POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Assistive controller for physical human-robot interaction based on cooperative game theory and human intention estimation

Tesi di Laurea Magistrale in
Automation and Control Engineering - Ingegneria dell'Automazione

Author: **Davide Cassinelli**

Student ID: 945785
Advisor: Prof. Paolo Rocco
Co-advisors: Paolo Franceschi, Nicola Pedrocchi
Academic Year: 2021-2022

Abstract

This thesis aims at designing an assistive control for physical Human-Robot Interaction (pHRI) based on Differential Cooperative Game Theory (DCGT). In particular, a distributed Model Predictive Control (dMPC) is formulated based on the DCGT principles. For proper implementation of the DCGT method, one crucial piece of information regards human intention, defined as the desired trajectory that a human wants to follow by deforming online a nominal trajectory of a manipulator robot over a finite rolling prediction horizon. The method used for human intention estimation is based on recurrent neural networks (RNNs), which have the special feature of keeping track of historical information that evolves over time to produce predictions about future evolution. Specifically, it is planned to use information about a portion of the trajectory before the current state to predict a future portion of the trajectory. The RNN model is composed of cascaded Long-Short Term Memory (LSTM) and Fully Connected (FC) layers (RNN+FC). The prediction of the RNN+FC model directly influences the control framework and robot behavior, so iterative training is proposed to adapt the model. The iterative learning procedure is time-consuming since it requires a relatively large amount of data and computational time. It also learns a model of a specific user performing a specific task. Therefore, Transfer Learning (TL) is proposed to quickly adapt the model to different users and tasks.

The behavior of the dMPC framework proposed is thoroughly analyzed with simulations to understand its applicability and parameters tuning to a pHRI assistive controller. Moreover, real experiments were carried out on a UR5 robotic arm to which a force sensor was installed. First, the RNN+FC model and the iterative training procedure are validated with specific experiments on two dimensions (x–y plain). Then, the TL technique is evaluated for the RNN+FC model adaptation to a new trajectory, different users, and co-manipulation of a large object. Finally, an application scenario is proposed for co-manipulating two different objects and comparing the obtained results with other controllers typically used in the pHRI.

Keywords: Physical human-robot Interaction, Impedance Control, Cooperative Game Theory, human intention prediction, MPC, RNN, Transfer Learning.

Abstract in lingua italiana

Questa tesi ha come obiettivo la progettazione di un controllo assistivo per l'interazione fisica uomo-robot (pHRI) basato sulla teoria dei giochi cooperativi differenziali (DCGT), attraverso l'implementazione di un controllo predittivo (dMPC). Per l'implementazione del metodo DCGT, molta importanza viene data all'intenzione umana, definita come la traiettoria che un uomo intende seguire modificando una traiettoria nominale di un robot manipolatore, in un dato intervallo di tempo. Il metodo utilizzato per la stima dell'intenzione umana si basa su reti neurali ricorrenti (RNN), che hanno la particolarità di tenere traccia delle informazioni che evolvono nel tempo per produrre previsioni sull'evoluzione futura. In particolare, si intende utilizzare le informazioni relative a una porzione di traiettoria precedente allo stato corrente per formularne una su una porzione futura di traiettoria. Il modello RNN è costituito da una parte di Long-Short Term Memory (LSTM) e da una di Fully Connected (FC) connesse in cascata (RNN+FC). La previsione fatta dal modello RNN+FC influenza direttamente il controllo e il comportamento del robot, pertanto viene proposto un processo iterativo per adattare il modello al meglio. Questa procedura viene svolta da uno specifico utente e richiede parecchio tempo poiché deve gestire un'elevata quantità di dati. Pertanto, viene proposto il metodo di Transfer Learning (TL) per adattare rapidamente il modello a utenti e oggetti diversi.
Il comportamento del dMPC è stato analizzato con alcune simulazioni e sono state sperimentate diverse soluzioni nella scelta dei parametri per un controllore assistivo pHRI. Inoltre sono stati condotti esperimenti reali su un braccio robotico UR5 su cui è stato installato un sensore di forza. In primo luogo, il modello RNN+FC e la procedura iterativa sono stati convalidati con esperimenti su due dimensioni. Successivamente, la tecnica TL viene valutata per l'adattamento del modello RNN+FC su una nuova traiettoria, con diversi utenti e con un oggetto di grandi dimensioni. Infine, viene proposta un'applicazione per la co-manipolazione di due oggetti diversi, confrontando i risultati con controllori tipici del pHRI.

Parole chiave: Interazione fisica uomo-robot, teoria dei giochi cooperativa, controllo ad impedenza, predizione dell'intenzione umana, MPC, RNN, Transfer Learning.

Contents

1 | Introduction

The application of robotics from industry to human environments is constantly growing due to the requirement of automatizing tasks and the high cost of experienced human resources. In this view, computers and machines are now taking place as assistive tools for humans in order to reduce fatigue and stress and to increase safety conditions in areas where robots are used, especially in modern factories. This need for collaborative manufacturing tasks shared between humans and robots leads to the development of what is called human-robot interaction (HRI). This interaction can be separated into two general categories: Remote interaction and Proximate interaction. Remote interaction is often referred to as teleoperation, supervisory control, or telemanipulation. Proximate interaction takes the form of a robot assistant and social interaction includes social, emotive, and cognitive aspects of interaction. Proximate interaction may also include physical interaction. In the case of physical interaction, we refer to a physical Human-Robot Interaction (pHRI).
The physical aspect is intended to improve the quality of life and human capabilities in terms of strength, speed, and precision. In general, on the one hand, robots can support the worker in terms of force and precision. On the other hand, humans have experience, general knowledge, and versatility to perform different and new tasks. The combination of these advantages has led to the growth of this branch of robotics.

One of the main objectives of pHRI is to make the robot's motion smooth and natural for the human interacting with it. In this context, new and possibly adaptive controllers are needed to assist human operators in performing shared tasks. Currently, the standard of industrial robot control is position/velocity commands at the joint level, and successful execution can be obtained only if the tasks can be accurately planned.

On the other hand, in pHRI, force controls are widely used because they allow a compliant behavior of the manipulator, leading to a more natural physical interaction. Two main classes of force control strategies can be identified: *indirect force control* and *direct force control*. The main difference between the two categories is that the former achieves force control indirectly via a motion control loop; the latter offers the possibility of controlling

directly the force and contact moment to a desired value. To the category of indirect force control belongs impedance control, where the position error is related to the contact force through a mechanical impedance of adjustable parameters.

1.1. Game Theory approach

Among the various control schemes, Game-Theory (GT) based controllers are nowadays under investigation thanks to their capability to model the interaction between multiple players. The Game-Theory (GT) approach provides valuable tools to analyze complex interactive behaviors between rational players and provides mathematical models that provide players with ”optimal” policies to minimize their objectives, taking interaction into account.

Three aspects are the core of a dynamic game, as in [45] and in [10]: the *mode of play*, the *equilibrium type*, and the *information pattern*. The *mode of play*, describes each player's attitude toward his/her own objective, as well as the other player's interest in a game. The *mode play* could be: *Cooperative*, where each player has a sense of collective and attempts to enter into a binding agreement, or *Non-Cooperative*, where players consider themselves individuals and concentrate on pursuing their own interests. The *equilibrium type* concerns each player's strategy adopted for pursuing his/her goal, as presented in [63]. Therefore, an equilibrium strategy of a player is his/her strategy that constitutes the equilibrium. The *Nash Equilibria*(NE) and *Stackelberg equilibria*(SE) are two typical equilibria solutions for the non-cooperative concept. In the NE, each player derives his/her strategy by taking the others' strategies into account, and all the players act simultaneously. In the SE, one player serves as the leader and the others serve as followers. The leader derives his strategy by taking into account all the followers' optimal responses, while the followers react to the leader's action by simply using their individual optimal responses. Comparable to the Nash and Stackelberg equilibrium is the *Pareto equilibrium*, a solution that emerges in a Cooperative Game and that is generated when there is no other outcome that makes every player at least as well off and at least one player strictly better off. Last, *information pattern* describes each player's knowledge of the states of the game system. The players are defined as possessing the "open-loop" information pattern when only the initial state of the game is known to them, otherwise, the players are considered as having the "closed-loop" information.

The robot and the human, in this context, adopt a cooperative scenario, working in "a game of cooperation agreement". The cooperative game is focused on how to maximize the interests of the participants in the game and how to distribute the benefits to each

participant. In fact, the benefit of participants through the cooperative game is typically higher than the one obtained by independent work, otherwise, participants will abandon cooperation.

To better explain this concept, consider the example of the prisoners. In this problem, there are two players, separated in different rooms, that are accused of a crime and interrogated at the same time. The player who confesses the crime by accusing the other gets immediate release while the accused player suffers the maximum penalty. If both players avoid confessing, the two are given a very light penalty. Conversely, if both confess, both are sentenced to the ordinary penalty.
According to Nash equilibrium, each player chooses his dominant strategy on the basis of the opponent's choice expectations. The end result is a stable NE but not optimal from an individual point of view.
In order to achieve the best option, the two players should cooperate and thus should not confess. In this latter Cooperative case, the outcome of the game is better if compared with the NE for both players. Therefore Cooperation is convenient against Non-Cooperation. This example makes clearer the choice of adopting a cooperative approach to describe pHRI. The robot, in this case, assists the human in performing a task, and benefits are gained when they cooperate.

One of the assumptions underlying the GT problem is that each player has knowledge of the opponents' goal. The goal of a human, in pHRI scenarios, can be described as the human's desire to move an object from a starting point to an endpoint, with the assistance of a robot. In particular, it is assumed that the human wants to follow a trajectory, over a finite rolling prediction horizon, and that the robot wants to assist in following it, with the objective to reduce the human's effort. The desired trajectory that the human wants to follow, can be defined as human intention and requires a lot of attention as it has to be detectable and interpretable by the robot and because it could be different depending on the type of goal.

1.2. Learning process

To understand human intention two main approaches can be pursued: a *model-based* and a *learning-based* approach. In the former, a physical model is obtained, and the aim is to identify its physical parameters. Such parameters are typically specific and vary according to the task and the different humans, making it complex to generalize the model. The latter is the process of using data to derive a function that maps human dynamics according to input-output relation, without any regard for the physical model. Such

an approach is capable of generalizing, at the cost that requires many data to recover a general function capable of modeling different input-output relationships. Regarding the second approach, Neural Networks (NN) represent a universal approach that can be applied to almost any field.

According to [77] and [1], a NN is a parallel distributed information processing structure in the form of a directed graph. A directed graph is a geometrical object consisting of a set of points called nodes connected through weights.

