constant weight values, even when varying the system's parameters.

A measure of the control input is defined to evaluate the goodness of the recovered costs as

$$\mathcal{E}_u = \frac{\int_{t_0}^{t_f} u(t) - \hat{u}(t)\, dt}{u_{max}} \tag{4.27}$$

where u_{max} is the maximum measured control effort, and it is used to normalize the errors to make the comparison fair. The recovered control input $\hat{u}(t)$ is computed by running a system simulation. In the ideal case,

this value should be zero. Defining the set-point as $x_{sp} = x_{ref} - x_0$, and the Raise Time RT as the time required to move the system from x_0 to $0.95\ x_{sp}$, it is possible to define a ratio

$$RT_{sp} = \frac{RT}{x_{sp}}, \tag{4.28}$$

to check how the different systems influence the rise time by removing dependency from the set-point distance. In the ideal case, the ratio should be the same for a given system with any set-point and should increase by increasing the damping.

Finally, a possible relation between the distance from the set-point and the weight of the control is investigated. In the ideal case, the values of R should always be the same, without regard for the set-point.

4.2.3.2 RESULTS

As previously discussed, the weight q_1 is constrained to be $q_1 = 1$, and the other weights q_2 and r are computed consequently. From the experiments, all the subjects have a negligible q_2, with values in the order of $1e-4$. This means that human feedback is based mainly on position, as also happens in [57]. Since the values of q_2 are very close to 0, the rest of this paper will approximate them with 0 without further analysis.

Given $q_2 = 0$, the ratio between q_1 and r drives the system's response. The analysis of the values of R, visible in figure 4.8a, shows that humans have a low weight also in how much effort is put into the task. The low value of r, compared with q_1, means that a fast set-point reaching is more favorable than saving effort.

Moreover, three subjects tend to decrease the value of r as the damping increase, which means that reaching the desired position at a specific time should remain almost constant against changes in the system.

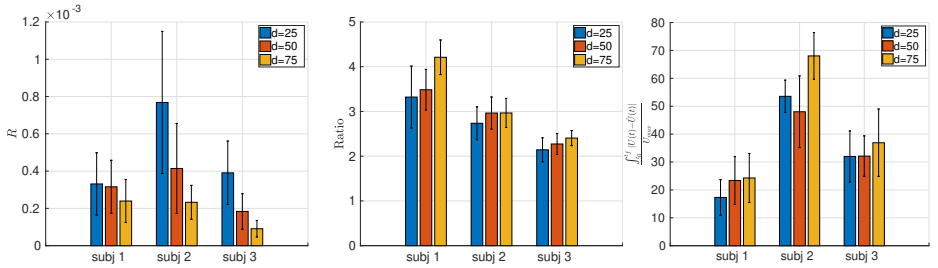
Figure 4.8b also displays such a behavior, showing the ratio RT_{sp}.

The Figure shows a minimal increment in the ratio compared with the increment of damping, particularly in subjects 2 and 3, confirming that humans appear more interested in time than in the effort required.

This study also shows a correlation between the set-point distance and the weight a human gives to the control effort. The farther the set-point is, the higher the control weight, as shown in figure 4.9.

The error between the measured and reconstructed control input is analyzed. Figure 4.8c shows the normalized error between the measured and the simulated control effort of the three subjects in the three scenarios.

Finally, figure 4.10 presents one control history measured and recovered for each subject in the study, with the corresponding normalized error.



(a) Mean and standard deviations of the recovered control weights

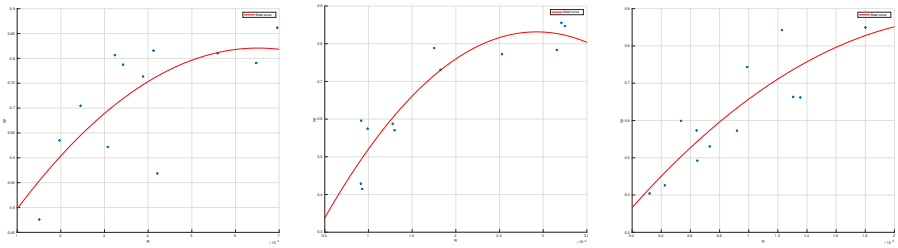
(b) rise time over set-point computation

(c) Mean and standard deviations of the control inputs

Figure 4.8: The results of the performance indexes

4.2.3.3 DISCUSSION

This work's analysis suggests human modeling as an optimal controller and hints about real human intention. The most relevant result is that humans tend to keep the required time as constant as possible when per-



(a) control weight identified with D=25 **(b)** control weight identified with D=50 **(c)** control weight identified with D=75

Figure 4.9: Correlation between control weight identified and different set-points, for subject 3

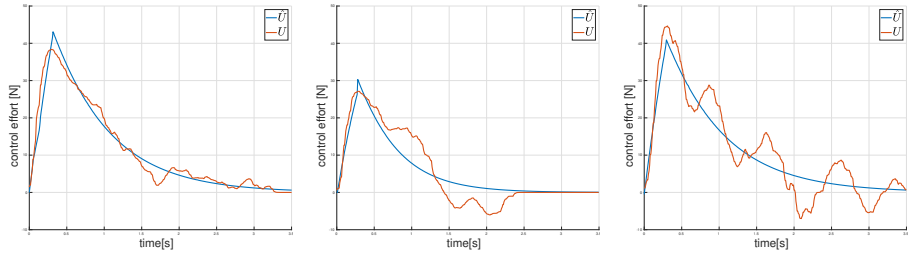
forming a task. This is probably because a human control works as feedback on a visual stimulus and keeps this control -which is not considered in the presented study- constant.

Despite this, optimal control modeling of a human interacting with a robot can still be adopted, keeping in mind some baseline rules shown in the results section. The human's weight on control input decreases if the system becomes more rigid against motions and decreases as the desired position becomes closer to the current one.

4.2.3.4 CONCLUSIONS

This work presents a study of human behavior modeled as optimal control during physical Human-Robot Interaction. Theoretical modeling of the system and control model is presented, and the formulation of the Inverse Optimal Control is presented. IOC is used to recover the cost functions of three subjects performing a reaching task interacting with different robot behavior. The analysis shows that OC modeling of a human may be used and gives good results with some caution.

Such a model presents applications in designing game-theoretical-



(a) Computed and measured effort of subject 1. $\mathcal{E}_u = 17.81$ **(b)** Computed and measured effort of subject 2. $\mathcal{E}_u = 46.32$ **(c)** Computed and measured effort of subject 3. $\mathcal{E}_u = 40.02$

Figure 4.10: Example of one trial of each subject.

based controllers for pHRI, in which the knowledge of the other's cost function is necessary. Moreover, it can also be used as a reference in modeling human behavior for digital twin simulations.

4.3 LEARNING HUMAN INTENTION PREDICTION VIA RECURRENT NEURAL NETWORK

This chapter investigates the human intention prediction via Recurrent Neural Networks (RNN) and Transfer Learning (TL), proposing a method to iteratively train the model to adapt it to the interaction and TL to adapt the model to new users and situations. The proposed predictive module builds upon the Game-theoretic control frameworks presented in 5.2 and 5.3. It integrates with both the control frameworks. Please refer to those models for completeness.

4.3.1 MODEL DESCRIPTION AND TRAINING

This subsection introduces the two main components of the model and their training procedure, based on iterative training and transfer learning.

4.3.1.1 RNN+FC MODEL

As also discussed in [91], the following can express the human limb dynamics in interaction with a robot at the robot's tip

$$-C_h\dot{x} - K_h(x_{ref,h} - x) = u_h, \tag{4.29}$$

with C_h and K_h damping and stiffness matrices. Assuming that $C_h = C_h(x, \dot{x})$ and $K_h = K_h(x)$, the desired human motion can be defined as

$$x_{ref,h} = F(x, \dot{x}, u_h). \tag{4.30}$$

The function F is nonlinear and time-variant. The problem becomes even more complicated if the human and the robot interact while transporting a large object, with additional inertia and different contact points. Therefore, We propose using a Recurrent Neural Network (RNN), cascaded with a Fully Connected (FC) layer (RNN+FC), to learn the complex human dynamics and provide the robot with the desired trajectory over the next horizon. In particular, among the various types, this work proposes using Long-Short Term Memory (LSTM), which has proven to have better performances for long-time series than the basic RNN.

The proposed method aims to identify and predict, over a finite rolling horizon, the desired human trajectory, given the history over a finite horizon. The RNN+FC model takes the last k time instant as input and predicts the human desired trajectory over the next N steps. The model accepts as inputs the actual robot positions and velocities $\mathbf{x} = [x, y, z, R, P, Y]^T$ and $\mathbf{v} = [\dot{x}, \dot{y}, \dot{z}, \omega_x, \omega_y, \omega_z]^T$, respectively, the force the human is exerting $\mathbf{u_h} = [f_x, f_y, f_z, \tau_x, \tau_y, \tau_z]^T$, and the nominal robot trajectory $\mathbf{x_{ref,r}} = [x_{ref,r}, y_{ref,r}, z_{ref,r}, R_{ref,r}, P_{ref,r}, Y_{ref,r}]^T$. Defining with T the current time, the input data are defined as $\underline{\mathbf{x}} = \{\mathbf{x}_{T-k}, ..., \mathbf{x}_T\}$ and $\underline{\mathbf{v}} = \{\mathbf{v}_{T-k}, ..., \mathbf{v}_T\}$ the vector containing the positions and velocities of the past k time instant, respectively, $\underline{\mathbf{u_h}} = \{\mathbf{u_h}_{T-k}, ..., \mathbf{u_h}_T\}$ the vector containing the hu-

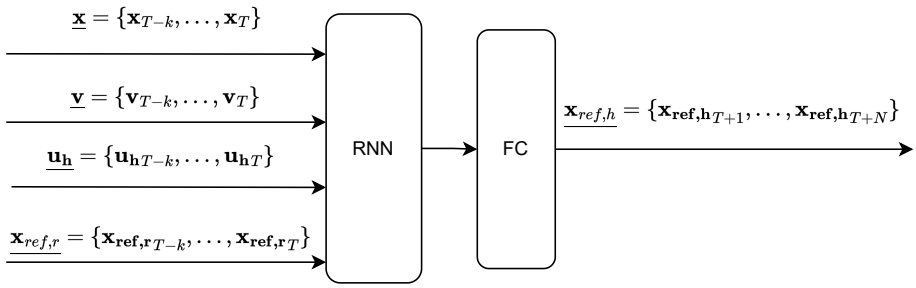


Figure 4.11: Human trajectory identification using the RNN+FC.

man interaction wrench over the past k time instant and $\underline{\mathbf{x}_{\mathbf{ref,r}}} = \{\mathbf{x_{ref,r}}_{T-k}, \ldots, \mathbf{x_{ref,r}}_T\}$ the vector containing the reference trajectory of the robot of the past k time instant. The output of the model is a finite sequence of reference positions in the horizon that goes from time $T+1$ to $T+N$, defined as $\underline{\widehat{\mathbf{x}}_{\mathbf{ref,h}}} = \{\mathbf{x_{ref,h}}_{T+1}, \ldots, \mathbf{x_{ref,h}}_{T+N}\}$, where $\widehat{(\cdot)}$ denotes an estimate. A schema of the proposed model with inputs and output is visible in Figure 4.11.

4.3.1.2 ITERATIVE TRAINING

Before using the model, data have to be acquired and processed to train the input-output relationship, *i.e.* the function F in (4.30) in our case. Collecting data without using the model (assuming, for example, $\underline{\widehat{\mathbf{x}}_{\mathbf{ref,h}}} = \underline{\mathbf{x}_{\mathbf{ref,r}}}$) allows the model training. However, once the model is being used, the robot behavior is no more the same as during the data collection phase because the assumption $\underline{\widehat{\mathbf{x}}_{\mathbf{ref,h}}} = \underline{\mathbf{x}_{\mathbf{ref,r}}}$ does not hold anymore. The robot assists using that estimate. The model can predict $\underline{\mathbf{x}_{\mathbf{ref,h}}}$ only with the original robot behavior based on such an assumption.

We propose an iterative training procedure to face this problem, We first collect an initial dataset $\mathcal{D}_0$ without using any model, letting $\underline{\widehat{\mathbf{x}}_{\mathbf{ref,h}}} = \underline{\mathbf{x}_{\mathbf{ref,r}}}$. We train a model, call it $\mathcal{M}_0$ with the subscript $_0$ indicating it is the

model trained with no model, which depends on the first dataset only $\mathcal{M}_0 = \mathcal{M}_0(\mathcal{D}_0)$. Once we have a model, a new data collection phase begins. After the second dataset $\mathcal{D}_1$ is collected, it is possible to train a second model $\mathcal{M}_1 = \mathcal{M}_1(\mathcal{M}_0, \mathcal{D}_0)$. This procedure is iterated for K times until a stop criterion is reached, and the model $\mathcal{M}_K = \mathcal{M}_K(\mathcal{M}_{k-1}, \mathcal{D}_{k-1})$ is finally ready for usage. A stopping criterion can be the prediction error, defined as the average of the Root Mean Square Error (RMS),

$$e_{RMS} = \frac{1}{L}\sum_{T=1}^{L}\sqrt{\frac{1}{N}\sum_{k=T}^{T+N}(\|\widehat{\mathbf{x}}_{\mathbf{ref},\mathbf{h_k}} - \mathbf{x}_k\ \|)^2}, \tag{4.31}$$

where $\widehat{\mathbf{x}}_{\mathbf{ref},\mathbf{h_k}}$ is the predicted human intention, $\mathbf{x}_k$ the measured poses, L is the length of the trajectory, and N is the prediction horizon. The iterative procedure stops when $||e^{k+1}_{RMS} - e^{k}_{RMS}|| < tol$.

4.3.1.3 TRANSFER LEARNING

Equation (4.29) differs from human to human. It describes human dynamics when the human is grasping the robot's tip but does not consider possible co-carried large objects that can be grasped from one side by the robot and from the other by the human, making the assumption invalid. The model $\mathcal{M}_K$ obtained after the iterative procedure might require additional training to adapt to new users and objects.

The main drawback of iterative procedures is they are time-consuming, requiring time for data collecting and model training. Transfer learning is a widely adopted method to speed up training starting from a pre-trained model.

Among the different TL techniques, We propose adopting a widely used strategy in computer vision or NLP domains, which consists of "freezing" some model layers and re-train on new data only a few layers, which means having fewer parameters to be tuned compared with

Algorithm 1 The iterative training procedure

Require: Sample records
Ensure: Trained model $\mathcal{M}$
1: Record dataset $\mathcal{D}_0$ without any model train model $\mathcal{M}$, with dataset $\mathcal{D}_0$
2: **while** $||e^{k+1}_{RMS} - e^{k}_{RMS}|| < tol$ **do**
3: Record dataset $\mathcal{D}_k$ with model $\mathcal{M}_k$;
4: train model $\mathcal{M}_{k+1}$ with dataset $\mathcal{D}_k$;
5: **if** new user or object **then**
6: Record dataset $\mathcal{D}_{TL}$ with model $\mathcal{M}_{k+1}$;
7: train model $\mathcal{M}_{TL}$ with dataset $\mathcal{D}_{TL}$;

the complete model. In this work, we propose to freeze the RNN part of the model and fine-tune only the final FC layer, with the insight that the RNN learns the features of the pHRI (*e.g.*, a force in a direction means that the human wants to steer the system in the same direction, an increasing force means the human is accelerating), while the FC layer learns how a specific user does interact (*e.g.*, how much force a particular user uses to steer the system, how fast a specific user accelerate the system, etc.). With the TL approach, it is possible to quickly adapt the model $\mathcal{M}_K$ to new users and additional objects co-carrying, with little data collection and fast training, leading to $\mathcal{M}_{TL} = \mathcal{M}_{TL}(\mathcal{M}_K, \mathcal{D}_K)$. The iterative procedure is summarized in 1, and a schematic presentation is visible in 4.12.

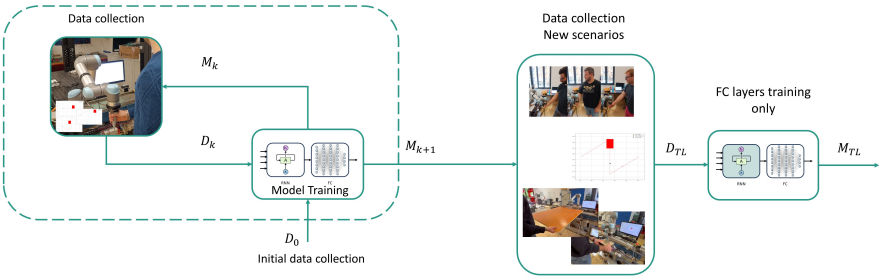


Figure 4.12: The schema of the training procedure. On the left, the iterative training. On the right, the Transfer Learning.

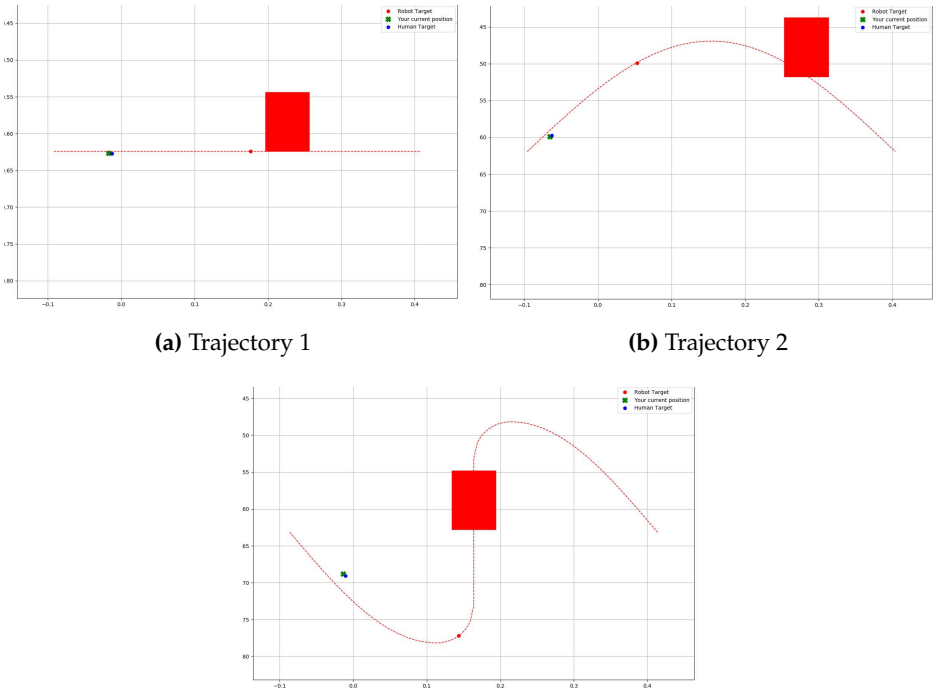
4.3.2 MODEL EVALUATION

The presented method is evaluated with real experiments. The robotic platform is a UR5 robot, with a Robotiq FT300 mounted at the tip for measuring the human interaction force. For this study, we want to simulate a collaborative motion along the x–y plane, which is typical for applications such as large object co-transportation. Moreover, the experiments want to simulate a situation in which a human and a robot are moving together along a trajectory, and the human, at a certain point, needs to deform the trajectory (e.g., because there is an obstacle that the robot doesn't know in between).

4.3.2.1 DATA COLLECTION

For the model's training, we collect the robot's actual poses and velocities from the robot's controller. The interaction force is measured at the robot tip via the FT sensor. An external computer computes the robot's nominal trajectories and streams the commands in real time. The data are sampled at 8 milliseconds. Three nominal robot trajectories are defined: linear, curved, and sinusoidal. The three trajectories are visible in Figure 4.13a, 4.13b, and 4.13c, respectively. The human must follow three trajectories visible on a monitor and deform them to avoid an obstacle that appears randomly at some point in the trajectory. The robot does not know the position of the obstacle.

To train the model, a single operator performed 20 repetitions for each trajectory for 60 trials. Despite the stop criterion (4.31) being matched after 1 iteration, we decided to perform 4 iterations to evaluate if iterating more can produce some improvement. The RNN+FC model has input data collected in the 125 time instant precedent, and the prediction horizon is set to 50 time steps. The LSTM model comprises 3 layers with 250 nodes, and the FC comprises two connected layers. The learning rate, decay function, and the optimizer of the RNN+FC model are obtained with



(a) Trajectory 1

(b) Trajectory 2

(c) Trajectory 3

Figure 4.13: The trajectories visible on the monitor. The red box is the obstacle, the green cross is the current position, and the red dot is the robot reference. The training uses trajectories 4.13a, 4.13b, and 4.13c.

Optuna [2]. The model is trained for 25 epochs.

The impedance parameters are $M_i = diag(10, 10)$, $C_i = diag(100, 100)$, and the stiffness is set to null $K_i = diag(0, 0)$, as typically in pHRI. The mass and damping coefficients have been hand-tuned to allow smooth motions. The cost of the two players are set as $Q_{h,h} = Q_{r,r} = diag([1, 1, 0.0001, 0.0001])$, $Q_{h,r} = Q_{r,h} = 0^{2\times 2}$ and $R_h = R_r = diag([0.0005, 0.0005])$. The human cost function parameters $Q_{h,h}$, $Q_{h,r}$ and R_h are recovered via Inverse Optimal Control (IOC) as in [41],

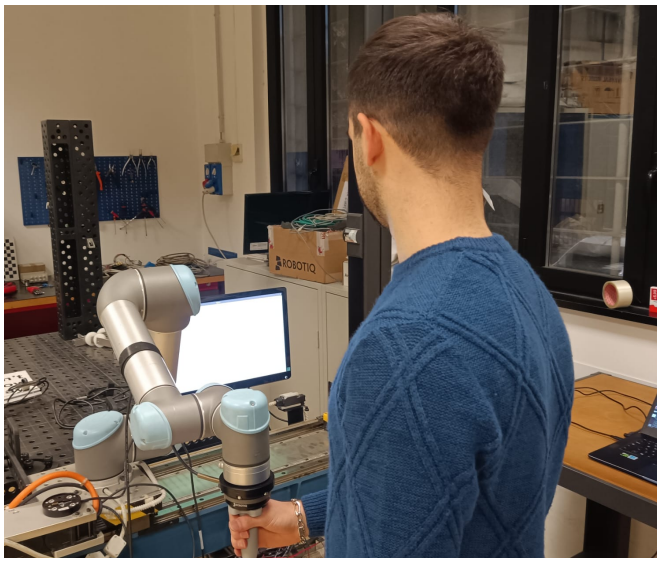


Figure 4.14: Experimental setup: a Robotiq FT300 sensor is at the robot tip; a monitor shows the reference trajectory.

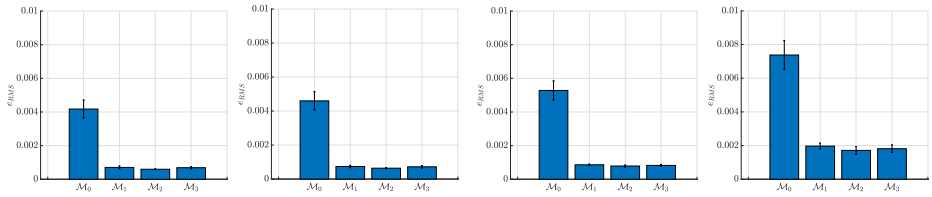
and an average value is used. The robot parameters $Q_{r,r}$, $Q_{r,h}$, and R_r are set equal to the human's to mimic a person. Different tuning may result in more assistive behavior, which might be desirable. Since the DGT controller accepts as a reference for (5.15) only one set-point rather than a prediction horizon, we take the 20^{th} point of the prediction horizon. Finally, $\alpha = 0.8$ is chosen to allow sufficient assistance. This value allows high assistance and the robot to recover the position of the robot set-point autonomously. Moreover, this value and higher allows the assumption $z_{ref} \simeq z_{ref,h}$ to hold. Figure 4.14 shows the setup.

4.3.2.2 ITERATIVE TRAINING EVALUATION

To evaluate the model's prediction capability, we measure the average RMS along each trajectory, computed as in (4.31). First of all, we evaluate the improvement provided by the iterative training. We compare the

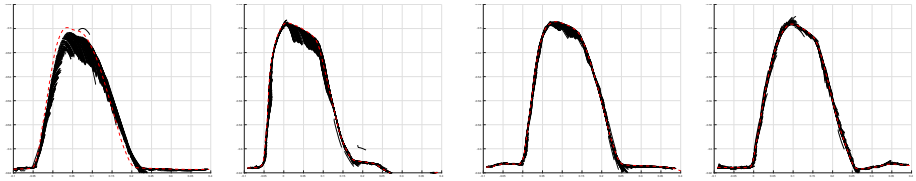
e_{RMS} for four iterations. At each iteration k, data $\mathcal{D}_k$ are collected with the model trained on the previous iterations $\mathcal{M}_{k-1}$.

We also want to evaluate the error to the width of the prediction horizon. This information is relevant, for example, in a Model Predictive Control (MPC) implementation or to foresee in advance dangerous situations (*e.g.,* collisions that the human does not expect, proximity to the robot workspace boundaries, etc.). Therefore, we measure the e_{RMS} with different prediction horizons. Note that we use the same model, so a horizon of 50 time steps is computed in all the cases, but we evaluate just the first n samples of it for horizons of $H = \{5, 10, 20, 50\}$ time steps, which corresponds to 0.04, 0.08, 0.16, and 0.4 seconds. Figure 4.15 shows the results of the iterative training evaluations, also with the time horizon dependence. On the one hand, by iterating the model, the e_{RMS} decreases significantly in two iterations, after that, remains stable, and no sensible improvements are visible after one iteration of the model. On the other hand, for a wide prediction horizon, the error increases. This is mainly because predicting human deviations from the nominal trajectory in advance is complex. Indeed, the e_{RMS} increases just before the human starts the deviation from the nominal trajectory, and the robot cannot know it in advance. Despite this, the average error is about one millimeter and increases to about two with the maximum prediction horizon considered. Some improvement can be achieved in terms of prediction direction. This is visible qualitatively from Figure 4.16 and can be explained by evaluating the maximum prediction error. More results relative to the other trajectories are presented in the appendix A.3 . Therefore, the maximum expected error e_{MAX} should be considered. Indeed, a significant error may cause an unwanted robot behavior that may cause it to go in the wrong direction because the human intention is misunderstood, even for some time instants. Results are visible in table 4.3. It is visible that by iterating more, the maximum error decreases. In general, both e_{RMS} and



(a) Prediction horizon of 0.04s **(b)** Prediction horizon of 0.08s **(c)** Prediction horizon of 0.16s **(d)** Prediction horizon of 0.4s

Figure 4.15: Model Evaluation: e_{RMS} for four models with different prediction horizons and model iterations $\mathcal{M}_k$.



(a) $\mathcal{M}_0$ **(b)** $\mathcal{M}_1$ **(c)** $\mathcal{M}_2$ **(d)** $\mathcal{M}_3$

Figure 4.16: Model Evaluation: the prediction at the various model iterations $\mathcal{M}_k$. The maximum prediction horizon is considered (0.4s). In solid black, the prediction at each time instant. In dashed red, the executed trajectory.

e_{MAX} should be considered in the evaluations. Indeed, even though the e_{RMS} is similar, this might be because a significant error (but limited in time) "vanishes" if the e_{RMS} is computed over a long trajectory where the average error is very low.

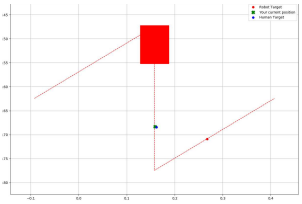
4.3.2.3 TRANSFER LEARNING EVALUATION

To evaluate the capabilities of generalization of the proposed method, three different new situations are addressed: (i) a trajectory not used during the training, (ii) five new subjects, and (iii) a different co-manipulated object. First, we evaluate the model's capability to predict human intent with a new trajectory. Then, TL learns the new trajectory and compares

Table 4.3: Model Evaluation: e_{MAX} for different models $\mathcal{M}_k$ and horizons H_k. Values are in millimeters

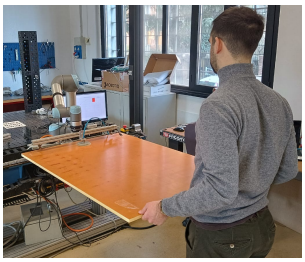
	$\mathcal{M}_0$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$
H_5	11.1 ± 1.9	4.7 ± 3.5	4.1 ± 1.5	3.0 ± 0.4
H_{10}	12.0 ± 1.8	5.0 ± 3.4	4.2 ± 1.5	3.2 ± 0.4
H_{20}	14.3 ± 1.8	5.9 ± 3.3	4.9 ± 1.4	3.9 ± 0.5
H_{50}	24.4 ± 2.9	13.7 ± 4.1	13.5 ± 2.2	10.6 ± 1.6

the errors. As for the previous case, different prediction horizons are evaluated. Results are in table 4.4. The e_{RMS} is similar to the prediction made on the training trajectories, around 1 mm for small prediction horizons (up to H_{20}), and becomes larger for wider prediction horizons. The TL allows for reducing the error, making it comparable to that of the known trajectories. TL procedure improves the maximum error of the prediction, and its results are comparable to that of the known trajectories. The experimental setup of this case is visible in figure 4.17a.



(a) The new trajectory used for transfer learning.

(b) Different subjects performing the task.



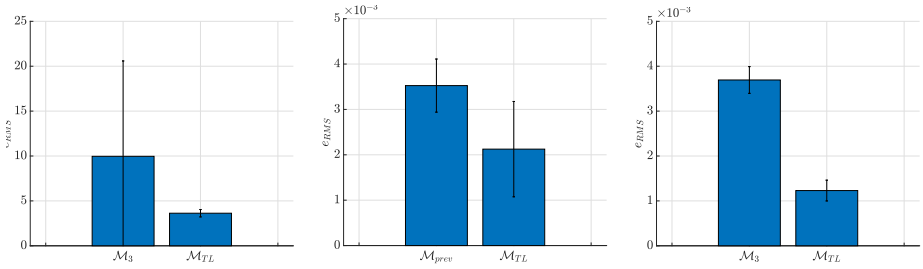
(c) TL with different objects experiments.

Figure 4.17: The three situations addressed with transfer learning.

Then, five subjects are asked to perform the three trajectories used during the training and deform them by directly grasping the robot at the tip. The experimental setup of this case is visible in figure 4.17b.

Finally, the subject that trained the base model grasps a panel of 106x82cm

assisted by the robot. By adding the object, different forces are exchanged. The experimental setup of this case is visible in figure 4.17c. Figure 4.18 shows the prediction error relative to the maximum prediction horizon considered (50 time steps, or 0.4 s) before and after applying the TL. The TL improves the predicting performances for the case of different subjects, making the average error after the TL comparable to the error of the iterated model, about 2mm. The TL also decreases the errors also for the case of the co-manipulated object.



(a) Transfer to different trajectory (Case I).

(b) Transfer to different subjects (Case II).

(c) Transfer to different objects (Case III).

Figure 4.18: Transfer learning Evaluation: value of e_{rms}

Table 4.5 shows the time required for each iteration's dataset collection and model training, comparing it with the TL time. The iterative procedure takes about two hours for each iteration. The TL approach allows performing this procedure only once. After that, the model can be adapted to new situations in about 10 minutes, thanks to the TL approach. The computation runs on a laptop with Intel i7 and NVIDIA GeForce 1050.

Table 4.4: Transfer Learning Evaluation, Case I: e_{RMS} and e_{MAX} before and after TL, for different models $\mathcal{M}_k$ and horizons H_k. Values are in millimeters.

	$\mathcal{M}_3$		$\mathcal{M}_{TL}$	
	e_{RMS}	e_{MAX}	e_{RMS}	e_{MAX}
H_5	1.13 ± 0.034	9.60 ± 10.9	0.89 ± 0.049	3.15 ± 0.79
H_{10}	1.15 ± 0.026	9.65 ± 10.8	0.95 ± 0.056	3.36 ± 0.60
H_{20}	1.15 ± 0.017	9.96 ± 10.6	1.03 ± 0.060	3.62 ± 0.40
H_{50}	2.71 ± 0.052	19.4 ± 5.43	1.68 ± 0.092	12.6 ± 2.54

Table 4.5: Transfer Learning Evaluation; time required for the data collection and model training.

	Iterations	TL subjects	TL object
dataset collection	60 ± 10 min	5 ± 2 min	5 ± 2 min
training	45 ± 5 min	4 ± 1 min	4 ± 1 min

CHAPTER 5

DIFFERENTIAL/DYNAMICS GAME THEORY FOR PHRI

This chapter is based on findings described in [40] and [36]. This chapter investigates the modeling of the interaction between humans and robots from a game-theoretic point of view. It formulates the basic problem and shows how to compute the robot's input in the GT framework. First, a continuous time dynamics model is used, and solutions are derived for the cooperative and non-cooperative cases. Then, the discrete-time model is analyzed, proposing the solution of a distributed Model Predictive control.

5.1 COOPERATIVE AND NON-COOPERATIVE GAME-THEORY

This section introduces the two main modes of play in Game Theory.

To clearly explain and analyze the two main possible interaction models, let's introduce a classic example: the rope-pulling game. The following example takes inspiration and partially follows the example in the introduction presented in [48]. In particular, let's focus on the non-zero-

sum game, which will be the situation addressed in the proposed thesis for the pHRI. The game is formulated as follows. Two players push a point mass by exerting forces f_1 and f_2. Both players exert forces with the same magnitude ($||f_1|| = ||f_2||$), but they pull in different directions θ_1 and θ_2. The game is played for 1 second. Initially, the mass is at rest, and for simplicity, we assume unit forces and a unit mass. According to Newton's law, the point mass moves according to

$$\ddot{x} = cos\theta_1(t) + cos\theta_2(t), \dot{x}(0) = 0, x(0) = 0$$

$$\ddot{y} = sin\theta_1(t) + sin\theta_2(t), \dot{y}(0) = 0, y(0) = 0$$

Player P_1 wants to maximize $x(1)$, whereas player P_2 wants to maximize $y(1)$.

Let's first consider the non-cooperative case. Suppose that player P_1 follows the course of action $\theta_1(t) = 0$ throughout the whole time period and therefore $\ddot{x} = 1 + cos\theta_2(t)$, $\ddot{y} = sin\theta_2(t)$, $\forall t \in [0, 1]$. In this case, the best course of action for P_2 so as to maximize $y(1)$ is precisely to choose $\theta_2(t) = \frac{\pi}{2}, \forall t \in [0, 1] \ddot{y}(t) = 1, \forall t \in [0, 1]$. Moreover, any deviation from this will necessarily lead to a smaller value of $y(1)$. In this sense, once P_1 decides to stick to their part of the solution in (1.3), a rational P_2 must necessarily follow their policy in (1.3).

Conversely, suppose that player P_2 follows the course of action $\theta_2(t) = \frac{\pi}{2}$ throughout the whole time period and therefore $\ddot{x} = cos\theta_1(t)$, $\ddot{y} = sin\theta_1(t) + 1$, $\forall t \in [0, 1]$. In this case, the best course of action for P_1 so as to maximize $x(1)$ is precisely to choose $\theta_1(t) = 0, \forall t \in [0, 1] \ddot{x}(t) = 1, \forall t \in [0, 1]$ Moreover, any deviation from this will necessarily lead to a smaller value of $x(1)$. Also now, once P_2 decides to stick to their part of the solution in (1.3), a rational P_1 must necessarily follow their policy in (1.3).

A pair of policies that satisfy the above properties is called a *Nash equi-*

librium solution. The key feature of Nash equilibrium is that it is stable, in the sense that if the two players start playing at the Nash equilibrium, none of the players gains from deviating from these policies.

Let's now consider the cooperative case. The pair of policies P_1 and P_2 that the players should pursue are $\theta_1(t) = \frac{\pi}{4} \forall t \in [0,1]$ and $\theta_2(t) = \frac{\pi}{4} \forall t \in [0,1]$, respectively. This pair of policies leads to constant accelerations $\ddot{x} = \ddot{y} = \sqrt{2}$ and therefore $x(1) = y(1) = \frac{\sqrt{2}}{2} > \frac{1}{2}$. This policy is interesting because both players do strictly better than with the Nash policies. This is not a Nash equilibrium solution, in spite of the fact that both players can do better than with the Nash solution. Solutions such as this one are the subject of cooperative game theory, in which one allows negotiation between players to reach a mutually beneficial solution. However, this requires faith/trust among the players. As noted above, solutions arising from cooperation are not robust with respect to cheating by one of the players.

A solution like the cooperative one is called Pareto-optimal because it is impossible to further improve one player's gain without reducing the gain of the other. The problem of finding Pareto-optimal solutions can typically be reduced to a single-criteria-constrained optimization. Indeed, for the non-zero-sum rope-pulling game, all Pareto-optimal solutions can be found by solving the following constrained optimization problem $max_{\theta_1(t),\theta_2(t)} x(1) : y(1) \geq \alpha$ with $\alpha \in R$. Pareto-optimal solutions are generally not unique and different values of α result in different Pareto-optimal solutions. All Pareto-optimal solutions can also be found by solving unconstrained optimization problems. This is the case for this example, where all Pareto-optimal solutions can also be found by solving

$$max_{\theta_1(t),\theta_2(t)} \; \alpha x(1) + (1-\alpha) y(1)$$

The different solutions are found by picking different values for α in the interval $[0,1]$.

In figure 5.1, the two outcomes (cooperative and non-cooperative) of the rope-pulling game are displayed. It is visible that, by cooperation, both players improve their objectives.

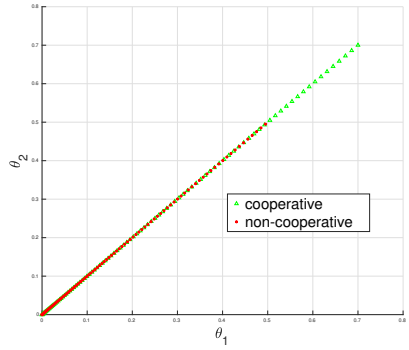


Figure 5.1: The cooperative and non-cooperative outcome of the rope-pulling game

This simple example highlights some crucial themes that the proposed work will address. First of all, it clarifies the main differences, advantages, and disadvantages of the two modes of play, the *Cooperative* and the *Non-Cooperative*, namely. It introduces the concepts of *Nash equilibrium* and *Pareto optima* solutions. It introduced the concept of games (*i.e.,* the system we are dealing with), players (*i.e.,* the agents acting on the system), and their policies (*i.e.,* the actions they apply to the system) and objectives (*i.e.,* the rules to determine the agents' actions).

5.2 DIFFERENTIAL GAME THEORY

This section describes the differential (continuous-time) problem, addressing its game-theoretical formulation and cooperative and non-cooperative solutions. Finally, an analysis of the system response with different parameters tuning is given.

Given the system dynamics (3.8), it is now important to define the

players' objective. Usually, agents' objectives are formulated as cost functions that must be minimized. The objectives can be modeled as functions containing just quadratic terms. There are two main reasons for such a formulation. First, these differential games are analytically and numerically solvable. Second, this linear quadratic problem setting appears if the agents' objective is to minimize the effect of a perturbation of their nonlinear optimally controlled environment.

Depending on the problem description, three main types of games are proposed. In the case players do not make any agreement and seek the optimization of their own cost without trusting each other for cooperation, we are in a Non-Cooperative situation. Conversely, in the Cooperative case, players agree to cooperate because cooperation can improve the outcome for all players with respect to non cooperate. This situation requires trust and agreement from the opponents. The third situation describes the leader-follower case, where the followers minimize their cost function, and the leader minimizes their own based on the follower's choice. The so-called Stackelberg solutions represent its solutions. This work analyzes only the first two models, Non-Cooperative and Cooperative, because they allow easier Role Arbitration and switch from one mode to another due to intrinsic peer interaction. At the same time, the leader-follower situation requires subsequent choices. Moreover, one player always has to anticipate the action of the other, making difficult the actual implementation.

A final consideration of the human and robot objectives should be done. In general, one wants to follow a trajectory during the motion from a pose to a target one. In this paper, we define as $z_{ref,h}$ and $z_{ref,r}$ the desired trajectories that the human and the robot would follow if they were the only agent acting on the system (3.6). In particular, standard motion planners can compute the robot trajectory, while the human trajectory must be identified. The objective of the two presented GT models is, in

the end, to let the system in (3.6) evolve according to a trajectory that is obtained by a combination of the two desired trajectories.

5.2.1 NON-COOPERATIVE GAME THEORY

The non-cooperative formulation of the problem involves competition between players. The objective for each player is the minimization of his cost function. The non-cooperative aspect implies that the players are assumed not to collaborate to attain this goal. In the following, we formulate the non-cooperative problem as a two-player, non-zero-sum game. A complete treatment of non-cooperative game theory is in [11].

In the non-cooperative case, the human and the robot aim to minimize their cost functions, subject to the other influence which are given by

$$J_{h,nc} = \int_0^\infty \left[(z - z_{ref,h})^T Q_h (z - z_{ref,h}) + u_h^T R_h u_h + u_r^T R_{h,r} u_r \right] dt \quad (5.1)$$

and

$$J_{r,nc} = \int_0^\infty \left[(z - z_{ref,r})^T Q_r (z - z_{ref,r}) + u_r^T R_r u_r + u_h^T R_{r,h} u_h \right] dt \quad (5.2)$$

where $z_{ref,h}$ and $z_{ref,r}$ are the human and robot reference targets, Q_h and $Q_r \in \mathbb{R}^{12\times12}$ are weight matrices on the state, R_h and $R_r \in \mathbb{R}^{6\times6}$ are weight matrices on the player's control input and $R_{h,r}$ and $R_{r,h} \in \mathbb{R}^{6\times6}$ are weight matrices on the opponents' control input.

The non-cooperative differential game problem can be summarized as

$$\begin{aligned} &\min_{u_h} J_h(z, u_h, u_r) \\ &\min_{u_r} J_r(z, u_h, u_r) \\ s.t.\ &\dot{z} = Az + B_h u_h + B_r u_r \\ &z(t_0) = z_0 \end{aligned} \quad (5.3)$$

Definition 1: Nash Equilibrium

The so-called Nash equilibrium is the set of solutions to (5.3). A Nash equilibrium is a situation where no player has convenience in changing its control action, formally defined as

$$\begin{aligned} J_h(z, u_h^*, u_r^*) &\leq J_h(z, u_h^*, u_r) \\ J_r(z, u_h^*, u_r^*) &\leq J_r(z, u_h, u_r^*) \end{aligned} \tag{5.4}$$

and the control actions u_h and u_r are the Nash equilibrium policies.

Two types of solutions exist based on the information players have on the current state of the system: open-loop and feedback. We consider only the feedback solutions in this work since position and force measurements are available online to the robot, and the human has direct visual feedback of the task. In the case of linear systems as (3.8) and quadratic cost functions as (5.1) and (5.2), the control policies of the players are computed as

$$u_h = -K_{h,nc}\left(z - z_{ref,h}\right) \tag{5.5}$$

and

$$u_r = -K_{r,nc}\left(z - z_{ref,r}\right) \tag{5.6}$$

The matrices $K_{h,nc}$ and $K_{r,nc}$ are the full-state feedback matrices, computed as $K_{h,nc} = R_h^{-1}B_h^T P_h$ and $K_{r,nc} = R_r^{-1}B_r^T P_r$, where P_h and P_r are solutions of coupled Riccati equations. For simplicity, define with $S_i = B_i R_{i,i} B_i^T$ and $S_{i,j} = B_i R_{i,i}^{-1} R_{j,i} R_{i,i}^{-1} B_i^T$, with $i = \{h,\ r\}$. The two coupled Riccati equations are

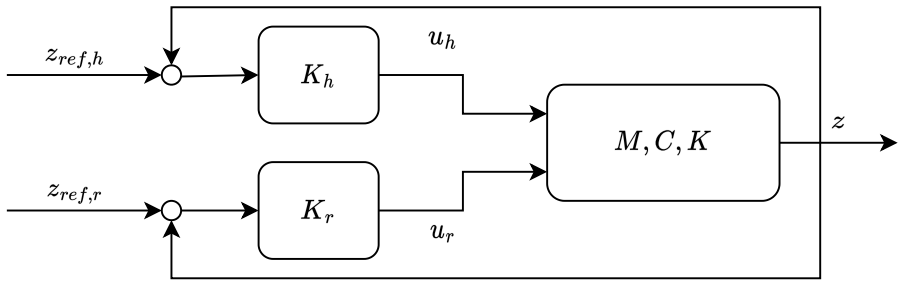


Figure 5.2: The block diagram of the non-cooperative model. The $K_{h,nc}$ and $K_{r,nc}$ are obtained by the minimization of (5.3)

$$\begin{aligned} 0 &= (A - S_r P_r)^T P_h + P_h (A - S_r P_r)^T - P_h S_h P_h + P_r S_{r,h} P_r + Q_h \\ 0 &= (A - S_h P_h)^T P_r + P_r (A - S_h P_h)^T - P_r S_r P_r + P_h S_{h,r} P_h + Q_r \end{aligned} \tag{5.7}$$

which can be solved as in [30]. The block diagram of the non-cooperative game interaction is in figure 5.2.

5.2.2 COOPERATIVE GAME THEORY

The Cooperative formulation of the problem allows agreement between the players to define a shared objective and work together toward it. The objective of each player is shared with the others, and a final, common objective is defined according to the agreement found. Players trust each other, and cooperating can improve their outcomes without hurting others. Each player is generally confronted with a whole set of possible outcomes from which somehow one outcome (which generally does not coincide with a player's overall lowest cost) is cooperatively selected. In the following, We present the Cooperative formulation of the two-players game, based on [29, 31], with an extension to the agreement of a shared reference.

In a cooperative framework, the human and the robot can be seen as two agents, each one to minimize a quadratic cost function, defined as

$$\begin{aligned} J_{h,c} = \int_0^\infty \big[(z - z_{ref,h})^T \, Q_{h,h} \, (z - z_{ref,h}) + \\ + (z - z_{ref,r})^T \, Q_{h,r} \, (z - z_{ref,r}) + u_h^T \, R_h \, u_h \big] \, dt \end{aligned} \tag{5.8}$$

and

$$\begin{aligned} J_{r,c} = \int_0^\infty \big[(z - z_{ref,h})^T \, Q_{r,h} \, (z - z_{ref,h}) + \\ + (z - z_{ref,r})^T \, Q_{r,r} \, (z - z_{ref,r}) + u_r^T \, R_r \, u_r \big] \, dt \end{aligned} \tag{5.9}$$

where $J_{h,c}$ and $J_{r,c}$ are the cost that the human and the robot incur, $Q_{h,h}, Q_{h,r} \in \mathbb{R}^{12\times12}$ and $Q_{r,h}, Q_{r,r} \in \mathbb{R}^{12\times12}$ matrices that weight the state and references and $R_h, R_r \in \mathbb{R}^{6\times6}$ weights on the control input.

By cooperating, a shared objective is defined as

$$J_c = \alpha \, J_h + (1 - \alpha) \, J_r = \int_0^\infty \left(\tilde{z}^T \, Q_c \, \tilde{z} + u^T \, R_c \, u \right) \, dt \tag{5.10}$$

with $\tilde{z} = z - z_{ref}$, where z_{ref}, Q_c and R_c must be defined, and $\alpha \in (0, 1)$ represents the weight each player's cost has in the overall cost. Combining (5.8) and (5.9) into (5.10), after some calculations, can be obtained

$$Q_c = \alpha \, (Q_{h,h} + Q_{h,r}) + (1 - \alpha) \, (Q_{r,h} + Q_{r,r}) \tag{5.11}$$

and

$$R_c = \begin{bmatrix} \alpha \, \hat{R}_h & 0^{6\times6} \\ 0^{6\times6} & (1 - \alpha) \, R_r \end{bmatrix} \tag{5.12}$$

Finally, defining

$$Q_h = \alpha \, Q_{h,h} + (1 - \alpha) Q_{h,r} \tag{5.13}$$

and

$$Q_r = \alpha\, Q_{r,h} + (1-\alpha) Q_{r,r} \tag{5.14}$$

the reference z_{ref} is a weighted composition of the human and robot references that can be expressed as

$$z_{ref} = Q_c^{-1}(z_{ref,h}\, Q_h \; + \; z_{ref,r}\, Q_r) \tag{5.15}$$

With a further step, the system in (3.8) becomes

$$\dot{z} = Az + Bu \tag{5.16}$$

with $A \in \mathbb{R}^{12\times 12}$ defined as before, $B \in \mathbb{R}^{12\times 12} = [B_h \;\; B_r]$ and $u = [u_h \; u_r]^T \in \mathbb{R}^{12\times 1}$.

The Linear Quadratic Differential Game problem can be finally formulated as a classical LQR problem:

$$\begin{gathered} \min_u J_c = \int_0^\infty \left(z^T\, Q_c\, z + u^T\, R_c\, u \right)\, dt \\ s.t.\; \dot{z} = A\, z + Bu \\ z(t_0) = z_0 \end{gathered} \tag{5.17}$$

The problem in (5.17) has infinite solutions lying on the Pareto frontier, depending on the parameter α.

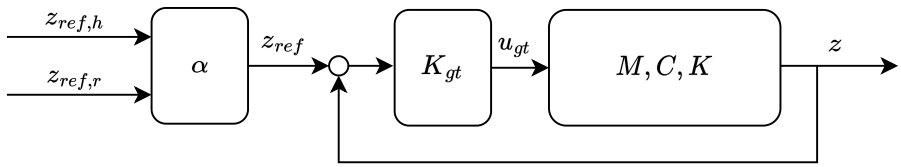


Figure 5.3: The block diagram of the cooperative model. The K is obtained by the minimization of (5.17)

Definition 2: Pareto Efficiency

A set of strategies $\mathcal{U}^* = \{u_h^*,\ u_r^*\}$ is called Pareto efficient if there not exists another set $\mathcal{U} = \{u_h,\ u_r\}$ such that

$$\begin{aligned} J_h(\mathcal{U}) &\leq J_h(\mathcal{U}^*) \\ J_r(\mathcal{U}) &\leq J_r(\mathcal{U}^*) \end{aligned} \tag{5.18}$$

with at least one strict inequality.

All the solutions of problem (5.17) are Pareto efficient, and the choice of one or another opens a new problem, the so-called Bargaining theory. In an LQ-CGT framework, the control action u is defined as full-state feedback as

$$u = -K_{gt}\,\tilde{z} = -K_{gt}\,z + K_{gt}\,z_{ref} \tag{5.19}$$

with $K_{gt} = R_c^{-1}B^T P$ and matrix P solution of the Algebraic Riccati Equation (ARE)

$$0 = A^T P + PA^T - PBR_c^{-1}B^T P + Q_c$$

Note that $u = [u_h, u_r]^T$. Therefore, the human and robot control inputs can be extracted by slicing the vector of control inputs. The block diagram of the cooperative interaction model is visible in figure 5.3.

5.2.3 DGT CONTROLLER PARAMETERS TUNING PERFORMANCES

Since there are many parameters, analyzing how the robot's behavior changes depending on various parameter tuning is interesting. The human parameters cannot be imposed and are recovered as described in the previous chapter. The Impedance system presented in Chapter 3 is simulated.

5.2.3.1 EXPERIMENTS

The CGT and NCGT approaches are also compared with the classical feedback LQR in simulations and preliminary experiments with a trained human.

In the LQR case, the human and the robot aim at minimizing their cost functions without any regard for the other player's actions, which are given by

$$J_{h,lqr} = \int_0^\infty \left((z - z_{ref,h})^T Q_h (z - z_{ref,h}) + u_h^T R_h u_h \right) dt \tag{5.20}$$

and

$$J_{r,lqr} = \int_0^\infty \left((z - z_{ref,r})^T Q_r (z - z_{ref,r}) + u_r^T R_r u_r \right) dt \tag{5.21}$$

which have solutions $K_{h,lqr} = R_h^{-1} B_h^T P_{h,lqr}$ and $K_{r,lqr} = R_r^{-1} B_r^T P_{r,lqr}$, where $P_{h,lqr}$ and $P_{r,lqr}$ are solutions of the AREs.
For all the cases, the matrices Q_h and Q_r are defined as in the cooperative case, as in (5.12), (5.13) and (5.14).

5.2.3.2 SIMULATIONS

We designed the virtual experiments to show the system response according to the tuning of (i) α, (ii) $Q_{r,r}$, and (iii) R_r. Indeed, the robot

cost function can be varied at will, while the human cost function cannot be imposed. The simulation parameters are $z_{ref,h} = 1\ m$; $z_{ref,r} = 0.5m$; state weight matrices $Q_{h,h} = diag([1, 0.0001])$, $Q_{h,r} = diag([0, 0])$, $Q_{r,h} = diag([0, 0])$ and $Q_{r,r} = diag([1, 0.0001])$; control weight matrices $Rh = 0.0005$ and $Rr = 0.0005$[1]. The system (impedance) parameters are $M_i = 10\,N$, $D_i = 25\,Ns/m$ and $K_i = 0\,N/m$. Setting the stiffness coefficient null is common when pHRI is involved, as in [34, 89]. However, this particular stiffness choice is not mandatory, and the method is still valid otherwise. Finally, the control input of the robot represents an additional stiffness, with a stiffness matrix equal to the control gain matrix.

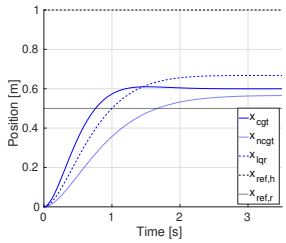
5.2.3.3 SIMULATION 1 (α MODIFICATION)

Figure 5.6 shows the position trajectory and the control inputs for three different values of $\alpha = \{0.2, 0.5, 0.9\}$. The LQR (dashed lines) is less affected by the change of α since the other's control input can be seen as an external disturbance, as the player does not know the opponent's influence in the system. NCGT (dotted line), instead, shows a significant variation. In NCGT and LQR, both players keep putting effort even when an equilibrium is reached. In NCGT, no player has an advantage in reducing his effort when Nash equilibrium is achieved. In LQR, an equilibrium is reached against external disturbance, and the player's effort can be seen as a disturbance rejection. Finally, in CGT, both players reduce their effort to zero after reaching the equilibrium point because they reach the shared reference together, and no one is interested in changing it.

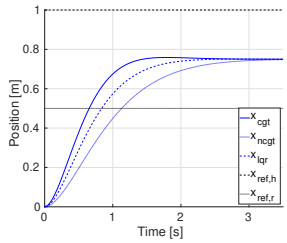
5.2.3.4 SIMULATION 2 (SMALL $Q_{r,r}$)

Figure 5.8 shows the case in which $Q_{r,r} = diag([0.1, 0.0001])$, and $\alpha = \{0.01, 0.05, 0.1, 0.2, 0.5, 0.9\}$. The figure plots only CGT variations since

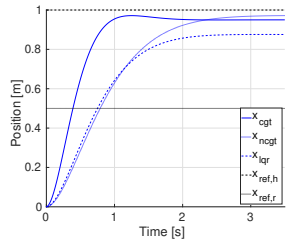
[1] These values are computed via Inverse Optima Control techniques on a trained human. The robot is set to mimic the human cost function.



(a) shared position with $\alpha = 0.2$.

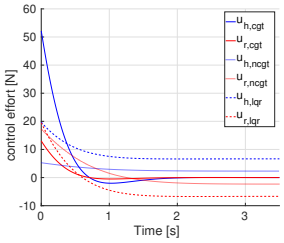


(b) $R_r = 0.0005$

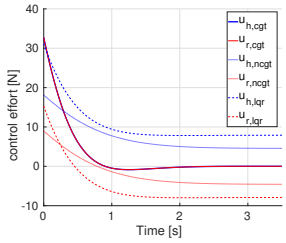


(c) shared position with $\alpha = 0.9$.

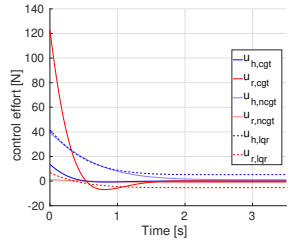
Figure 5.4: Positions



(a) control effort with $\alpha = 0.2$. Blue lines human control actions, red lines robot



(b) control effort with $\alpha = 0.5$. Blue lines human control actions, red lines robot



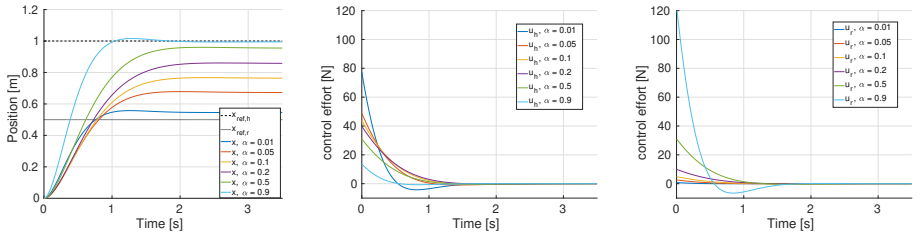
(c) control effort with $\alpha = 0.9$. Blue lines human control actions, red lines robot.

Figure 5.5: Efforts

Figure 5.6: position and control effort comparison varying weight parameter α.

the other controllers have similar behaviors to the one presented for the CGT. Figure 5.8(a) shows the variation in the state according to α by lowering the robot's state weight matrix. Because the weight on the robot's reference is low, the state goes quickly towards the direction of the human's reference even for small values of α (> 0.1). The robot reference becomes relevant only with minimal values of α. In this simulation, $\alpha = 0.1$ allows for making them comparable, being the robot's state weight one-tenth of the human's, Figures 5.8(b-c) show the control efforts are similar

to Simulation 1 (*e.g.*, see the case $\alpha = 0.5$).



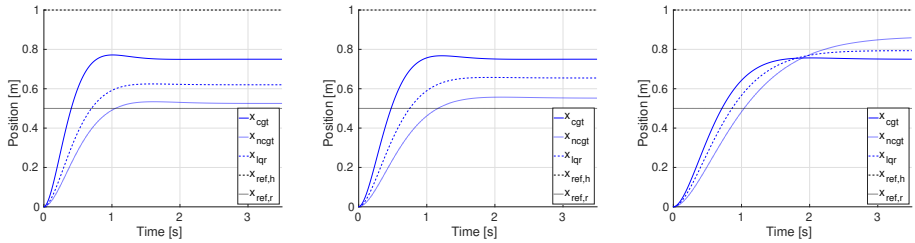
(a) position state history with increasing α **(b)** human control action with increasing α. **(c)** robot control action with increasing α.

Figure 5.7: Positions

Figure 5.8: Position and control effort comparison varying weight parameter α with low weight on the robot state $Q_{r,r} = diag([0.1, 0.0001])$.

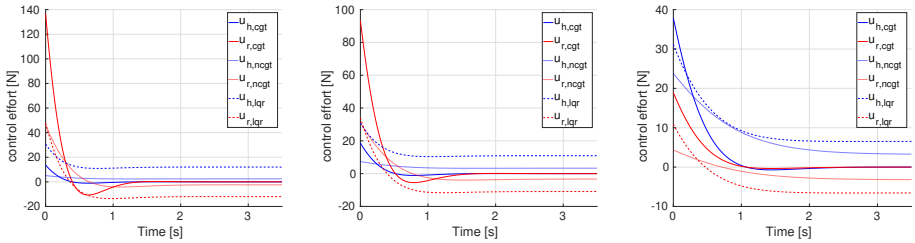
5.2.3.5 SIMULATION 3 (SMALL $R_r = \{5e^{-5}, 1e^{-4}, 1e^{-3}\}$)

Figure 5.11 shows the results by keeping $\alpha = 0.5$. On the one hand, CGT keeps constant the equilibrium position, resulting from an agreement depending on α. On the other hand, the human and robot control inputs vary according to R_r. The effort of the robot increases as the Rr decreases, and vice versa for the human effort. Conversely, LQR and NCGT reach different equilibrium points according to R_r. Indeed, LQR can partially compensate for the human control effort like an external disturbance. Figure 5.12 shows the costs according to α. The human cost of the cooperative case is lower for $\alpha > 0.6$. Increasing α is preferable for the human when he wants to lead because the robot assists more humans in reaching their target. Conversely, low α means that the human is more willing to help the robot reach its target.



(a) state with $Rr = 0.00005$. **(b)** state with $Rr = 0.0001$ **(c)** state with $Rr = 0.001$.

Figure 5.9: Positions



(a) control actions with $Rr = 0.00005$. **(b)** control actions with $Rr = 0.0001$. **(c)** control actions with $Rr = 0.001$.

Figure 5.10: Efforts

Figure 5.11: Position and control efforts according to different robot weight on control action R_r.

5.2.3.6 SIMULATIONS NOTES

CGT minimizes both players' control effort, at least at the equilibrium. CGT is more favorable for humans with high values of α, possibly with the human leader. At the same time, NCGT or LQR are more likely to be adopted with low values of α for the robot-leading situation. Low values of R_r allow the robot to be more helpful with high α (human leading) and be more capable of driving the system to its reference with low values of α (robot leading).

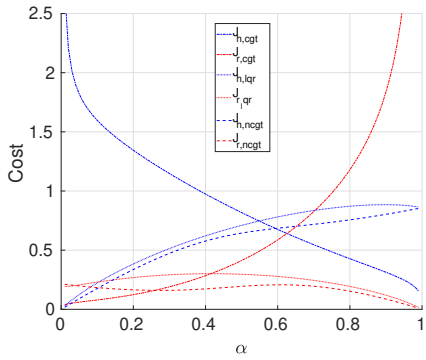


Figure 5.12: Human (blue lines) and robot (red lines) costs for the three controllers for various α in the simulated cases.

5.2.3.7 REAL-WORLD EXPERIMENTS

Experiments are performed with a human operator cooperating with a UR5 robot. Two relevant components of the proposed controller must be addressed when dealing with real humans: knowledge of the human's reference and cost function. An Inverse Optimal Controller (IOC) [41, 144] recovers an estimate of the human parameters, $\hat{Q}_h$ and $\hat{R}_h$. The robot knows the human target that defines the reference in 5.15. This might seem too restrictive, but it allows fair testing of the controller's performance. Human target pose identification is a well-investigated topic [21, 90].

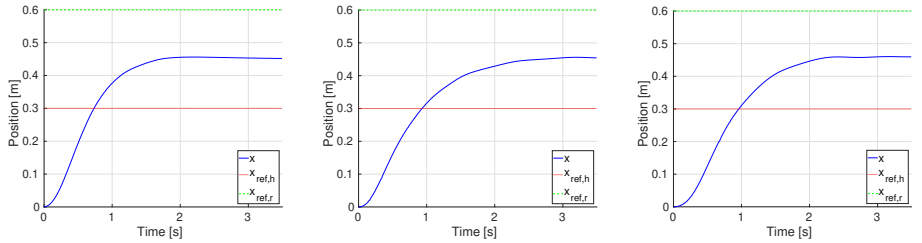
The humans performing the experiments know the robot's behavior and the different control techniques. This awareness is achieved after proper training, even if the human does not know anything initially. This choice is made for different reasons. (1) the methodology copes with a human aware of the robot, so it is expected that the operator knows (at least approximately) the robot's behavior and has some knowledge. (2) The knowledge of the opponent's intention is among the hypothesis of the GT formulation. The purpose of the experiments is to verify if the CGT describes pHRI and to avoid additional noise. This hypothesis must

be relaxed in real-world applications, which will be addressed in future works. (3) If the human is aware of the CGT behavior of the robot, better results can be achieved thanks to cooperation. This experiment can be seen as a test for the method's long-term applicability. This assumption is also in [121].

The operator's target is shown on a monitor, with the robot one and the robot tip's current position. The human is asked to reach the target position in physical interaction with the robot. Such a target position is not visible on the monitor. The human only knows the weight $\alpha = \{0.2, 0.5, 0.9\}$ (defined offline). The target poses are $z_{ref,h} = 0.6m$ and $z_{ref,r} = 0.3m$.

The robot applies a virtual force u_r, while the human applies a real, measured force to the system. The balance of the two forces is turned into a velocity command through an impedance controller. The human and the robot move the system toward their goals (which are different for NCGT and LQR, the agreed one for CGT), knowing each other's goals.

Figure 5.15 plots the state and efforts for three controllers with $\alpha = 0.5$. As from 5.15(a-b-c), the equilibrium solution is reached in all three cases. The human measured and nominal control efforts are in 5.15(d-e-f). The nominal control effort is computed with the gain matrix $K_{h,i} = R_h^{-1} B_h^T P_{h,i}$ with $i = \{cgt, lqr, ncgt\}$. As expected from simulations, CGT shows that once the equilibrium (*i.e.*, agreed reference) is reached, neither the human nor the robot puts any more effort into the system. On the contrary, the other two controllers still inject effort into the system to keep the equilibrium. Moreover, this continuous pulling of the robot towards its reference introduces oscillations, as the human does not exactly behave as modeled. On the contrary, CGT appears more smooth. Humans and robots share the same target and cooperatively steer the system towards it without "pulling" it in different directions. The human costs of performing real-world experiments are computed for a time window of 3.5s.

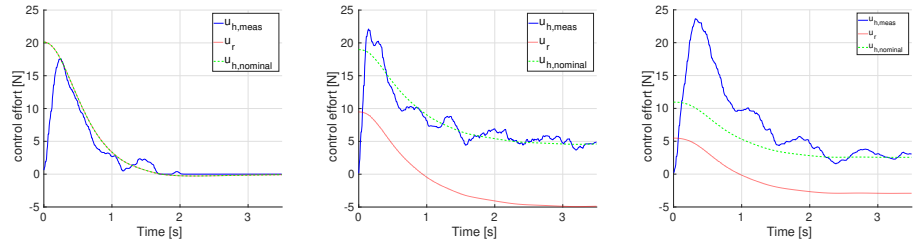


(a) measured position for CG with $\alpha = 0.5$.

(b) measured position for LQR with $\alpha = 0.5$.

(c) measured position for NCG with $\alpha = 0.5$.

Figure 5.13: Positions



(a) CG measured human and robot control efforts and nominal human control effort.

(b) LQR measured human and robot control efforts and nominal human control effort.

(c) NCG measured human and robot control efforts and nominal human control effort.

Figure 5.14: Efforts

Figure 5.15: position and control effort comparison for the three controllers with the weight parameter $\alpha = 0.5$

The results are in figure 5.16. This measurement reflects what is observed in the simulated scenarios: the human incurs high costs for low values of α, and vice versa, low costs for high values. The opposite happens for the LQR and NCGT because it is more convenient to work with high values of α for humans. In such a case, a lower effort produces better results. High values of α lead to a shared target close to the human's, and the robot drives the system toward it.

Finally, the error between measured and nominal human control ef-

forts is computed for the three controllers for three values of α. The errors are the sum over the time windows of 3.5s and normalized by the maximum nominal control input:

$$\mathcal{E}_u = \frac{1}{max(u_{h,n})} \int_{t_i}^{t_f} \|u_h - u_{h,n}\| \, dt \tag{5.22}$$

Figure 5.17 shows that the CG better fits with the pHRI, making it more applicable to real-world applications.

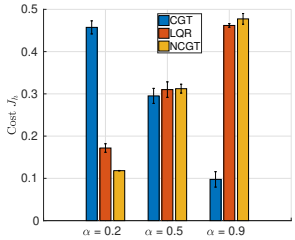


Figure 5.16: Computed Human cost varying α.

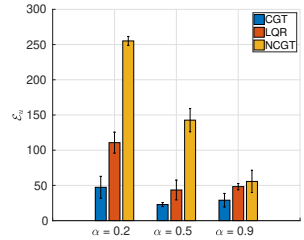


Figure 5.17: human's control errors for the three controllers varying α.

5.2.3.8 DISCUSSION

Consider that a high value of α corresponds to the human-leading, while a low value corresponds to the robot-leading. In the real world, a human likely wants the help of the robot to pursue his target but would probably not accept putting too much effort into the task to help the robot when the α is low. In this sense, a scenario suggests using CGT when the human leads and NCGT or LQR when the robot leads. Therefore, the robot helps the human in the first case and drives the system toward its reference when it wants to lead.

As simulations show, a small robot's control action weight increases the robot's reactiveness, helping humans. For high values of α, the robot is more willing to help the human reach its target, while for low values, the robot is more willing to reach its target and can exert higher forces

compared with the case in which $R_r = R_h$. This behavior could make the cooperation between a human and a robot more comfortable.

Real-world experiments confirm that for the CGT case, the human cost is higher for low values of α and vice versa for the other two. Comparing the three real-world experiments, CGT fits the model better, allowing modeling the pHRI as a cooperative game, which might be helpful for simulations and tuning of the controllers. Moreover, having a model that reasonably fits with data possibly allows for online prediction of human behavior. This prediction capability can improve assistive controllers' design and provide the robot with some information about the future behavior of its opponents.

Limitations to real-world applications are mainly due to human modeling. First, the experiments are performed by a trained human, aware of CGT methods, while operators do not precisely follow the CGT formulation. Second, human costs vary from person to person and possibly along the work shift. Third, the task could make it vary. For this purpose, the presented method should be enriched with a robustness term. Furthermore, it could be possible to investigate the performance of online IOC techniques to identify the cost function in run-time. Finally, as stated above, the human target is taken for granted in this work. In real-world applications, this is typically not true and predictive models for the human target should be investigated and integrated.