Let's focus on the basic model, visible in fig 1.1, the *feed forward neural network*. The additional intermediate layers (between input and output) are referred to as *hidden layers* because the computations performed are not visible to the user. The default architecture of feed-forward networks assumes that all nodes in one layer are connected to those of the next layer. Therefore, the architecture of the neural network is almost fully defined once the number of layers and the number/type of nodes in each layer have been defined

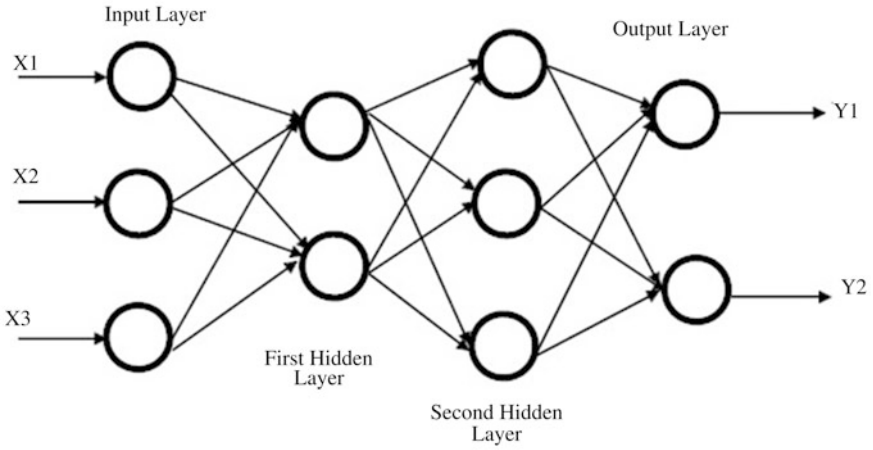


Figure 1.1: Basic architecture of a two-layers Feed Forward Neural Network, from [77]

One of the main drawbacks is that Neural Networks are time-consuming, require a lot of data, and may produce wrong outputs as they tend to overfit in low data regimes, making it challenging to apply to actual robot manipulation.

To solve this problem, Transfer Learning (TL) is proposed. According to [69], Transfer Learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned. The meaning of this and what the TL process consists of is an attempt to be more efficient during the learning process and collecting data, in particular when facing new situations. In fact, collecting new data is typically an expensive and time-consuming activity, because of the necessity of creating

a new dataset and re-training the NN model.

The benefits that transfer learning brings can be identified by three measures: *Initial performance*, the initial performance achievable in the target task using only the transferred knowledge, compared to the initial performance of an ignorant agent; the *Improvement rate*, the amount of time it takes to fully learn the target task and the *Final performance level*, achievable in the target task compared to the final level without transfer.

So the transfer learning work with a model pre-trained on a particular part of the knowledge that can be transferred to the new domain. As presented in [48] different types of Transfer learning methods exist, such as *inductive transfer learning*, *transductive transfer learning*, *unsupervised transfer learning*.

In inductive transfer learning the objective is to induce a predictive model from a set of training examples while in transductive transfer learning refers to the situation where all test data are required to be seen at training time, and the learned model cannot be reused for future data. Unsupervised transfer learning assumes that a reasonably sized dataset exists in the target task, and the target-task learner requests labels for examples only when necessary.

1.3. Motivation

Considering an industrial scenario, the mutual benefits that humans and robots can obtain through cooperation are multiple. In particular, considering collaborative Robots, safe interaction can be easily achieved, but they typically have a limited payload and workspace and cannot handle too complicated tasks. On the other hand, humans can have beneficial help from robots as they are more precise and do not suffer from fatigue that may arise during repetitive tasks.

As an example, consider the case of co-carrying and precise positioning of a large and/or deformable object. The robot, particularly if it is small-medium size as typical collaborative robots, is likely to grasp it from one side. The human can safely grasp the same object on the other side and they can both cooperatively transport it to the target pose. In doing this, during the motion, the cognitive capabilities of the human allow him/her to see and avoid possible obstacles that the robot might not know. On the other hand, if the target position is known to the robot, the robot can help in positioning the object with the required precision. This may occur in the case of strict tolerances imposed by an assembly process, where the human grasping an object has difficulties in positioning it precisely, and should be guided by the robot.

Therefore, it appears clear that interaction and cooperation can provide both mutual benefits, and the need to define a method for the robot to assist the human must be defined.

Thanks to its capability of describing interactions at various levels, Game Theory comes into play in this context. In particular, among the various types of solutions, the cooperative approach shows better outcomes if an agreement is possible. In using this approach, some assumptions must be considered. Is well known, one of the major assumptions behind GT is that the players know the opponents' objective. This is, in general, not true for the robot. Therefore, recovering the human objective represents a challenge to be solved. Defining the human objective as his/her intention to follow a desired trajectory over a finite rolling prediction horizon, a method capable to predict such a trajectory is required.

To do this, neural networks, and particularly RNNs provide a solution to the problem. Being training an RNN time consuming as it requires collecting a lot of data, some efficient method is required to adapt such an RNN model to new users or new situations that may appear. This is done by using the concept of transfer learning, which helps with time-saving and decreasing the size of the dataset.

To conclude, the motivation behind this work is to design a pHRI application capable to handle the co-manipulation of large objects improving benefits to humans thanks to the cooperation. To do this, different modules are deployed, to make the approach adaptable to different users.

The thesis was carried out at CNR-STIIMA in Milano and, in particular, at the personal robotics for manufacturing (PERFORM) laboratory.

1.4. Contribution

Before going through a literature review we analyze the thesis achievements. We can express that the Game-Theoretic framework shows potential for being implemented in a pHRI scenario. As one of the main hypotheses of GT is that the players have knowledge of the opponent's objective, a method to allow the robot to know the human intention is required. In the literature, different studies investigate human intention with different definitions. In the pHRI field, human intention is typically defined as the desired trajectory that a human wants to follow. Learning techniques have been shown to provide a good approximation to predict human intention.
Within this context, the main contributions of the proposed work are identified.

The first contribution is the adoption of the dMPC method in a pHRI scenario. In the literature, dMPC formulations are proposed for similar tasks as shared driving, but no applications to the pHRI problem are proposed yet.
Second, GT is used to describe pHRI and derive role-arbitration laws, focusing on the Non-Cooperative description of the interaction. Very little effort and only a few studies investigate the Cooperative GT, which compared to the NC typically provides better outcomes for all the players.
Finally, in the literature, different learning models are proposed, but the prediction is limited to one step ahead. In this work, to implement the dMPC logic, the prediction is made over a horizon of 0.4 seconds.
To summarize, the main contributions are:

1. adoption of the dMPC method in a pHRI scenario, with analysis of its behavior and performance;
2. Game-Theoretical Cooperative description of the pHRI, and dMPC problem formulation;
3. long-term human intention prediction based on RNN+FC model and its integration with the dMPC control framework.

1.5. Organization of the thesis

The thesis is organized as follows.

The first section introduces the main concepts and the motivations behind this work. Finally, the contribution of this thesis is analyzed.

The second chapter analyzes the background and the state of the art of pHRI methods and applications. The first part is a collection of work on the interaction between humans and robots and the various classification of the various control type used in pHRI. Then it discusses the Game-Theoretical description of human-machine interaction, with a focus on pHRI applications. In the end, are present the works on human intention estimation and the way it could be solved, focusing on the neural network approach.

Chapters 3 and 4 will explain the core methods used in the thesis. In particular, chapter 3 derives the cooperative game-theoretic distributed Model Predictive Control formulation of the pHRI problem. Chapter 4 presents the formulation of the human intention detection problem and the proposed solution based on the learning process. The RNN+FC model and the Transfer Learning technique are presented.

Chapter 5 is dedicated to presenting and discussing the results and real experiments conducted. At first, a section is dedicated to presenting the setup and the parameters tuning. Then the validation of the model through the iterative procedure is discussed. And a part of the application process is studied to check the goodness of the control In the end, a section with a discussion of the results obtained is presented.

To finish, a conclusion part is treated to summarize the concept of the work done.

2 | State of the Art

This chapter analyzes previous works and offers useful insights for the work presented in this thesis. The controllers typically used for the pHRI case are discussed. The use of Game Theory to describe humans interacting with machines is analyzed. Then the different techniques for learning human behavior interacting with A robot are analyzed using neural networks.

2.1. Physical Human Robot Interaction Controllers

The subject of pHRI controllers is deeply analyzed as a research topic, due to the need for smooth, safe, and natural interaction and it is a topic constantly developing in the world of robotics.

The pHRI could be considered an advantage, as it looks at various ways the robot can assist humans during different tasks. In [61], it is explained how the robot can provide a force to support the human work that helps to perform more challenging tasks and how, conversely, the human provides his knowledge to the robot to get the most out of the interaction of the two. Another reason to include robots in manufacturing is to lighten human physical and cognitive efforts. For this purpose, the operator's activities should be reorganized in order to assign more cognitive and control tasks to humans and to assign repetitive tasks and all operations that require greater accuracy, speed, and repeatability to robots. For example, in [51], a method for an industrial case focused on assembly operations supported by collaborative robots, is adopted. It adopts a user experience (UX)-oriented structured method to investigate the human-robot dialogue to map the interaction with robots during the execution of shared tasks. Additionally, to take full advantage of human skills, it is important that intuitive user interfaces (UI) are appropriately designed, as discussed in [71], so that human operators can easily program and interact with the robot, as it aims to be as intuitive as possible to be as efficient as possible.

Moreover, given the wide application in the world of pHRI, there is not only one related

to the industrial aspect. In the field of healthcare robotics, we find several applications. The telerobotic devices in the operating room facilitate dexterous surgical procedures, as present in [76], the exoskeletons in the rehabilitation domain as walking aids, and the upper-limb movement assist devices, as discussed in [9].

These aspects are very interesting and open the discussion to various fields. Despite this, this work focuses on the industrial aspects. There is more than one way in which the robot can cooperate with humans in the industrial scenario, according to [24]. In fact, there can be human-robot coexistence, cooperation, and collaboration. In the coexistence scenarios, the robots and humans are in the same environment but generally do not interact with each other. In cooperation and collaboration scenarios, as the case considered in this work, robots and humans share the workspace. In particular, in the cooperative one, the human operator and robot work in the same workspace at the same time, though each focuses on separate tasks. In the collaboration one, they execute a task together and the action of one has immediate consequences for the other.

This type of collaboration finds wide use in industries and factories and therefore assumes high importance in how it can be inserted into an industrial scenario. In [51], this analysis is discussed going to evaluate the actors involved, the cycle time, and the duration of the various tasks.

2.2. Control Techniques for pHRI

In the vast universe of human-robot interaction, analyzing various controllers used in different works is interesting. This subsection focuses on the various type of Impedance/Admittance Control used in the field of robotics. Moreover, it analyzes a particular description of interaction given by the Game Theoretic modeling. An introduction to the game theory approach is given, focusing on how a human-robot collaboration problem can be reformulated as a multi-agent game.