5.2.3.9 CONCLUSIONS

This section presented a controller for the pHRI based on Cooperative Game Theory. A shared reference is computed, and a solution to the LQ-CGT reference tracking problem is obtained. Starting from two different references, the controller converges to an agreed one, sharing the required effort to drive the system to the target. The proposed method is compared in simulations and real-world experiments with the two

controllers, LQR and NCGT. A set of simulations investigates the performances of the proposed controller, varying the control parameters. Real-world experiments partially confirm what was observed in the simulated cases and show that CGT better fits with the pHRI task. Finally, a discussion of the results obtained is proposed, addressing the possibility and limitations of the proposed methodology. Future works will involve online human-robot role arbitration, making variable the parameter α in run-time, possibly switching between CGT and LQR/NCGT controllers. Finally, online human target position prediction will be included in the control scheme, as it is crucial for real-world applications.

5.3 GAME-THEORETIC DISTRIBUTED MODEL PREDICTIVE CONTROL

This section presents the formulation of the distributed Model Predictive Control (dMPC) problem. Then, it presents the solution to the dMPC problem in a Game-Theoretic (GT) framework for both the Non-Cooperative and the Cooperative cases, leading to the GT-dMPC framework. Implementing a Model Predictive Control framework is particularly interesting in the pHRI scenario since it allows to continuously update the control inputs according to the system's current state. This allows to take into account human uncertainties and unpredictability easily. Moreover, it fully exploits the long-term human intention prediction presented in chapter 4.

5.3.1 DISTRIBUTED MODEL PREDICTIVE CONTROL - DMPC

According to the dMPC presented in [142], consider the discrete version of the Cartesian impedance 3.12, written as

$$\begin{aligned} z(k+1) &= A_d z(k) + B_{h,d} u_h + B_{r,d} u_r \\ y(k) &= C_d\, z(k) \end{aligned} \tag{5.23}$$

with z, A_d, $B_{h,d}$ and $B_{r,d}$ defined as in chapter 3, and $C_d \in \mathbb{R}^{m\times 12}$ is defined according to the desired output. According to the MPC formulation, define the prediction horizon as N_p and the control horizon as N_c. The output prediction in the future N_p sampling times, calculated at time k is given by:

$$\begin{cases} y(k+1) = C_d A_d z(k) + C_d B_h u_h(k) + C_d B_r u_r(k) \\ y(k+2) = C_d A_d^2 z(k) + C_d A_d B_h u_h(k) + \\ \qquad\quad + C_d A_d B_r u_r(k) + C_d B_h u_h(k+1) + C_d B_r u_r(k+1) \\ \quad \vdots \\ y(k+N_p) = C_d A_d^{N_p} z(k) + \cdots + C_d A_d^{N_p - N_c} B_h u_h(k+N_c-1) + \\ \qquad\quad + C_d A_d^{N_p - N_c} B_r u_r(k+N_c-1) \end{cases} \tag{5.24}$$

which can be written in compact matrix form as

$$Y(k) = Fz(k) + \Phi_h U_h(k) + \Phi_r U_r(k) \tag{5.25}$$

where $Y \in \mathbb{R}^{mN_p}$ is the predicted output and is equal to

$$Y(k) = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N_p) \end{bmatrix} = \begin{bmatrix} C_d\, z(k+1) \\ C_d\, z(k+2) \\ \vdots \\ C_d\, z(k+N_p) \end{bmatrix} \tag{5.26}$$

$F \in \mathbb{R}^{mN_p \times 12}$ is the free response matrix, equal to:

$$F = \begin{bmatrix} C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_d^{N_p} \end{bmatrix} \tag{5.27}$$

and $\Phi_i \in \mathbb{R}^{mNp \times 6N_c}$, with subscript $i = h, r$ denoting the human and robot, are matrices representing the forced response

$$\Phi_i = \begin{bmatrix} C_d B_{i,d} & 0 & \dots & 0 \\ C_d A_d B_{i,d} & C_d B_{i,d} & \dots & 0 \\ C_d A_d^2 B_{i,d} & C_d A_d B_{i,d} & \dots & 0 \\ \vdots & \vdots & \ddots & \\ C_d A_d^{N_p-1} B_{i,d} & C_d A_d^{N_p-2} B_{i,d} & & C_d A_d^{N_p-N_c} B_{i,d} \end{bmatrix} \tag{5.28}$$

The two vectors $U_h(k)$ and $U_r(k) \in \mathbb{R}^{6N_c}$ are the input vectors along the horizon that must be defined.

5.3.2 NON-COOPERATIVE GT-DMPC

Consider also that the human and the robot have their reference trajectories. Define with $Y_{ref,h}(k) = [y_{ref,h}(k+1),\ y_{ref,h}(k+2),\ \dots,\ y_{ref,h}(k+N_p)]^T$ the trajectory a human would like to follow over the next N_p timesteps if there is no interaction with the robot. Similarly, $Y_{ref,r}(k) = [y_{ref,r}(k+1),\ y_{ref,r}(k+2),\ \dots,\ y_{ref,r}(k+N_p)]^T$ denotes the trajectory planned by a motion planner that the robot would like to follow over the next N_p timesteps if there is no interaction with the human. The predicted track-

ing errors at each timestep are

$$E_i(k) = \begin{bmatrix} e(k+1) \\ e(k+2) \\ \vdots \\ e(k+N_p) \end{bmatrix} = \begin{bmatrix} y(k+1) - y_{ref,i}(k+1) \\ y(k+2) - y_{ref,i}(k+2) \\ \vdots \\ y(k+N_p) - y_{ref,i}(k+N_p) \end{bmatrix} \tag{5.29}$$

with subscript $i = h, r$ denoting the human and robot. The tracking errors in compact form are:

$$E_h = Y(k) - Y_{ref,h}(k) \tag{5.30}$$

and

$$E_r = Y(k) - Y_{ref,r}(k) \tag{5.31}$$

According to [75, 121], the human and the robot have the objective of minimizing a cost function that depends on the tracking error and the control effort required, defined as

$$\begin{aligned} J_h(k) &= \sum_{l=1}^{N} e_h(k+l)^T \, Q_h \, e_h(k+l) + u_h(k+l)^T \, R_h \, u_h(k+l) \\ &= E_h^T \tilde{Q}_h E_h + U_h^T \tilde{R}_h U_h \end{aligned} \tag{5.32}$$

and

$$\begin{aligned} J_r(k) &= \sum_{l=1}^{N} e_r(k+l)^T \, Q_r \, e_r(k+l) + u_i(k+l)^T \, R_i \, u_i(k+l) \\ &= E_r^T \tilde{Q}_r E_r + U_r^T \tilde{R}_r U_r \end{aligned} \tag{5.33}$$

with

$$\tilde{Q}_i = \begin{bmatrix} Q_i & & \\ & \ddots & \\ & & Q_i \end{bmatrix} \quad \text{and} \quad \tilde{R}_i = \begin{bmatrix} R_i & & \\ & \ddots & \\ & & R_i \end{bmatrix} \tag{5.34}$$

The NGT-dMPC problem for the Cooperative Game Theoretic pHRI can then be summarized as

$$\begin{aligned} \min_{u_h} J_h &= E_h(k)^T\, Q_h\, E_h(k) + U_h(k)^T\, R_h\, U_h(k), \\ \min_{u_r} J_r &= E_r(k)^T\, Q_r\, E_r(k) + U_r(k)^T\, R_r\, U_r(k), \\ s.t.\ Y(k) &= Fz(k) + \Phi_h U_h(k) + \Phi_r U_r(k). \end{aligned} \tag{5.35}$$

Following [142] and [75], the solution of problems (5.35) can be computed as

$$U^* = \begin{bmatrix} U_h^* \\ U_r^* \end{bmatrix} = \begin{bmatrix} I & K_h \\ K_r & I \end{bmatrix}^{-1} \begin{bmatrix} L_h & 0 \\ 0 & L_r \end{bmatrix} \begin{bmatrix} Z_h \\ Z_r \end{bmatrix} \tag{5.36}$$

in which, defining

$$\begin{aligned} S_h &= (\Phi_h^T Q_h \Phi_h + R_h)^{-1} \Phi_h^T Q_h \\ S_r &= (\Phi_r^T Q_r \Phi_r + R_r)^{-1} \Phi_r^T Q_r \end{aligned} \tag{5.37}$$

the gains are computed as

$$\begin{aligned} K_h &= S_h \Phi_h \\ L_h &= [-S_h F_h \quad S_h] \end{aligned} \tag{5.38}$$

$$\begin{aligned} K_r &= S_r \Phi_r \\ L_r &= [-S_r F_r \quad S_r] \end{aligned} \tag{5.39}$$

and the states are

$$Z_h = \begin{bmatrix} z(k) \\ y_{ref,h}(k+1) \\ \vdots \\ y_{ref,h}(k+N) \end{bmatrix} = \begin{bmatrix} z(k) \\ Y_{ref,h}(k) \end{bmatrix} \tag{5.40}$$

and

$$Z_r = \begin{bmatrix} z(k) \\ y_{ref,r}(k+1) \\ \vdots \\ y_{ref,r}(k+N) \end{bmatrix} = \begin{bmatrix} z(k) \\ Y_{ref,r}(k) \end{bmatrix} \tag{5.41}$$

Finally, to implement the receding horizon logic, only the components of U_h^* and U_r^* relative to the next step are used, hence $u_h(k) = U^*(1)$ and $u_r(k) = U^*(1+N)$.

5.3.3 COOPERATIVE GT-DMPC

According to [142], to implement the cooperative GT dMPC, the system in 3.12 can be augmented as

$$\begin{aligned} z_a(k+1) &= A_a z_a(k) + B_{h,a} u_h + B_{r,a} u_r \\ y_a(k) &= C_a\, z_a(k) \end{aligned} \tag{5.42}$$

with $z_a = [z^T \; z^T]^T$, $A_a = blkdiag(A_d,\ A_d)$, $B_{h,a} = [B_h \; B_h]^T$ and $B_{r,a} = [B_r \; B_r]^T$, and $C_a \in \mathbb{R}^{2m \times 24}$ is defined according to the desired output. Following the same procedure in (5.24), the system in (5.42) can be written as

$$Y_a(k) = F_a z_a(k) + \Phi_h U_h(k) + \Phi_r U_r(k) \tag{5.43}$$

where the predicted output $Y_a \in \mathbb{R}^{2mN_p}$, the free response matrix $F_a \in \mathbb{R}^{2mN_p \times 24}$, and the forced responses Φ_h and $\Phi_r \in \mathbb{R}^{2mNp \times 6N_c}$ are com-

puted similarly to (5.26), (5.27), and (5.28), respectively. The two vectors $U_h(k)$ and $U_r(k) \in \mathbb{R}^{6N_c}$ are the input vectors along the horizon that must be defined.

Consider also that the human and the robot have their reference trajectories, defined with $Y_{ref,h}(k)$, and $Y_{ref,r}(k)$ as in the previous section. Recalling the tracking errors from (5.30) and (5.31), the augmented system error can be defined as

$$E_a(k) = \begin{bmatrix} y(k) - y_{ref,h}(k+1) \\ y(k) - y_{ref,r}(k+1) \\ y(k+1) - y_{ref,h}(k+2) \\ y(k+1) - y_{ref,r}(k+2) \\ \vdots \\ y(k+N_p-1) - y_{ref,h}(k+N_p) \\ y(k+N_p-1) - y_{ref,r}(k+N_p) \end{bmatrix} \tag{5.44}$$

According to [75, 121], the human and the robot have the objective of minimizing a cost function that depends on the tracking error and the control effort required, defined as

$$\begin{aligned} J_i(k) &= \sum_{l=1}^{N} e_i(k+l)^T\, Q_{i,i}\, e_i(k+l) + e_j(k+l)^T\, Q_{i,j}\, e_j(k+l) + \\ &+ u_i(k+l)^T\, R_i\, u_i(k+l) \\ &= \sum_{l=1}^{N} \begin{bmatrix} e_i(k+l)^T & e_j(k+l)^T \end{bmatrix} \begin{bmatrix} Q_{i,i} & 0 \\ 0 & Q_{i,j} \end{bmatrix} \begin{bmatrix} e_i(k+l) \\ e_j(k+l) \end{bmatrix} + \\ &+ u_i(k+l)^T\, R_i\, u_i(k+l) \\ &= \sum_{l=1}^{N} e_a(k+l)^T\, Q_i\, e_a(k+l) + u_i(k+l)^T\, R_i\, u_i(k+l) \end{aligned} \tag{5.45}$$

with $i, j = \{h, r\}$ subscripts denoting the human and robot matrices. In equations 5.45, $Q_{i,j}$ defines the weight that the human and the robot as-

sign to their own and the opponent's reference tracking error, $e_i(k+l) = y(k+l) - y_{ref,i}(k+l)$ refers to the tracking errors foreseen for the human and the robot at time step $k+l$, with $e_a(k+l) = \begin{bmatrix} e_h(k+l) \\ e_r(k+l) \end{bmatrix}$, and $u_h(k+l)$ and $u_r(k+l)$ are the control inputs of the human and the robot at time step $k+l$. Matrices $\tilde{Q}_i$ and $\tilde{R}_i$ are defined as in (5.34). Finally, equations (5.45) can be rewritten in compact form as

$$J_i(k) = E_a(k)^T \, \tilde{Q}_i \, E_a(k) + U_i(k)^T \, \tilde{R}_i \, U_i(k) \tag{5.46}$$

5.3.3.1 COOPERATIVE GAME-THEORETIC PROBLEM FORMULATION

In a cooperative game, the players agree to cooperate by sharing a common objective. The common objective is a weighted sum of the singular objectives, and the weights are the bargaining outcome. The bargaining is defined by a set of parameters $\{\alpha_i,\ i = 1 : N_{players},\ \sum_{i=1}^{N} \alpha_i = 1,\ 0 < \alpha_i < 1\}$, which weights the contribution of each player.

The shared objective is defined as

$$J_{gt}(k) = \alpha \, J_h(k) + (1-\alpha) \, J_r(k) \tag{5.47}$$

It is then possible to define $Q_{gt} = \alpha \, \tilde{Q}_h + (1-\alpha) \, \tilde{Q}_r$, $R_{gt,h} = \alpha \, \tilde{R}_h$ and $R_{gt,r} = (1-\alpha) \, \tilde{R}_r$. Therefore, from equation 5.47, the Cooperative Game-Theoretic single objectives of the two players are defined as

$$J_{h,gt}(k) = E_a(k)^T \, Q_{gt} \, E_a(k) + U_h(k)^T \, R_{gt,h} \, U_h(k) \tag{5.48}$$

and

$$J_{r,gt}(k) = E_a(k)^T \, Q_{gt} \, E_a(k) + U_r(k)^T \, R_{gt,r} \, U_r(k) \tag{5.49}$$

The GT-dMPC problem for the Cooperative Game Theoretic pHRI can then be summarized as

$$\begin{aligned} \min_{u_h} J_{h,gt} &= E_a(k)^T\, Q_{gt}\, E_a(k) + U_h(k)^T\, R_{gt,h}\, U_h(k), \\ \min_{u_r} J_{r,gt} &= E_a(k)^T\, Q_{gt}\, E_a(k) + U_r(k)^T\, R_{gt,r}\, U_r(k), \\ s.t.\ Y_a(k) &= F_a z(k) + \Phi_h U_h(k) + \Phi_r U_r(k). \end{aligned} \tag{5.50}$$

Following [142] and [75], the solution of problems (5.50) can be computed as

$$U^* = \begin{bmatrix} U_h^* \\ U_r^* \end{bmatrix} = \begin{bmatrix} I & K_h \\ K_r & I \end{bmatrix}^{-1} \begin{bmatrix} L_h & 0 \\ 0 & L_r \end{bmatrix} \begin{bmatrix} Z_h \\ Z_r \end{bmatrix} \tag{5.51}$$

in which, defining

$$\begin{aligned} S_h &= (\Phi_h^T Q_{gt} \Phi_h + R_{gt,h})^{-1} \Phi_h^T Q_{gt} \\ S_r &= (\Phi_r^T Q_{gt} \Phi_r + R_{gt,r})^{-1} \Phi_r^T Q_{gt} \end{aligned} \tag{5.52}$$

the gains are computed as

$$\begin{aligned} K_h &= S_h \Phi_h \\ L_h &= [-S_h F_h \quad S_h] \end{aligned} \tag{5.53}$$

$$\begin{aligned} K_r &= S_r \Phi_r \\ L_r &= [-S_r F_r \quad S_r] \end{aligned} \tag{5.54}$$

and

$$Z_h = Z_r = \begin{bmatrix} z_{gt}(k) \\ y_{ref,h}(k+1) \\ y_{ref,r}(k+1) \\ \vdots \\ y_{ref,h}(k+N) \\ y_{ref,r}(k+N) \end{bmatrix} \tag{5.55}$$

Finally, to implement the receding horizon logic, only the components of U_h^* and U_r^* relative to the next step are used, hence $u_h(k) = U^*(1)$ and $u_r(k) = U^*(1+N)$.

5.4 GT-DMPC PARAMETERS TUNING PERFORMANCES

This section presents simulations of the GT-dMPC frameworks presented in section 5.3 to understand its performances for different values of the main parameters, with the objective of understanding the set that better fits with the pHRI application. Then the training procedure of the RNN+FC model is explained and evaluated.

The GT-dMPC frameworks presented in section 5.3 is simulated with different parameters tuning. For the simulation, only one dof is considered. The system is discretized at 0.008 seconds. The parameters analyzed are the prediction horizon $N_p = \{0.04,\ 0.16,\ 0.4\}$ which correspond to 5, 20, and 50 timesteps horizons, the bargaining solution parameter $\alpha = \{0.2,\ 0.5,\ 0.9\}$, and the weight of the robot's cost function for the control input $R_r = \{0.001,\ 0.0005,\ 0.0001\}$. The parameters of the human cost function cannot be made variable arbitrarily because they are descriptive of the intrinsic human behavior. Such parameters can only be recovered as in [41, 82]. In this work, an average sufficiently descriptive is used, resulting in $Q_{h,h} = diag([1, 0.0001])$, $Q_{h,r} = diag([0,0])$ and $R_h = 0.0005$. The impedance parameters of (3.7) are set to $m_i = 10$, $c_i = 100$ and $k_i = 0$. The two references are given as $Y_{ref,h}(k) = sin(k : k + N_p)$ and $Y_{ref,r}(k) = 0.5\, sin(k : k + N_p)$.

To compare the tracking performances of the proposed controller with different tuning, we define a tracking error index as

$$e_{trac} = \int_0^t \|Y_{ref,h} - x\| dt \tag{5.56}$$

Note that we are interested in designing an assistive controller, and the

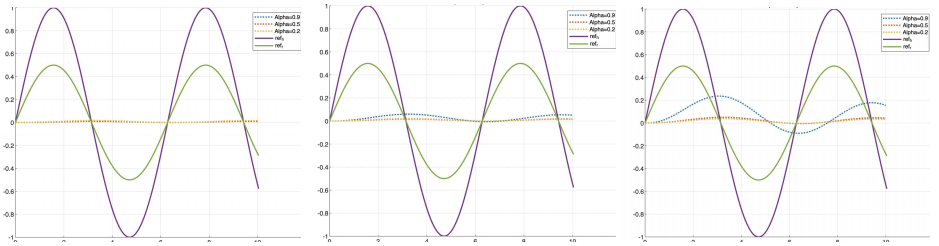
objective is to reduce the tracking error of the desired human reference trajectory. Table 5.1 presents results for the simulated scenarios of the e_{trac} index. As visible, with high values of $\alpha(= 0,9)$, the system tends to fol-

Table 5.1: tracking performances

	$R_r = 0.001$			$R_r = 0.0005$			$R_r = 0.0001$		
α	0.2	0.5	0.9	0.2	0.5	0.9	0.2	0.5	0.9
N_5	6.15	6.15	6.14	6.15	6.15	6.14	6.14	6.14	6.09
N_{20}	5.99	6.07	5.93	5.96	6.03	5.60	5.66	5.56	2.86
N_{50}	4.45	5.16	3.77	4.25	4.73	2.48	3.28	2.71	0.61

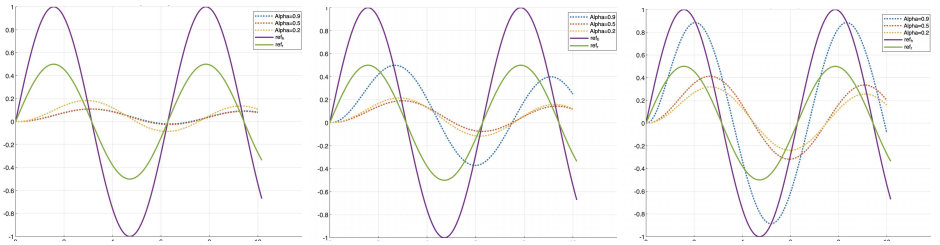
low the human reference closer than the robot's one. Therefore, since this work aims to define an assistive controller for the human, it is suggested to use high values of α, thus allowing for better assistance from the robot. On the contrary, according to GT formulation, with $\alpha = 0,2$, it should be the human that helps the robot follow its reference. This situation is not realistic. In fact, humans do not know the robot's reference. Moreover, it is unnatural for the human to assist the robot, and it prevents the human from being assisted. So low values of α should be discarded.

Figure 5.21 shows some of the system tracking simulations.



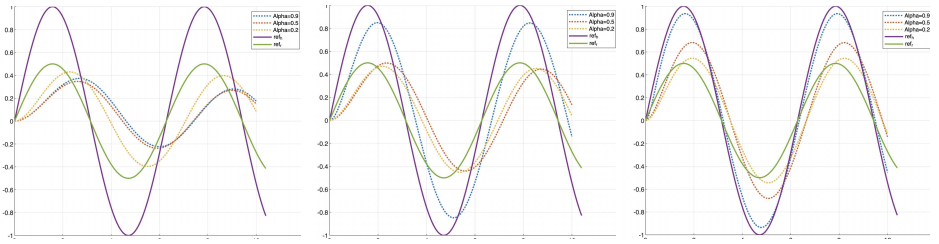
(a) $R_r = 0.01$ **(b)** $R_r = 0.0005$ **(c)** $R_r = 0.0001$

Figure 5.18: $N_p = 5$



(a) $R_r = 0.01$ **(b)** $R_r = 0.0005$ **(c)** $R_r = 0.0001$

Figure 5.19: $N_p = 20$



(a) $R_r = 0.01$ **(b)** $R_r = 0.0005$ **(c)** $R_r = 0.0001$

Figure 5.20: $N_p = 50$

Figure 5.21: Qualitative evaluation of tracking performances according to different tuning parameters of the GT-dMPC controller. Low values of R_r and long prediction horizons generally allow human reference tracking performances.

CHAPTER 6

ASSISTIVE CONTROLLERS IMPLEMENTATION

This chapter shows the application of the interaction models presented in chapters 5.2 and 5.3 to assistive controllers. First, it presents applications of the GT controllers as purely assistive controllers. Such controllers are assistive in that they help humans pursue a goal, reducing the required load (e.g., by reducing the required force). Then, it addresses role arbitration methodologies. Finally, it presents an industrial use case in which applying the methodologies presented in the previous chapters allows one to perform a complex task such as precise positioning of a large patch of carbon fiber material with high precision.

6.1 DIFFERENTIAL GT ASSISTANCE CONTROLLER

This section presents the controller implementation proposed in 5.2, enhanced by the human intention prediction of section 4.3. In particular, the Cooperative model is considered. Indeed, under certain parameters tuning, it allows for the robot to assist the human in pursuing the goal.

Please refer to 5.2.3 for more details and considerations on this.

This section considers a planar motion. Therefore, the impedance parameters are $M_i = diag(10, 10)$, $C_i = diag(100, 100)$, and the stiffness is set to null $K_i = diag(0, 0)$, as typically in pHRI. The mass and damping coefficients have been hand-tuned to allow smooth motions. The cost of the two players are set as $Q_{h,h} = Q_{r,r} = diag([1, 1, 0.0001, 0.0001])$, $Q_{h,r} = Q_{r,h} = 0^{2\times2}$ and $R_h = R_r = diag([0.0005, 0.0005])$. The human cost function parameters $Q_{h,h}$, $Q_{h,r}$ and R_h are recovered via Inverse Optimal Control (IOC) as in [41], and an average value is used. The robot parameters $Q_{r,r}$, $Q_{r,h}$, and R_r are set equal to the human's to mimic a person. Different tuning may result in more assistive behavior, which might be desirable. Since the DGT controller accepts as a reference for (5.15) only one set-point rather than a prediction horizon, we take the 20^{th} point of the prediction horizon. Finally, $\alpha = 0.8$ is chosen to allow sufficient assistance. This value allows high assistance and the robot to recover the position of the robot set-point autonomously. Moreover, this value and higher allows the assumption $z_{ref} \simeq z_{ref,h}$ to hold.

We compare the proposed approach with two common strategies, Impedance Control (IMP) and Manual Guidance (MG). Both controllers rely on (3.7), with the difference that the robot assistive contribution is null ($u_r = 0$) for the IMP, and the stiffness matrix is null $K_i = 0^{6\times6}$ for MG. We maintain the same mass for the IMP case, superimpose the stiffness matrix to $K_i = diag([200, 200])$, and the damping to the 90% of the critical damping.

We compare the Root Mean Square of the interaction force, a common performance index for pHRI controllers,

$$f_{RMS} = \sqrt{\frac{1}{L}\sum_{k=0}^{L} f_k^2\, dt}, \tag{6.1}$$

with L length of the trajectory as before and f_k the module of the measured interaction force at time instant k.

Figure 6.1 reports the comparison of f_{RMS} with the other IMP and MG, considering our model after TL. The t-test shows no difference between MG and IMP, while between the TL and the other two, there is. Although the improvement might seem minor, with a different tuning of the controller (higher α or smaller R_r) the robot might increase the assistance so that the force decrease even more.

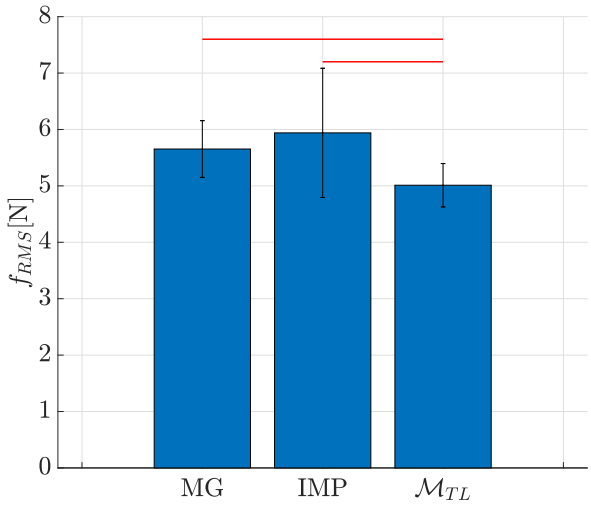


Figure 6.1: The f_{RMS} for Manual Guidance (MG), Impedance (IMP), and our assistive control with TL. A red bar indicates p-values < 0.05. The null hypothesis is rejected between $\mathcal{M}_{TL}$ and both MG and IMP, while it is not rejected between MG and IMP.

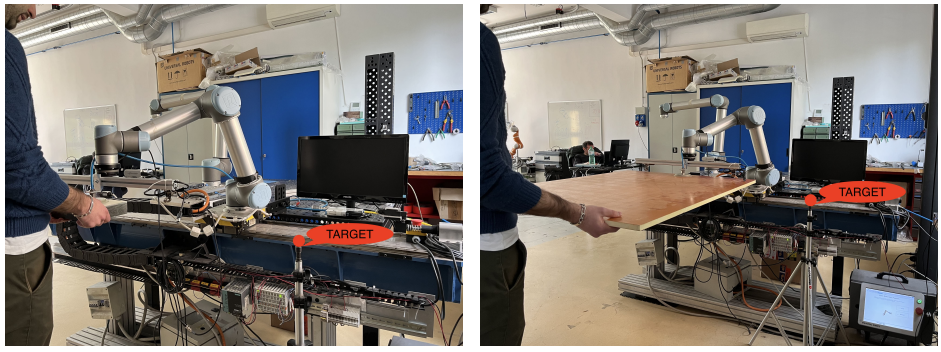
We selected some parameters to tune the robot's level of assistance, such as α and R_r. Changing them, also the assistance that the robot can provide will vary. Applying the TL to fine-tune the model on humans and robots could be interesting and reasonable.

6.2 GT-DMPC ASSISTANCE CONTROLLER

This section shows results relative to the implementation of the GT-dMPC in section 5.3, along with the human intention predictive model of section 4.3.

The proposed approach exploiting the RNN+FC predictive model into the GT-dMPC framework is evaluated with experiments involving two different co-manipulated objects for three sets of experiments each. Five subjects aged between 28 and 36 years old performed the experiments.

The experiments involve the full motion along the Cartesian positions (x–y–z). Three sets of experiments are designed, involving two different components. The first component is a granite brick weighing about 3.6 kg, which is usually used for calibration. The second component is a composite board (size 900x700 mm), an aeronautical component assembled in the cargo area of airplanes. The two objects and the collaborative setup are visible in figure 6.2.



(a) Mass setup **(b)** Panel setup

Figure 6.2: The two setups used to perform Transfer Learning and comparison with MG and IMP controllers.

The experiments are carried out as follows. First of all, the iterative training procedure is performed. The training is carried on in this

stage without any tool or co-manipulated object. The robot mounts a lightweight 3d-printed handle to allow the human to grasp it right after the FT sensor. The trajectories are computed with Moveit!, and the human is allowed to deform the nominal robot trajectory in the x–y–z Cartesian space along the translational degrees of freedom. The RNN+FC model is increased to have 375 hidden layers in the RNN, and it outputs the prediction of the human desired positions in the x–y–z coordinates. A total of five iterations are performed to train and adapt the model to learn the actual interaction. The iterative training procedure output is the model $\mathcal{M}_4$, which is trained to collaboratively reach a precise target T_0 without any additional inertia.

After that, the model $\mathcal{M}_4$ can be adapted via TL to co-manipulate different objects with different users. Each new subject is asked to reach the target point co-manipulating an object, using the GT-dMPC controller, and the model $\mathcal{M}_4$ predicts the desired motion. The target point is the same one used for the iterative training procedure T_0. Each subject, for each of the two objects, performs 15 reaching tasks to the target T_0. After the dataset collection, TL training is quickly performed, and the comparison of the GT-dMPC controller with the IMP and MG can be performed.

Three different reaching tasks are defined. The first requires the co-manipulation of the objects and the reaching of the same target used during the training T_0. Each subject performs such a reaching task five times with the three controllers. After that, a brief questionnaire is filled out. Then, a new target T_1 is defined. The subjects are asked to reach it in two different conditions. In the first case, the target T_1 is also known to the robot, defined in this case as $T_{1,k}$, and Moveit! computes a nominal trajectory from the home position to $T_{1,k}$. The second case still involves reaching the target T_1, but now it is unknown to the robot ($T_{1,u}$), and the trajectory computed connects the home position to the target pose T_0. After each set, the same questionnaire is filled out, and the subjects are

asked to score different performances for each controller.

The following performance indexes are defined to evaluate the performances of the proposed controller compared with the MG and the IMP controllers. Being the objective of the proposed work the design an assistive controller, We consider as performance indexes the required force to move the object, the precision of positioning such an object to a target position, and the smoothness of motion to measure the naturalness of the interaction. The force Root Mean Squared error is considered over the entire trajectory, defined as

$$f_{RMS} = \sqrt{\frac{1}{L}\sum_{k=0}^{L} \|f_k\|^2}, \tag{6.2}$$

with L denoting the length of the trajectory and $\|f_k\|$ the module of the measured interaction force at time instant k.

To measure the precision in reaching the target pose, we defined a tolerance δ. The distance from the target pose is defined as the Euclidean distance as $d(c,t) = \|x_c - x_t\|$, with subscripts $_{c,t}$ denoting the current and target pose, respectively. As a performance index, we compute the average $d(c,t)$ for two seconds after the first occurrence of $d(c,t) < \delta$. The precision performance index is defined as

$$d_{avg} = \frac{1}{n}\sum_{k=0}^{n} d(c,t)_k. \tag{6.3}$$

It is important to note that the average over a time horizon gives a measure of the precision of reaching a target and the stability with which such a target is reached over time.

To measure the smoothness of the interaction, the Spectral-Arc Length (SAL), as defined in [8].

Finally, we proposed a questionnaire to the subjects to evaluate sub-

jective performance indexes which give an intuition of the naturalness of the interaction. In the proposed questionnaire the subjects are asked to score five indexes:

- *assistance*: measures the overall assistance felt by the users provided by the different controllers;
- *naturalness*: measures how natural the interaction was perceived by the subjects in completing the tasks;
- *smoothness*: measures how smooth the interaction was;
- *effort*: measures the required force perceived by the subjects during the interaction;
- *detection of intention*: measures how well the robot correctly intended the subjects' motion intention during motion.

6.2.1 RESULTS

As an illustrative example, figure 6.3 shows the actually executed trajectory, the predicted portion of the trajectory each time instant, and the nominal robot trajectory, during the execution of the first reaching task, involving the lumped mass and the initial target pose T_0.

The results of the f_{RMS} index are presented in figure 6.4. In general, it is visible that the IMP controller is the one that requires the higher force to co-manipulate the objects in both cases. This is because the IMP controller has a virtual spring that always tries to restore the current pose to the nominal robot trajectory and applies a force opposite to the trajectory deformation imposed by the subject. The GT-dMPC and MG controllers show similar performances. In the case of lumped mass, the GT-dMPC slightly performs better, showing that the RNN+FC model has learned the interaction model even if a payload is applied.

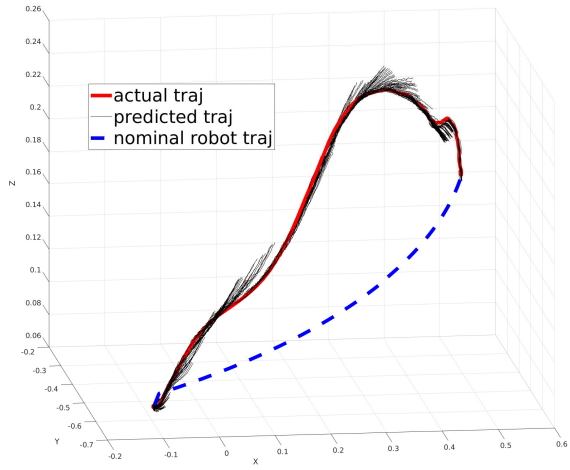
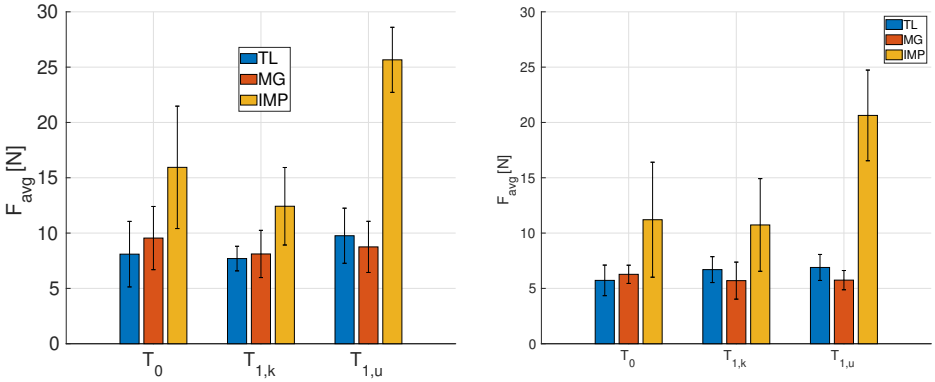


Figure 6.3: Predicted, executed, and nominal trajectories for the co-manipulation of a lumped mass object.

The results of the d_{avg} are shown in figure 6.5. In this case, the IMP controller performs well when the target pose is known to the controller, as the virtual spring applies a force to move the robot towards it. Despite this, the additional payload is not considered by the IMP controller. Therefore, humans must still apply vertical force to reach the target pose properly. Moreover, when the human wants to reach a target different from the nominal one, the IMP cannot provide any assistance. On the contrary, the IMP controller tends to reach its known target, deriving from the human one. The MG controller cannot provide any assistance to reach the target pose, and its performances strictly depend on each subject's capabilities. The GT-dMPC controller performs better than the two others in almost all three cases regarding the mass task.

The SAL index shows that the proposed GT-dMPC controller performs comparably to the other two controllers in most cases. In general, the MG controller allows smooth motions because it has a passive be-

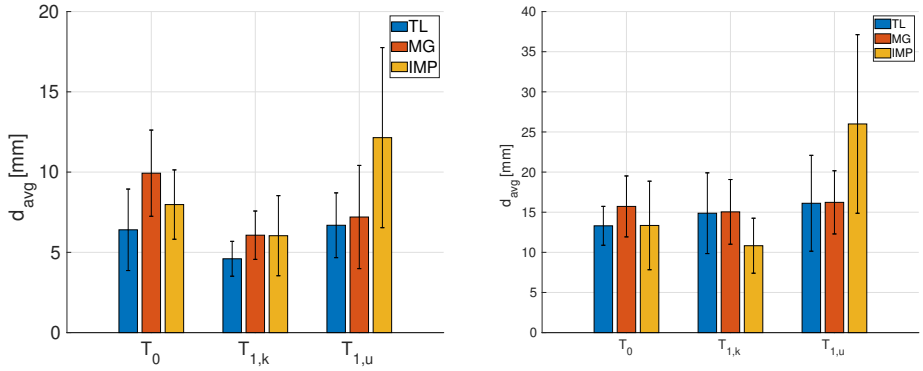


(a) average force mass **(b)** average force panel

Figure 6.4: Force RMS, measured as in (6.2), for the three tasks and the three controllers

havior, and the IMP controller allows smooth motion because it tends to follow the nominal trajectory, which is typically computed to avoid too high acceleration in pHRI tasks. It is interesting to see that the GT-dMPC with the active assistive contribution shows comparable performances and that the introduced assistance does not worsen the SAL index. Moreover, in the unknown reaching task, the IMP controller is incapable of any assistance, making it hard to deform the trajectory and precisely reach the target. In this case, the SAL is more negative (indicating more jerky trajectory execution), and the robot's behavior introduces oscillations. Results of the SAL computation are visible in figure 6.6.

Finally, the questionnaire results are presented in table 6.1. Note that the presented results are an aggregate of the singular scores given after each task to each controller. Therefore, the results presented are, in some cases, contradictory. This happens because, for example in the IMP case, the IMP controller is very assistive for the tasks involving the reaching of a known target but is not assistive at all if the target is unknown to



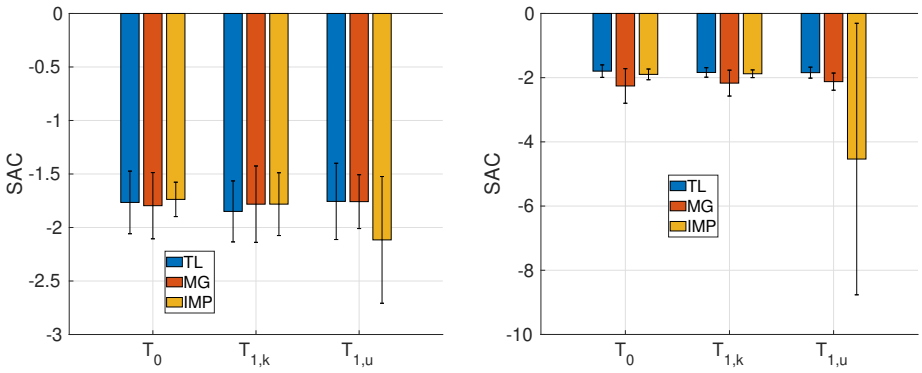
(a) average distance mass

(b) average distance panel

Figure 6.5: Average distances, measured as (6.3) for the three tasks and the three controllers

the controller. Therefore the same subject can perceive high assistance in some cases and low in others. We present an aggregate of the results to give an overall performance of the controllers, and to evaluate also their flexibility.

The GT-dMPC controller is the one that shows higher assistance, with most of the scores ranging between 3 and 4. The MG controller also shows good results, with about one-third of the scores equal to 3. This can be interpreted as the humans like a slightly damped interaction as this can sometimes help in being precise and damps the load descent, letting the human perceive a little assistance in terms of "gravity compensation". The IMP controller shows various and different scores. This happens because it can be quite assistive in cases where the target pose is known. In such cases, the IMP controller helps with precise positioning by applying an attractive virtual force. For the same reason, it applies an attractive virtual force in the vertical direction to the nominal vertical target, partially relieving the human operator and giving the impression of sustaining the weight.



(a) SAL mass **(b)** SAL panel

Figure 6.6: average SAL for the three tasks and the three controllers

Table 6.1: Subjective questionnaire showing the percentage of responses count.

Score %		0	1	2	3	4
Assistance	GT-dMPC	0	6.67	20.0	40.0	33.3
	MG	13.3	13.3	13.3	60.0	0
	IMP	13.3	26.7	13.3	20.0	26.7
Naturalness	GT-dMPC	0	0	33.3	20	46.7
	MG	0	13.3	13.3	66.7	6.67
	IMP	13.3	26.7	53.3	6.67	0
Smoothness	GT-dMPC	0	0	6.67	66.7	26.7
	MG	6.67	13.3	26.7	53.3	0
	IMP	0	40.0	20.0	40.0	0
Effort	GT-dMPC	0	46.7	46.7	6.67	0
	MG	0	13.3	60.0	26.7	0
	IMP	6.67	33.3	0	26.7	33.3
DoI	GT-dMPC	0	0	26.7	26.7	46.7
	MG	0	6.67	33.3	46.7	13.3
	IMP	20.0	13.3	53.3	6.67	6.67

Naturalness evaluates how natural is perceived the interaction by the user. The GT-dMPC controller shows great performance, with the majority of the scores in 4, indicating that the proposed controller allows natural pHRI. Also the MG controller shows good performance, as its passivity guarantees a natural behavior of the robot. Despite this, it is perceived as a little less natural since it does not provide any support and the humans felt to be the only ones in charge of the task. Compared with the other two, the IMP controller does not show significant results. This is mainly because it has a reference trajectory that tends to follow, introducing forces that sometimes are not following the desired motion of the human.

Regarding smoothness, the GT-dMPC performs slightly better than the MG controller, with great results for both. In this case, the IMP controller is also evaluated with good scores, only slightly lower than the other two controllers. This can still be because the IMP controller, reacting to external forces that the additional inertia can produce, might oscillate a little, even in proximity to the target position.

The effort is measured as the lower, the better since it was asked to score how much effort each subject should put into completing each task. In this case, the GT-dMPC scores are in the lower half, showing that the proposed controller is capable of relieving the users from carrying the weight and also allows trajectory deformation according to the desired user intention without requiring excessive interaction force. The MG controller performances are in the upper half of the range. This is because it does not require any particular force to deviate from the nominal trajectory. Still, it cannot assist in reducing the perceived weight of the co-transported object. The IMP controller has different scores, either they are high or low. This happens because in some tasks when the target is known, the IMP controller assists in quickly reaching the target and lifting the weight, and applying a reactive force in the vertical direction. On

the contrary, when the user wants to deviate from the nominal trajectory, as in the case of an unknown target, the user should apply a higher force to counteract the attractive virtual force to the nominal trajectory.

Finally, the perceived Detection of Intention is evaluated. The GT-dMPC is the controller that better gives the impression of detecting where the human wants to go and provides proper assistance. The MG controller also shows good results because it smoothly allows motions according to human desire. The IMP scores are slightly different, as in the case the target is known, the human can think that the robot properly understands the intention of reaching a desired point, but when the target it is not known it does not.

6.3 ADAPTIVE IMPEDANCE CONTROLLER FOR HUMAN-ROBOT ROLE ARBITRATION BASED ON COOPERATIVE DIFFERENTIAL GAME THEORY

This section presents a controller that allows role arbitration between humans and robots. It implements the cooperative controller as presented in 5.2. The role arbitration is mitigated by the interaction force that the human imposes on the robot tip. The resulting behavior is a robot capable of completing a task autonomously but also allowing the human to modify the trajectory easily.

6.3.1 PRELIMINARIES

Some preliminary consideration must be made to understand the elements that compose this controller properly.

6.3.1.1 BARGAINING

In a cooperative environment, it is rational to consider the set of Pareto solutions. Since there are infinite Pareto-optimal solutions, we enter the

bargaining theory arena to decide which is the most effective one. The bargaining problem is how to define the appropriate α. In classical bargaining solutions (Nash, Kalay-Smorodinsky, egalitarian), α is defined such that the cost of all players decreases compared to their non-cooperative cost.

In this work, the two players bargain on who leads and follows the task; in this sense, one can accept a higher cost if he is the follower. Hence, α is used as a weighting factor to move the control authority from the robot to the human and vice-versa. Indeed, for high values of α, the robot cost tends to disappear from the overall cost computation J_α. Hence it will be less costly for the robot to put much effort into the system, becoming the leader, resulting in a higher cost for the human. On the contrary, for low values of α, the robot cost increases, and its control input will be dramatically reduced, leading the control authority to the human. Each situation in between represents a cooperative solution where the control authority is shared appropriately.

The selection of the weight parameter α depends on the force applied by the human and is processed by the sigmoid function:

$$\alpha = d - \frac{a}{1 + e^{-b(\|u_h\| - c)}} \tag{6.4}$$

where the constant parameters a, b, c, d are used to shape the function properly. In particular, a defines the height of the function, b the width of the transition phase, c represents an offset in the x direction, and d is an offset in the y direction, moreover the negative sign after d means that α is decreasing as $\|u_h\|$ increases.

6.3.1.2 TRAJECTORY TRACKING

For trajectory tracking purposes, it is sometimes useful to divide the control input into a feedforward and a feedback part.

Starting from the solution of the ARE in (4.19), the feedback gain matrix is defined as

$$K^{fb} = R_{\alpha}^{-1} B^T P \tag{6.5}$$

and the feedback control actions are given by

$$u^{fb} = -K^{fb} z(t) \tag{6.6}$$

From (6.6) the feedback control terms demanded from the human and the robot can be computed as

$$u^{fb} = \begin{bmatrix} \bar{u}_h^{fb} \\ u_r^{fb} \end{bmatrix} \tag{6.7}$$

where $\bar{u}^{fb}$ denotes the nominal effort demanded to the human. A feedforward term is added to allow trajectory tracking.

The feedforward gain matrix is defined as

$$K^{ff} = [K^{fb}\ I] \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{6.8}$$

with A and B as in (5.16), and C and D output and feedthrough matrices of the state-space system description, respectively. The feedforward term results in

$$u^{ff} = K^{ff} z_{ref}. \tag{6.9}$$

Since the system matrix is typically not square in this formulation, the pseudoinverse can be used instead of the inverse. This results in a vector of dimension 12, and its components are

$$u^{ff} = \begin{bmatrix} \bar{u}_h^{ff} \\ u_r^{ff} \end{bmatrix} \tag{6.10}$$

with the superscripts h and r denoting the human and robot contributions.

Finally, the total control input results in $u = u^{ff} + u^{fb}$, and the control action of the robot can be computed as

$$u_r = u_r^{ff} + u_r^{fb} \tag{6.11}$$

6.3.1.3 THE CONTROL LAW AS A VARIABLE IMPEDANCE

Looking at (6.7), can be divided into two components, and the control inputs demanded from the human and the robot are computed as

$$\begin{bmatrix} \bar{u}_h^{fb} \\ u_r^{fb} \end{bmatrix} = - \begin{bmatrix} K_h^{fb} \\ K_r^{fb} \end{bmatrix} z(t) \tag{6.12}$$

where $K_h \in \mathbb{R}^{6\times m}$ and $K_r \in \mathbb{R}^{6\times m}$ are the components of the matrix K defined in (6.5). Looking at the robot control input, the feedback part can be rewritten as

$$u_r^{fb} = -K_r^{fb} \begin{bmatrix} x - x_0 \\ \dot{x} \end{bmatrix} \tag{6.13}$$

Given $K_r^{fb} = [K_{r,k}^{fb}\ K_{r,d}^{fb}]$, the two components associated with the stiffness and damping of the variable impedance cooperative system are defined as

$$K_{imp}^{fb} = K_{r,k}^{fb} \tag{6.14}$$

and

$$D_{imp}^{fb} = K_{r,d}^{fb} \tag{6.15}$$

A similar computation can be done for the feedforward terms, resulting in

$$u_r^{ff} = K_r^{ff} x_{ref} \tag{6.16}$$

where K_r^{ff} represents the components relative to the robot of the matrix (6.8) and

$$K_{imp}^{ff} = K_r^{ff} \tag{6.17}$$

The robot control input results in

$$u_r = -D_{imp}^{fb}\dot{x} - K_{imp}^{fb}(x - x_0) + K_{imp}^{ff}x_{ref} \tag{6.18}$$

Substituting (6.18) into (3.7) results in

$$\begin{aligned} M_i\,\ddot{x}(t) + (D_i + D_{imp}^{fb})\dot{x}(t) + \\ +(K_i + K_{imp}^{fb})\,(x(t) - x_0(t)) - K_{imp}^{ff}\,x_{ref}(t) = u_h(t) \end{aligned} \tag{6.19}$$

Because varying α varies matrix P and matrix K accordingly, (6.19) represents a variable impedance system subject to the human force, with the values of D_{imp} and K_{imp} updated according to the human will to lead or follow, detected by force applied.

6.3.2 EXPERIMENTS

An experiment is designed to test the proposed control method for sharing control authority in human-robot collaboration. The robot has to follow a planar circular trajectory, while the human has a different path to follow, which partially overlaps with the robot one. The nominal robot trajectory is defined as

$$x_{ref,r}(t) = \begin{bmatrix} -\rho\,sin(\omega\,t) \\ -\rho\,cos(\omega\,t) \end{bmatrix} \tag{6.20}$$

where $\rho = 0.2$ is the radius of the circumference and ω is the angular velocity.

The human desired trajectory is piecewise-defined as

$$x_{ref,h}(t) = \begin{cases} x_{ref,r}(t), & t_0 < t < t_1 \\ \begin{bmatrix} -\rho\, sin(\omega\, t) \\ x_1 + \frac{x_2 - x_1}{t_2 - t_1}(t - t_0) \end{bmatrix}, & t_1 < t < t_2 \\ \begin{bmatrix} -\rho\, sin(\omega\, t) \\ x_2 + \frac{x_3 - x_2}{t_3 - t_2}(t - t_1) \end{bmatrix}, & t_2 < t < t_3 \\ x_{ref,r}(t), & t_3 < t \end{cases} \tag{6.21}$$

with $t_0 = 0s$, $t_1 = 3s$, $t_2 = 5s$, $t_3 = 7s$.

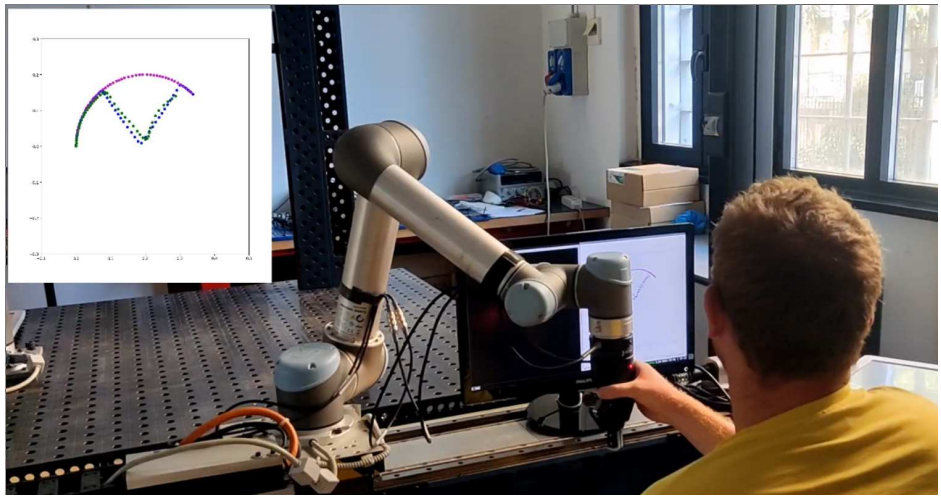


Figure 6.7: The experimental setup, showing the application and the monitor displaying the three trajectories: magenta - robot desired trajectory, blue - human desired trajectory, green - actual trajectory.

The nominal robot trajectory, the desired human trajectory, and the current position are shown in real-time on a screen, and the human has the goal to make the current trajectory as close as possible to the desired human trajectory. Figure 6.7 shows the experimental setup.

In Figure 6.8, the nominal trajectory of the robot (dashed red line), the desired trajectory of the human (dashed yellow line), and the actual robot end-effector positions (solid blue line) are shown.

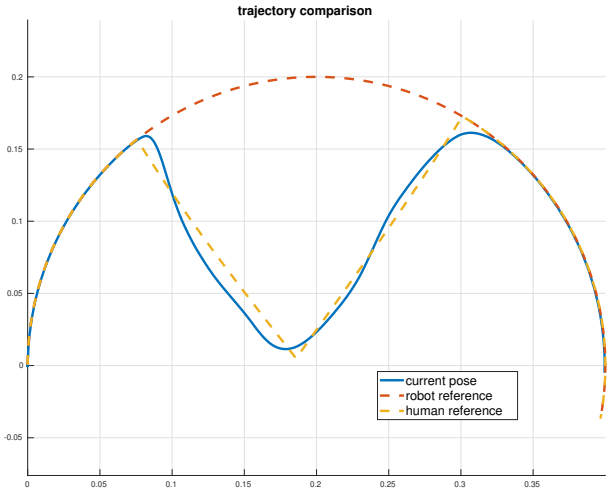


Figure 6.8: Comparison between the nominal trajectory of the robot, the nominal trajectory of the human, and the measured

To test and compare the proposed approach, the following indices are computed:

- Trajectory following error, measured as

$$\mathcal{E}_{track} = \int_{T_{start}}^{T_{end}} \left\| x_{ref,h}(t) - x(t) \right\| dt \tag{6.22}$$

  As low the $\mathcal{E}_{track}$ is, as close the actual trajectory is to the nominal one

- Geometrical following error, measured as

$$\mathcal{E}_{Xcorr} = \int_{T_{start}}^{T_{end}} \left\| x_{ref,h}(t + \delta t) - x(t) \right\| \, dt \tag{6.23}$$

  where δt represents the time delay of the actual trajectory with respect to the nominal one, computed by applying cross-correlation between the nominal trajectory of the human and the measured one. In this way, it is possible to compare the capability of the human to track the geometrical reference. As low the $\mathcal{E}_{Xcorr}$ is, as close the actual trajectory is to the geometrical one

- Measured interaction force, measured as

$$\mathcal{F} = \int_{T_{start}}^{T_{end}} \| f(t) \| \, dt \tag{6.24}$$

  As low the $\mathcal{F}$ is, as less effort the human has to put in the cooperative task

- Mechanical work, measured as

$$\mathcal{W} = \int_{T_{start}}^{T_{end}} \vec{f}(t) \cdot d\vec{S} \, dt \tag{6.25}$$

  The lower the $\mathcal{W}$ is, the less energy the human has to put into the cooperative task

The goal of the proposed controller is to allow smooth interaction and leader-follower transition with the human, reducing the effort required and allowing good trajectory tracking. Hence the goal is to minimize the indexes defined above together.

As a comparison, three different types of impedance controllers are used. The CGT controller is compared with an LQR controller, which can be seen as the CGT with $\alpha = 1$ always, and two classical impedance

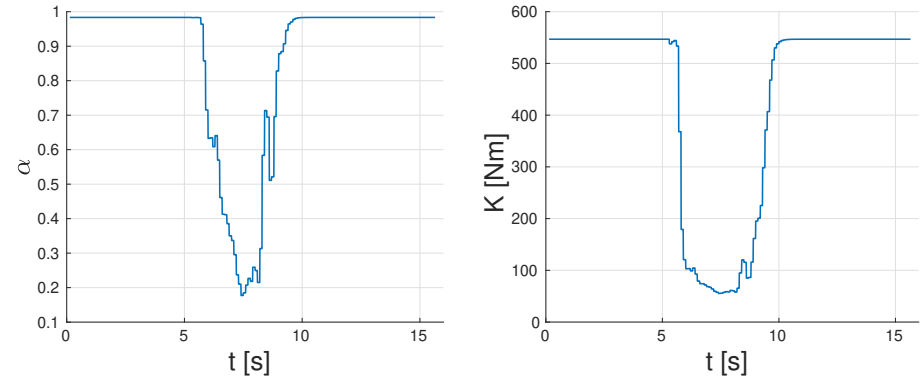
controllers with different values of stiffness K. One has high stiffness (HIC), and the other has a low stiffness (LIC) value. For the two classical impedance controllers (HIC, LIC), the value of the damping to achieve the critical damping $D = 2D_{cr}\sqrt{KM}$. For the LQR and CGT controllers, the values of damping and stiffness depend on the matrices $\hat{Q}_h$, $\hat{R}_h$, Q_r and R_r, defined as $\hat{Q}_h = diag(50, 50, 1, 1)$, $\hat{R}_h = diag(10, 10, 0, 0)$, $Q_r = diag(20, 20, 1, 1)$ and $R_r = diag(1, 1, 0.01, 0.01)$ for the CGT case, $R_r = diag(1, 1, 0.1, 0.1)$ for the LQR case, otherwise the stiffness was too high and too much force was required to move the robot barely. The $\hat{Q}_h$ and $\hat{R}_h$ used are average values. The mass matrix for all the cases is defined as $M = diag(10, 10)$, and the base damping and stiffness for the CGT and LQR cases are $K = diag(0, 0)$ and $D = diag(40, 40)$. The mass-spring-damper parameters used or computed for the experiments are presented in Table 6.2, while Figure 6.9 shows the changing parameters during the task. For the CGT case, the value of α is bounded such that $0.01 \leq \alpha \leq 0.99$, and the sigmoid parameters are chosen to be $a = 0.98$, $b = 0.7$, $c = 7$, $d = 0.99$.

Table 6.2: The mass, spring, and damping parameters used for the experiments.

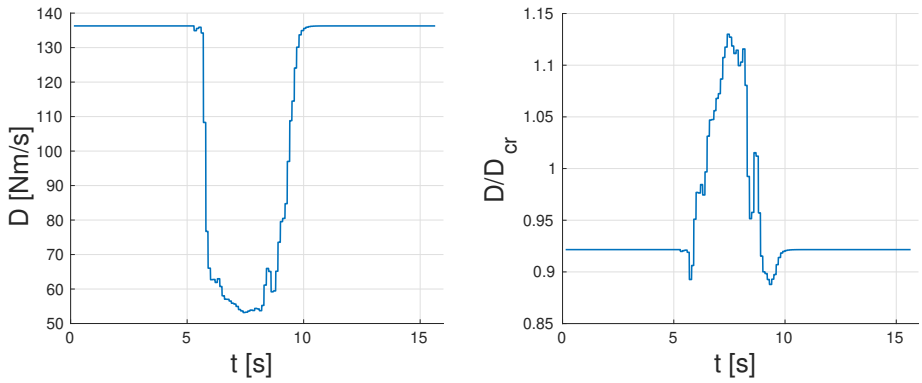
	CGT	HIC	LIC	LQR
K	$55 \div 550$	100	20	222
D	$53 \div 135$	57	25.5	84
D/D_{cr}	$1.13 \div 0.92$	0.9	0.9	0.89
M	10	10	10	10

6.3.3 RESULTS

In the test campaign, five subjects (age 30 ± 1) are asked to perform the task five times for each controller. Before starting the test, the subjects are allowed to practice as long as needed to feel confident with the current



(a) Arbitration parameter α **(b)** Variable stiffness

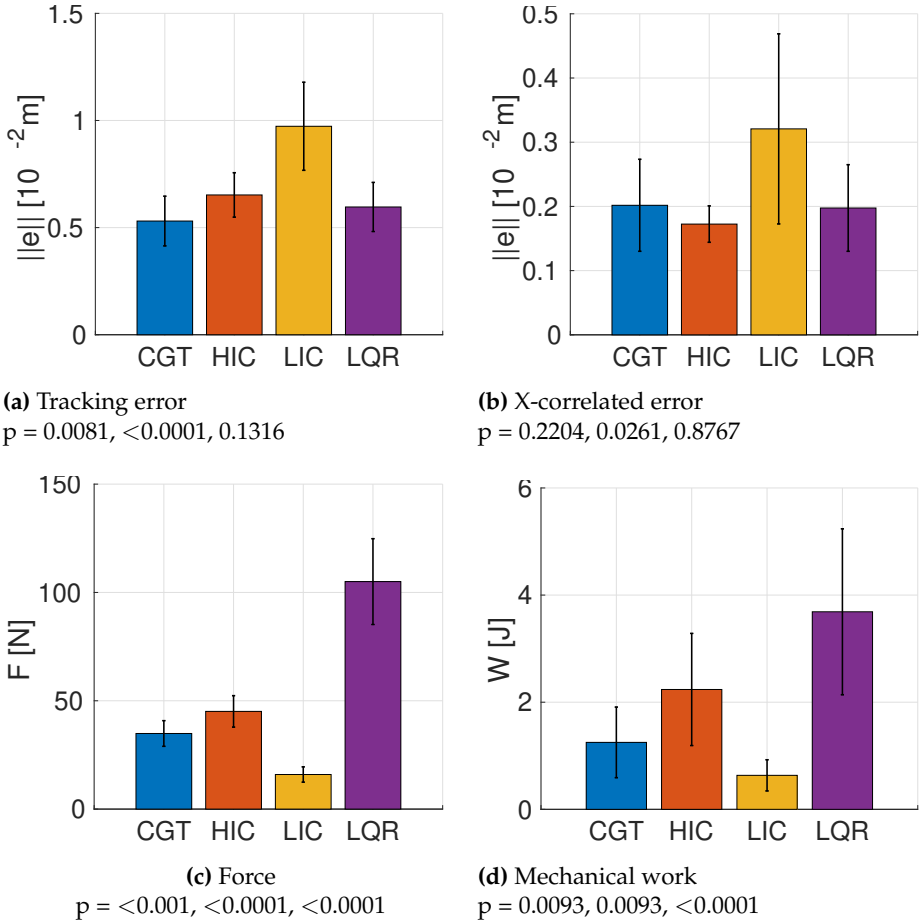


(c) Variable damping **(d)** Variable critical damping

Figure 6.9: Variable parameters for the CGT

controller, then five trials in a row with the same controller are recorded. This procedure is used for each of the four controllers, selected randomly for each subject.

A UR5 robot is used, controlled by joint velocities. (3.8) is used to compute the Cartesian reference positions, and (3.11) provides the robot joint velocity tracking controller with the reference values. A Robotiq FT 300, mounted on the robot tip, is used to measure the force applied by



(a) Tracking error
p = 0.0081, <0.0001, 0.1316

(b) X-correlated error
p = 0.2204, 0.0261, 0.8767

(c) Force
p = <0.001, <0.0001, <0.0001

(d) Mechanical work
p = 0.0093, 0.0093, <0.0001

Figure 6.10: Indexes comparison

the human.

A t-test compares the CGT controller in pairs with all the others for each computed index. The t-test allows for defining the statistical relevance of the computed values. The results are in Figure 6.10, with the mean and standard deviations and the relative p-values.

The tracking error, computed as in (6.22), is shown in Figure 6.10a.

The CGT controller shows the best performance, and the t-test assesses the statistical relevance of the measured values. This result can be explained by the robot's high stiffness/low damping ratio while the robot is leading and the low stiffness/high damping ratio while the human is leading. On the one hand, the high stiffness/low damping ratio allows fast and reliable trajectory tracking and fast recovery after a trajectory modification (*i.e.* the robot is faster in getting back to its nominal trajectory after the human leaves control). On the other hand, a low stiffness/high damping ratio allows the human to move the robot smoothly and precisely while leading. Compared to the LIC, constant high stiffness values allow better trajectory tracking in HIC and LQR.

The geometrical error, computed as in (6.23), is shown in 6.10b. The t-test shows that no significant differences between the four controllers can be appreciated. This shows that the tasks are executed correctly, and comparing the controllers for the computed indexes is fair. Indeed, this shows that similar results can be obtained in terms of the geometrical path following with all the controllers.

The force evaluation is presented in 6.10c. Since the exchanged force mainly depends on the robot stiffness (measured as $f = K(x_{ref,r} - x)$), the lower the stiffness is, the lower the exchanged force. This effect results in the LIC with the best performance concerning the required force. Despite this, the CGT controller shows better performance with respect to LQR and HIC because when interaction happens, its stiffness lowers. Notably, performances similar to the LIC can be obtained by a different tuning of the values of matrices Q_r and R_r.

Finally, similar considerations can be derived for the mechanical work shown in Figure 6.10d. Indeed, similar behavior is observed concerning the force results depending on the interaction force.

6.3.4 DISCUSSION

The results show that being equal to the capability of the four controllers in following the nominal geometrical path under human-robot cooperation, the CGT controller outperforms the others in trajectory following. Moreover, even if the LIC requires less interaction force to accomplish the task, it shows the lowest capability in trajectory tracking, and the CGT also requires low interaction forces. As a remark, a similar interaction force can be obtained for the CGT controller by tuning the parameters differently.

Interestingly, Table 6.2 and Figure 6.9d show that the critical damping value varies from lower than critical to higher than critical, according to the role adaptation. This effect allows the robot to have a more damped behavior when the human is leading, allowing smooth motion, while a low damping ratio allows better and faster trajectory tracking when the robot is leading.

Some limitations of the proposed control can be identified. First, the human control objective (*i.e.* the human cost function parameters) can only be estimated offline with some inverse techniques. Future works will address online estimation techniques for the human control objective estimation to adapt the robot behavior accordingly. Indeed, it is reasonable to suppose that the human may change his cost function parameters due to fatigue by repeating a task all day long. Secondly, many parameters have to be defined arbitrarily. In particular, predicting the variable damping and stiffness values is difficult. Hence, defining some tuning rules according to the required performances may be interesting. Moreover, the arbitration law introduced in this work is a simple function of the force, but it can be improved and modified according to other sensing capabilities and different human intention estimations. Future work will address a more complex and improved arbitration law involving visual feedback, human intention estimation, and force feedback. In such a way,

with a more complex sensor fusion, a more precise estimate of the human will to lead or follow can be obtained. Some future applications of this framework will involve the co-manipulation and the co-transportation of heavy objects, as well as lightweight parts and deformable objects as composite materials plies. Another interesting application can be for rehabilitation purposes, where the robot behavior is adjusted by tuning the parameter α to make the robot contribution high when the patient can barely perform a task and gradually lower the robot contribution as the patient starts to recover some motion capabilities.

6.4 ROLE ARBITRATION WITH DIFFERENTIAL GAME THEORY

This section presents a controller that fully exploits the two interactive models presented in 5.2 to allow efficient Role Arbitration. The proposed controller allows switching from the Cooperative to the Non-cooperative model, to change the robot's behavior from follower assistant to leader, according to the situation.

6.4.1 PRELIMINARIES

Some additional considerations are required for proper controller implementation.

6.4.1.1 HUMAN REFERENCE TRAJECTORY ESTIMATION

The definition of a method for detecting and predicting the desired human trajectory is out of the scope of this work. Moreover, many accurate techniques exist in the literature, as proposed in [21, 85, 90], so the choice of the best human trajectory identification techniques is left to the reader. Despite this, since it is a piece of essential information to apply the control scheme proposed, it is necessary to identify a method for the

prediction. In the proposed approach, we decided to implement an easy yet powerful method based on the direction of the force interaction. The human reference state is composed of position and velocity vectors as $z_{ref,h} = [x_{ref,h}^T \ \dot{x}_{ref,h}^T]^T$. The velocity component has always had a minor and typically negligible influence on human behavior, as they are proven to care only about the position [41, 57]. Therefore it can be set to zero without any loss of generality, as $\dot{x}_{ref,h} = 0^{6\times1}$. This is a typical choice made by various works addressing human-machine interaction [58, 120, 122]. On the contrary, the pose is updated at each cycle by the following:

$$x_{ref,h}^{+} = x_{ref,h}^{-} + K_{p,h}\, u_h \tag{6.26}$$

with $(^+)$ and $(^-)$ referring to the updated and previous poses, respectively, and $K_{p,h}$ defines a coefficient proportional to the human exerted force.

6.4.1.2 ROLE ARBITRATION LAW

As already said, the solution of (5.17) strictly depends on the choice of α. In the Cooperative Differential Game Theory, the *Bargaining Problem* refers to the problem of choosing the best appropriate α, and different solutions are available in the literature (Nash bargaining solution [123], Kalai-Smorodinsky [69], egalitarian [68]). These methods aim at identifying the best compromise between players so that everyone has the incentive to cooperate rather than compete against each other. The solution found is static, and the game proceeds by minimizing (5.17).