2.2.1. Impedance/Admittance Controls techniques

Considering a classical pHRI scenario as in fig 2.1, humans and robots perform coordinated operations. Robots can be industrial manipulators, and humans operate the robot's end-effector to perform some tasks collaboratively. One difficulty in such tasks lies in controlling manipulators complying with the operator and constraining it simultaneously in the predefined task space. In order to solve issues of compliance in pHRI, Impedance/Admittance Control is becoming one of the most efficient control methods.

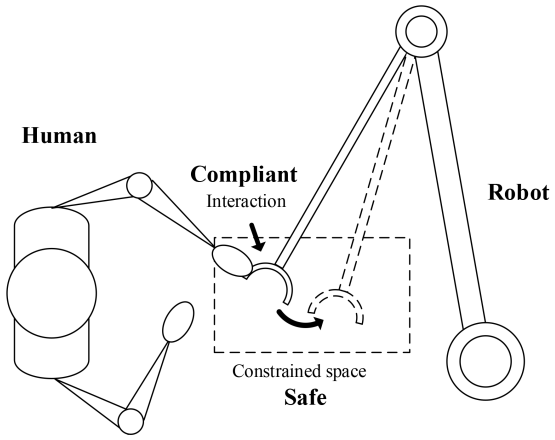


Figure 2.1: Typical human-robot interactive scenario from [18]

The *Impedance Control*, proposed first by Hogan in [20], expresses the relationships between the contact force and state based on an impedance model. In particular, Impedance control aims at ensuring that the manipulator, controlled in position and in interaction with the environment, manifests the behavior of a mechanical impedance, described as a generalized mass-spring-damping system.

An interesting evolution of impedance/admittance controls is adaptive impedance/admittance control. It deserves attention because it integrates the strength and accuracy of the robot with the human operator's ability of task cognition in order to transfer the detected force into the reference position and velocity of the robot. This controller is verified in [18], on a robotic manipulator with seven flexible joints and with advanced sensors, including position, velocity, and torque sensors. The admittance control ensures that the end-effector motion of the manipulator complies with the human operation and avoids collision with surroundings. The admittance control is also used in [78], where the damping is updated online in pHRI to reduce the contact force change and the contact force and thus make pHRI easier and more natural, showing the effectiveness of the proposed dynamic model and the adaptive control method.

A method that provides an application for impedance control is discussed in [11]. This work presents an experimental study on human–robot co-manipulation in the presence of kinematic redundancy. In particular, Cartesian impedance control is employed to achieve a compliant behavior of the robot's end effector in response to forces exerted by the human operator. It proposes different impedance modulation strategies, which take into account human behavior during the interaction and establish the most effective redundancy resolution strategy. Although it gets good results, the application is different and addresses problems far from what is proposed in this thesis.

Impedance control is not only used in the robotics field, as presented in [68], where a Pneumatically Actuated Antagonistic Robot Joint is analyzed to test a torque/impedance controller that commands a desired torque to two cylinder-based force controllers.

Many other works present examples where impedance control is applied in different methods. In [33], a learning control has been proposed to govern a robot arm dynamics to follow the target impedance model not requiring the knowledge of the robot structure. It can be also combined with an algorithm for physically interactive trajectory deformations, which allows the human to modulate both the actual and desired trajectories of the robot as discussed in [37].

The impedance control is related to the use of a force sensor as in [8], where the variable impedance control problem of commercial industrial robots is studied, to investigate the uncertain contact impedance characteristics between the human palm and a robot end-effector in the pHRI process. This approach is widely used in robotic arm applications, with differences in each work. In particular, it finds a lot of efficiency in manual guidance control with force tracking as in [56], or [58], giving good results in actively assisting the human in the target task, compensating for the unknown part weight. In addition, [59], studied the possibility to keep the learning of the human-robot interaction dynamics active, allowing accounting for the adaptation of the human motor system.

An interesting approach is to use hybrid controllers that combine two different controllers as the impedance set-point and adaptation of the mass-spring-damper parameters, as in [57] and [60], where the operator guides the robot manually to manipulate heavy components.

As demonstrated, impedance control is widely used specifically on robotic arm models and human interaction and it has performed well in the applications where it has been used.

2.2.2. Game-Theory based controllers with human

Among the various control schema, Game-Theory (GT) based controllers are widely used in order to model the interaction between different users, as it offers a solution to the problem of a multi-agent system.

Previous literature studied applications of GT models to the interaction between a human driver and an autonomous vehicle. The methodologies proposed in this field are very interesting and also useful for pHRI applications as it shows the advantages of this approach. In [45], GT is used for modeling a driver's steering interaction with vehicle

collision avoidance control in path-following scenarios through four different paradigms, namely decentralized, Non-Cooperative Nash, Non-Cooperative Stackelberg, and Cooperative Pareto. For each strategy, are analyzed the optimization problems and their resulting steering strategies. Then, two control approaches, applicable to these optimization problems, namely the distributed Model Predictive Control (dMPC) and the Linear Quadratic (LQ) dynamic optimization approaches, are described in detail. The solutions obtained imply a variety of driver steering control behaviors that can be yielded by varying driver path-error weights. There are many choices that can incur depending on the situation in which we are acting, for example in [46], a cooperative Pareto steering strategy when a driver interacts with an automated steering, is discussed. Here the driver can improve his/her performance in following a target path by increasing the effort in pursuing his/her own interest under the driver-automation cooperative control goal. Another choice is to choose a Nash approach as in [47]. Here six drivers are recorded while using a driving simulator and compared with the "conventional" optimal-control driver model. Their model fitting errors are analyzed, resulting in the game-theoretic driver model being statistically significantly better than the conventional driver model for three out of six individuals. Sometimes, is possible to actively switch between the Cooperative and the Non-Cooperative scenario, as treated in [26]. The Game Theory formulation, here, allows for lateral control tasks-lane keeping and obstacle avoidance where the shared control strategy for this mixed driving authority is based on TTC (time to collision) and tracking error. It is interesting because one application, which is not based on the robot model, uses the game theoretic approach of distributed predictive model control.

An interesting comparison between non-cooperative and cooperative is done in [6], starting from the human intent defined as cooperative and not. This work is very interesting because it gives an example of a criterion of choice and choosing if one is better than the other. The criterion is the knowledge of the players. In fact, if they can communicate and trust each other, a cooperative solution is better. On the other hand, if this communication doesn't occur and no agreement exists, a non-cooperative approach is preferable.

In addition to this, Game Theory also finds much use in pHRI, an application of most interest to this thesis.

A Non-Cooperative Nash equilibrium is studied in [34], where the robot is able to adjust its own role according to the human's intention detectable from the measured interaction force. With no human interaction forces, the adaptive scheme allows the robot to take the lead. Otherwise, when the human exerts strong forces, the robot becomes the follower. Despite the approach addressing the Non-Cooperative case, it provides an example of how human-robot interaction can take place even from this point of view.

In [4], the GT is studied on an upper limb exoskeleton robot or in [35], where develops an interactive robot controller able to understand the control strategy of the human user and react optimally to their movements. It precisely identifies each other's control law and allows them to perform the task with minimum effort successfully. In [81], an online estimation method to identify unknown humans' control objectives is used to solve an N-player linear quadratic differential game theoretic problem. Here, the effectiveness of the proposed method is demonstrated by rigorous theoretical analysis and simulations where the humans cooperate with the robot to transport the object back and forth along the horizontal direction.

A work very similar to our approach is [12], which discusses the arbitration of the role between a robot and a human during physical Human-Robot Interaction, sharing a common task. It is very valuable because formulates the problem following a Cooperative Differential scenario. It discusses the arbitration of the role between a robot and a human during pHRI finding multiple solutions on the Pareto frontier. The tests are done on a robotic arm that has to follow a planar circular trajectory, while the human has a different path to follow, which partially overlaps with the robot one. The proposed method is capable of managing the leader-follower transition continuously, but it addresses the problem using an LQR controller and not with an MPC approach. It also does not address a solution with shared reference tracking.

The GT finds many applications in pHRI and obtains promising results for each work. In particular, the ones that use the cooperative scenario give a very efficient solution.

2.3. Human intention estimation/identification

As mentioned so far, human-robot interaction involves studies dedicated to understanding, designing, and evaluating robotic systems for use with humans. A very important part is the communication between the human and the robot and, in particular, what is called *human intent*. This concept assumes a different meaning depending on the action that a human takes, so is interesting to see a variety of cases of different definitions of human intent.

In order to predict this human intent, we can classify the solution into different concepts. The approaches can be classified as *model-based* and *learning based* or *model free*. The model-based work with a physical model with the aim to identify its physical parameters. It expresses a method to design a model based on the estimation of the motion intention of the human partner. In fact, a general model to describe the dynamics of a human limb is supposed to include its mass-damper-spring property. The damper and spring components

usually dominate human limb mode, as explained in [31] or in[53]. The human partner may change his limb impedance during the collaboration. This makes the estimation very difficult. Indeed, it is conditioned by the interaction history and the robot's future action choices, but it works in short time horizons in nominal operating conditions, and it learns only the relative likelihoods of future human actions and responses.

So a better option to solve the human intention issue is a learning-based approach. They became popular in particular approaches such as deep learning with neural networks. It requires a large amount of training data, but data acquisition better generalizes the function that links input with output.

2.3.1. Intent Detection

Based on the literature, human intention can have multiple meanings, so the first part will be devoted to its definition. After that, we will be able to concentrate on the aspects more on the side of the robots, which aim to look at how they measure these intentions and with what "channels" they do it, and how they then interpret that data. In figure 2.2, an easy representation of these three steps is visible.

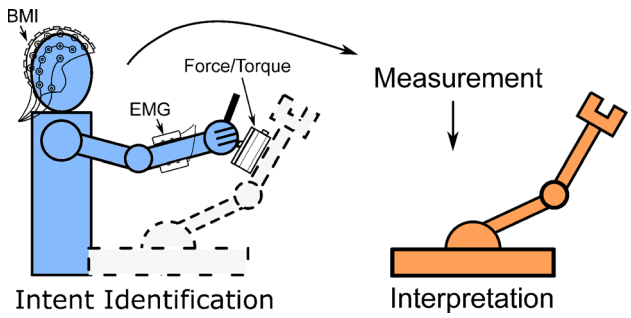


Figure 2.2: The three steps for conveying the human's intent to the robot: identification, measurement, and interpretation [38]

The human intention has several definitions. One is the ability of the robot to imitate the human arm behavior, for example, during handshaking, as discussed in [74]. It models the interaction through the handshake gesture between the two agents, relying on a large amount of data collected from human-human handshake experiments. Intention can also deal with recognizing a series of predefined motions. In [62], the robot has to identify eight motions related to sitting, standing, or walking and control a wearable lower extremity assistive device intended to aid stroke patients during activities of daily living or rehabilitation based on sensor readings from the limb attached to the assistive device.

A more "clinical" definition is detectable in [41], where the human intent is defined as a trigger to initiate motion, often ascertained from a brain-machine interface (BMI). These reported works show how human intention detection is a very complicated aspect and how important it is to define it.

In this thesis, human intention is defined as the desired trajectory that the human wants to follow over a finite rolling prediction horizon.

A similar definition is discussed in [21], where the intent is defined as a predicted forward path over a short time horizon in terms of a velocity or position trajectory. Also, in [50], it is defined as the interaction force between the person and the wrist exoskeleton at the handle or, in [28], as the effective torque about the user's elbow within an elbow exoskeleton.

Once this intention has been defined, the question arises as to how to measure and interpret this. A large category of examples relates to the so-called Myography *i.e.,* the measurement of the activation of human muscles and their resulting contractile force. The more common technique is electromyography (EMG) or Surface electromyography (sEMG), which is less invasive as it is limited to measuring superficial muscles near the surface of the skin that measures the changing electrical potential. An example of this is seen in [52]. Recently also the force myography (FMG), also known as topographic force mapping or muscle pressure mapping, used in [55], has received attention because it infers muscle forces by detecting changes in muscle volume underneath tactile sensors placed on the surface of the limb. Similar to myography, there is Sonomyography, a method used in [2], based on ultrasound imaging. Examples of these are little used in human-robot collaboration as we understand it but more in the medical aspect and, in particular, in the manufacture of prostheses and technologies to support the movement of body parts. Another method is called pattern recognition, used in [22]; an approach that maps patterns of any number of signal features to desired prosthesis poses, grasps, or functions and uses artificial neural networks to learn this mapping varies.

Hence, within this research, it is argued that the robot can only provide minimal assistance to the human, specifically only on the part involved in the rehabilitation process. So although it occupies a large set of cases, our discussion will move to a more industrial and robot-oriented approach and through approaches that use methods to learn models from data.

Great space is given to it based on motion capture with an RGB-D camera. Used in [43], but also in [30], consists of a tool able to detect the colors and the distance of the point from the cameras. The main drawback is that it requires specific hardware, and image

data increase the duration of the training phase due to the high complexity.

A great method of measuring and interpreting the intent relies on the use of a 6DOF force/torque sensor mounted on the robot to measure the dynamical forces during the interaction, as in [74]. It aims to examine the use of force and position to estimate the human impedance during the handshaking task. Particularly, it measures the relationship between the position and orientation of the robot end-effector and the resulting forces and torques at the end-effector, resulting from interaction with the human. The human is then modeled as a linear impedance with three parameters (mass, damping, and stiffness). Using the recursive least squares algorithm for online parameter estimation, the current human impedance parameters are heuristically classified as being "low" or "high". They are then used as the inputs to a hidden Markov model (HMM) to decide if the person intends to be "active" or "passive" in the handshake interaction with the robot.

Instead of measuring the robot position to estimate the interaction force, one can measure the interaction force between the human and the robot at the end-effector and estimate the desired human position. In [14], the user's desired position is extracted by assuming a model of the user's control and is then assumed to be the equilibrium point of the spring. Just as researchers have used interaction force measurements to estimate a desired position of the human, the measured interaction force can be used to estimate other forms of motion intention, as treated in [31]. For example, in [5], the interaction forces measured in the handles of an intelligent walker are fed into a model of the nonholonomic walker dynamics to predict the walker's forward path over a short time horizon. Force/torque sensing in the handle of an assistive cane is used in [72] to ascertain the hidden walking state of the user.

2.3.2. Neural Networks Models

We have introduced arguments involving the computation of certain parameters useful for estimating human and robot behavior. These parameters are time-varying and related to impedance one and the model's output strongly depends on these parameters expressed in understanding and studying human intention. To overcome this complex modeling and parameter estimations, Machine Learning is gaining popularity in addressing such problems. In particular Neural networks achieve excellent results in approximating complex nonlinear systems.

One possible approach is to use the *CNN (Convolutional Neural Networks)* trained to predict the existence of parts to be picked. It works well for applications, as discussed in [49], where tries to perform, a pick operation autonomously. It uses the initial human data,

to find the pixel position of an object from the obtained image and the estimated pose of the object from physical space. Despite this, they are not suitable for predicting time series. Wide use is based on *Artificial Neural Networks (ANN)* in predicting the intended direction of human movement. In [65], is computed by utilizing electromyography (EMG) signals acquired from human arm muscles. It proposes only motion classification to define direction, and the robot assists in the detected directions only but does not predict future motion intentions. Also, in [23], an Adaptive Neural Network estimates the next joint coordinate of the human motion intention represented with the desired angle of the human lower limb interacting with an exoskeleton for rehabilitation. Or the *Deep Reinforcement Learning*, as in [15], where a human and a PR2 robot jointly control the ball position on a plank based on vision and force/torque data. Unfortunately, they operate in a delayed-return environment. A way to estimate human intention is with *RBFNN (Radial Basis Function Networks)*, as studied in [14], during and handshake act between a robot arm and a human using interaction force, position, and velocity at the interaction point. Even if the simulation result has shown the validity of the proposed method, it is just one step ahead and so is not applicable to this thesis.

The examples presented gives an overview of possible existing neural networks and some of their applications, but do not consider long-term prediction.

When dealing with sequence-to-sequence learning applications, *Recurrent Neural Networks (RNN)* turn out to be very effective. Finds much use in machine translation or for predicting the next elements in a sequence. In a recurrent neural network, there is a one-to-one correspondence between the layers in the network and the specific positions in the sequence the network contains a variable number of layers, each with a single input corresponding to that time stamp. A key point here is the presence of the self-loop, as represented in 2.3, which will cause the hidden state of the neural network to change after the input of each word in the sequence. The weight matrices in different temporal layers are shared to ensure the same function at each time stamp.

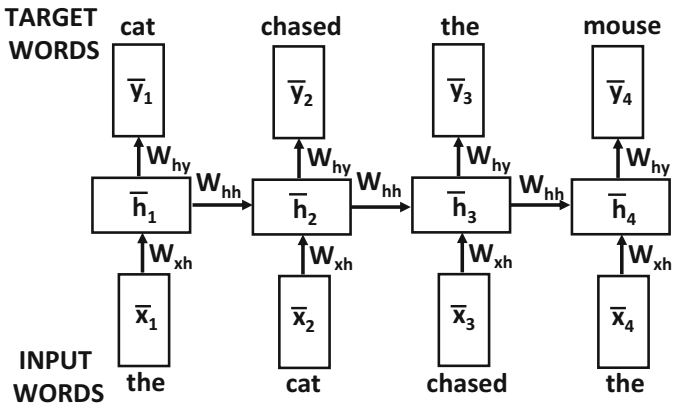


Figure 2.3: Time-layered representation of an RNN with the encoded word and its predicted word likelihoods from [1]

A classic application of the RNNs is for Natural Language Processing(NLP) *i.e.* a way for computers to analyze, understand, and derive meaning from human language in a smarter way. In fact, they work very well in labelings sequence and sequence prediction tasks, such as handwriting recognition, language modeling, machine translation, phonetic labeling of acoustic frames and etc. In [67] and in [44], some experiments are conducted with a small (short-term) spoken dialogues data set and a large (long-term) textual document corpus showing that the proposed RNN reaches optimal performances in learning complex internal structures to expose relevant information.

Thanks to the fact that they can handle time-series data, RNNs found their way also in pHRI to predict human behavior during various tasks. In [80], is used to analyze visual observations of human actions in an assembly setting and predict the human operator's motion future trajectory for online robot action planning and execution. This collaborative assembly uses the pickup and handover of a screwdriver as an example. These analyses are done through information taken from a camera to monitor the human positions and predict the next steps. The use of a camera is always a tough issue but it is an interesting way to collect data. A camera is used also in [70] or in [43], to predict the trajectory of a human who follows a haptic robotic guide. Here the experiment was carried out in the way that the robotic guide was set to move along a random path for 20–30 s in each session. The participants were instructed to follow a randomly moving haptic robotic guide while blindfolded and then apply a deep learning method based on RNN. The robot could collaborate directly from demonstrations, as discussed in [75], without pre-programming and shows a strong ability to adapt to movements with multiple time scales.

Furthermore, among the various types of RNNs, is now widely adopted the Long-Short-Term Memory (LSTM) architecture that outperforms classical RNNs and is used to solve various problems where the sequential data model can store important information. In [39], the LSTM is used to predict the reference one time instant in the future, based on the human limb dynamics and then an assistant controller is proposed to help humans complete collaboration tasks. It validates the performance of this prediction algorithm on a 7 d.o.f Franka Emika robot equipped with joint torque sensors. The goodness of the approach is discussed in [19] while in [40], an interesting LSTM architecture is proposed to predict the desired reference set-point at the next step but does not address any adaptation to new users or objects. The problems, in this last work, are that they do not address the problem of adapting the model to new users, so each new human has to record the full dataset. The LSTM is used not only in the pHRI but finds wide usage in the automobile environment, as proposed in [73]. This article it's proposing a novel method for identifying the driver's braking intention. In order to improve the identification accuracy of driving intention, a braking intention identification model based on Long Short-Term Memory (LSTM) Network is constructed.

2.3.3. Transfer Learning with Neural Networks

One problem that has been identified in the use of neural networks is the lack of adaptability for new users or objects with the goal of reducing computation time and facilitating network training. For this reason, one approach that has been identified is transfer learning, which is already used in many areas and models. In [27], is used to automate domain mapping for value function transfer and speed up reinforcement learning on variants of previously played games. The approach detailed in this work transfers a learned value function from a source task to initialize the value function of a target task, identified to be similar through a graph-based method.

[54] makes use of the *inductive transfer learning*, where some labeled data are required in the target domain to induce a predictive model or the *unsupervised transfer learning* used in [7], to solve learning tasks in the target domain, such as clustering. The knowledge could be transferred by pooling together the rating data from multiple rating matrices in related domains. In [29], is used a model that can share rating knowledge in the form of a latent cluster-level rating model, trained on the pooled rating data from multiple related rating matrices.

There exist methods proposed to "fine-tune" all network parameters [16], tune only the parameters of the last few layers [36], or just use the pre-trained model as a fixed feature

extractor like in [17]. In particular, in this last work, it's explained an approach of *adaptive fine-tuning* called "SpotTune", which consists of taking an image from the target task and using a network to make the decision on whether to pass the image through the fine-tuned layers or the pre-trained layers.

It could be possible to use a pre-trained model as the starting point for the upcoming training phase as in [79]. This application is suitable for those cases in which is possible to train the model on a relatively large dataset. Differently, from freezing the last layers, this algorithm may lead to an overfitted model, *i.e.* a model which fits extremely well on the training dataset but achieves poor performance when generalized to different data.

An objective of this thesis is to reduce the time when the model is adapted to new users or new situations, such as different trajectories or co-manipulated objects. And so, looking at the various cases is used a pre-trained model for new trajectories, users, and co-manipulated objects by freezing the LSTM layer and fine-tuning a cascaded Fully Connected layer, which makes the procedure faster.