In the proposed control schema, the value of α is changed dynamically on the fly according to some defined law, allowing the Role arbitration between the human and the robot. Indeed, this work aims to make the robot assistive, capable of following human intentions, assisting the human, and taking control over the human to avoid undesired situations.

Four main undesired situations are identified:

a) singularities

b) proximity to objects

c) proximity to workspace limits

d) distance to the reference target position

Remark 6

Proximity to joint limits is not directly considered, as in the computation of the manipulability index, it is already included, as described below.

To avoid configurations close to singularities, the manipulability index μ is taken into account, defined as in [39]:

$$\mu = P\sqrt{det(\mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T)}, \tag{6.27}$$

with penalty factor P, similar to the one introduced in [168] used to scale the manipulability to account for joint limits. P are computed separately for each joint j, as

$$P_j = 1 - e^{-k\frac{(q_j - q_j^{lb})(q_j^{ub} - q_j)}{(q_j^{ub} - q_j^{lb})^2}}, \tag{6.28}$$

where k is a scaling factor that can be used to adjust the behavior near joint limits, q_j is the current position of the joint j, and q_j^{ub} and q_j^{lb} are the upper and lower bounds, respectively.

The proximity to objects d_o is measured as the minimum distance between any robot link and any object in the environment, possibly excluding objects that must be manipulated.

$$d_o = \min(robot - objects) \tag{6.29}$$

The proximity to workspace limits d_{ws} represents the Euclidean distance between the current Cartesian position of the robot TCP and the boundaries specified according to the application.

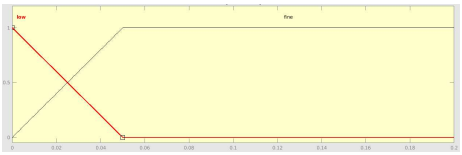
$$d_{ws} = \left\| x^{ee} - x^{ws} \right\| \tag{6.30}$$

with x^{ee} and x^{ws} denoting the Cartesian positions of the end-effector and the workspace boundaries, respectively.

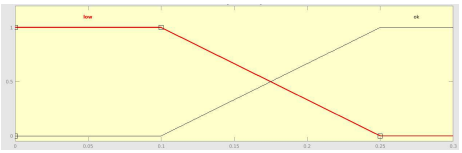
The distance to the reference target position d_{trg} is measured as the Euclidean distance between the current end-effector position and the target pose of the task (*e.g.*, pick/place pose).

$$d_{trg} = \left\| x^{ee} - x^{trg} \right\| \tag{6.31}$$

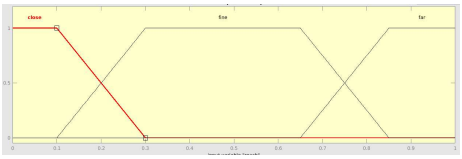
with x^{trg} denoting the target position of the end-effector.



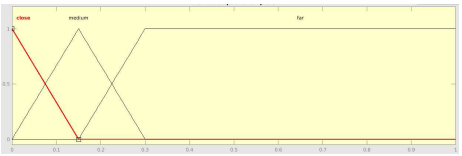
(a) manipulability membership function



(b) distance to collisions membership function



(c) distance to workspace boundaries membership function



(d) distance to target pose membership function

Figure 6.11: Input membership functions to the Fuzzy Logic System

These four indices allow defining the proper α, allowing for Role Arbitration. It is defined as a Fuzzy Logic System (FL) that accepts as inputs

the four indices and returns the value of α. In figure 6.11, the four membership functions related to the four indices are visible. The following rules are defined:

$$\begin{cases} \text{if } (\mu \text{ is } low) \text{ or } (d_o \text{ is } low) \text{ or } (d_{ws} \text{ is } close) \text{ or } (d_{trg} \text{ is } close) \text{ then } \alpha \text{ is } low \\ \text{if } (\mu \text{ is } low) \text{ or } (d_o \text{ is } low) \text{ or } (d_{ws} \text{ is } far) \text{ or } (d_{trg} \text{ is } close) \text{ then } \alpha \text{ is } low \\ \text{if } (\mu \text{ is } fine) \text{ or } (d_o \text{ is } ok) \text{ or } (d_{ws} \text{ is } fine) \text{ or } (d_{trg} \text{ is } far) \text{ then } \alpha \text{ is } high \\ \text{if } (\mu \text{ is } fine) \text{ or } (d_o \text{ is } ok) \text{ or } (d_{ws} \text{ is } fine) \text{ or } (d_{trg} \text{ is } medium) \text{ then } \alpha \text{ is } shared \end{cases} \tag{6.32}$$

Processing the four membership functions through the above rules allows defining the proper α, with its membership function visible in figure 6.12. Identifying a threshold value of $\alpha = \alpha_{th}$ makes switching from the Cooperative to the non-cooperative case possible, allowing Role Arbitration. When α is high, the cooperative interaction model is selected, and the human can fully control the task and move the robot freely, enhanced by the robot assistance u_r. Conversely, when α is low, the non-cooperative interaction model is selected, and the robot takes control over the human by applying a virtual force u_r to recover the original safe trajectory. The block diagram of the arbitration control is in figure 6.13.

Remark 7

By varying the parameter α, also the matrices Q_h, Q_r, R_h and R_r vary. In the Cooperative case, how they vary is straightforward, and it is described by (5.11) and (5.12). In this case, the parameter α also represents the solution to the Bargaining problem. In the Non-Cooperative case, the matrices are computed as (5.13) and (5.14). The value of the matrices R_h and R_r are kept constant.

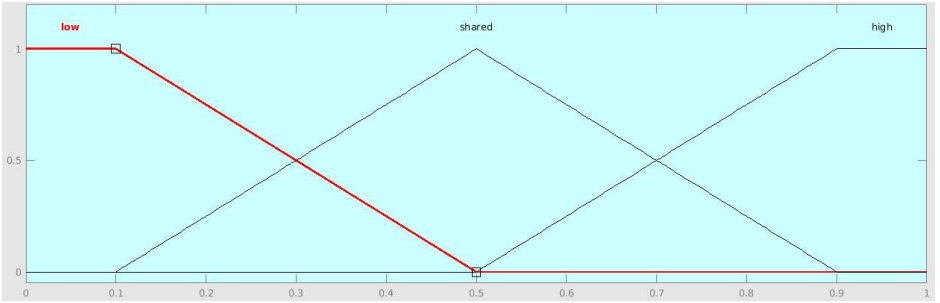


Figure 6.12: output membership function of the FLS that defines α.

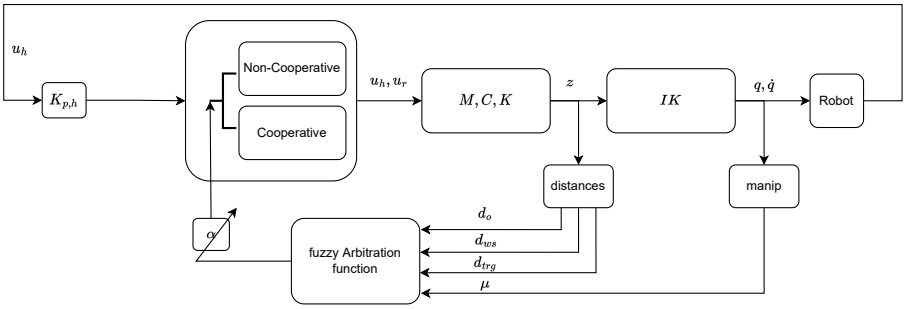


Figure 6.13: The block diagram of the arbitration mechanism. The variable α allows switching between cooperative and non-cooperative models.

6.4.1.3 SAFETY CONSIDERATIONS

The method proposed does not directly addresses the safety issues that arise when pHRI is considered. Therefore, some considerations are given. According to the standard rules defined by the ISO in [59–61], safety depends on and is evaluated considering the entire application and not the single modules such as the robotic platform, its control, and the other modules alone.

According to the standards, the robot at the end-effector should not move faster than 250 mm/s to certify the application. It should also stop if the external forces and power exceed a threshold according to the so-called Power Force Limitation (PFL).

Despite being hard to make sure that the application is safe according to the standards, the proposed FL arbitration law can handle such limitations within certain limits. Indeed, by defining some additional rules, the FL arbitration can prevent the robot from moving too fast or colliding unsafely with the human. By preventing such events, robot safety stops triggered by modules and sensors certified according to the standards can be reduced.

For example, the proposed fuzzy logic arbitration can handle the speed limitation. Indeed, the nominal trajectory is computed to comply with the limited speed constraint. The FL can include monitoring the speed, and the role of the leader can be assigned to the robot as close as the end-effector speed approaches the limit if the human tries to move faster.

PFL requires that, in the case of a collision with a human, the power and force are granted to be below a certain threshold. Such a threshold depends on the type of collision (free body or pinch) and the body area where it might happen. To avoid direct collisions with the human body, a camera system can track the human body position, the human body can be modeled as an environmental obstacle, and (6.29) allows for taking this distance into account. Moreover, a rule that monitors the robot's

position to the human body parts can also be added. For example, if it comes too close to the human head, the FL arbitration should allow the robot to lead the action far from it. It can be handled by measuring the height of the robot end-effector or directly by using a skeleton tracker if available. It is also possible to add danger zones as described in [83] and include them in the FL arbitration law. Of course, this approach should not consider the human arm that is directly in contact with the robot (if no co-manipulated object is considered). Otherwise, the distance between humans and robots will always be zero.

The presented cases do not guarantee the certification of the collaborative application, which requires the safety stops of the robot. On the contrary, they can prevent the robot from incurring such situations, allowing it also to reduce safety stops.

6.4.2 EXPERIMENTS

Three sets of experiments are designed to test the method's validity. In all scenarios, the human and the robot must reach a target position and return to the starting point. In the first case, a collision object is added to the scene, but the human does not know its presence. In the second scenario, the human knows the presence of an object, while the robot doesn't. The third case does not involve collision objects but requires reaching an intermediate target point that the robot does not know. The test cases are designed to simulate these three real-world scenarios:

a) the human and the robot are co-manipulating an object, possibly significant, that may impede the human's view of some environmental feature. The robot knows the obstacle's presence and helps the human avoid it. Another situation can happen when the obstacle is behind the human, and he does not see it;

b) a dynamical collision object is placed in the middle of the robot's

trajectory, and the human sees it and drives the robot far from possible collisions;

c) a task is mainly repeatable. Still, some exceptions can sometimes require an additional operation (e.g., during object sorting, most objects must be placed in a container, but some objects must be redirected for some additional operation due to poor quality).

The proposed approach is compared with two other controller methodologies typically used to execute such tasks, standard manual guidance (MG) and a standard impedance control (IMP). The first controller (MG) represents a standard when an operator guides the robot, and it acts as a follower. Typical applications are learning-from-demonstration or robotic assistance for load reduction. The second controller (IMP) is typical in pHRI as it allows compliancy and a robot that modifies its trajectory according to external forces.

Remark 8

The same control can obtain the three controller behavior by switching off the robot contribution for IMP (*i.e.*, $u_r = 0$ in (3.7)), switching off the robot contribution, and setting null the robot stiffness for MG (*i.e.*, $u_r = 0$, and $K_i = 0$ in (3.7)).

Remark 9

The trajectories are pre-computed using Moveit! and then used for all the cases to make the experiments fully comparable.

6.4.2.1 EVALUATION CRITERIA

The following indexes are defined and evaluated to compare the three controllers. The interaction force is evaluated as a measure of the robot's assistance to the human. As less the force is, as better the robot assists the human. The interaction force is measured as

$$\mathcal{F} = \int_{T_{start}}^{T_{end}} \frac{\|f(t)\|}{T} \, dt \tag{6.33}$$

it is normalized over T in which a force greater than a threshold (1N) is measured.

An index measures the percentage of points in the measured trajectory in which the robot is closer than 1mm to the obstacle to measure the obstacle avoidance capability. The index is defined as

$$\mathcal{C}^{\%} = \frac{n_c}{n_p} 100 \tag{6.34}$$

with n_c and n_p indicating the number of points in a collision (*i.e.* the robot is closer than 1mm to the obstacle) and the total number of points in the trajectory, respectively.

An index measures the percentage of points in the measured trajectory in which the robot's end-effector violates the safe workspace boundaries (in this work, it is set to 850mm as from the UR5 datasheet) to measure the capability in avoiding the violation of the workspace boundaries. The index is defined as

$$\mathcal{WS}^{\%} = \frac{n_{ws}}{n_p} 100 \tag{6.35}$$

with n_{ws} and n_p indicating the number of points that violates workspace boundaries (*i.e.* the robot end-effector is far more than 850mm from the robot base) and the total number of points in the trajectory, respectively.

Finally, since the proposed control also allows precise robot positioning at target points (e.g., pick/place target poses), an index evaluates the capability to precisely reach a target pose, defined as (6.31), measured when the action is considered concluded.

$$d_{trg} = \left\| x^{ee} - x^{trg} \right\| \tag{6.36}$$

This same index is used to measure the capability of reaching target points known to the robot and the additional via point in case (c), which is unknown to the robot.

6.4.2.2 EXPERIMENTAL SETUP

A total of 5 participants (age 28 ± 3.5) were involved in the testing. The participants have different confidence and previous experience with using robots in general and pHRI tasks. In particular, one participant had no experience with robots at all. The participants were instructed to move the robot tip from start to target pose and were told that the robot could take control over the human in some situations. Participants were asked to let the robot lead if they sensed that the robot was taking control.

The robotic platform is a Universal Robot 5 controlled in joint velocity at a frame rate of 125 Hz. The interaction force is measured at 100 Hz with a Robotiq FT300 sensor mounted at the robot end-effector. A handle is mounted after the sensor, allowing the human to grasp and interact with the robot. The robotic setup during the execution of experiments when the human is leading the robot to avoid an obstacle is visible in figure 6.14.

The parameters used are as follows. $Q_{h,h} = diag([1, 1, 1, 0.0001, 0.0001, 0.0001])$, $Q_{h,r} = 0^{6x6}$ and $R_h = diag([0.0005, 0.0005, 0.0005])$ and $Rr = diag([0.0001, 0.0001, 0.000$

[1] All these values are normalized with respect to $Q_{h,h}(1,1)$. In Optimal Control problems, the minimization of J_i equals the minimization of $\lambda\, J_i$, with λ positive value. Therefore, since the parameters $Q_{h,h}$ and R_h comes from Inverse Optimal Control studies, they

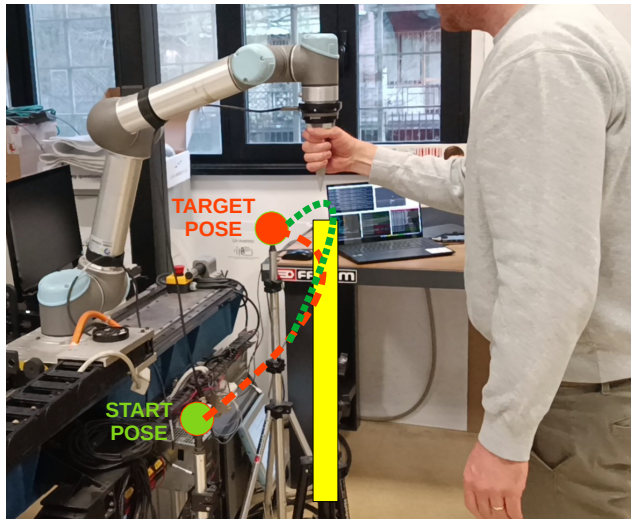


Figure 6.14: Experimental setup during the execution of obstacle avoidance experiment. The human is driving the robot to avoid the physical obstacle highlighted by the yellow rectangle in the picture. The nominal robot's trajectory is in dashed red. In green is the new trajectory after the human intervention to avoid the obstacle. In the real setup, the start pose (green circle), the target pose (red circle), and the obstacle (yellow rectangle) are indicated by tripods to give the human a reference.

The values of the impedance parameters in (3.7) are set to $M_i = diag([10, 10, 10])$, $D_i = diag([100, 100, 100])$ and $K_i = diag([0, 0, 0])$ for the standard manual guidance and the GT experiments, while $K_i = diag([200, 200, 200])$ for the standard impedance control.

The values for the human are computed offline via Inverse Optimal Control as in [41]. Despite the human cost function values possibly changing according to the task, the subject, and the different stages of the task, the choice of using fixed values can be justified for several reasons. First

can be normalized to any arbitrary value. The small values of $Q_{h,h}$ relative to the velocity components of the state indicate that humans do not care about the velocity compared with the tracking error. The small values of R_h indicate that humans prefer to minimize the tracking error rather than the effort required to track it.

of all, the role arbitration mechanism is the fundamental module that changes the assistance level of the robot, which strictly depends on the value of α. Both the value of α and the human cost function influence the outcome of the game, but the value of α has a significantly higher impact with respect to the human cost function, provided that the values used to describe the human cost function are reasonable within certain tolerances (we selected an average value, it is true that such a value can possibly change, but changes are within a restricted range around the value we selected). Note that it is always possible to implement online techniques to recover such parameters on the fly and for different phases of the task, as demonstrated by different works [42, 56, 67, 174]. The robot's parameters are set according to previous studies to assist when required and provide strong interaction when there is no agreement. The matrices Q_h and Q_r are defined as in the cooperative case for all the cases. This means that, in the cooperative case, Q_h and Q_r are defined as in 5.2.2 following (5.13) and (5.14). In the non-cooperative case, equations (5.1) and (5.2) in section 5.2.1 use Q_h and Q_r. To let the Role Arbitration be smooth and continuously variable, we also update values of Q_h and Q_r according to the same update used for the cooperative case, following (5.13) and (5.14).

During the experiments, the FL arbitration module is in charge of selecting the appropriate role for the human and the robot (*i.e.*, the value of α), according to the current status of the tasks. Therefore, during each experiment, the value of α varies. By defining a threshold value of α ($\alpha_{th} = 0.5$, in this work), the robot selects the cooperative or the non-cooperative behavior according to $\alpha > \alpha_{th}$ and $\alpha < \alpha_{th}$, respectively.

6.4.3 RESULTS

This section presents the results relative to the three sets of experiments.

6.4.3.1 EXPERIMENT WITH THE OBJECT UNKNOWN TO THE HUMAN

In the first set of experiments, the robot knows the presence of an obstacle and leads the human away from an object known to the robot. The goal is to move the robot end-effector from an initial position to a target position, then, after a slight pause, move back to the initial position. In the middle of the trajectory is placed a virtual object. The robot knows the obstacle's presence and position, but the human does not. The robot's nominal trajectory is computed to be collision-free, considering the obstacle known to the robot.

Figure 6.16a shows the arbitration parameter α variation during the task. In particular, around second 4, the robot approaches the object, and d_o is low. Moreover, d_{ws} increases, making α low and the robot close to its nominal trajectory.

This experiment computes the capability of the three controllers to avoid any collision. Figure 6.16b shows an average of the index (6.34) computed for all the subjects. As clearly visible, in the MG case, because the robot cannot exert any force, the human is not aware of the robot's intention and collides with the virtual obstacle. In the GT and IMP cases, the human can almost always avoid the obstacle, even if it is unknown, because the robot pulls the human to its collision-free trajectory, helping him avoid it. In this case, GT and IMP behave similarly in proximity to the obstacle.

Figure 6.15 shows the trajectories followed by the robot tip with the three controllers and the nominal collision-free trajectory computed by the motion planner. As visible, the IMP and GT controllers allow the robot to pull the human and its tip close to the nominal collision-free trajectory, allowing the task's success. On the contrary, in the MG case, no force can be exerted, and the robot cannot prevent collision with the obstacle.

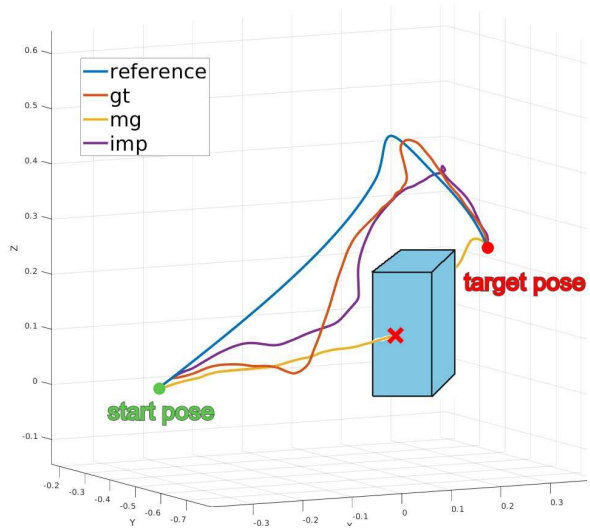
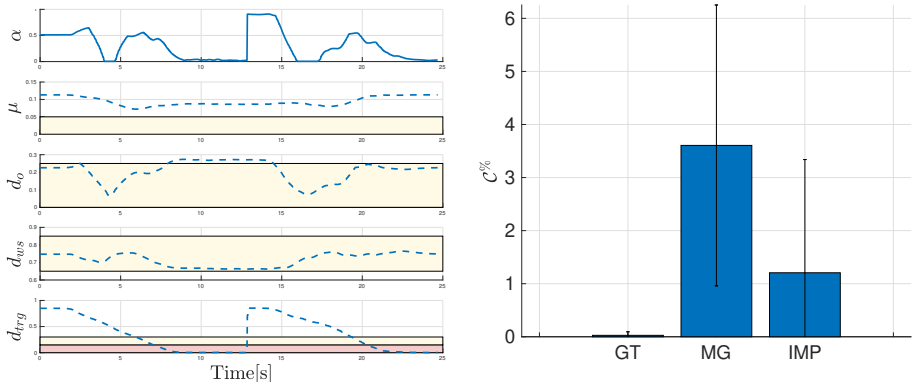


Figure 6.15: The end-effector trajectories in the Cartesian space in the first set of experiments. The GT and IMP controllers can safely avoid the obstacle by imposing force on the human. The MG controller cannot impose any force and collides.

6.4.3.2 EXPERIMENT WITH THE OBJECT UNKNOWN TO THE ROBOT

In this experiment, the human leads the robot away from an object known only to the human.

In this case, because the human knows precisely where the obstacle is, it is easier for them to avoid it. Hence the index (6.34) was found to be always zero, and it is not shown here. Conversely, because the human can move the robot to avoid the obstacle, this may violate workspace boundaries. Indeed, during the test with the MG controller, two subjects exceeded workspace boundaries while avoiding the obstacle (the robot was straight), leading to swift joint motions of the elbow joint and finally to the robot's emergency stop. Figure 6.18b shows an average of the workspace boundaries violation for the three controllers, computed as (6.35) (the two occurrences described above are not used for index com-



(a) Variation of α during the task

(b) Collision percentage

Figure 6.16: Experiment A: in this case, the human does not know the obstacle's presence.

putation, an additional trial was done). The GT controller is the one that better avoids workspace boundaries violation, allowing safer deformations compared with the other two controllers.

As visible from figure 6.17, in the case of MG and partially in the IMP, the trajectory modified by the human exceeds the safe workspace boundaries. The workspace boundaries are, in this work, designed according to the UR specifications. This means the robot can exceed it, as happens in this experiment. Most positions can be reached but with restrictions on the tool orientation because the robot cannot reach far enough in some situations.

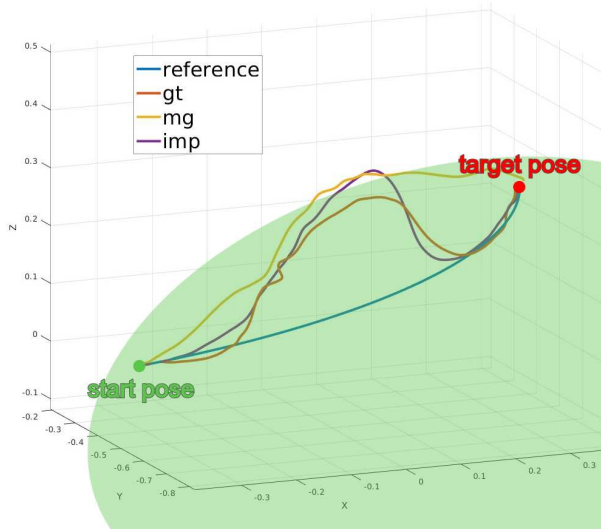
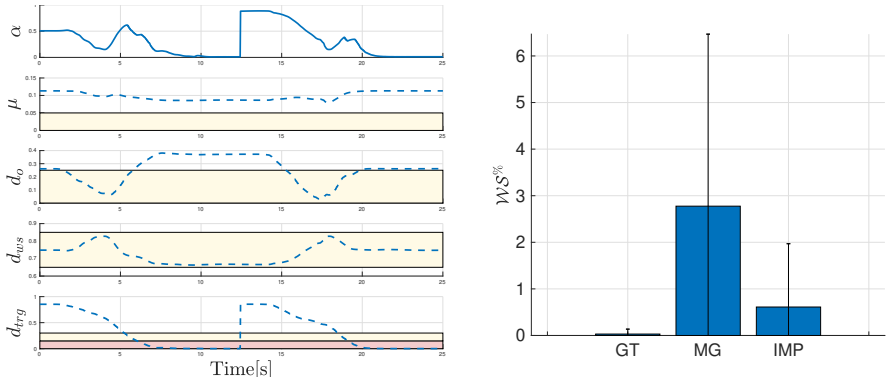


Figure 6.17: The end-effector trajectories in the Cartesian space in the second set of experiments (humans must avoid an obstacle). The MG and IMP controllers slightly exceed the workspace boundaries. The RA framework keeps the GT controller below the dangerous robot's over-extension. The light green area represents the allowed workspace.

6.4.3.3 EXPERIMENT WITH ADDITIONAL TARGET POINT

The third case is presented here. The initial and final points of the task are the same as in the previous experiment, and the robot's nominal trajectories are the same. An additional target point is in between, and the human is asked to reach it. Therefore, instead of directly moving toward the target point, the human has to modify the robot's nominal trajectory to reach an intermediate way-point precisely. No obstacles are involved in the scene, and the robot allows trajectory deformations as long as the tip is far from the target pose.

The variation of the various indices is visible in figure 6.20a. As visible, the only phases where the robot leads happen near the target pose, granting precise position reach. Instead, during the task execution, the



(a) Variation of α during the task

(b) Workspace boundaries.

Figure 6.18: Experiment B: the human deviates from nominal trajectory to avoid an obstacle unknown to the robot

arbitration parameter lets the human lead (with an exception at the beginning of the task because the starting position is close to the WS boundaries), allowing the robot's behavior to be comparable to the one of the MG control. The precision of reaching the intermediate way-point is measured by (6.36), and results of the comparison with MG and IMP are shown in 6.20b. The robot's behavior is comparable to the one of the MG. Therefore, the precision of reaching the target point is comparable between GT and MG, showing good performances. On the contrary, the IMP control does not allow substantial trajectory deformations nor the robot to slow down/stop at a precise way-point, leading to bad performances.

Figure 6.19 shows the trajectory executed with the three controllers to perform this task. As visible, the target via point is unreachable in the IMP case. This happens because the robot prevents the human from reaching it since it does not adapt to the human's intentions. Indeed, the IMP controller aims at following the predefined trajectory allowing just minor adjustments given by the interaction, and too high forces are

required to modify the trajectory substantially.

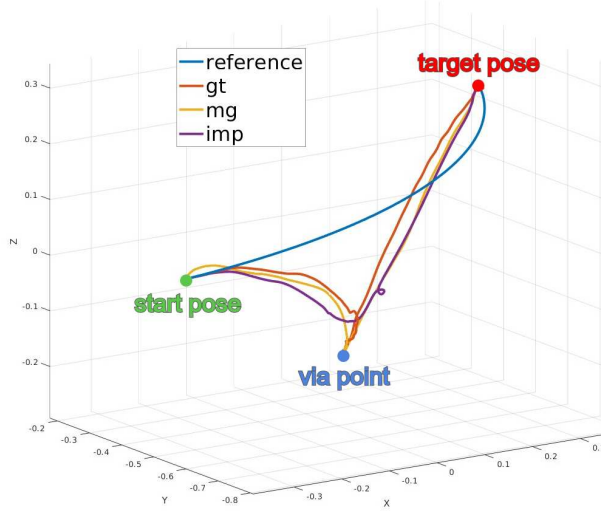
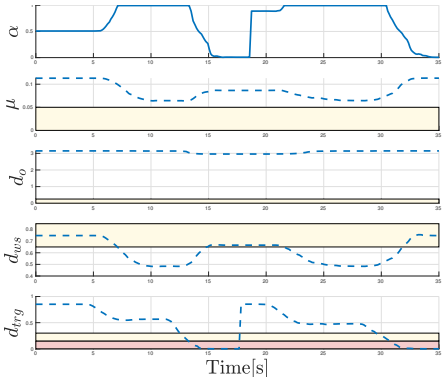
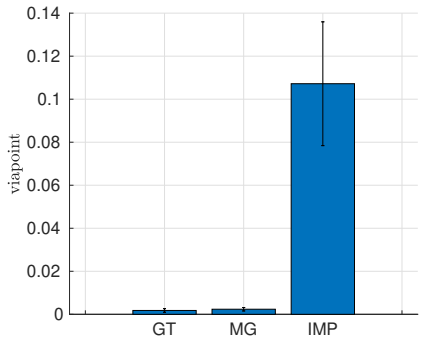


Figure 6.19: The end-effector trajectories in the Cartesian space in the third set of experiments (humans must reach an additional via point). The IMP controller does not allow reaching the desired via point because it pulls the end-effector toward the nominal trajectory far from the via point. The GT and MG controllers allow for trajectory modification and reach precisely the additional via point.



(a) Variation of α during the task.



(b) Proximity to the additional via-point not known to the robot.

Figure 6.20: Experiment C: trajectory deformation

6.4.3.4 GENERAL CONSIDERATIONS

In this subsection, performances comparable for the three controllers are analyzed.

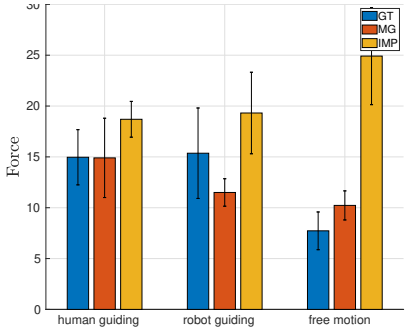
Figure 6.21a shows results for the force index in (6.33). In general, the IMP controller is the one that presents a higher required force to complete the task for all the cases. This happens because the controller treats human forces as external disturbances and the virtual spring always tends to steer the system to the reference position.

MG and GT show comparable forces in the case the human knows the obstacle and moves the robot to avoid it. Considering the case where the robot knows the obstacle's presence, the GT case exchanges more force because the robot must impose a force to attract the robot end-effector on the safe collision-free nominal trajectory. Such a force represents a haptic channel of communication that makes the human aware of the robot wanting to take the lead. In the MG case, the exchanged force is lower because the robot follows the human without steering capability. This situation is not applicable because it leads the system to a collision. Fi-

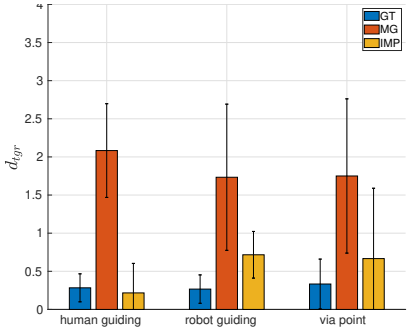
nally, in the case of additional via point reaching, the MG control requires some additional force if compared with the GT control because it always requires force to move in the same direction. In the GT case, the force can be reduced as the robot assists in moving in the desired direction.

The capacity to reach with precision the final point is in figure 6.21b. As expected, in the MG case, the capacity to match the final point is left to human ability, leading to the highest values for all three experiments. The performances of GT and IMP are comparable, as the two methods are very similar in behavior when the arbitration parameter α is low.

Finally, note that the robot's nominal paths are precomputed for each experimental scenario and kept the same for each experiment. This choice is made to compare each experiment avoiding dependencies of the various indexes (in particular (6.33), (6.34) and (6.35)) from different path lengths. This may raise the reasonable question of how different paths and path lengths influence the indexes evaluations. In general, different values for the indexes are expected with different path lengths, as they strictly depend on the interaction time or the percentage of interaction along the full path. Despite this, similar results are expected when comparing the same controllers on the same path, even if it is shorter or longer than the ones proposed in this work. For example, consider a longer path for the experiment presented in section IV-A. For sure, the values visible in 6.16b will be different, as the percentage computed depends on the path length. Despite this, the proposed method still allows the user to avoid the unknown obstacle better than the MG and IMP controllers.



(a) Forces exerted by the human while performing the task with the three controllers. P-values are (0.9560, 0.0067, 0.0130, 0.0126, 0.0005, 0.0003). The null hypothesis is rejected between GT and MG if the human deviates from the robot. The null hypothesis is rejected between GT and IMP if the robot deviates from the human trajectory. The null hypothesis is confirmed for the free robot motion



(b) Target position reaching.

Figure 6.21: Evaluation of indexes for the three trajectories

6.5 USE-CASE SCENARIO

Finally, this section presents the application of the Role Arbitration controller presented in the section above to a real industrial use-case. The role arbitration method is implemented in a real scenario for the manipulation of flexible materials. In particular, the co-manipulation of large carbon fiber ply is addressed.

6.5.1 TASK DESCRIPTION

The selected use-case scenario involves a human and a robot that must co-manipulate a set of large carbon fiber plies of different shapes. The material is pre-preg, meaning that the fibers are pre-impregnated with

the polymer matrix. Such material is commonly used for applications that require strong mechanical properties and low weight, such as in the aerospace and high-performance automotive industries. The specific use case is designed for the production of the spoiler of the Dallara "La stradale" car. One of the main issues in composite material production is the precise positioning of the carbon fibers. Indeed, carbon fibers have exceptional mechanical characteristics but can only bear tensile stresses, not compression or shear. Therefore, the fibers must be aligned according to the design phase to guarantee that the most stressed points do not break. Moreover, the specific use-case also must fulfill aesthetic requirements.

In this context, the robot is fundamental because of its precision. It can indeed position precisely the so-called First Lamination Point (FLP) of carbon fiber plies. The FLP is the point of the ply from which the draping process begins. If the FLP is positioned with the required precision, then the draping is very likely to be made as designed. Therefore, the robot must (i) pick the ply in the precise position and (ii) move the ply into the mold to guarantee that the FLP is in the correct position. The main issue when the robot has to manipulate large plies is that it has a limited workspace, and it might be difficult for the robot to properly grasp and position the entire ply.