3 | Cooperative Game-Theoretic formulation of the distrubuted Model Predicive Control

This Chapter describes the cooperative controller used. At first, the motion of the robot tip, described as a Cartesian Impedance control, is presented. Then the formulation of the distributed Model Predictive Control (dMPC) and its implementation through the Cooperative Game Theory (CGT) is studied.

3.1. System Modeling

This section presents the modeling of the robot motion subject to external forces. The robot motion at the end effector is modeled as a Cartesian impedance, because more natural for the human operator. The desired robot motion at the end-effector in fact, as explained in [20], can be described by the equation of mechanic impedance implemented in the Cartesian space:

$$M_i\, a(t) + D_i\, v(t) + K_i\, \Delta x(t) = u_h(t) + u_r(t) \tag{3.1}$$

where M_i, D_i and $K_i \in \mathbb{R}^{6\times 6}$ are the desired inertia, damping, and stiffness matrices, respectively; $a(t)$, $v(t)$ and $\Delta x(t) \in \mathbb{R}^6$ are the Cartesian accelerations, velocities and delta positions at the end-effector, with $\Delta x(t) = x(t) - x_0(t)$ with $x_0(t)$ the equilibrium position of the virtual spring, and $u_h(t) \in \mathbb{R}^6$ and $u_r(t) \in \mathbb{R}^6$ represent the measured human and virtual robot effort applied to the system. The robot contribution u_r can be seen as an additional assistance that the robot provides to the human. The Cartesian coordinates in x are defined according to [64], with the vector $x = [p^T\ \theta^T]^T$ where p^T are the position coordinates and θ^T the set of Euler angles[1] that defines the rotation matrix

[1]This choice assumes that the angular rotation maintains limited values in the target applications, mainly along one rotation axis, as they work when taken far from the critical points.

describing the end-effector orientation. Also, we can write the vector containing the linear and angular velocities, as $v = [\dot{p}^T \ \omega^T]^T$ with $\dot{p}$ the linear velocity, ω angular velocity.

We start with the equation already mentioned and represent the system with the state-space formulation. Starting from the equation 3.1 we can linearize the system around a working point and write:

$$\begin{aligned} \dot{z} &= Az + B_h u_h + B_r u_r \\ y &= C\, z \end{aligned} \tag{3.2}$$

where $z = [\Delta x^T \ v^T] \in \mathbb{R}^{12}$ is the state space vector.

A is the state matrix

$$A^{12\times12} = \begin{bmatrix} 0^{6\times6} & I^{6\times6} \\ -M_i^{-1}K_i & -M_i^{-1}D_i \end{bmatrix} \tag{3.3}$$

and B is the input matrix

$$B_h^{12\times6} = B_r^{12\times6} = \begin{bmatrix} 0^{6\times6} \\ M_i^{-1} \end{bmatrix} \tag{3.4}$$

with $0^{6\times6} \in \mathbb{R}^6$ denoting and $I^{6\times6} \in \mathbb{R}^6$ the Identity matrix and C the output matrix of the system that converts z to y.

Finally, since the robot controllers accept commands in discrete time, and also data are collected in discrete time, we rewrite the system described in 3.2 in discrete time.

$$\begin{aligned} z(k+1) &= A_d\, z(k) + B_{d,h} u_h(k) + B_{d,r} u_r(k) \\ y(k) &= C_d\, z(k) \end{aligned} \tag{3.5}$$

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 of the system that converts $z(k)$ to $y(k)$.

It is worth noticing that it is more common to feed the robot with the reference position in the joint space, rather than in Cartesian space. It is possible to obtain the reference velocity in the joint space through the following transformation.

$$\dot{q}_{ref}(t) = J(q)^{+}\dot{x}(t) \tag{3.6}$$

where the $\dot{q}_{ref}(t) \in \mathbb{R}^n$ are the reference velocities in the joint space, n represents the number of joints, and $J(q)^+$ is the pseudoinverse of the analytical Jacobian matrix.

Joint positions are then computed via a simple integration. Assume $\dot{q} \simeq \dot{q}_{ref}(t)$ considering that today's robots have great tracking performance in the frequency range excitable by the operator.

3.2. Dynamic Cooperative Game-Theoretic MPC

This subsection presents the GT-based control modeling proposed with Distributed Model Predictive Control.

The Game Theory framework provides definitions and models to better understand how rational decision makers (i.e. the two players), choose their strategy and predict the results of their interaction. The GT framework is subject to dynamic equations that describe the state variables evolution of both the players (also called agents).

We now describe the structure and the formulation of the dMPC, as expressed in [26], using the cooperative scenario. To do this, we enlarge the system and augment it. The system express in 3.5 became:

$$\begin{aligned} z_{gt}(k+1) &= A_{gt} z_{gt}(k) + B_{h,gt} u_h + B_{r,gt} u_r \\ y_{gt}(k) &= C_{gt}\, z_{gt}(k) \end{aligned} \tag{3.7}$$

with $z_{gt} = \begin{bmatrix} z \\ z \end{bmatrix}$, $A_{gt} = \begin{bmatrix} A_d & 0^{12\times12} \\ 0^{12\times12} & A_d \end{bmatrix}$, $B_{h,gt} = \begin{bmatrix} B_h \\ B_h \end{bmatrix}$, $B_{r,gt} = \begin{bmatrix} B_r \\ B_r \end{bmatrix}$ and $C_{gt} \in \mathbb{R}^{m\times24}$ is defined according to the desired output.

Having done all this we can formulate the MPC method: we define the predicted horizon as N_p and the control horizon as N_c and then we write the equation that predicts the future steps:

The output prediction in the future N_p sampling times, calculated at time i is given by:

$$\begin{aligned}
&y(k+1) = C_{gt}A_{gt}x(k) + C_{gt}B_h u_h(k) + C_{gt}B_r u_r(k) \\
&y(k+2) = C_{gt}A_{gt}^2 x(k) + C_{gt}A_{gt}B_h u_h(k) + C_{gt}A_{gt}B_r u_r(k) + C_{gt}B_h u_h(k+1) + C_{gt}B_r u_r(k+1) \\
&\quad \vdots \\
&y(k+N_p) = C_{gt}A_{gt}^{N_p} x(k) + \cdots + C_{gt}A_{gt}^{N_p-N_c} B_h u_h(k+N_c-1) + C_{gt}A_{gt}^{N_p-N_c} B_r u_r(k+N_c-1)
\end{aligned} \tag{3.8}$$

(3.8) can be written in compact matrix form as

$$Y(k) = Fz(k) + \Phi_h U_h(k) + \Phi_r U_r(k) \tag{3.9}$$

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

$$Y(k) = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N_p) \end{bmatrix} \tag{3.10}$$

$F \in \mathbb{R}^{mN_p \times 24}$ is the free response matrix, equal to:

$$F = \begin{bmatrix} C_{gt}A_{gt} \\ C_{gt}A_{gt}^2 \\ \vdots \\ C_{gt}A_{gt}^{N_p} \end{bmatrix} \tag{3.11}$$

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

representing the forced response, defined as

$$\Phi_i = \begin{bmatrix} C_{gt}B_{i,gt} & 0^{m\times 6} & \dots & 0^{m\times 6} \\ C_{gt}A_{gt}B_{i,gt} & C_{gt}B_{i,gt} & \dots & 0^{m\times 6} \\ C_{gt}A_{gt}^2B_{i,gt} & C_{gt}A_{gt}B_{i,gt} & \dots & 0^{m\times 6} \\ \vdots & \vdots & \ddots & \\ C_{gt}A_{gt}^{N_p-1}B_{i,gt} & C_{gt}A_{gt}^{N_p-2}B_{i,gt} & & C_{gt}A_{gt}^{N_p-N_c}B_{i,gt} \end{bmatrix} \tag{3.12}$$

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

$$U_h(k) = \begin{bmatrix} u_h(k+1) \\ u_h(k+2) \\ \vdots \\ u_h(k+N_c) \end{bmatrix} \tag{3.13}$$

$$U_r(k) = \begin{bmatrix} u_r(k+1) \\ u_r(k+2) \\ \vdots \\ u_r(k+N_c) \end{bmatrix} \tag{3.14}$$

To recall we are dealing with the interaction with a GT model approach. This implied the presence of two players as in this case are human and a robot. A goal of these two players is to minimize a cost function in order to resolve the GT problem. This cost function can be defined as a quadratic cost function, as also studied in [45] and in [34], with respect to the control state. So the two cost function applied to a 2-player game could be expressed as:

$$\begin{aligned} J_h(k) &= \sum_{i=1}^{N} e_h(k+i)^T\, Q_{h,h}\, e_h(k+i) + e_r(k+i)^T\, Q_{h,r}\, e_r(k+i) + \\ &+ u_h(k+i)^T\, R_h\, u_h(k+i) \\ &= \sum_{i=1}^{N} \begin{bmatrix} e_h(k+i)^T & e_r(k+i)^T \end{bmatrix} \begin{bmatrix} Q_{h,h} & 0 \\ 0 & Q_{h,r} \end{bmatrix} \begin{bmatrix} e_h(k+i) \\ e_r(k+i) \end{bmatrix} + \\ &+ u_h(k+i)^T\, R_h\, u_h(k+i) \\ &= \sum_{i=1}^{N} e_{gt}(k+i)^T\, Q_h\, e_{gt}(k+i) + u_h(k+i)^T\, R_h\, u_h(k+i) \end{aligned} \tag{3.15}$$

and

$$\begin{aligned} J_r(k) &= \sum_{i=1}^{N} e_h(k+i)^T\, Q_{r,h}\, e_h(k+i) + e_r(k+i)^T\, Q_{r,r}\, e_r(k+i) + \\ &+ u_r(k+i)^T\, R_r\, u_r(k+i) \\ &= \sum_{i=1}^{N} \begin{bmatrix} e_h(k+i)^T & e_r(k+i)^T \end{bmatrix} \begin{bmatrix} Q_{r,h} & 0 \\ 0 & Q_{r,r} \end{bmatrix} \begin{bmatrix} e_h(k+i) \\ e_r(k+i) \end{bmatrix} + \\ &+ u_r(k+i)^T\, R_r\, u_r(k+i) \\ &= \sum_{i=1}^{N} e_{gt}(k+i)^T\, Q_r\, e_{gt}(k+i) + u_r(k+i)^T\, R_r\, u_r(k+i) \end{aligned} \tag{3.16}$$

where $Q_{h,r}$, defines the weight that the human assigns to their own and the robot's reference tracking, $Q_{r,h}$, defines the weight that the robot assign to their own and the human's reference tracking, $Q_{h,h}$, defines the weight that the human assigns to their own and itself reference tracking, $Q_{r,r}$, defines the weight that the human assigns to their own and itself reference tracking, and $e_h(k+i) = y(k+i) - y_{ref,h}(k+i)$ and $e_r = y(k+i) - y_{ref,r}(k+i)$ refers to the tracking errors foreseen for the human and the robot at time step $k+i$, with $e_{gt}(k+i) = \begin{bmatrix} e_h(k+i) \\ e_r(k+i) \end{bmatrix}$, and $u_h(k+i)$ and $u_r(k+i)$ are the control inputs of the human and the robot at time step $k+i$. With $y_{ref,r}$ and $y_{ref,h}$ we define the robot and the human trajectory that each one has to follow.

Now we formulate the equations of MPC in a cooperative control case. In fact in the cooperative game players communicate with each other and share the common objective as expressed by a set of parameters that is α, where

$$\alpha_i,\ i = 1 : N_{players},\ \sum_{i=1}^{N} \alpha_i = 1,\ 0 < \alpha_i < 1$$

which weighs the contribution of each player. The common objective to be shared between the players can be defined as a weighted summation of each cost function as follows:

$$\begin{aligned} Q_{gt} &= \alpha\, \tilde{Q}_h + (1-\alpha)\, \tilde{Q}_r, \\ R_{gt,h} &= \tilde{R}_h \\ R_{gt,r} &= \tilde{R}_r \end{aligned} \tag{3.17}$$

and writing,

$$E_{gt}(k) = \begin{bmatrix} y(k) - y_{ref,h}(k+1) \\ y(k) - y_{ref,r}(k+1) \\ y(k+1) - y_{ref,h}(k+2) \\ y(k+1) - y_{ref,r}(k+2) \\ \vdots \\ y(k+N-1) - y_{ref,h}(k+N) \\ y(k+N-1) - y_{ref,r}(k+N) \end{bmatrix} \tag{3.18}$$

To express the final cost function formula we first define the following matrices:

$$\tilde{Q}_h(k) = \begin{bmatrix} Q_h & & \\ & \ddots & \\ & & Q_h \end{bmatrix} \tag{3.19}$$

$$\tilde{Q}_r(k) = \begin{bmatrix} Q_r & & \\ & \ddots & \\ & & Q_r \end{bmatrix} \tag{3.20}$$

and

$$\tilde{R}_h(k) = \begin{bmatrix} R_h & & \\ & \ddots & \\ & & R_h \end{bmatrix} \tag{3.21}$$

$$\tilde{R}_r(k) = \begin{bmatrix} R_r & & \\ & \ddots & \\ & & R_r \end{bmatrix} \tag{3.22}$$

and $U_r(k)$ and $U_h(k)$ are as in 3.13 and 3.14

The two cost functions 3.15 and 3.16 can be finally written in compact form as:

$$\begin{aligned} J_h(k) &= E_{gt}(k)^T \, \tilde{Q}_h \, E_{gt}(k) + U_h(k)^T \, \tilde{R}_h \, U_h(k) \\ &= E_{gt}(k)^T \, Q_{gt} \, E_{gt}(k) + U_h(k)^T \, R_{gt,h} \, U_h(k) \end{aligned} \tag{3.23}$$

$$\begin{aligned} J_r(k) &= E_{gt}(k)^T \, \tilde{Q}_r \, E_{gt}(k) + U_r(k)^T \, \tilde{R}_r \, U_r(k) \\ &= E_{gt}(k)^T \, Q_{gt} \, E_{gt}(k) + U_r(k)^T \, R_{gt,r} \, U_r(k) \end{aligned} \tag{3.24}$$

So the Distributed Model Predictive Control (DMPC) problem for the Cooperative Game Theoretic pHRI can then be summarized as:

$$\begin{aligned} \min_{u_h} J_h &= E_{gt}(k)^T\, Q_{gt}\, E_{gt}(k) + U_h(k)^T\, R_{gt,h}\, U_h(k) \\ s.t.\ Y(k) &= Fz(k) + \Phi_h U_h(k) + \Phi_r U_r(k) \end{aligned} \tag{3.25}$$

$$\begin{aligned} \min_{u_r} J_r &= E_{gt}(k)^T\, Q_{gt}\, E_{gt}(k) + U_r(k)^T\, R_{gt,r}\, U_r(k) \\ s.t.\ Y(k) &= Fz(k) + \Phi_h U_h(k) + \Phi_r U_r(k) \end{aligned} \tag{3.26}$$

Following [26], the solution of problems (3.25) and (3.26) can be computed as

$$U^* = \begin{bmatrix} U_h^* \\ U_r^* \end{bmatrix} = \begin{bmatrix} I & K_h \\ K_r & I \end{bmatrix}^{-1} \begin{bmatrix} L_h & 0 \\ 0 & L_r \end{bmatrix} \begin{bmatrix} Z_h \\ Z_r \end{bmatrix} \tag{3.27}$$

in which, defining

$$\begin{aligned} S_h &= (\Phi_h^T Q_{gt} \Phi_h + R_{gt,h})^{-1} \Phi_h^T Q_{gt} \\ S_r &= (\Phi_r^T Q_{gt} \Phi_r + R_{gt,r})^{-1} \Phi_r^T Q_{gt} \end{aligned} \tag{3.28}$$

the gains are computed as

$$\begin{aligned} K_h &= S_h \Phi_h \\ L_h &= [-S_h F_h \quad S_h] \end{aligned} \tag{3.29}$$

$$\begin{aligned} K_r &= S_r \Phi_r \\ L_r &= [-S_r F_r \quad S_r] \end{aligned} \tag{3.30}$$

and

$$Z_h = Z_r = \begin{bmatrix} z_{gt}(k) \\ y_{ref,h}(k+1) \\ y_{ref,r}(k+1) \\ \vdots \\ y_{ref,h}(k+N) \\ y_{ref,r}(k+N) \end{bmatrix} \tag{3.31}$$

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 $u_r(k) = U^*(1+N)$. But the human cannot be programmed, so a way of estimating z_ref is needed.

4 | Learning human intention for trajectory prediction

In this Chapter, it is shown how the proposed RNN+FC model is applied. Starting from the human intention identification we go through the iterative approach analysis. Finally, Transfer learning of the proposed model to other users/objects is presented.

4.1. Human Intention Estimation

In order to develop a suitable controller, it is very important to know the desired trajectory of the human. This is because the robot must provide adequate assistance to the human during the work it has to perform.

As also expressed in this work [31], we can collect in an equation the possibility of expressing the interaction between a robot and a human limb.

$$-C_h\dot{x} - K_h(x_{ref,h} - x) = u_h \tag{4.1}$$

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

$$x_{ref,h} = F(x, \dot{x}, u_h) \tag{4.2}$$

The function F, being non-linear and time-varying, is quite complicated. In fact, it depends on many factors such as the task to be performed, the user doing that particular task, the user's arm configuration, and many others. In addition, if we consider the fact that this research intends to allow the robot to carry objects, it is not sufficiently descriptive because it takes on meaning in correspondence with the contact between the robot end-effector and the human.

To overcome this problem, the solution found is to develop a neural network because excel-

lent at approximating complex non-linear systems with high uncertainties. In particular, it is used the RNN cascaded with a fully connected(FC) part that learns human behavior and provides the robot with the necessary information to assist the human. The main advantage of RNNs is that they take into account previous events allowing information to persist. Among the various RNN types, the LSTM (Long-Short Term Memory) is used for this specific task as it performs better in solving problems involving long time series.

Therefore, the proposed method aims to identify and predict, over a finite rolling horizon, the desired human trajectory, given the past history over a finite horizon. This means that the system takes the last k instants of time as input and predicts N instant of time of human's trajectory. In particular, T is the current time, and the input data are collected in an interval from $T-k$ to T.

As input, we have the actual robot position and velocities:

$$x = [x, y, z, R, P, Y]^T \tag{4.3}$$

$$v = [\dot{x}, \dot{y}, \dot{z}, \omega_x, \omega_y, \omega_z]^T, \tag{4.4}$$

the force exerted by the human:

$$[f_x, f_y, f_z, \tau_x, \tau_y, \tau_z]^T \tag{4.5}$$

and the nominal robot trajectory:

$$x_{ref,r} = [x_{ref,r}, y_{ref,r}, z_{ref,r}, R_{ref,r}, P_{ref,r}, Y_{ref,r}]^T \tag{4.6}$$

The complete control framework is visible in figure 4.1. The estimated human position and the robot reference are the two inputs in the MPC block. The robot reference comes from a block called motion planner which is the system that provides the trajectories for the robot to follow; while the human estimation is what we train from the RNN. The two cost functions, that we imposed in the previous paragraph, also are part of the MPC block, precisely the robotic cost function and human cost function. Finally, the output we have from the MPC, as the virtual robot effort, with the measured human, became the input for the impedance control. And through a kinematic inversion, we control the robot.

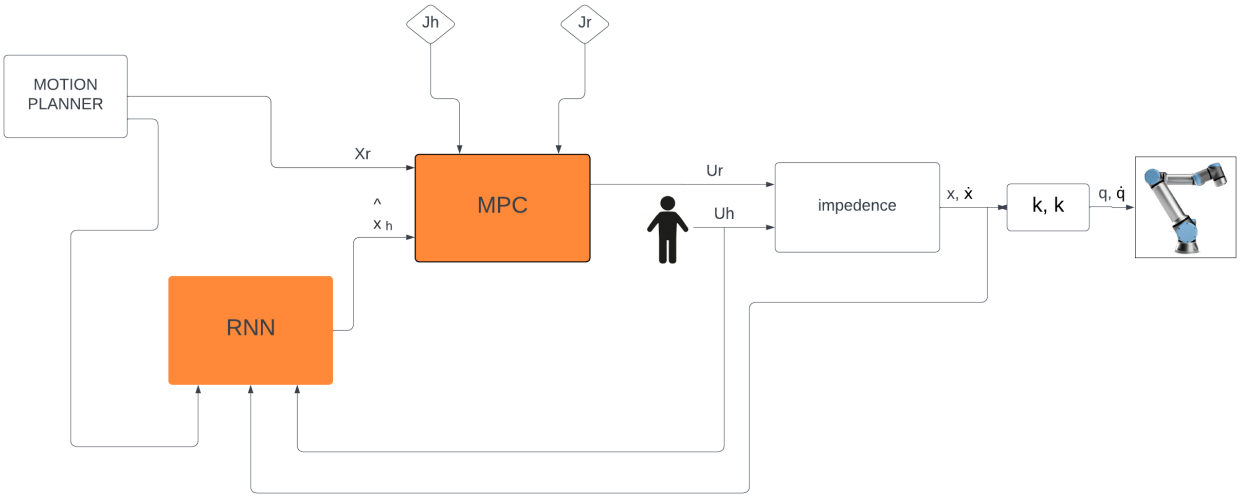


Figure 4.1: Structure of the system

4.1.1. Iterative Training

As we have seen in the section on Game Theory Control 3.2, and thanks to the equation used in 3.31 we can formulate the control as expressed in 3.27 in a cooperative Game Theory approach. We now know that to tune the parameter of the RNN, the machine learning approach requires a procedure that trains the model in order to obtain the input-output relationship as the function in 4.2. So before using the model we have to acquire the data needed to train the model and to do this, we have to collect the data without using the model. The dataset we create is called D_0 and it is done assuming that this equation applies:

$$\hat{x}_{ref,h} = x_{ref,r} \tag{4.7}$$

where $\hat{x}_{ref,h}$ stands for the predicted trajectory of the human and $x_{ref,r}$ the actual trajectory done by the robot. That means that with no model loaded the robot has no element to predict the human trajectory. So, it assumes that the human predicted trajectory is equal to the robot one.