The human, for his part, has a large workspace, as it can freely move. Another relevant property of humans is that they can easily handle unforeseen situations, such as an obstacle in between the path, and take action to avoid it. Finally, regarding the specific composite material manufacturing, the human operator must perform the draping task, which is too complex a task to be automatized because the material is sticky and flexible, and the mold might have angles and slopes that the robot cannot handle.

For these reasons, it is reasonable for humans and robots to cooperate

in handling and manufacturing large plies. The complete task is composed of various sub-tasks, that require different levels of autonomy for both agents in the manufacturing scene. The full draping procedure steps are detailed in the Appendix A.4.

The most delicate sub-task is collaborative transport. In this situation, the role of leader continuously moves from the human to the robot and vice versa. This happens particularly because the workspace of the robot can be limited, the robot can be huge, and the ply and the gripper and other features might partially occlude the human sight. In particular, in the proposed use-case, the workspace is limited by walls and cell dimensions, and the gripper is particularly massive. These make the co-manipulation particularly challenging because the human has limited control of the rear of the gripper, which is not visible and far from the rotation point. This fact that the rear of the gripper is far from the rotation point means that even small rotations of the ply imply a large linear motion of the rear of the gripper. Therefore, the Role Arbitration law must avoid any collisions, still allowing the human to lead the task when far from collisions.

6.5.2 VIRTUAL FORCE ESTIMATION

This thesis's previous work relies on force exchange between humans and robots, measured through the force-torque sensor. Forces are exchanged either directly grasping the robot tip as in 4.5, or with a rigid object in between, as in 4.17c. Nevertheless, the proposed use case addresses the co-manipulation of a flexible, large object. Being the object flexible, with a stiffness comparable to the one of a cloth or sheet, the force can only be imposed with traction when the fabric is fully extended and has some tension. This makes the measurement complex, even impossible, and unreliable when the fabric is not fully extended. To overcome this issue, a method to estimate a virtual exchanged force u_h has to

be defined.

The method proposed in this work relies on the relative distance between the human and the robot. This implies that the ply, during the co-manipulation, must always be without tension. Otherwise, the ply deformation will always be the same for any applied force. Therefore, we define a nominal distance lower than the maximum allowed by the ply. That distance, call it d_0[2] represents the zero-wrench state. Therefore, any deformation with respect to such a pose is defined by the distance between the human and the robot d. The difference $\Delta d = d - d_0$ can be seen as a force that the human imposes on the system. Finally, since typically wrenches have magnitudes different from the deformations allowed by the co-manipulated plies, the relative distances are multiplied by a vector of gains K_d to convert them into wrenches, resulting into

$$u_{h,d} = K_d \, \Delta d \tag{6.37}$$

The distance d is measured from the middle point between the hands of the human, as shown in figure 6.22. To evaluate the position of the

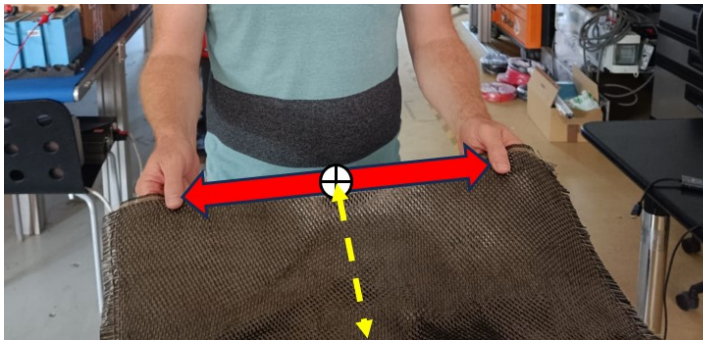


Figure 6.22: The distance between the human and the robot in yellow. The distance is measured from the median point between the two hands.

[2]the distance is described as a vector containing position and rotation information

hands, several methods can be used. It is possible to use skeleton tracking methods to find the hands' positions but this approach typically does not guarantee a high frame rate because it requires relatively long image elaboration. Another possibility is the use and tracking of fiducial markers [115], which allows a higher frame rate. A different and novel approach relies on mapping deformations of the ply and distances from the human [125]. This approach requires the training of a Convolutional Neural Network. After that, it is capable of estimating the distance just by looking at the shape of the ply. The last approach is used for the experiments presented in this section.

6.5.3 ROLE ARBITRATION LAW

The Role-Arbitration law, for the proposed use-case takes inspiration from what is proposed in subsection 6.4.1.2, simplifying it a little. We do not consider singularities and joint limits for the use-case since the proposed task does not involve dangerous motions that can reach such limitations. Moreover, to properly consider the size of the gripper, we also do not directly consider workspace boundaries. Indeed, workspace boundaries are computed for a single point (the TCP) and do not consider oversized grippers. Therefore, we consider workspace boundaries as collision objects. In this way, managing proximity is easier both from a logical and a computational point of view. The cell is visible in figure 6.23, the yellow walls represent the workspace boundaries as collision objects.

The role arbitration law, in the end, reduces to measure the distance from target d_{trg} as in (6.31), and the distance from collisions d_o as in (6.29). The two values are modulated by a sigmoid function defined as in (6.4), to obtain $\alpha_{trg} = \sigma(d_{trg})$, and $\alpha_o = \sigma(d_o)$, with σ denoting the sigmoid function. Finally, the parameter α is computed as

$$\alpha = min(\alpha_{trg}, \alpha_o) \tag{6.38}$$

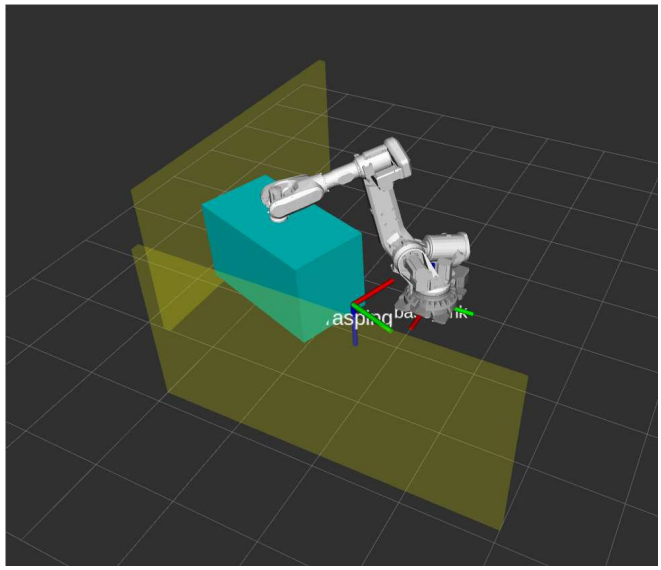


Figure 6.23: The cell and its boundaries, defined as collision objects in yellow.

6.5.4 EXPERIMENTS

To verify the proposed method in the use-case scenario, we also compare with results obtained using a pure Manual Guidance (MG) controller. When the MG controller is used, the robot only follows human intentions and reduces the ply deformation by trying to keep it undeformed. Therefore, it allows co-transportation of large plies that otherwise would be unfeasible for a single human.

6.5.4.1 PERFORMANCE INDEXES

We compare the success rate of the task by using the proposed Role arbitration method and the standard MG controller. The success rate is measured as three divided by the number of trials n_t required to reach

three times the target pose of the FLP, as

$$Sr = \frac{3}{n_t} \tag{6.39}$$

A task is not considered complete if the robot stops due to safety stops caused by the gripper exceeding some safe zone installed into the robot controller. Such safe zones are necessary to prevent the robot from colliding with the external walls and are conservatively positioned 10 centimeters from the wall and cell boundaries. We use the same distance to add the yellow collision walls into the scene for the role arbitration.

We also measure the absolute time T required to reach the target pose within a certain tolerance. The tolerance is set to be $tol = 30mm$, and the distance is measured as the Euclidean distance between the current pose x_c and the target x_{trg} of the FLP. When $d(x_{trg} - c_c) < tol$ the task ends. T is measured for the three times the task is a success, with the same data used to compute Sr. To summarize, T is measures as

$$T = min\{t | d(x_{trg} - c_c) \leq tol\} \tag{6.40}$$

Another interesting measure is the time required to complete the task in the collaborative modes with respect to the nominal computed trajectory for the robot only. We measure it as

$$\Delta T_{\%} = \frac{T - T_n}{T_n} \tag{6.41}$$

where T and T_n represent the time required to perform the actual trajectory and the nominal time, respectively.

To see how much the actually executed trajectory deviates from the nominal, we also compute the Dynamic Time Warping (DTW) [112]. The DTW provides information about the geometrical distance between two trajectories without taking into account the time and possible time shifts

between the two trajectories. It is useful to see how well the cooperative trajectory executed tracks the original path. Indeed, following the nominal path represents a great advantage in the case the trajectory is computed to comply with ergonomic constraints. In this case, the role arbitration algorithm can also help the human work in ergonomic conditions that might be lost if the human guides the robot.

We do not directly measure the precision with which the robot positions the FLP with respect to its target pose. This is because the collaborative transportation ends when the FLP approaches the goal pose within a tolerance. Therefore, the tolerance represents the maximum precision that we can reach.

A total of three subjects performed the experiments. Each subject is let to practice with the controller before collecting data. Each subject knows which control mode is active - either MG or RA - during each task. This is necessary to let the human know that the robot wants to lead when it requires control to avoid collisions. One of the subjects has previous experience with this kind of controller, while it is the first time for the other two.

6.5.4.2 RESULTS

Table 6.3 shows results relative to the success rate Sr from (6.39). In the manual guidance case, not all the users completed the task successfully at each trial. This is mainly due to the fact that, after picking up the ply, it was very complex to lead the robot far from the picking pose. This is because the gripper is huge and has to rotate by more than $90°$. Such a rotation applies around the grasping point of the ply, located on one side of the gripper. Therefore, even a small rotation around that point causes a large motion at the rear of the gripper. The grasping point is close to a wall, and this makes it extremely complex to move the robot. In the failure cases, the gripper hit the safety fences placed to prevent the robot

from colliding with the wall, and the task stopped. On the contrary, in the RA case, when the gripper approaches the wall (*i.e.*, the safety fences), the robot knows the risk of a collision and takes control over the human to recover the pre-computed safe and collision-free nominal trajectory. in this way, it is possible to avoid hitting the safety, and 100% of the trials succeeded.

	MG	GT
usr1	100%	100%
usr2	75%	100%
usr3	60%	100%
total	75%	100%

Table 6.3: The success rate for the three users and the global using the two controllers.

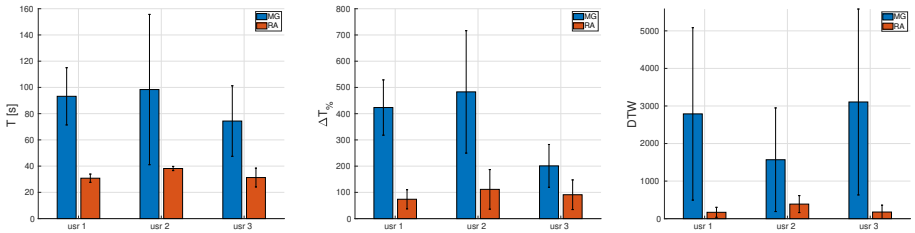
In figure 6.24, the indexes relative to the total time T, the percentage increment $\Delta T_{\%}$, and the DTW are visible. In particular, figure 6.24a shows the time required to perform the task with the two controllers, for the three users. It is visible that the MG case takes about three times more than the time required to reach the target pose x_{trg} in the RA case. This is because of two main reasons. First of all, as already discussed, it is quite complicated to rotate the large gripper in the narrow spaces of the cell. In the MG controller case, the operator needs to be careful and care for avoid collisions during the co-manipulation. Therefore, each motion performed is small and slow, and sometimes the operator must also plan in advance and reason about the next move. Instead, in the RA case, the robot helps the human remain in a safe operating zone, and the operator is more confident in performing faster and larger motions. The second reason for the great improvement provided by the RA compared to the MG controller is because when approaching the x_{trg}, the robot helps the human and moves the FLP towards its target pose. In the MG case, the

human must place the ply exactly at the FLP pose without any help. This task is also very complex because even with a visual reference of where the robot must go, it is not easy to move with precision the huge gripper when the only communication channel is a large, deformable ply. Indeed, if the object were rigid, force feedback would be more intuitive for manual guidance. Moreover, being the ply large and long, the human is far from the target point, and from the operator's point of view, it might not be clear where the FLP is with respect to its target pose.

Figure 6.24b shows results for $\Delta T_{\%}$. In this case, it is clearly visible that the RA controller allows for reducing the execution time if compared with the MG, with respect to the nominal execution time. Note that the nominal trajectories are computed by a sample-based motion planner (RRT), therefore the trajectories are not always repeatable, and their execution time might vary a lot. The interesting result, in this case, is that the actual execution time with respect to the nominal execution time is more close in the RA case. Despite being almost double with the RA controller, with the MG controller, the execution time rises up to five times more than the nominal one. Consider that the motion planner should also be "human-aware", and should plan a trajectory that is ergonomic and natural for the human [3]. In this case, it is expected that executed and planned times do not differ that much. This is very useful for plant scheduling of tasks and making reliable forecasts on production times, for example.

Finally, figure 6.24c presents the DTW index. The lower the DTW value is, the more similar the two trajectories are. This shows that, in the RA case, the actually executed trajectory is more similar than in the MG case with respect to the nominal one. In general, it is not fundamental that the executed and the planned trajectories are similar. Despite this, it can present some advantages. Consider a human-aware, ergonomic,

[3] The design of a human-aware, ergonomic, natural motion planner is out of the scope of this work. Despite this, its integration is foreseen and more results will be presented in future works.



(a) The time required to perform the task with the two controllers.

(b) The $\Delta T_{\%}$ from (6.41)

(c) The DTW

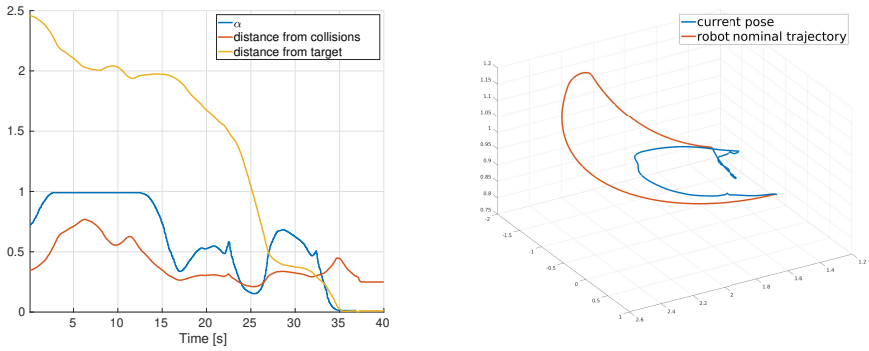
Figure 6.24: Evaluation of the indexes for the three subjects.

motion planner. In this case, it is very useful that the human follows the trajectory which is planned to be ergonomic. This can avoid long-term injuries due to wrong postures and working conditions. Moreover, consider an optimal motion planner. In this case, the closer the actual trajectory is with respect to the planned one, the more close-to-optimal it is. This can further improve cycle times if the trajectory is computed with that objective. Then consider that some areas in the cell might not be safe for working. The motion planner can plan a trajectory to avoid such areas. The closer the actual trajectory is, the safer it is. This last example is actually what happens in the proposed use-case scenario when the robots avoid the human from hitting the safety fences by recovering its nominal trajectory.

Figure 6.25a shows the variation of the arbitration parameter α during the execution of collaborative transportation. In the beginning, the pick pose is close to the safety fence, α is not exactly one, and the human is helped in moving away from that situation until α is one. As the human moves the robot, it approaches the safety fences, the distance from collisions drops, and so it does α, around second 13. After this point, the robot is blandly in command of the task and moves slowly toward

the target pose. As soon as the d_{trg} component gets lower than the d_o, the robot strongly takes control of the task, and α quickly goes to zero, around seconds 33. After this point, the robot is in almost full command of the task and gets precisely to the target pose x_{trg}.

Figure 6.25b shows the nominal and actual trajectories based on the arbitration in 6.25a. At the beginning (rear part of the image), the human can command the robot, and the two trajectories substantially differ. In particular, in this case, the motion planner (recall that is it sample-based, not human-aware, etc.) plans a trajectory (in red) that is not comfortable for the human as it rises a lot before descending to the target pose. Therefore, the human is allowed to modify it and the current trajectory (in blue) seems more reasonable. In the final part of the trajectory, in the bottom right, the two trajectories slowly rejoin until the final point where they match exactly.



(a) The RA arbitration parameters variation **(b)** The nominal and modified trajectories along the task.

Figure 6.25: The RA parameters and the corresponding trajectories

Finally, figure 6.26, shows the two trajectories in the real scenario.

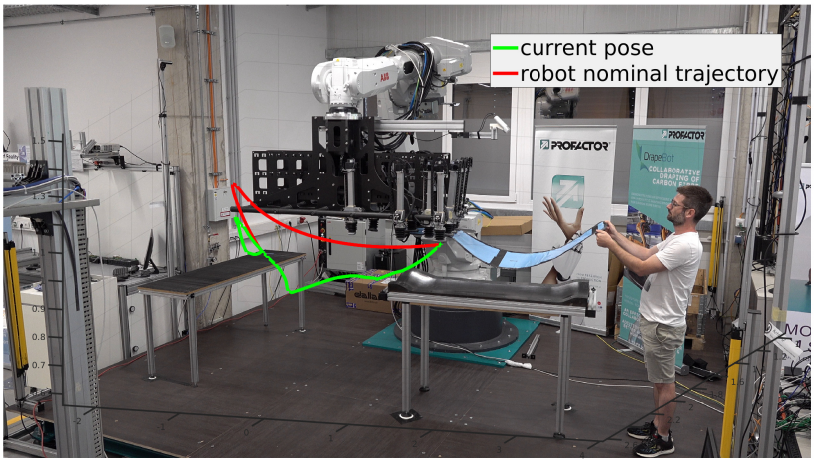


Figure 6.26: The nominal and actual trajectories in the cell.

6.5.4.3 CONCLUSIONS

This use-case shows that the RA mechanism is useful in handling real industrial scenarios. It is shown that with such an approach, the success rate of the task can be significantly improved. Moreover, the execution time drops compared to the standard Manual Guidance control. Also the time increment and the DTW with respect to a nominal planned trajectory indexes show that the RA control overcomes the MG control. This becomes even more interesting when an optimal, human-aware motion planner is used to compute the nominal trajectory. In this case, it is possible to make reliable forecasting on execution times and it is possible to let operators work in safe conditions, avoiding dangerous areas of the workcell. A novel motion planner will be integrated in future work to show that it can further improve the execution time and safe working conditions.

CHAPTER 7

PREFERENCE BASED OPTIMIZATION OF PHRI CONTROLLERS

The aim of this chapter is to present a method for tuning a pHRI controller based on the preferences of different subjects. Because different tasks may require different tuning, it is important that such a procedure is fast and easy. Moreover, interesting results can be observed as general human preferences. We tested our methodology on two different tasks, one requiring precise path following and the other requiring a fast and large motion toward a target position. The data are analyzed to see general human behavior and preferences. Finally, a questionnaire is proposed to the users to check this method's applicability in real environments, involving time required for tuning and satisfaction.

7.1 METHOD

Consider the Cartesian impedance system in 3.8, and the Cooperative Game theoretic framework of 5.2.

7.1.0.1 HUMAN REFERENCE ESTIMATION

In hand-guiding applications, the reference trajectory is typically not known a priori, and the robot must follow as well as possible human desired reference. So the problem of estimating the human reference arises. In this work, We decided to implement an easy yet powerful method based on the interaction force. The human reference pose is updated at each cycle by the following:

$$x_{ref}^{+} = x_{ref}^{-} + K_{p,h}\, u_h \tag{7.1}$$

with superscripts $^{+}$ and $^{-}$ referring to the updated and previous poses, respectively, and $K_{p,h}$ defines a coefficient proportional to the human exerted force. $K_{p,h}$ is an arbitrary parameter that can be tuned based on human preference.

7.1.1 PREFERENCE-BASED OPTIMIZATION

The PBO algorithm employed in this paper is based on the methodology developed in [15] by some of the authors, where the GLISp algorithm is introduced. In the following, the algorithm is briefly recalled. Please refer to the original paper for full treatment.

7.1.1.1 BUILDING A SURROGATE FUNCTION FROM PREFERENCES

The objective of the GLISp algorithm is to learn and minimize a surrogate function $\hat{J} : \mathbb{R}^{n_\theta} \to \mathbb{R}$ of an (unknown) underlying performance index J based on the observed preferences. Given a set of parameters

$\Theta = \{\theta_1, \ldots, \theta_N\}$ The surrogate function $\hat{J}$ is parametrized as a linear combination of Radial Basis Functions (RBFs):

$$\hat{J}(\theta) = \sum_{k=1}^{N} \beta_k \phi(\gamma d(\theta, \theta_k)), \tag{7.2}$$

where $d : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$ is the squared Euclidean distance $d(\theta, \theta_i) = \|\theta - \theta_i\|_2^2$, $\gamma > 0$ is a scalar parameter, $\phi : \mathbb{R} \to \mathbb{R}$ is an RBF, and $\boldsymbol{\beta} = [\beta_1 \ \ldots \ \beta_N]^T$ are the unknown coefficients to be computed based on the available user's preferences. In this work, the Gaussian RBF $\phi(\gamma d) = e^{-(\gamma d)^2}$ is used, for more examples and explanations of RBFs please refer to [14].

Given two sets of parameters θ_i and θ_j with $i \neq j$, the preference function $\pi : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\theta} \to \{-1, 0, 1\}$ is defined as:

$$\pi(\theta_i, \theta_j) = \begin{cases} -1 & \text{if } \theta_i \text{ "better" than } \theta_j \\ 0 & \text{if } \theta_i \text{ "as good as" } \theta_j \\ 1 & \text{if } \theta_i \text{ "worst" than } \theta_j. \end{cases} \tag{7.3}$$

Therefore, the surrogate function $\hat{J}$ has to satisfy the following constraints:

$$\begin{aligned} \hat{J}(\theta_{i(h)}) &\leq \hat{J}(\theta_{j(h)}) - \sigma + \varepsilon_h \ \text{ if } \ \pi(\theta_{i(h)}, \theta_{j(h)}) = -1 \\ \hat{J}(\theta_{i(h)}) &\geq \hat{J}(\theta_{j(h)}) + \sigma - \varepsilon_h \ \text{ if } \ \pi(\theta_{i(h)}, \theta_{j(h)}) = 1 \\ |\hat{J}(\theta_{i(h)}) - \hat{J}(\theta_{j(h)})| &\leq \sigma + \varepsilon_h \ \text{ if } \ \pi(\theta_{i(h)}, \theta_{j(h)}) = 0 \end{aligned} \tag{7.4}$$

for all $h = 1, \ldots, M$ with M number of expressed preferences, where $\sigma > 0$ is a tolerance, and ε_h is a positive slack variable which is used to relax the preference constraints.

Defining with $\Delta\hat{J}_h = \hat{J}(\boldsymbol{\theta}_{i,h}) - \hat{J}(\boldsymbol{\theta}_{j,h}) = \phi\gamma \sum_{k=1}^{N} (d(\theta_{i(h)}, \theta_k) - (d(\theta_{j(h)}, \theta_k)\beta_k,$

the coefficient vector $\boldsymbol{\beta}$ describing the surrogate $\hat{J}$ is the solution of the Quadratic Programming (QP) problem, constrained by (7.4):

$$\begin{aligned} &\min_{\beta,\varepsilon} \sum_{h=1}^{M} \varepsilon_h + \frac{\lambda}{2} \sum_{k=1}^{N} \beta_k^2 \\ &s.t. \Delta\hat{J}_h \leq -\sigma + \varepsilon_h, \forall h: \ b_h = -1 \\ &\quad \Delta\hat{J}_h \ \geq \sigma - \varepsilon_h, \forall h: \ b_h = 1 \\ &\quad \left|\Delta\hat{J}_h\right| \leq \sigma + \varepsilon_h, \forall h: \ b_h = 0 \\ &\quad h = 1, \ldots, M. \end{aligned} \tag{7.5}$$

with $b_h = \pi(\theta_{i(h)}, \theta_{j(h)})$. The scalar $\lambda > 0$ in the cost function (7.5) is a regularization parameter that guarantees uniqueness in the solution of the QP problem.

7.1.1.2 ACQUISITION FUNCTION

To guarantee a tradeoff between *exploration* and *exploitation*, to generate a new set of parameters, an *acquisition function* $a(\theta)$ can be defined such that

$$\theta_{N+1} = \arg\min_{\theta \in \Theta} a(\theta). \tag{7.6}$$

Let's define an *exploration function* as the Inverse Distance Weighting (IDW)

$$z(\theta) = \begin{cases} 0 & \text{if } \theta \in \{\theta_1, \ldots, \theta_N\} \\ \tan^{-1}\left(\frac{1}{\sum_{i=1}^{N} w_i(\theta)}\right) & \text{otherwise} \end{cases} \tag{7.7}$$

where $w_i(\theta) = \frac{1}{d^2(\theta,\theta_i)}$.

The *acquisition function* $a : \mathbb{R}^{n_\theta} \to \mathbb{R}$ is constructed as:

$$a(\theta) = \frac{\hat{J}(\theta)}{\Delta\hat{J}} - \delta z(\theta), \tag{7.8}$$

where $\delta \geq 1$ is an arbitrary exploration parameter and $\Delta\hat{J} = \max_i\{\hat{J}(\theta_i)\} - \min_i\{\hat{J}(\theta_i)\}$ is the range of the surrogate function on the samples in Θ and used in (7.8) as a normalization factor to simplify the choice of the exploration parameter δ.

After an initialization phase, the following steps are iterated for the optimization:

i) generate a new sample by (7.6),

ii) ask the user to express a preference $\pi(\theta_{N+1}, \theta_N^\star)$;

iii) update the estimate of $\hat{J}$ through (7.5);

iv) iterate over N.

7.1.2 PARAMETER FOR OPTIMIZATION

To conclude, in this work We are seeking the optimization of the robot's weight on the control action (*i.e.* R_r), the weighting factor of the cost functions (*i.e.* α), the proportional gain $K_{p,h}$ of eq. (7.1).

The R_r is chosen because it affects the cost the robot has on the control it can provide. As a consequence, the robot's behavior is more reactive and can put more effort into the task to track x_{ref}. This is the only parameter of the robot's cost function because it is assumed that the Q_r is always 1 on the positions and about 0 on the velocities, null all the rest. In this way, there is a strict relation between Q_r and R_r, in the sense that R_r is always a fraction of the Q_h. Note that this does not change the outcome of the optimization as $min(J) = min(\lambda J)\ \forall\ \lambda > 0$.

The parameter α represents the solution to the Bargaining problem. It directly modifies the contribution of the robot into the global cost function (5.10) and consequently the global optimization (4.16). In general, high values of α represents more robot contribution and in this work, more robot assistance.

Finally, the parameter $K_{p,h}$ modifies the set-point. At each control step, the set-point is updated according to (7.1). It is clear that high values of $K_{p,h}$ make the updated set-point far from the current one, while low values make it close. This means that at each control cycle, the farther the set point is, the higher the control input required to reach it is.

7.2 EXPERIMENTAL VALIDATION

To evaluate the proposed method, We asked 5 healthy subjects, aged between 27 and 36 years old, to perform two sets of experiments. The subjects have different knowledge and experience in the pHRI field, from zero experience to some. We defined some numerical indices and a questionnaire to evaluate objective and subjective performances.

7.2.0.1 DESIGN OF EXPERIMENT

Two sets of experiments are proposed to evaluate the procedure in two different scenarios. The first experiment wants to evaluate the preferences of humans in performing a precise task, as a precise path following in the x–y plane. Subjects are asked to track the path visible in Fig. 7.1. This task is selected to mimic some typical industrial applications such as painting, cutting, material deposition, or even teaching-by-demonstration in hand-guiding applications, where the human drives the robot towards a path. In these cases, the robot can possibly hold a heavy tool relieving the operator from the weight, and the human just has to focus on the guidance asppect. The second task is a reaching task to evaluate the preferences of humans in performing relatively fast motions without the necessity to be extremely precise. Subjects are asked to reach a set point at 650 mm, in the x–y plane. This task is selected to mimic the approach phase to a target position. In general, these two tasks are selected to mimic a scenario where the human moves the robot close to a target posi-

tion, then performs a precise task. It is expected that different preferences are assigned to different tasks with different goals.

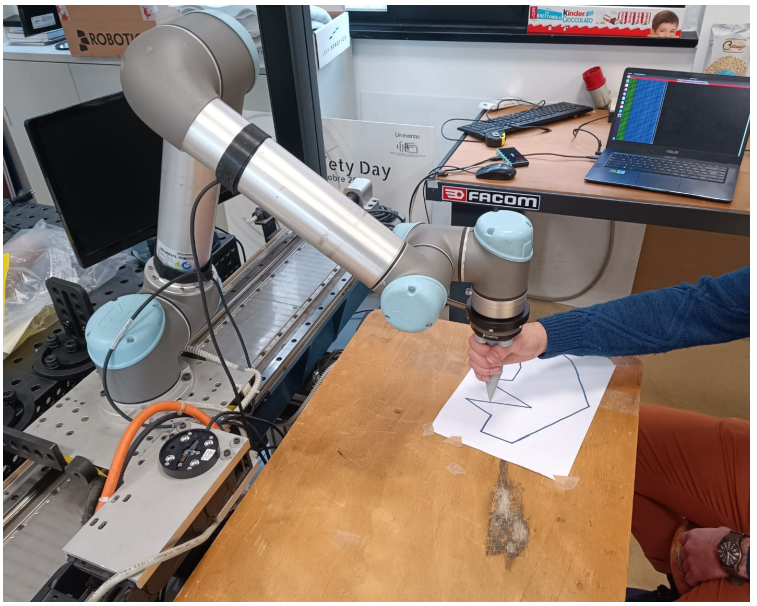


Figure 7.1: A subject performing the path following task. The complete setup is visible.

For both cases, the human's cost function is composed of $Q_h = diag([1, 1, 0.0001, 0.0001])$ and $R_h = diag([0.0005, 0.0005])$, recovered via Inverse Optimal Control experimentally in a previous work [41]. The impedance parameters are $M_i = 10$ and $D_i = 50$. The only parameters related to the robot which are not optimized during the procedure are the components of the Q_r in the robot's cost function, which are set $Q_r = diag([1, 1, 0.0001, 0.0001])$. The robot used for the experiments is a Universal Robot 5 controlled in velocity at 125 Hz. The human interactive force is measured by a Robotiq FT300 force sensor mounted at the robot's end-effector.

The subjects are asked to perform the two tasks as well as possible

and are asked to give a preference based on the satisfaction of performing the overall task. In the first case, the subjects are asked to evaluate their performance in the path following, along with the stress (physical and mental) required to perform it. In the second case, subjects are asked to evaluate the support that the robot is able to provide to reach the target pose quickly and with low effort.

The preferences are expressed, for both experiments, in two steps. An initial step is necessary to initialize the GLISp. Four experiments with different parameter sets are performed, and the user is asked to rank these experiments. After that, the best candidate solution found is compared with a new set, and the user is asked to express a preference. Based on the preference, the candidate solution is either updated or kept, and the process is iterated until convergence. We consider convergence when (i) three times in a row the preferences expressed are "as good as", (ii) three times in a row the new set of parameters is very similar to the ones suggested previously and discarded, (iii) a maximum number of iteration is reached.

7.2.0.2 EVALUATION CRITERIA

We used some performance indexes to evaluate the final results and to analyze relevant trends.

For the first case only, the Dynamic Time Warping (DTW) method [112] is used to compare the trajectories with the nominal one, to evaluate how well the preference expressed by the human is able to track it. This is not applicable to the reaching case because no predefined trajectory is used, and the subjects are asked to reach approximately a final position.

For both cases, it is possible to measure the force required to complete the task, which is computed as the Root Mean Square (RMS) of the

measured force as follows:

$$f_{RMS} = \sqrt{\frac{1}{t_{end} - t_0} \int_{t_0}^{t_{end}} f(t)^2 \, dt} \tag{7.9}$$

We measure the f_{RMS} to evaluate how much effort humans like to put into a task. Indirectly, We want to analyze humans' preference for hard or soft haptic feedback in manual guidance applications.

Finally, a quick questionnaire is proposed to the users to evaluate their feelings on (i) how long was the tuning procedure, (ii) how tiring it was, and (iii) how much satisfied they are with the optimized parameters set. Users are asked to rate on an interval [0,4], with 0 meaning "no long at all", "no tiring at all", and "not satisfied at all", respectively. The questionnaire is asked after each optimization (path following and reaching) to check if there are differences.

7.2.1 RESULTS

Figure 7.2 shows the nominal trajectory (red-dashed), the trajectory with the best DTW (solid blue), and all the other trajectories performed by a subject with different parameters during optimization.

We evaluate the DTW to check if the perception of the human on the best task is also confirmed numerically. In figure 7.3, it is visible for the five subjects the range of DTW measured with different parameters (for subjects 2 and 4, the range is cut because a trial was close to instability and too much high DTW is scored). In red is highlighted the execution corresponding to the preferred by the human, and in three subjects coincide with the best measured. This study is useful to check that the preferences expressed by the users can also be measured and verified so that the subjective PBO, in this case, corresponds to objective better results.

A parameter typically used to show the goodness of a proposed pHRI controller, a measurement of the force exchanged is given, assuming that

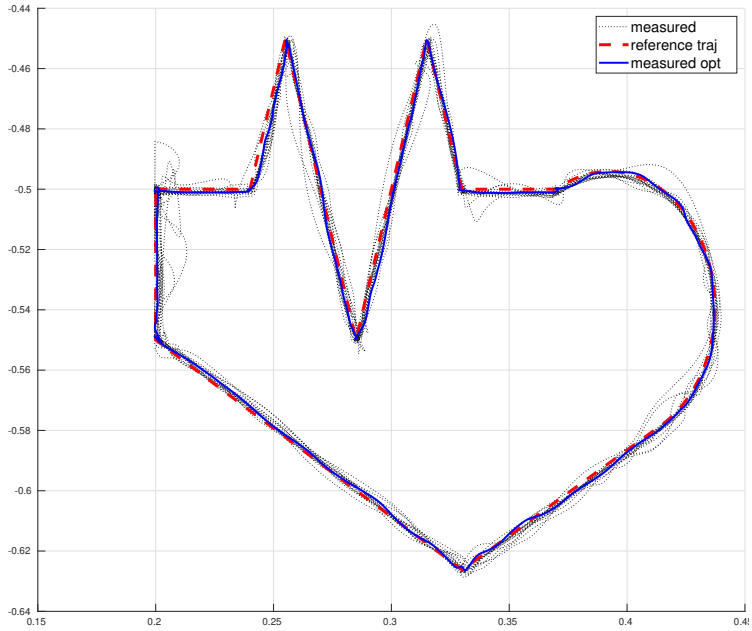


Figure 7.2: Trajectories comparison relative to subject 3. Dashed red is the reference trajectory, solid blue is the measured trajectory with optimized parameters, and dotted black are all the other trajectories.

the lower is the best. So, based on the preferences expressed, We want to see if this sentence is true. The f_{RMS} is visible in figure 7.4, for both tasks. It is, in general, true that low forces correspond to more pleasant interaction. Despite this, in a couple of cases, this is not true. This happens in particular for the reaching task. It seems that humans prefer to reach the set point quickly rather than save effort. A similar result was also recovered in [41]

Finally, the preferred parameters set are presented in figure 7.5 for both tasks for the three subjects. In general, high values of R are preferred for the path-following task, while low/very low values are preferred for the reaching task. Recalling that the parameter R is responsible

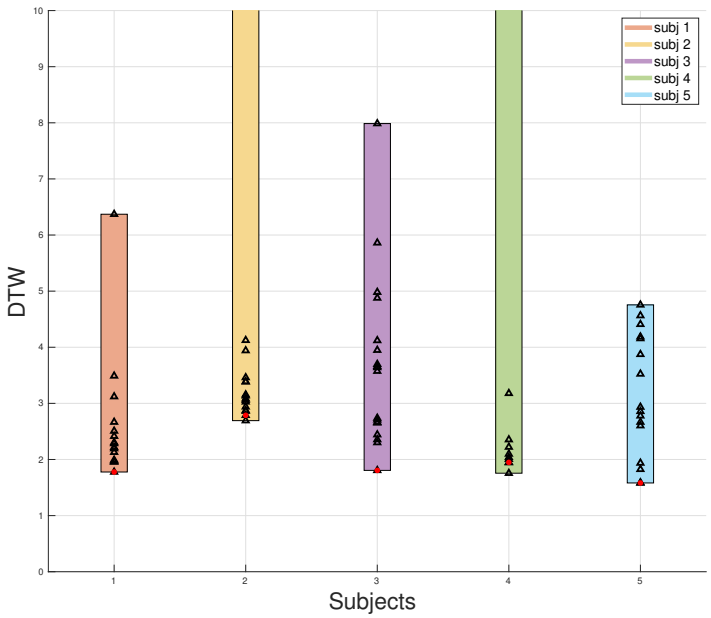


Figure 7.3: The DTW relative to various repetitions of the task with different parameters. In red is highlighted the preference expressed by the user. In three cases, the preference expressed coincides with the best result, while in the other two, the preference expressed is anyway very close.

for weighting the robot's effort in the cost function, high values of R mean low effort put into the task by the robot. So it makes sense that humans prefer high assistance for large, fast movements and very little assistance for small and precise motions. High assistance indeed means faster and easier reaching of the set point while can introduce oscillation in the case of small, precise motions.

An opposite behavior is observed for the parameter α, which is, in general, preferred low for precise path following and high for fast motions. The reason is the same as for the R. High values of alpha mean, in the end, higher assistance.

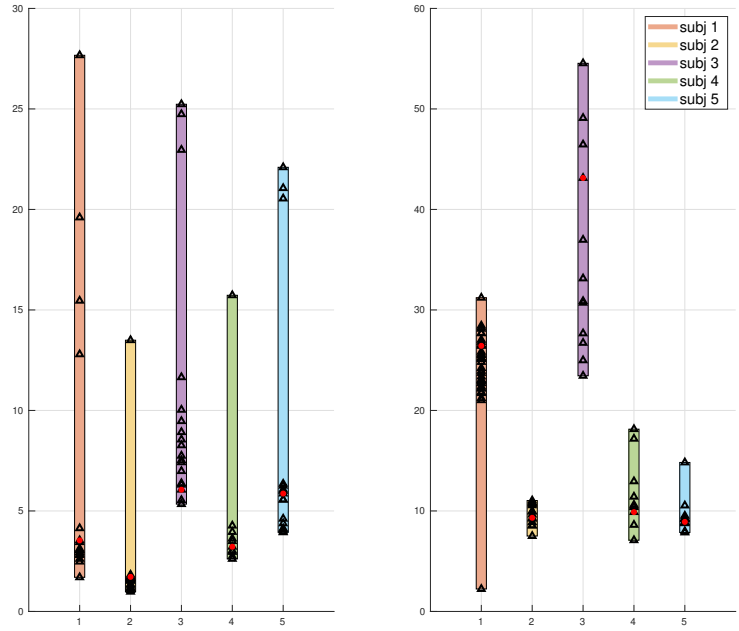


Figure 7.4: Forces measured. Comparison between the range tested and the preferred one. Not in all cases, the minimum force means the human preference (indicated in red).

Finally, for the $K_{h,p}$ parameter, low values are preferred for both tasks. This is due to the fact that too high values lead to jerky and unstable behavior since the set point is put too far and always overshoots the real desired target pose.

In conclusion, the questionnaire results are presented in table I for the two tasks. On average, users are pretty satisfied with the results obtained with the PBO procedure for both cases. Users also found the procedure not so long and not so tiring for both tasks. The only slight variation observed is in the length of the procedure, which was found to be a little shorter for the reaching case. This can be explained easily because the time required to perform a single reaching task was way shorter than a

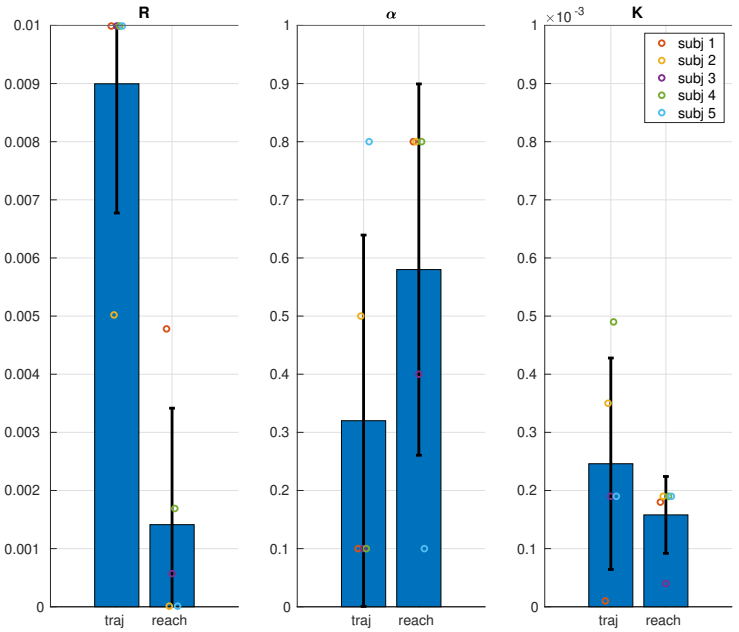


Figure 7.5: Mean optimal values found for the three parameters. On the left are the parameters related to the path-following task, on the right to the reaching task

complete path-following task.

	Long	Tiring	Satisfied
Path following	1.25	1.00	2.75
Reaching	0.75	1.00	2.75

Table 7.1: Questionnaire evaluation results.

7.2.2 CONCLUSION

This work presents a method for tuning an assistive game-theoretical cooperative controller for pHRI. The GLISp algorithm is used to perform PBO for two different sets of tasks. Results show that the procedure is

appreciated by the users and is not tiring. Moreover, it is shown that subjective human preferences are, in general, coherent with numerical performance indexes. It was also shown that for each subject, different sets of parameters are preferred for the different tasks, while similar results are shared between the same task. This suggests that it is possible to extract general preferences for tuning such controllers based on the task. To make this approach more general, online variations of such parameters will be considered in future works for different tasks. For instance, a complete task may require fast motion to a target pose and then precise path following, so parameters can be changed smoothly on-the-fly. In that case, optimization of the transition function will be investigated.

CHAPTER 8

CONCLUSIONS AND FUTURE WORKS

8.1 CONCLUSIONS

This thesis aims to implement controllers for physical Human–Robot Interaction (pHRI) by describing the interaction within a Differential/Dynamic Game-Theoretic (DGT) framework. A DG involves several components; some are imposed by the game, such as the system dynamics. Some others depend on each player, the mode of the game, and the information available. Therefore, this thesis, on the one hand, addresses the modeling of the human as a rational player to include it in a DGT framework; on the other hand, it investigates the robot's parameters tuning to make such an interaction smooth and comfortable for the human.

The model of the human represents a crucial piece of information to formulate the DGT problem. In 4.1, this thesis studies human behavior as a linear state-feedback controller. The human is modeled as a gain matrix that multiplies the system's state. Such a gain matrix, which in control theory is typically obtained by appropriate tuning procedure (e.g.,

Pole Placement, LQR), is unknown in the case of a human. Moreover, it is time-variant according to the actual human behavior. This thesis addresses the identification of such an unknown, time-variant gain matrix. Starting from observations of the interaction, an Extended Kalman Filter (EKF) is implemented to show the feasibility of online recovery of the matrix. Augmenting the state with the unknown control gains makes it possible to identify them online. Simulations and experiments involving a reaching task and time-varying control laws are proposed, comparing EKF with a modified Least Squares method (LS) and a Recursive Least Squares (RLS). The results show that the EKF approach performs better than the LS and RLS methods, particularly in identifying the gains in the time-variant cases. Finally, experimental results show that the EKF approach is more stable than LS and performs better than RLS and LS.

This study opens the way to Inverse Optimal Control (IOC) methodologies, which are also investigated in 4.2. Given that a human can be described as a linear-state feedback controller, it is possible also to answer the question, "How does a human obtain such a control law?". This question is investigated in 4.2. It is hypothesized that a human behaves as an optimal controller. Therefore, he/she has a cost function that wants to minimize. As a first approximation, this work assumes that the cost function of the human is quadratic in the state and control effort, leading to an LQR formulation. Since the objective is to recover the cost function, the problem is proposed as an Inverse Optimal Control (IOC) problem. Experiments are proposed to show that, under some approximation, even a simple quadratic cost function can describe human behavior. IOC is used to recover the cost functions of three subjects performing a reaching task interacting with different robot behaviors. The results show that even if a single cost function cannot capture the human complex behavior, there are certain relations between the cost parameters and different (in magnitude of motion) tasks. The most relevant result is that humans tend to

keep the required time as constant as possible when performing a task. The recovered cost function is used for the Game-Theoretical formulation of the interaction in the experimental section of this thesis, in 6 and 6.3.

Another fundamental piece of information to formulate the Game-Theoretical problem is the knowledge of the intention. This thesis defines human intention as the desired trajectory that the human wants to follow over a finite rolling prediction horizon. Chapter 4.3 presents a method for predicting the desired human trajectory over a finite time horizon. The model of the human is described with a Recurrent Neural Network (RNN) trained on real interacting data. The model needs an iterative training procedure to reduce the prediction error, which is time-consuming and does not generalize to different users or situations. To overcome this issue, we propose a Transfer Learning method that adapts the model to new users and situations. The approach is validated with real-world experiments. Different prediction horizons are also evaluated to show the dependency of the error. The time required by the different training steps is evaluated, showing that the TL approach reduces the time necessary to train the model. Finally, the assistive controller enhanced by the RNN+FC model is compared to standard controllers used in pHRI.

Once all the necessary blocks of the GT controllers are available, it is possible to formulate the Game-Theoretical problem for the Human–Robot interaction. This thesis proposes two different approaches, one time-continuous and the other time-discrete. The two models are presented in 5 and 5.3. The models are also deeply analyzed with simulations to show how they can be used according to different selections of the parameters that allow selection. The most relevant finding, which is the same for both models, is that the Cooperative formulation perfectly suits the situation where the robot should assist the human partner. Conversely, the Non-Cooperative formulation allows the robot to lead the

action when required.

6 proposes different GT-based controller implementations of the cooperative mode considered. The two control schemas, continuous and discrete, are used with the human intention estimation module to evaluate the assistive capabilities of the GT-based controllers compared with standard controllers, Manual Guidance (MG), and Impedance control (IC). Results show that the performances improve when using the assistance provided by the GT modeling. In the GT-dMPC case, results show that the proposed controller can provide better assistance in various tasks by reducing the required interaction force and allowing better precision in stable positioning around a target pose. The assistive component does not introduce any additional oscillation, allowing natural interaction with the human. The final subjective questionnaire shows that the users generally appreciate the proposed controller better than the other two.

Then, the Cooperative scenario is also studied for a trajectory following task, with role arbitration. The proposed controller is compared with other MG and IC controllers with different tuning parameters. The results show that being equal to the capability of the four controllers in following the nominal geometrical path under human-robot cooperation, the CGT controller outperforms the others in trajectory following. Moreover, even if the LIC requires less interaction force to accomplish the task, it shows the lowest capability in trajectory tracking, and the CGT also requires low interaction forces. As a remark, a similar interaction force can be obtained for the CGT controller by tuning the parameters differently. Interestingly, it can be noted that the critical damping value varies from lower than critical to higher than critical, according to the role adaptation. This effect allows the robot to have a more damped behavior when the human is leading, allowing smooth motion. In contrast, a low damping ratio allows better and faster trajectory tracking when the robot is

leading.

Finally, the Role Arbitration is achieved by dynamically switching between the Cooperative and the Non-Cooperative models. A Fuzzy Logic System defines an arbitration function capable of considering various dangerous situations to modify the arbitration parameter α. This arbitration function is a hybrid controller that behaves as empowering manual guidance when dangerous situations are not foreseen while moving to a behavior closer to impedance control when the robot approaches dangerous situations. Experiments show that the proposed controller outperforms the other two (MG and IC) in various situations.

The same approach is also used in an industrial use-case to co-manipulate large, flexible carbon fiber patches. In the application scenario, results show that with Role Arbitration, it is possible to reduce the number of errors during the execution of the collaborative transportation. The time required reduces dramatically compared to Manual guidance and also allows for lower deviations from a nominal trajectory.

This thesis's last chapter is dedicated to studying human preferences. All the proposed control schema and applications require tuning of a set of parameters that govern the robot behavior. Therefore, it is essential for a personalized interaction to allow users to tune such parameters according to their preferences. This thesis proposes Preference-Based Optimization (PBO) to tune the parameters according to human desire. Moreover, the set of parameters analysis shows that different sets should be used for different tasks. Two tasks are proposed to the users: one that requires precise path following and the other that requires fast teaching of a target. A general trend demonstrates that humans require different robot behaviors depending on the task.

8.2 FUTURE WORKS

Because of the main limitations due to the general human modeling, future works will investigate online Inverse Reinforcement Learning (IRL) and Inverse Optimal Control (IOC). In this way, it will be possible to recover the human cost function online, adapting the robot's behavior to the current status of the human, to different subjects, and so on.

Significant improvement will also be implemented within the GT-dMPC framework, the Role Arbitration mechanism. In doing this, an online motion planning module can bring significant improvement. Indeed, in this way, it will always be possible to compute, online at each time step, a collision-free connecting trajectory from the current status of the robot to the nominal pre-computed (offline) trajectory of the robot. So, when the robot requires control, it can reconnect with the initially planned trajectory in a safe way.

Finally, possible nonlinear modeling of human behavior and adopting more complex cost functions that consider other components than only the current state and the effort will give the robot more complete information and make the interaction more natural.

BIBLIOGRAPHY

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the twenty-first international conference on Machine learning, page 1, 2004.

[2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD Int Conf knowledge discovery & data mining, pages 2623–2631, 2019.

[3] M. B. Alatise and G. P. Hancke. A review on challenges of autonomous mobile robot and sensor fusion methods. IEEE Access, 8:39830–39846, 2020.

[4] A. Alili, V. Nalam, M. Li, M. Liu, J. Feng, J. Si, and H. Huang. A novel framework to facilitate user preferred tuning for a robotic knee prosthesis. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 31:895–903, 2023.

[5] A. Alili, V. Nalam, M. Li, M. Liu, J. Si, and H. H. Huang. User controlled interface for tuning robotic knee prosthesis. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6190–6195, 2021.

[6] M. Anvaripour, M. Khoshnam, C. Menon, and M. Saif. Fmg- and rnn-based estimation of motor intention of upper-limb motion in human-robot collaboration. Frontiers in Robotics and AI, 7, 2020.

[7] A. Attanasio, B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri. Autonomy in surgical robotics. Annual Review of Control, Robotics, and Autonomous Systems, 4(1):651–679, 2021.

[8] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet. A robust and sensitive metric for quantifying movement smoothness. IEEE transactions on biomedical engineering, 59(8):2126–2136, 2011.

[9] C. Bartneck, T. Belpaeme, F. Eyssel, T. Kanda, M. Keijsers, and S. Šabanović. Human-robot interaction: An introduction. Cambridge University Press, 2020.

[10] D. Bazzi, G. Priora, A. M. Zanchettin, and P. Rocco. Rrt* and goal-driven variable admittance control for obstacle avoidance in manual guidance applications. IEEE Robotics and Automation Letters, 7(2):1920–1927, 2022.

[11] T. Başar and G. J. Olsder. Dynamic Noncooperative Game Theory, 2nd Edition. Society for Industrial and Applied Mathematics, 1998.

[12] A. Bechar and C. Vigneault. Agricultural robots for field operations: Concepts and components. Biosystems Engineering, 149:94–111, 2016.

[13] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and F. Tanaka. Social robots for education: A review. Science Robotics, 3(21):eaat5954, 2018.

[14] A. Bemporad. Global optimization via inverse distance weighting and radial basis functions. arxiv, 8(1):1–18, 2020.

[15] A. Bemporad and D. Piga. Global optimization based on active preference learning with radial basis functions. Machine Learning, 110(2):417–448, 2021.

[16] W. Bi, X. Wu, Y. Liu, and Z. Li. Role adaptation and force, impedance learning for physical human-robot interaction. In 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM), pages 111–117, 2019.

[17] D. A. Braun, P. A. Ortega, and D. M. Wolpert. Nash equilibria in multi-agent motor interactions. PLOS Computational Biology, 5(8):1–8, 08 2009.

[18] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse. Collaborative manufacturing with physical human–robot interaction. Robotics and Computer-Integrated Manufacturing, 40:1–13, 2016.

[19] Y. Choi, M. Choi, M. Oh, and S. Kim. Service robots in hotels: understanding the service quality perceptions of human-robot interaction. Journal of Hospitality Marketing & Management, 29(6):613–635, 2020.

[20] E. Coronado, T. Kiyokawa, G. A. G. Ricardez, I. G. Ramirez-Alpizar, G. Venture, and N. Yamanobe. Evaluating quality in human-robot interaction: A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0. Journal of Manufacturing Systems, 63:392–410, 2022.

[21] S. Cremer, S. K. Das, I. B. Wijayasinghe, D. O. Popa, and F. L. Lewis. Model-free online neuroadaptive controller with intent estimation for physical human-robot interaction. IEEE Tran on Robotics, 36(1):240–253, 2020.

[22] I. Daniyan, V. Balogun, A. Adeodu, B. Oladapo, J. K. Peter, and K. Mpofu. Development and performance evaluation of a robot for lawn mowing. Procedia Manufacturing, 49:42–48, 2020. Proceedings of the 8th International Conference on Through-Life Engineering Services – TESConf 2019.

[23] E. S. de Lima and B. Feijó. Artificial intelligence in human-robot interaction. In H. Ayanoğlu and E. Duarte, editors, Emotional Design in Human-Robot Interaction: Theory, Methods and Applications, pages 187–199. Springer International Publishing, Cham, 2019.

[24] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi. An atlas of physical human–robot interaction. Mechanism and Machine Theory, 43(3):253–270, 2008.

[25] F. Dehais, E. A. Sisbot, R. Alami, and M. Causse. Physiological and subjective evaluation of a human–robot object hand-over task. Applied Ergonomics, 42(6):785–791, 2011.

[26] J. Dong, J. Xu, Q. Zhou, and S. Hu. Physical human–robot interaction force control method based on adaptive variable impedance. Journal of the Franklin Institute, 357(12):7864–7878, 2020.

[27] A. D. Dragan, S. Bauman, J. Forlizzi, and S. S. Srinivasa. Effects of robot motion on human-robot collaboration. In 2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 51–58, 2015.

[28] V. Duchaine, B. Mayer St-Onge, D. Gao, and C. Gosselin. Stable and intuitive control of an intelligent assist device. IEEE Transactions on Haptics, 5(2):148–159, 2012.

[29] J. Engwerda. LQ dynamic optimization and differential games. John Wiley & Sons, 2005.

[30] J. Engwerda. Algorithms for computing nash equilibria in deterministic lq games. Computational Management Science, 4(2):113–140, 2007.

[31] J. Engwerda. The regular convex cooperative linear quadratic control problem. Automatica, 44(9):2453–2457, 2008.

[32] C. Fang, G. Rigano, N. Kashiri, A. Ajoudani, J. Lee, and N. Tsagarakis. Online joint stiffness transfer from human arm to anthropomorphic arm. In IEEE Int Conf Sys, Man, and Cyb (SMC), pages 1457–1464, 2018.

[33] I. A. Faruque, F. T. Muijres, K. M. Macfarlane, A. Kehlenbeck, and J. S. Humbert. Identification of optimal feedback control rules from micro-quadrotor and insect flight trajectories. Biological cybernetics, 112(3):165–179, 2018.

[34] F. Ficuciello, L. Villani, and B. Siciliano. Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. IEEE Transactions on Robotics, 31(4):850–863, 2015.

[35] P. Franceschi, F. Bertini, F. Braghin, L. Roveda, N. Pedrocchi, and M. Beschi. Predicting human motion intention for phri assistive control, 2023.

[36] P. Franceschi, M. Beschi, N. Pedrocchi, and A. Valente. Modeling and analysis of phri with differential game theory, 2023.

[37] P. Franceschi and N. Castaman. Combining visual and force feedback for the precise robotic manipulation of bulky components. In E. Stella, editor, Multimodal Sensing and Artificial Intelligence: Technologies and Applications II, volume 11785, page 1178510. International Society for Optics and Photonics, SPIE, 2021.

[38] P. Franceschi, N. Castaman, S. Ghidoni, and N. Pedrocchi. Precise robotic manipulation of bulky components. IEEE Access, 8:222476–222485, 2020.

[39] P. Franceschi, S. Mutti, and N. Pedrocchi. Optimal design of robotic work-cell through hierarchical manipulability maximization. Robotics and Computer-Integrated Manufacturing, 78:102401, 2022.

[40] P. Franceschi, N. Pedrocchi, and M. Beschi. Adaptive impedance controller for human-robot arbitration based on cooperative differential game theory. In 2022 International Conference on Robotics and Automation (ICRA), pages 7881–7887, 2022.

[41] P. Franceschi, N. Pedrocchi, and M. Beschi. Inverse optimal control for the identification of human objective: a preparatory study for physical human-robot interaction. In 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), pages 1–6, 2022.

[42] P. Franceschi, N. Pedrocchi, and M. Beschi. Identification of human control law during physical human–robot interaction. Mechatronics, 92:102986, 2023.

[43] R. Galin and R. Meshcheryakov. Review on human–robot interaction during collaboration in a shared workspace. In Interactive Collaborative Robotics: 4th International Conference, ICR 2019, Istanbul, Turkey, August 20–25, 2019, Proceedings 4, pages 63–74. Springer, 2019.

[44] S. S. Ge, Y. Li, and H. He. Neural-network-based human intention estimation for physical human-robot interaction. In 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pages 390–395, 2011.

[45] S. Haddadin and E. Croft. Physical Human–Robot Interaction, pages 1835–1874. Springer International Publishing, Cham, 2016.

[46] W. He, C. Xue, X. Yu, Z. Li, and C. Yang. Admittance-based controller design for physical human–robot interaction in the constrained task space. IEEE Transactions on Automation Science and Engineering, 17(4):1937–1949, 2020.

[47] A. Hentout, M. Aouache, A. Maoudj, and I. Akli. Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017. Advanced Robotics, 33(15-16):764–799, 2019.

[48] J. P. Hespanha. Noncooperative Games, pages 3–12. Princeton University Press, 2017.

[49] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

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

[51] N. Hogan. Impedance Control: An Approach to Manipulation: Part I—Theory. Journal of Dynamic Systems, Measurement, and Control, 107(1):1–7, 03 1985.

[52] Y. Huo, X. Li, X. Zhang, and D. Sun. Intention-driven variable impedance control for physical human-robot interaction. In IEEE/ASME Int Conf Adv Intell Mechatronics (AIM), pages 1220–1225, 2021.

[53] R. Ikeura, T. Moriguchi, and K. Mizutani. Optimal variable impedance control for a robot and its application to lifting an object

with a human. In Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication, pages 500–505, 2002.

[54] M. Ilami, H. Bagheri, R. Ahmed, E. O. Skowronek, and H. Marvi. Materials, actuators, and sensors for soft bioinspired robots. Advanced Materials, 33(19):2003139, 2021.

[55] J. Inga, E. Bischoff, T. L. Molloy, M. Flad, and S. Hohmann. Solution sets for inverse non-cooperative linear-quadratic differential games. IEEE Control Systems Letters, 3(4):871–876, 2019.

[56] J. Inga, A. Creutz, and S. Hohmann. Online inverse linear-quadratic differential games applied to human behavior identification in shared control. In 2021 European Control Conference (ECC), pages 353–360, 2021.

[57] J. Inga, M. Eitel, M. Flad, and S. Hohmann. Evaluating human behavior in manual and shared control via inverse optimization. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 2699–2704, 2018.

[58] J. Inga, M. Flad, and S. Hohmann. Validation of a human cooperative steering behavior model based on differential games. In 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pages 3124–3129, 2019.

[59] ISO 10218-1:2011. Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots. Standard, International Organization for Standardization, Geneva, Switzerland, 2011.

[60] ISO 10218-2:2011. Robots and robotic devices — Safety requirements for industrial robots — Part 2: Robot systems and integra-

tion. Standard, International Organization for Standardization, Geneva, Switzerland, 2011.

[61] ISO/TS 15066:2016. Robots and robotic devices — Collaborative robots. Standard, International Organization for Standardization, Geneva, Switzerland, 2016.

[62] R. Jahanmahin, S. Masoud, J. Rickli, and A. Djuric. Human-robot interactions in manufacturing: A survey of human behavior modeling. Robotics and Computer-Integrated Manufacturing, 78:102404, 2022.

[63] N. Jarrassé, T. Charalambous, and E. Burdet. A framework to describe, analyze and generate interactive motor behaviors. PLOS ONE, 7(11):1–13, 11 2012.

[64] F. Jean and S. Maslovskaya. Inverse optimal control problem: the linear-quadratic case. In 2018 IEEE Conference on Decision and Control (CDC), pages 888–893, 2018.

[65] A. Ji and D. Levinson. A review of game theory models of lane changing. Transportmetrica A: transport science, 16(3):1628–1647, 2020.

[66] Z. Ji, Q. Liu, W. Xu, Z. Liu, B. Yao, B. Xiong, and Z. Zhou. Towards shared autonomy framework for human-aware motion planning in industrial human-robot collaboration. In 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), pages 411–417, 2020.

[67] W. Jin, D. Kulić, J. F.-S. Lin, S. Mou, and S. Hirche. Inverse optimal control for multiphase cost functions. IEEE Transactions on Robotics, 35(6):1387–1398, 2019.

[68] E. Kalai. Proportional solutions to bargaining situations: interpersonal utility comparisons. Econometrica: Journal of the Econometric Society, pages 1623–1630, 1977.

[69] E. Kalai and M. Smorodinsky. Other solutions to nash's bargaining problem. Econometrica: Journal of the Econometric Society, pages 513–518, 1975.

[70] T. Kanda and H. Ishiguro. Human-robot interaction in social robotics. CRC Press, 2017.

[71] J. Kim, H. Ladjal, D. Folio, A. Ferreira, and J. Kim. Evaluation of telerobotic shared control strategy for efficient single-cell manipulation. IEEE Transactions on Automation Science and Engineering, 9(2):402–406, 2012.

[72] J. Kim, A. K. Mishra, R. Limosani, M. Scafuro, N. Cauli, J. Santos-Victor, B. Mazzolai, and F. Cavallo. Control strategies for cleaning robots in domestic applications: A comprehensive review. International Journal of Advanced Robotic Systems, 16(4):1729881419857432, 2019.

[73] S. Ko and R. Langari. Shared control between human driver and automation in cooperative driving based on game theoretic model predictive control. In Dynamic Systems and Control Conference, volume 59162, page V003T18A004. American Society of Mechanical Engineers, 2019.

[74] S. Ko and R. Langari. Shared control between human driver and machine based on game theoretical model predictive control framework. In 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pages 649–654. IEEE, 2020.

[75] S. Ko and R. Langari. Shared control between human driver and machine based on game theoretical model predictive control framework. In 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pages 649–654, 2020.

[76] Y. Kobayashi, T. Sugimoto, K. Tanaka, Y. Shimomura, F. J. Arjonilla Garcia, C. H. Kim, H. Yabushita, and T. Toda. Robot navigation based on predicting of human interaction and its reproducible evaluation in a densely crowded environment. International Journal of Social Robotics, pages 1–15, 2022.

[77] F. Köpf, J. Inga, S. Rothfuß, M. Flad, and S. Hohmann. Inverse reinforcement learning for identification in linear-quadratic dynamic games. IFAC-PapersOnLine, 50(1):14902–14908, 2017.

[78] T. Kopp, M. Baumgartner, and S. Kinkel. Success factors for introducing industrial human-robot interaction in practice: an empirically driven framework. The International Journal of Advanced Manufacturing Technology, 112:685–704, 2021.

[79] K. Kronander and A. Billard. Stability considerations for variable impedance control. IEEE Transactions on Robotics, 32(5):1298–1305, 2016.

[80] J. Krüger, T. Lien, and A. Verl. Cooperation of human and machines in assembly lines. CIRP Annals, 58(2):628–646, 2009.

[81] H. Kwakernaak and R. Sivan. Linear optimal control systems, volume 1072. Wiley-interscience, 1969.

[82] F. Köpf, J. Inga, S. Rothfuß, M. Flad, and S. Hohmann. Inverse reinforcement learning for identification in linear-quadratic dynamic games. IFAC-PapersOnLine, 50(1):14902–14908, 2017. 20th IFAC World Congress.

[83] B. Lacevic, A. M. Zanchettin, and P. Rocco. Safe human-robot collaboration via collision checking and explicit representation of danger zones. IEEE Transactions on Automation Science and Engineering, 20(2):846–861, 2023.

[84] A. Lecours, B. Mayer-St-Onge, and C. Gosselin. Variable admittance control of a four-degree-of-freedom intelligent assist device. In 2012 IEEE International Conference on Robotics and Automation, pages 3903–3908, 2012.

[85] P. Leica, F. Roberti, M. Monllor, J. M. Toibero, and R. Carelli. Control of bidirectional physical human–robot interaction based on the human intention. Intelligent Service Robotics, 10(1):31–40, 2017.

[86] H.-Y. Li, I. Paranawithana, L. Yang, and U.-X. Tan. Variable admittance control with robust adaptive velocity control for dynamic physical interaction between robot, human and environment. In 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), pages 2299–2306, 2021.

[87] M. Li, H. Cao, X. Song, Y. Huang, J. Wang, and Z. Huang. Shared control driver assistance system based on driving intention and situation assessment. IEEE Transactions on Industrial Informatics, 14(11):4982–4994, 2018.

[88] S. Li, M. Bowman, and X. Zhang. A general arbitration model for robust human-robot shared control with multi-source uncertainty modeling. arXiv preprint arXiv:2003.05097, 2020.

[89] Y. Li, G. Carboni, F. Gonzalez, D. Campolo, and E. Burdet. Differential game theory for versatile physical human–robot interaction. Nature Machine Intelligence, 1(1):36–43, Jan 2019.

[90] Y. Li, J. Eden, G. Carboni, and E. Burdet. Improving tracking through human-robot sensory augmentation. IEEE Robotics and Automation Letters, 5(3):4399–4406, 2020.

[91] Y. Li and S. S. Ge. Human–robot collaboration based on motion intention estimation. IEEE/ASME Transactions on Mechatronics, 19(3):1007–1014, 2014.

[92] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu. Continuous role adaptation for human–robot shared control. IEEE Transactions on Robotics, 31(3):672–681, 2015.

[93] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu. Continuous role adaptation for human–robot shared control. IEEE Transactions on Robotics, 31(3):672–681, 2015.

[94] Y. Li, K. P. Tee, R. Yan, W. L. Chan, and Y. Wu. A framework of human–robot coordination based on game theory and policy iteration. IEEE Transactions on Robotics, 32(6):1408–1418, 2016.

[95] Y. Li, K. P. Tee, R. Yan, W. L. Chan, Y. Wu, and D. K. Limbu. Adaptive optimal control for coordination in physical human-robot interaction. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 20–25, 2015.

[96] Z. Li, Q. Xu, and L. M. Tam. A survey on techniques and applications of window-cleaning robots. IEEE Access, 9:111518–111532, 2021.

[97] J. F.-S. Lin, V. Bonnet, A. M. Panchea, N. Ramdani, G. Venture, and D. Kulić. Human motion segmentation using cost weights recovered from inverse optimal control. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pages 1107–1113, 2016.

[98] C. Lindig-León, G. Schmid, and D. A. Braun. Nash equilibria in human sensorimotor interactions explained by q-learning with intrinsic costs. Scientific reports, 11(1):1–15, 2021.

[99] Y. Long, Z.-j. Du, W.-d. Wang, and W. Dong. Human motion intent learning based motion assistance control for a wearable exoskeleton. Robotics and Computer-Integrated Manufacturing, 49:317–327, 2018.

[100] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley. A Review of Intent Detection, Arbitration, and Communication Aspects of Shared Control for Physical Human–Robot Interaction. Applied Mechanics Reviews, 70(1), 02 2018. 010804.

[101] D. P. Losey and M. K. O'Malley. Trajectory deformations from physical human–robot interaction. IEEE Transactions on Robotics, 34(1):126–138, 2018.

[102] W. Lu, Z. Hu, and J. Pan. Human-robot collaboration using variable admittance control and human intention prediction. In 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), pages 1116–1121, 2020.

[103] J. Luo, D. Huang, Y. Li, and C. Yang. Trajectory online adaption based on human motion prediction for teleoperation. IEEE Transactions on Automation Science and Engineering, 19(4):3184–3191, 2022.

[104] J. Mainprice, R. Hayne, and D. Berenson. Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces. IEEE Transactions on Robotics, 32(4):897–908, 2016.

[105] H. Maithani, J. A. C. Ramon, and Y. Mezouar. Predicting human intent for cooperative physical human-robot interaction tasks. In IEEE Int Conf Contr and Aut (ICCA), pages 1523–1528, 2019.