With this data collected, we can train the model we called M_0 where the "0" indicate that it is the iteration done with no model loaded.
It depends on the first data set only:

$$M_0 = M_0(D_0) \tag{4.8}$$

physical

This new model is used for collecting new data but the robot behavior is no more the same as the 0 one. In fact, the assumption $\hat{x}_{ref,h} = x_{ref,r}$ does not hold anymore because the robot now can predict $\hat{x}_{ref,h}$. The new assumption follow this equation:

$$\hat{x}_{ref,h} \neq x_{ref,r} \tag{4.9}$$

Now the robot, due to the fact that we change the parameters of the model, starts to predict the human trajectory, and so thanks to the new model the robot can assist better the human during its task.

The process that is being created follows an iterative procedure where the first step is the one already explained and so the next step is to collect a new data-set D_1, and train a second model on these data.

$$M_1 = M_1(M_0, D_0) \tag{4.10}$$

.

This procedure is done to find the best model possible and this process can be done K times:

$$M_K = M_K(M_{k-1}, D_{k-1}) \tag{4.11}$$

The iteration process can be stopped by a stop criterion *i.e.* the law that indicates at which iteration the model doesn't improve itself. To express this with math formulation we have to find a value that indicates the stopping criterion. An example of this could be the average of the Root Mean Square Error (RMS), computed as

$$e_{RMS} = \frac{1}{L}\sum_{T=1}^{L}\sqrt{\frac{1}{N}\sum_{K=T}^{T+N}(\|\hat{x}_{ref,h} - x_k\|^2)} \tag{4.12}$$

The stopping criterion can be expressed with this assumption

$$\|e_{RMS}^{k+1} - e_{RMS}^{k}\| < toll \tag{4.13}$$

We can summarize the procedure algorithm in this way:

Algorithm 4.1 The iterative procedure proposed for the training

Data: Sample Records
Result: Trained Model M
Record data-set D_0 without any model;
Train model M_1 with data-set D_0
while $||e^{k+1}_{RMS} - e^{k}_{RMS}|| < tol$ **do**
    Record data-set D_k with model M_k;
    Train model M_{k+1} with data-set D_k;
end while

4.1.2. Transfer Learning

In this part, we discuss how we can make the model as general as possible as soon as it is trained because the equation written before in 4.1 depends on a human to human.

In addition to this, it describes the behavior of the human in the specific case we have tried so far, i.e. in the situation where the man grabs the robot tip. The situation in fact changes, as the values of the trained model change, when the person using the robot is different or when it is no longer the human directly grasping the robot but the robot assists the human in carrying large objects.
An example of this second case is the possibility of maneuvering an object in two in particular where the object is held by the robot on one side and follows the movement of the human holding it on the other side.

To do this, the model we trained in the process before, is no longer suitable and therefore needs to be reformulated and requires further training. Doing all this, takes a lot of time, as we have to collect new data, and re-train the model several times until the tolerance we set in the stop criterion satisfies us.

The solution to improve this long model is therefore to use the theory of *transfer learning*, through a model, based on RNN+FC, for the new user/object. A representation is expressed in the next figure 4.2

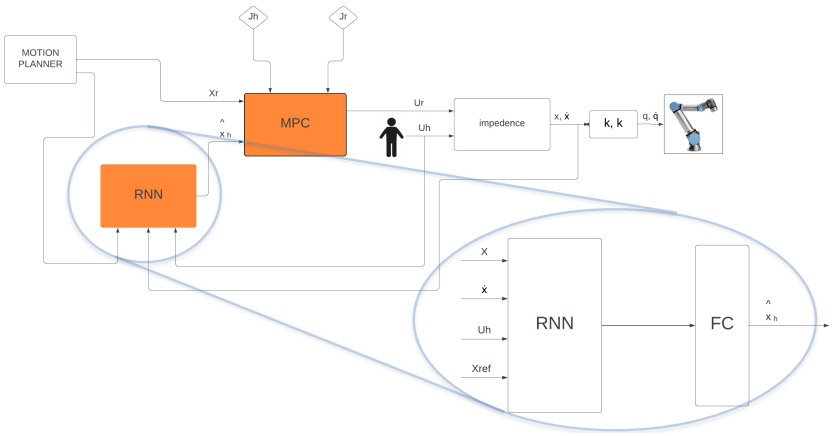


Figure 4.2: Representation of RNN+FC inside the control scheme

This procedure, therefore, allows a considerable reduction in process time. This is because we only perform a single iteration, due to the fact that we start from a model that has already been trained and because we collect less data

As we saw in section 2.3.3 of the state of the art, there are various strategies that can be adopted for transfer learning. The process adopted in this thesis falls into a widely used case in NLP and computer vision, in which what we do is "freeze" some layers of the model we are using and train only a few more layers, which we keep free to adjust their parameter. In fact, by doing this, the training time is greatly reduced, because if before we had to tune all the model's parameters, now, in order to adapt our model to the new user, we only have to re-tune a much smaller number of layers.

Specifically, in our case, we freeze the part of RNN, and the part we consider the *fully connected* is the part we train again. It is reasonable to think that the RNN part learns the pHRI features while the FC part is responsible for adapting to the new user/object. The difference could be that the RNN learns that, when the human imposes a force in a certain direction has a certain type of consequence. While the FC looks at how much force the specific user imparts to the robot in order to personalize the model as much as possible. The difference lies in the fact that the RNN part learns the pHRI features while the FC part is responsible for adapting our user or the new object in order to learn the new interaction with it. The RNN for example learns that the fact that the human imposes a force in a certain direction has a certain type of consequence while the FC looks at how much force the specific user imparts to the robot in order to personalize the model as much as possible.

So using the transfer learning approach, we define the M_{TL} model equation as follows:

$$M_{TL} = M_{TL}(M_k, D_k) \tag{4.14}$$

The algorithm in this case can be summarised in the following way:

Algorithm 4.2 The iterative procedure proposed for the training with Transfer Learning

1: **Data:** Sample Records
2: **Result:** Trained Model M
3: Record data-set D_{TL} based on the model trained in Algorithm 4.1
4: Train model M_{TL} with data-set D_{TL}

It is interesting now to see the final structure of the proposed model as in fig 4.1.

5 | Experimental Results

The presented method is evaluated with simulations and real experiments. This chapter analyzes the performance of the dMPC controller for the pHRI approach of the cooperative assistant. Then, the RNN+FC model is evaluated, showing its performance for the iterative training and Transfer Learning procedures. Finally, a cooperative transport application that uses the RNN+FC module and the dMPC is presented, compared with a standard controller for pHRI.

5.1. Experimental Setup

The experimental setup is presented here. The robotic platform is a UR5 robot from Universal Robots, equipped with a Robotiq FT300 sensor mounted at the tip for measuring the human interaction force. The robot has 6 degrees of freedom, and it is controlled by an external computer at 125 Hz. The robot arm is mounted on a fixed base. It weighs about 20 kg with a payload of up to 5 kg and 850 mm reach. It is a versatile and flexible collaborative robotic arm that can combine very compact size and high performance

The collaboration and physical interaction with the human makes use of two different tools. One visible in figure 5.1a, allows being grasped by the human directly at the end effector of the robot. This tip is specifically made by a 3D printer, so it is made of resistant but very lightweight plastic. It permits free movement for the robot as the human can easily guide it according to his intentions. Collecting data for training the model results easier and faster with this handle, if compared to the application scenario that involves the grasping of a large object. The second tool is a suction cup, as shown in figure 5.1b, used explicitly for carrying large/heavy objects. It is mounted at the end effector and connected to an air generator to create the vacuum.

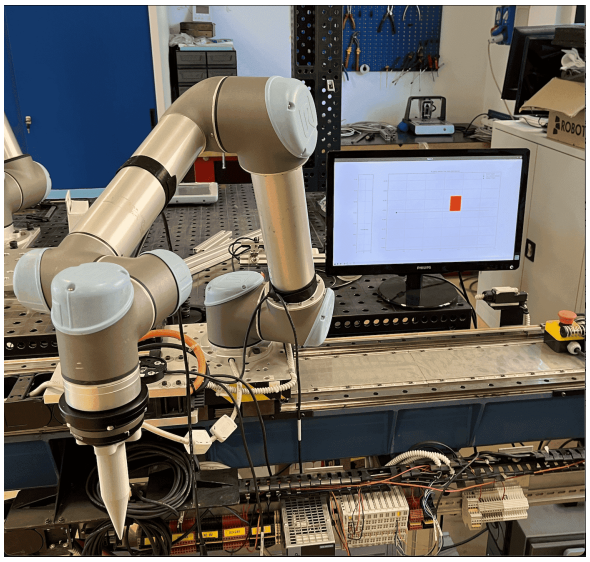
The user can stand or sit in front of the robot and handle it.

The robot's nominal trajectory, defined by the motion planner, is defined offline by an external computer.

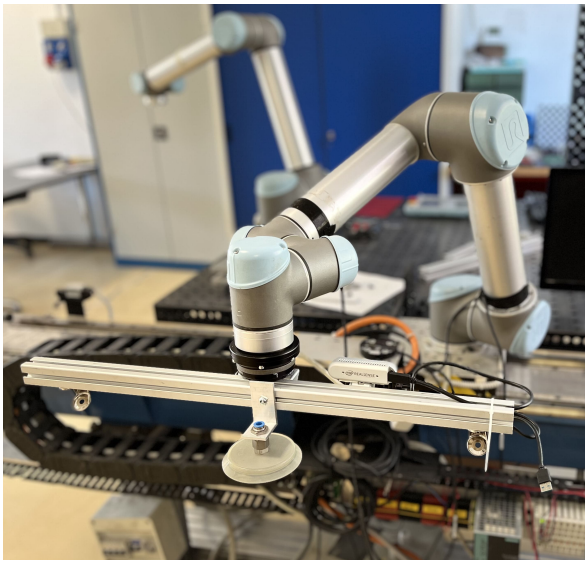
An external monitor positioned near the robot displays the trajectory and the obstacle.

This is used only for the validation procedure of the model as during the application case no trajectory and no obstacles are shown on the monitor.

The setup with the two tools is visible in figure 5.8.