[106] B. Mazzolai and C. Laschi. A vision for future bioinspired and biohybrid robots. Science Robotics, 5(38):eaba6893, 2020.

[107] J. R. Medina, S. Endo, and S. Hirche. Impedance-based gaussian processes for predicting human behavior during physical interaction. In IEEE Int Conf Rob and Aut (ICRA), pages 3055–3061, 2016.

[108] M. Menner and M. N. Zeilinger. Convex formulations and algebraic solutions for linear quadratic inverse optimal control problems. In 2018 European Control Conference (ECC), pages 2107–2112, 2018.

[109] H.-S. Moon and J. Seo. Prediction of human trajectory following a haptic robotic guide using recurrent neural networks. In 2019 IEEE World Haptics Conference (WHC), pages 157–162. IEEE, 2019.

[110] M. Mujica, M. Crespo, M. Benoussaad, S. Junco, and J.-Y. Fourquet. Robust variable admittance control for human–robot comanipulation of objects with unknown load. Robotics and Computer-Integrated Manufacturing, 79:102408, 2023.

[111] M. Mulder, D. M. Pool, D. A. Abbink, E. R. Boer, P. M. Zaal, F. M. Drop, K. van der El, and M. M. van Paassen. Manual control cybernetics: State-of-the-art and current trends. IEEE Transactions on Human-Machine Systems, 48(5):468–485, 2017.

[112] M. Müller. Dynamic time warping. Information retrieval for music and motion, pages 69–84, 2007.

[113] S. L. Müller-Abdelrazeq, K. Schönefeld, M. Haberstroh, and F. Hees. Interacting with collaborative robots—a study on attitudes

and acceptance in industrial contexts. Social robots: Technological, societal and ethical aspects of human-robot interaction, pages 101–117, 2019.

[114] S. Musić and S. Hirche. Haptic shared control for human-robot collaboration: A game-theoretical approach. IFAC-PapersOnLine, 53(2):10216–10222, 2020. 21st IFAC World Congress.

[115] S. Mutti, G. Dimauro, and N. Pedrocchi. Kinematic-aware UKF-based fast fiducial marker tracker. In E. Stella, F. Soldovieri, D. Ceglarek, and Q. Kemao, editors, Multimodal Sensing and Artificial Intelligence: Technologies and Applications III, volume 12621, page 1262115. International Society for Optics and Photonics, SPIE, 2023.

[116] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche. The role of roles: Physical cooperation between humans and robots. The International Journal of Robotics Research, 31(13):1656–1674, 2012.

[117] F. Müller, J. Janetzky, U. Behrnd, J. Jäkel, and U. Thomas. User force-dependent variable impedance control in human-robot interaction. In 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), pages 1328–1335, 2018.

[118] X. Na and D. Cole. Theoretical and experimental investigation of driver noncooperative-game steering control behavior. IEEE/CAA Journal of Automatica Sinica, 8(1):189–205, 2021.

[119] X. Na and D. J. Cole. Linear quadratic game and non-cooperative predictive methods for potential application to modelling driver–afs interactive steering control. Vehicle system dynamics, 51(2):165–198, 2013.

[120] X. Na and D. J. Cole. Game-theoretic modeling of the steering interaction between a human driver and a vehicle collision avoidance controller. IEEE T HUM-MACH SYST, 45:25–38, 2015.

[121] X. Na and D. J. Cole. Modelling of a human driver's interaction with vehicle automated steering using cooperative game theory. IEEE/CAA Journal of Automatica Sinica, 6(5):1095–1107, 2019.

[122] X. Na and D. J. Cole. Experimental evaluation of a game-theoretic human driver steering control model. IEEE Tran on Cyb, pages 1–14, 2022.

[123] J. F. Nash. The bargaining problem. Econometrica, 18(2):155–162, 1950.

[124] G. Nicola, E. Villagrossi, and N. Pedrocchi. Human-robot co-manipulation of soft materials: enable a robot manual guidance using a depth map feedback. In 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pages 498–504, 2022.

[125] G. Nicola, E. Villagrossi, and N. Pedrocchi. Co-manipulation of soft-materials estimating deformation from depth images. Robotics and Computer-Integrated Manufacturing, 85:102630, 2024.

[126] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco. Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots. IEEE Robotics and Automation Letters, 3(2):796–803, 2018.

[127] D. Nicolis, A. M. Zanchettin, and P. Rocco. Human intention estimation based on neural networks for enhanced collaboration with robots. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1326–1333, 2018.

[128] M. A. K. Niloy, A. Shama, R. K. Chakrabortty, M. J. Ryan, F. R. Badal, Z. Tasneem, M. H. Ahamed, S. I. Moyeen, S. K. Das, M. F. Ali, M. R. Islam, and D. K. Saha. Critical design and control issues of indoor autonomous mobile robots: A review. IEEE Access, 9:35338–35370, 2021.

[129] O. S. Oguz, Z. Zhou, S. Glasauer, and D. Wollherr. An inverse optimal control approach to explain human arm reaching control based on multiple internal models. Scientific reports, 8(1):1–17, 2018.

[130] R. Palm, R. Chadalavada, and A. J. Lilienthal. Recognition of human-robot motion intentions by trajectory observation. In 2016 9th International Conference on Human System Interactions (HSI), pages 229–235, 2016.

[131] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish. Recent progress on programming methods for industrial robots. Robotics and Computer-Integrated Manufacturing, 28(2):87–94, 2012.

[132] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani. Robot adaptation to human physical fatigue in human–robot comanipulation. Autonomous Robots, 42:1011–1021, 2018.

[133] B. S. Peters, P. R. Armijo, C. Krause, S. A. Choudhury, and D. Oleynikov. Review of emerging surgical robotic technology. Surgical endoscopy, 32:1636–1655, 2018.

[134] T. Petković, D. Puljiz, I. Marković, and B. Hein. Human intention estimation based on hidden markov model motion validation for safe flexible robotized warehouses. Robotics and Computer-Integrated Manufacturing, 57:182–196, 2019.

[135] J. W. Polderman and J. C. Willems. Pole placement by state feed-

back. In Introduction to Mathematical Systems Theory, pages 311–339. Springer, 1998.

[136] A. Popovici, P. Zaal, and D. M. Pool. Dual extended kalman filter for the identification of time-varying human manual control behavior. In AIAA Modeling and Simulation Technologies Conference, chapter Human Factors, Perception, and Cueing, pages 1–17. AIAA, 2017.

[137] A. Popovici, P. Zaal, and D. M. Pool. Dual extended kalman filter for the identification of time-varying human manual control behavior. In AIAA Modeling and Simulation Technologies Conference, page 3666, 2017.

[138] E. Prati, M. Peruzzini, M. Pellicciari, and R. Raffaeli. How to include user experience in the design of human-robot interaction. Robotics and Computer-Integrated Manufacturing, 68:102072, 2021.

[139] E. Prati, V. Villani, F. Grandi, M. Peruzzini, and L. Sabattini. Use of interaction design methodologies for human–robot collaboration in industrial scenarios. IEEE Transactions on Automation Science and Engineering, 19(4):3126–3138, 2022.

[140] M. C. Priess, R. Conway, J. Choi, J. M. Popovich, and C. Radcliffe. Solutions to the inverse lqr problem with application to biological systems analysis. IEEE Transactions on Control Systems Technology, 23(2):770–777, 2015.

[141] I. Ranatunga, S. Cremer, D. O. Popa, and F. L. Lewis. Intent aware adaptive admittance control for physical human-robot interaction. In IEEE Int Conf Rob and Aut (ICRA), pages 5635–5640, 2015.

[142] J. Rawlings, D. Mayne, and M. Diehl. Model Predictive Control: Theory, Computation, and Design. Nob Hill Publishing, 2017.

[143] J. Rojer, D. M. Pool, M. M. van Paassen, and M. Mulder. Ukf-based identification of time-varying manual control behaviour. IFAC-PapersOnLine, 52(19):109–114, 2019. 14th IFAC Symposium on Analysis, Design, and Evaluation of Human Machine Systems HMS 2019.

[144] S. Rothfuß, J. Inga, F. Köpf, M. Flad, and S. Hohmann. Inverse optimal control for identification in non-cooperative differential games. IFAC-PapersOnLine, 50(1):14909–14915, 2017. 20th IFAC World Congress.

[145] L. Roveda, N. Castaman, S. Ghidoni, P. Franceschi, N. Boscolo, E. Pagello, and N. Pedrocchi. Human-robot cooperative interaction control for the installation of heavy and bulky components. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 339–344, 2018.

[146] L. Roveda, S. Haghshenas, M. Caimmi, N. Pedrocchi, and L. Molinari Tosatti. Assisting operators in heavy industrial tasks: On the design of an optimized cooperative impedance fuzzy-controller with embedded safety rules. Frontiers in Robotics and AI, 6:75, 2019.

[147] L. Roveda, S. Haghshenas, A. Prini, T. Dinon, N. Pedrocchi, F. Braghin, and L. M. Tosatti. Fuzzy impedance control for enhancing capabilities of humans in onerous tasks execution. In 2018 15th International Conference on Ubiquitous Robots (UR), pages 406–411, 2018.

[148] L. Roveda, B. Maggioni, E. Marescotti, A. A. Shahid, A. Maria Zanchettin, A. Bemporad, and D. Piga. Pairwise

preferences-based optimization of a path-based velocity planner in robotic sealing tasks. IEEE Robotics and Automation Letters, 6(4):6632–6639, 2021.

[149] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L. Molinari Tosatti, and N. Pedrocchi. Model-based reinforcement learning variable impedance control for human-robot collaboration. Journal of Intelligent & Robotic Systems, 100(2):417–433, Nov 2020.

[150] L. Roveda, A. Testa, A. A. Shahid, F. Braghin, and D. Piga. Q-learning-based model predictive variable impedance control for physical human-robot collaboration. Art Intell, 312:32, 2022.

[151] L. Roveda, P. Veerappan, M. Maccarini, G. Bucca, A. Ajoudani, and D. Piga. A human-centric framework for robotic task learning and optimization. Journal of Manufacturing Systems, 67:68–79, 2023.

[152] S. Saeedvand, M. Jafari, H. S. Aghdasi, and J. Baltes. A comprehensive survey on humanoid robot development. The Knowledge Engineering Review, 34:e20, 2019.

[153] S. Saunderson and G. Nejat. How robots influence humans: A survey of nonverbal communication in social human–robot interaction. International Journal of Social Robotics, 11:575–608, 2019.

[154] J. Schmidtler, K. Bengler, F. Dimeas, and A. Campeau-Lecours. A questionnaire for the evaluation of physical assistive devices (quead): Testing usability and acceptance in physical human-robot interaction. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 876–881, 2017.

[155] A. Scibilia, N. Pedrocchi, and L. Fortuna. Human control model estimation in physical human;machine interaction: A survey. Sensors, 22(5), 2022.

[156] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano. Autonomy in physical human-robot interaction: A brief survey. IEEE Robotics and Automation Letters, 6(4):7989–7996, 2021.

[157] T. Seo, Y. Jeon, C. Park, and J. Kim. Survey on glass and façade-cleaning robots: Climbing mechanisms, cleaning methods, and applications. International Journal of Precision Engineering and Manufacturing-Green Technology, 6:367–376, 2019.

[158] M. Sharifi, S. Behzadipour, and G. Vossoughi. Model reference adaptive impedance control in cartesian coordinates for physical human–robot interaction. Advanced Robotics, 28(19):1277–1290, 2014.

[159] M. Sharifi, A. Zakerimanesh, J. K. Mehr, A. Torabi, V. K. Mushahwar, and M. Tavakoli. Impedance variation and learning strategies in human–robot interaction. IEEE Transactions on Cybernetics, 52(7):6462–6475, 2022.

[160] A. Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena, 404:132306, 2020.

[161] F. Sherwani, M. M. Asad, and B. Ibrahim. Collaborative robots and industrial revolution 4.0 (ir 4.0). In 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), pages 1–5, 2020.

[162] F. Sherwani, M. M. Asad, and B. S. K. K. Ibrahim. Collaborative robots and industrial revolution 4.0 (ir 4.0). In 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), pages 1–5. IEEE, 2020.

[163] B. Siciliano and L. Villani. Robot force control. 2000.

[164] D. Sirintuna, A. Giammarino, and A. Ajoudani. An object deformation-agnostic framework for human–robot collaborative transportation. IEEE Transactions on Automation Science and Engineering, pages 1–14, 2023.

[165] D. Sirintuna, I. Ozdamar, Y. Aydin, and C. Basdogan. Detecting human motion intention during phri using artificial neural networks trained by emg signals. In IEEE Int Conf Robot and Human Interactive Communication (RO-MAN), pages 1280–1287, 2020.

[166] G. L. Smith, S. F. Schmidt, and L. A. McGee. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle, volume 135. National Aeronautics and Space Administration, 1962.

[167] A. Takagi, G. Ganesh, T. Yoshioka, M. Kawato, and E. Burdet. Physically interacting individuals estimate the partner's goal to enhance their movements. Nature Human Behaviour, 1(3):1–6, 2017.

[168] M. J. Tsai and Y. H. Chiou. Manipulability of manipulators. Mechanism and Machine Theory, 25(5):575 – 585, 1990.

[169] F. Vicentini. Collaborative Robotics: A Survey. Journal of Mechanical Design, 143(4):040802, 10 2020.

[170] F. Vicentini. Terminology in safety of collaborative robotics. Robotics and Computer-Integrated Manufacturing, 63:101921, 2020.

[171] V. Villani, F. Pini, F. Leali, and C. Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics, 55:248–266, 2018.

[172] Y. Wang, Y. Sheng, J. Wang, and W. Zhang. Optimal collision-free robot trajectory generation based on time series prediction of human motion. IEEE Robotics and Automation Letters, 3:226–233, 2018.

[173] K. Westermann, J. F.-S. Lin, and D. Kulić. Inverse optimal control with time-varying objectives: application to human jumping movement analysis. Scientific reports, 10(1):1–15, 2020.

[174] H.-N. Wu. Online learning human behavior for a class of human-in-the-loop systems via adaptive inverse optimal control. IEEE Transactions on Human-Machine Systems, 52(5):1004–1014, 2022.

[175] X. Xing, K. Maqsood, D. Huang, C. Yang, and Y. Li. Iterative learning-based robotic controller with prescribed human–robot interaction force. IEEE Transactions on Automation Science and Engineering, 19(4):3395–3408, 2022.

[176] Y. Xu, H. Zhang, L. Cao, X. Shu, and D. Zhang. A shared control strategy for reach and grasp of multiple objects using robot vision and noninvasive brain–computer interface. IEEE Transactions on Automation Science and Engineering, 19(1):360–372, 2022.

[177] B. Yao, Z. Zhou, L. Wang, W. Xu, Q. Liu, and A. Liu. Sensorless and adaptive admittance control of industrial robot in physical human-robot interaction. Robotics and Computer-Integrated Manufacturing, 51:158–168, 2018.

[178] A. Zacharaki, I. Kostavelis, A. Gasteratos, and I. Dokas. Safety bounds in human robot interaction: A survey. Safety Science, 127:104667, 2020.

[179] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias. Safety in human-robot collaborative manufacturing environments:

Metrics and control. IEEE Transactions on Automation Science and Engineering, 13(2):882–893, 2016.

[180] J. Zhang, H. Liu, Q. Chang, L. Wang, and R. X. Gao. Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. CIRP annals, 69(1):9–12, 2020.

[181] X. Zhao, S. Chumkamon, S. Duan, J. Rojas, and J. Pan. Collaborative human-robot motion generation using lstm-rnn. In IEEE-RAS Int Conf Humanoid Robots (Humanoids), pages 1–9, 2018.

[182] M. Zhu, A. Bemporad, and D. Piga. Preference-based mpc calibration. In 2021 European Control Conference (ECC), pages 638–645, 2021.

[183] M. Zhu, D. Piga, and A. Bemporad. C-glisp: Preference-based global optimization under unknown constraints with applications to controller calibration. IEEE Transactions on Control Systems Technology, 30(5):2176–2187, 2022.

[184] R. Zou, Y. Liu, J. Zhao, and H. Cai. A framework for human-robot-human physical interaction based on n-player game theory. Sensors, 20(17), 2020.

*

APPENDIX A

APPENDIX

A.1 CONTINUOUS TO DISCRETE CONVERSION

Define the discretization interval as δt. This value is typically imposed by the controlled system. In most of the experiments in this thesis, $\delta t = 0.008s$ because the UR5 used is controlled at a frequency of 125Hz, which corresponds to 0.0008 s. The matrices' conversion from continuous to discrete is shown below.

$$A_d = (I + 0.5\,A\,\delta t)\;(I - 0.5\,A\,\delta t)^{-1} \tag{A.1}$$

and

$$Bd = A^{-1}\;(A_d - I) * B; \tag{A.2}$$

A.2 EKF MATRICES

In the following, the matrices used for the EKF implementation are provided for the one-DoF case. The observation function h is defined as

$$h(\zeta, w) = \begin{bmatrix} x + w_x \\ \dot{x} + w_{\dot{x}} \end{bmatrix} \tag{A.3}$$

The matrix Aa is computed as

$$Aa(\zeta, \nu) = \frac{\partial f(\zeta, \nu)}{\partial \zeta}\Big|_{\hat{\zeta}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-(k+k_{1,h})}{m} & \frac{-(d+k_{2,h})}{m} & \frac{x_{ref}-x}{m} & \frac{-\dot{x}}{m} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.4}$$

The matrix H is

$$H(\zeta, w) = \frac{\partial h(\zeta, w)}{\partial w}\Big|_{\hat{\zeta}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{A.5}$$

In the 3-DoFs case, such as in the simulation, the matrix A_a becomes

$$Aa(\zeta, \nu) = \frac{\partial f(\zeta, \nu)}{\partial \zeta}\Big|_{\hat{\zeta}} =$$

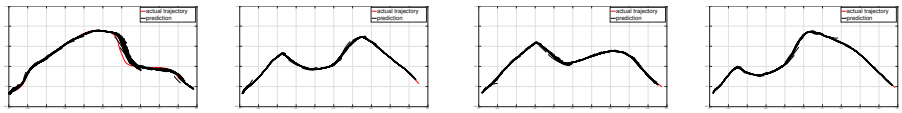
$$\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{-(k+k_{1,x})}{m} & 0 & 0 & \frac{-(d+k_{2,x})}{m} & 0 & 0 & \frac{x_{ref}-x}{m} & 0 & 0 & \frac{-\dot{x}}{m} & 0 & 0 \\
0 & \frac{-(k+k_{1,y})}{m} & 0 & 0 & \frac{-(d+k_{2,y})}{m} & 0 & 0 & \frac{y_{ref}-y}{m} & 0 & 0 & \frac{-\dot{y}}{m} & 0 \\
0 & 0 & \frac{-(k+k_{1,\theta})}{j} & 0 & 0 & \frac{-(d+k_{2,\theta})}{j} & 0 & 0 & \frac{\theta_{ref}-\theta}{j} & 0 & 0 & \frac{-\omega}{j} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{A.6}$$

A.3 ADDITIONAL RESULTS HUMAN MOTION INTENTION PREDICTION

This section presents additional results relative to the training of the RNN+FC model.

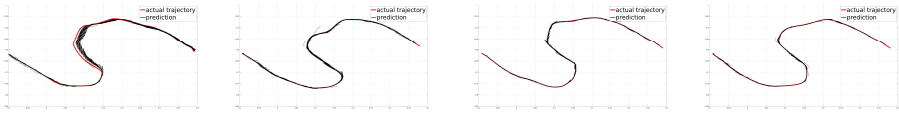
A.3.1 ITERATIVE TRAINING

Recalling the method and results presented in 4.3, it follows some results relative to the trajectories 2 and 3. Note that the various models $\mathcal{M}_K$ are trained on the full set composed of the three trajectories. Therefore, the model is the same for each iteration. Prediction of the four models for the trajectories 2 and 3 are visible in figures A.1 and A.2, respectively.



(a) $\mathcal{M}_0$ **(b)** $\mathcal{M}_1$ **(c)** $\mathcal{M}_2$ **(d)** $\mathcal{M}_3$

Figure A.1: Iterative training on trajectory 2

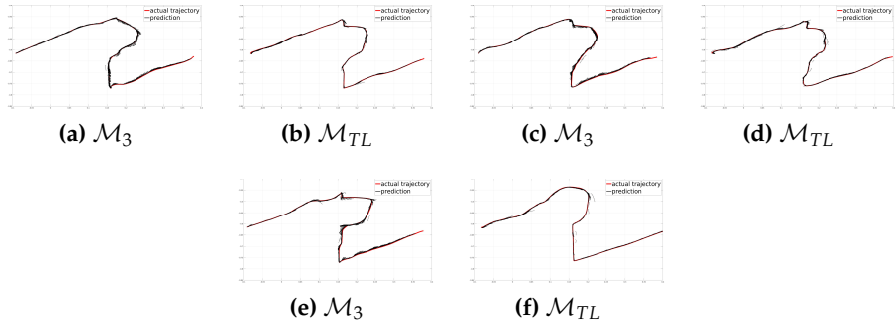


(a) $\mathcal{M}_0$ **(b)** $\mathcal{M}_1$ **(c)** $\mathcal{M}_2$ **(d)** $\mathcal{M}_3$

Figure A.2: Iterative training on trajectory 2

A.3.2 TRANSFER LEARNING

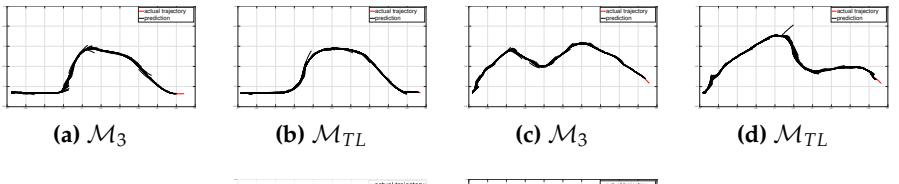
Transfer learning is tested with a new trajectory, new users, and co manipulated objects. In this section, results relative to those situations are presented. In particular, Figure A.3 presents results relative to the Transfer Learning to a new, unseen trajectory, Figure A.4 presents results relative to the Transfer Learning to new users, and Figure A.5 presents results relative to the Transfer Learning to new co-manipulated objects.



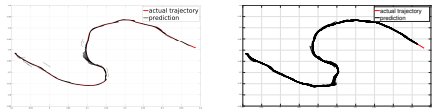
(a) $\mathcal{M}_3$ **(b)** $\mathcal{M}_{TL}$ **(c)** $\mathcal{M}_3$ **(d)** $\mathcal{M}_{TL}$

(e) $\mathcal{M}_3$ **(f)** $\mathcal{M}_{TL}$

Figure A.3: Transfer learning case I - new trajectory

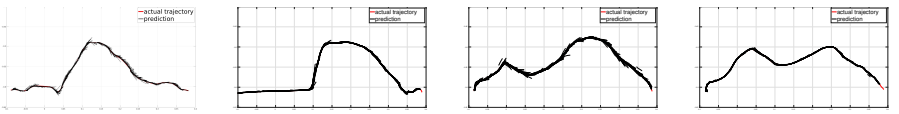


(a) $\mathcal{M}_3$ **(b)** $\mathcal{M}_{TL}$ **(c)** $\mathcal{M}_3$ **(d)** $\mathcal{M}_{TL}$

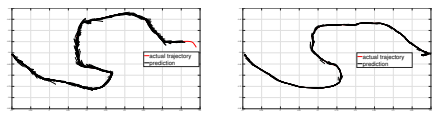


(e) $\mathcal{M}_3$ **(f)** $\mathcal{M}_{TL}$

Figure A.4: Transfer learning case II - new users



(a) $\mathcal{M}_3$ **(b)** $\mathcal{M}_{TL}$ **(c)** $\mathcal{M}_3$ **(d)** $\mathcal{M}_{TL}$



(e) $\mathcal{M}_3$ **(f)** $\mathcal{M}_{TL}$

Figure A.5: Transfer learning case III - new objects

A.3.3 COOPERATIVE GAME THEORY

This section briefly recalls some definitions relative to the Cooperative Game-Theory For a complete discussion and demonstrations, see [29].

Definition 3: Pareto optimality

A set of strategies $\hat{u} = \{\hat{u}_1, ..., \hat{u}_N\}$ is called Pareto optimal (or efficient) if the set of inequalities,

$$J_i(u) \leq J_i(\hat{u}) \ , \ i = 1, ..., N \tag{A.7}$$

where at least one of the inequalities is strict, does not allow for any solution $u \in U$. $(J_1(\hat{u}), ..., J_N(\hat{u}))$ is called Pareto solution, and the set of such tuple is the Pareto frontier.

Define the set A as

$$A := \left\{ \alpha = (\alpha_1 \ , ..., \ \alpha_N) | \alpha_i \geq 0 \ , \ and \ \sum_{i=1}^{N} \alpha_i = 1 \right\}$$

Lemma 1: Pareto optimal control

Let $\alpha \in A$. If $\hat{u} \in U$ is such that

$$\hat{u} = \arg\min_{u \in u} \left(\sum_{i=1}^{N} \alpha_i \, J_i(u) \right)$$

then $\hat{u}$ is Pareto optimal.

Corollary 1: Pareto optimality

Consider 3.8, 5.8 and 5.9. Under the assumption that $Q_i \geq 0$ and $J_i(u)$ are convex. The set of all cooperative Pareto solutions is given by

$$(J_1(u^*(\alpha)), ..., J_N(u^*(\alpha))) \ , \ \alpha \in A$$

where the corresponding Pareto Optimum is obtained as

$$u^*(\alpha) = \arg\min_{u \in U} \left(\sum_{i=1}^{N} \alpha_i \, J_i(u) \right)$$

A.3.4 CGT MATRICES COMPUTATION

In this section, the computations of Q_{gt}, R_{gt} and z_{ref} are detailed. Rewriting and rearranging 5.8 and 5.9 leads to

$$\begin{aligned} J_h = \int_0^\infty (\, & z^T \, (Q_{h,h} + Q_{h,r}) \, z + z_{ref,h}^T \, Q_{h,h} \, z_{ref,h} + z_{ref,r} \, Q_{h,r} \, z_{ref,r} + \\ & -2 \, z_{ref,h}^T \, Q_{h,h} \, z - 2 \, z_{ref,r}^T \, Q_{h,r} \, z + u_h^T \, R_h \, u_h \,) \ dt \end{aligned} \tag{A.8}$$

and

$$\begin{aligned} J_r = \int_0^\infty (\, & z^T \, (Q_{r,h} + Q_{r,r}) \, z + z_{ref,h}^T \, Q_{r,h} \, z_{ref,h} + z_{ref,r} \, Q_{r,r} \, z_{ref,r} + \\ & -2 \, z_{ref,h}^T \, Q_{r,h} \, z - 2 \, z_{ref,r}^T \, Q_{r,r} \, z + u_r^T \, R_r \, u_r \,) \ dt \end{aligned} \tag{A.9}$$

Combining these equations into 5.10, it results in

$$\begin{aligned} J_{gt} = \int_0^\infty (\, & z^T Q_{gt} z + z_{ref,h}^T \, Q_h \, z_{ref,h} + z_{ref,r}^T \, Q_r \, z_{ref,r} + \\ & -2 \, z_{ref,h}^T \, Q_h \, z - 2 \, z_{ref,r}^T \, Q_r \, z + u^T \, R_{gt} \, u \,) \ dt \end{aligned} \tag{A.10}$$

with

$$Q_{gt} = \alpha\ (Q_{h,h} + Q_{h,r}) + (1-\alpha)\ (Q_{r,h} + Q_{r,r}) \tag{A.11}$$

$$Q_h = \alpha\, Q_{h,h} + (1-\alpha)\, Q_{r,h} \tag{A.12}$$

$$Q_r = \alpha\, Q_{h,r} + (1-\alpha)\, Q_{r,r} \tag{A.13}$$

and

$$R_{gt} = \begin{bmatrix} \alpha\, R_h & 0 \\ 0 & (1-\alpha)\, R_r \end{bmatrix}. \tag{A.14}$$

Using 5.10, z_{ref} can be computed by:

$$\begin{aligned} J_{gt} &= \int_0^\infty \left((z - z_{ref})^T\, Q_{gt}\, (z - z_{ref}) + u^T\, R_{gt}\, u \right)\, dt \\ &= \int_0^\infty \left(z^T\, Q_{gt}\, z + z_{ref}^T\, Q_{gt}\, z_{ref} - 2\, z_{ref}^T\, Q_{gt}\, z + u^T\, R_{gt}\, u \right)\, dt \end{aligned} \tag{A.15}$$

Being constant $z_{ref,h}^T\, Q_h\, z_{ref,h}$, $z_{ref,r}^T\, Q_r\, z_{ref,r}$ and $z_{ref}^T\, Q_{gt}\, z_{ref}$, the solution of problem 4.16 is not affected, and the only components in A.10 and A.15 to be compared are

$$-2\, z_{ref}^T\, Q_{gt}\, z = -2\, (z_{ref,h}^T\, Q_h\ +\ z_{ref,h}^T\, Q_r)\, z \tag{A.16}$$

Simplifying, z_{ref} results in

$$z_{ref} = Q^{-1} (z_{ref,h}\, Q_h\ +\ z_{ref,h}\, Q_r) \tag{A.17}$$

A.4 DRAPING PROCEDURE

The proposed manufacturing of carbon fiber composite material task requires the following sub-tasks:

1. the ply, placed on a carrier, is localized, and the robot autonomously approaches it, positioning with precision with respect to the FLP - figure A.6a;

2. the robot grasps the ply autonomously with a gripper equipped with suction caps. One of the suction cups is directly placed on the FLP - figure A.6b;

3. the robot partially lifts the ply from the side it is attached to autonomously;

4. the human approaches the ply from the side opposite to the robot one and grasps it - figure A.6c;

5. the human and the robot together lift the ply to avoid any contact with the carrier, this is necessary to let the CNN model work - figure A.6d;

6. the human and the robot together move the ply towards the target pose. In this phase, the Role-Arbitration law is fundamental. Indeed, by letting the role continuously switch, the robot avoids collisions with the environment that might be occluded to the sight of the human, and vice-versa, the human can command the robot's motion to comply with the operator's preferences. Possibly, the operator can also handle unforeseen situations - figure A.6e;

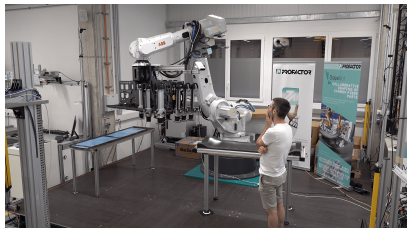
7. the human and the robot roughly approach the target pose of the FLP with the human leading - figure A.6f;

8. when the FLP approaches its goal pose within a predefined tolerance, the role of leader progressively moves from the human to the robot. The robot finally positions the PLF precisely to its target pose - figure A.6g;

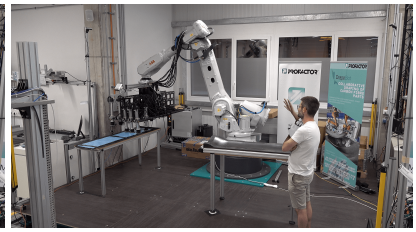
9. the robot moves toward the mold until the FLP is in contact - figure A.6h;

10. the robot stops any motion, it applies a force to keep the FLP in position while the human performs the draping.

This procedure is performed multiple times until all the plies are draped.

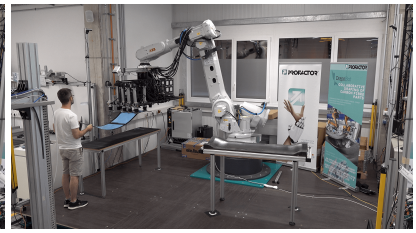


(a) The robot approaches the ply - autonomous motion

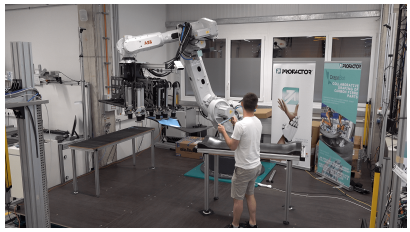


(b) The robot grasps the ply - autonomous motion

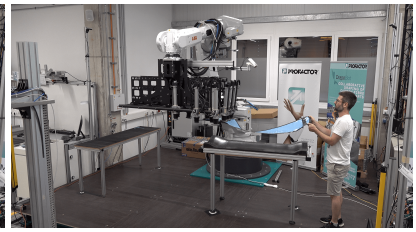
(c) The robot lifts the ply - autonomous motion



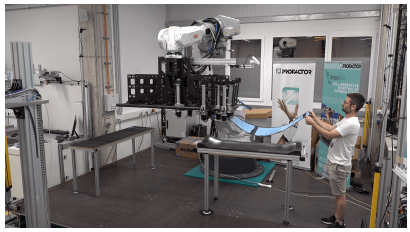
(d) The human and the robot lift the ply - autonomous motion



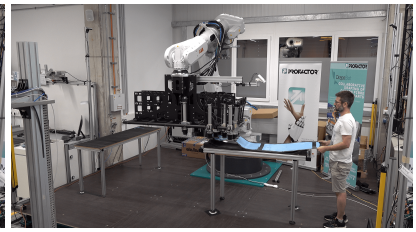
(e) The human and the robot transport the ply towards the goal pose - collaborative transport with role arbitration



(f) The human leads the task toward the goal pose - smooth leader role switch from human to robot



(g) The robot leads the task precisely to the goal pose - robot leader



(h) The robot approaches the mold to allow precise draping

Figure A.6: The various steps of the draping process

ACKNOWLEDGEMENTS

First of all, if you read until this point, thank you!

Then, I have to acknowledge multiple people who made this work possible. Thanks to my Ph.D. advisor, Manuel Beschi, for his help and support over the three years. His background support helped to make the math for multiple works.

Thanks to Nicola Pedrocchi for the daily support and for guiding my research without imposing his ideas. Moreover, super important, thanks for giving me money for this work.

Thanks to my colleagues at CNR-STIIMA in Milano Stefano, Piri, Adriano, Giorgio, and Enrico. Thanks to the technicians, Tito, Joao, and Roberto, who helped me realize the experiment setups. Thanks to my colleagues at CARI Lab in Brescia Cesare, Roberto, Michele, and Michele.

Thanks to Loris, for better or for worse, always present in my working career, from the beginning at ITIA to the visiting period at IDSIA-SUPSI, the ride after I-RIM, and the shining future.

And a non-institutional thanks to my family. Chiara my wife and my spouse always supported me. Giovanna, a smile. Alberto, welcome on board.

That's all folks !

paolo