(a) Experimental setup used for data collection and Iterative Training procedure.



(b) Experimental setup used for TL and object co-manipulation.

Figure 5.1: The experimental setup in the two configurations. The UR5 equipped with the FT sensor is visible. In the background, the monitor used to display the trajectory and the virtual obstacle.

The control framework is based on ROS nodes [66]. Three different ROS nodes are used: one to control the robot, one which implements the motion planner, and one where the RNN+FC model runs to predict human intentions. The ROS nodes are written either with python or C++. In particular, the controller node implementing the dMPC is written in C++ code, while the RNN+FC and the motion planner are implemented in python nodes. The analysis of the result is done using Matlab scripts. The Neural Network is implemented with the *PyTorch*[42] library. All the computations are performed on a standard ASUS laptop with an Intel i7 and with an Nvidia GeForce 1050 GPU running on a Linux/Ubuntu system operator.

5.2. Design of Experiments

For the evaluation of the RNN+FC model experiments are performed on the x–y plane, involving only two dofs. The impedance control parameters in (3.1) are set as follows:

$$M_i = \begin{bmatrix} 10 & \\ & 10 \end{bmatrix} \quad C_i = \begin{bmatrix} 100 & \\ & 100 \end{bmatrix} \quad K_i = \begin{bmatrix} 0 & \\ & 0 \end{bmatrix} \tag{5.1}$$

The choice of setting K_i to zero is typical in manual guidance applications, as in [32].

The two cost functions parameters in (3.15) and (3.16) of the two players are set as:

$$\begin{aligned} Q_{h,h} = Q_{r,r} &= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0.0001 & \\ & & & 0.0001 \end{bmatrix} \\ Q_{h,r} = Q_{r,h} &= 0^{2\times 2} \\ R_h &= \begin{bmatrix} 0.0005 & \\ & 0.0005 \end{bmatrix} \end{aligned} \tag{5.2}$$

In particular, the human cost function parameters $Q_{h,h}$,$Q_{h,r}$ andR_h are recovered via Inverse Optimal Control (IOC) as in[13], and an average value is used. The robot parameters $Q_{r,r}$ and $Q_{r,h}$ are set equal to the human's to mimic a person except for R_r.

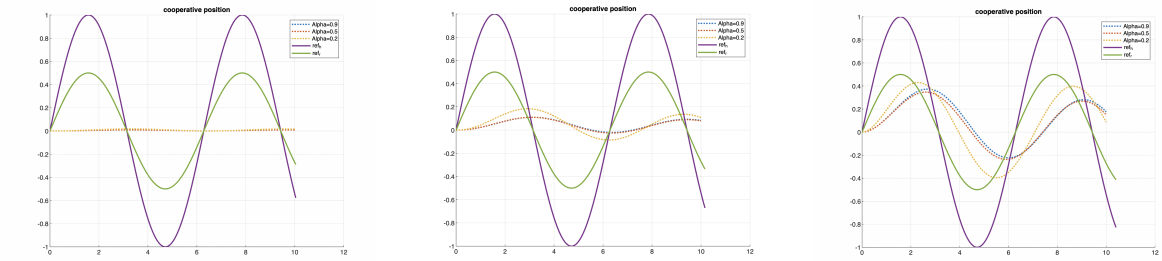
The value of the parameter α and R_r, will be discussed in the next section, where an analysis of the dMPC behavior is proposed by varying this value, according to (3.17) and (3.22). Different values are selected for the evaluation of the RNN+FC model and for the application scenario. In the first case, we set $\alpha = 0.8$ and $Rr = \begin{bmatrix} 0.0005 & \\ & 0.0005 \end{bmatrix}$ while in the second, we change a little based on the fact that more assistance can be provided, as discussed in 5.2.1, by setting them equal to $\alpha = 0.9$ and $Rr = \begin{bmatrix} 0.0001 & \\ & 0.0001 \end{bmatrix}$.

5.2.1. dMPC performance analysis

This section analyzes, with simulations, the dMPC performances according to different tuning parameters. Such an analysis allows for defining the parameters presented in the method 3.2 that better suit the robotic assistance in pHRI tasks. This section focuses on the dMPC performances varying those parameters that are free to choose in the pro-

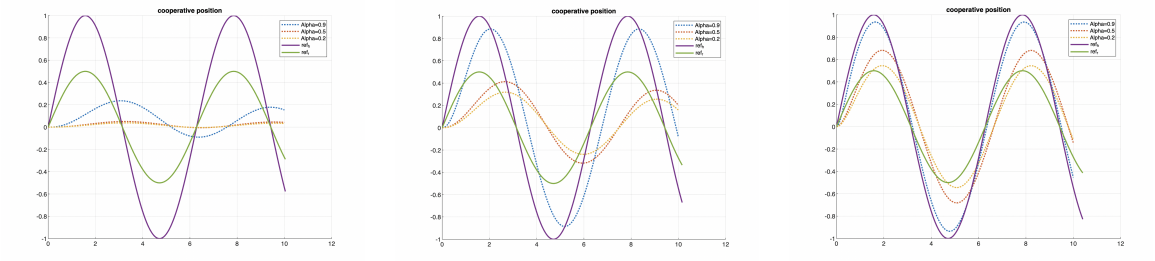
posed pHRI application. In particular, We analyze the system behavior according to the variation of α, R_r, and the prediction horizon $\mathcal{H}$. Indeed, these are the parameters that can be tuned, while the other ($Q_{h,h}$, $Q_{h,r}$, R_h) depend on the human and cannot be set arbitrarily. In particular, We simulate the system with $\alpha = \{0.2, 0.5, 0.9\}$. Low values of α correspond to the case where the shared cost approximates the robot's cost. On the contrary, high values of α correspond to the case where the shared cost approximates the human cost. We also simulates for values of $R_r = \{0.01, 0.0005, 0.0001\}$. The lower R_r is the higher the robot's assistance. Finally, we simulate different values of the prediction horizon as $H = \{0.04, 0.16, 0.4\}$ seconds, or $H = \{5, 20, 50\}$ samples, with a sampling period of 0.008 seconds. This simulation is performed to check the reference tracking capabilities of the dMPC framework, varying the prediction horizon. The test is conducted assuming a sinusoidal signal as a reference. The two references are defined as $x_{ref,h} = sin(t)$ and $x_{ref,r} = 0.5\, sin(t)$.

The figure in 5.2, 5.3, and 5.4 shows the result of this test.



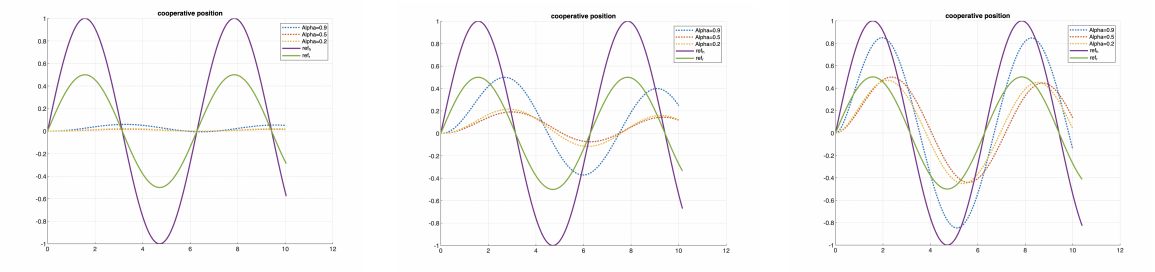
(a) Prediction Horizon of 0.04s. (b) Prediction Horizon of 0.16s. (c) Prediction Horizon of 0.4s.

Figure 5.2: Evaluation of the dMPC tracking performances at different values of α with $R_r = 0.01$ and on different prediction Horizon.



(a) Prediction Horizon of 0.04s. (b) Prediction Horizon of 0.16s. (c) Prediction Horizon of 0.4s.

Figure 5.3: Evaluation of the dMPC tracking performances at different values of α with $R_r = 0.0001$ and on different prediction Horizon.



(a) Prediction Horizon of 0.04s. (b) Prediction Horizon of 0.16s. (c) Prediction Horizon of 0.4s.

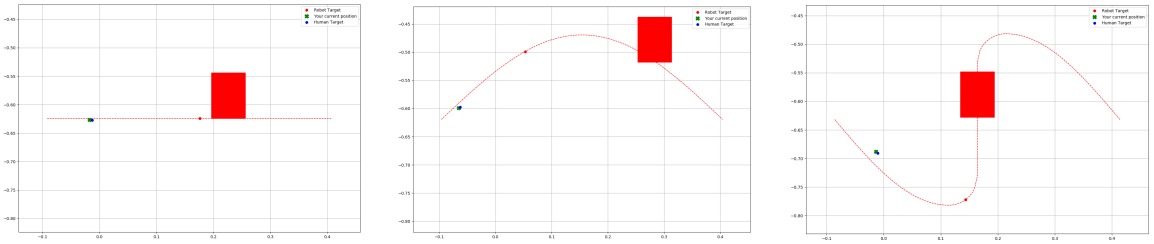
Figure 5.4: Evaluation of the dMPC tracking performances at different values of α with $R_r = 0.0005$ and on different prediction Horizon.

As we see from the figure with $\alpha = 0,9$ the curve follows the human reference more, so we have a more assistive controller that follows the human intention. With $\alpha = 0,2$, according to GT, it should be the human who puts much effort to help the robot track its reference. This is impossible for us because human does not know the reference of the robot. Moreover, it is unnatural for the human to assist the robot, so low values of α should not be chosen. Setting the R_r too high we have bad performance in every α choice, while the difference between the $R_r = 0.0001$ and $R_r = 0.0005$ is that the lower it is, the more responsive the robot behavior becomes, possibly leading to jerky motions for too small values. Varying the prediction horizon, we see that we have better results when we can predict as far ahead as possible in time.

5.2.2. Human Intention prediction evaluation

This section describes the procedure for data collection and RNN+FC model training both for the iterative and the TL training. All the datasets contain data on the robot's actual poses, velocities, reference robot's trajectory, and interactive force. The human force is measured at the robot tip via the FT sensor. The robot's nominal trajectory defined by the motion planner is defined offline and commanded in real-time. The data collected are sampled at 0.008 seconds, as this is the sampling time of the robot's controller. To evaluate the model's performances, preliminary experiments are performed only on the X-Y plane.

Three different nominal trajectories are computed, defined as linear, curved, and sinusoidal. They are represented in figure 5.5a, 5.5b, and 5.5c, respectively.



(a) Linear Trajectory. (b) Curve Trajectory. (c) Sine Trajectory.

Figure 5.5: The three trajectories used in the collecting phase

A single trial consists of following the given trajectory that appears on a monitor, starting from the initial point to the endpoint, while data are collected. The human has to avoid a virtual obstacle (the red box in the figures) that appears randomly along the path. A complete dataset, for the Iterative learning phase, is composed of 60 trials, 20 for each of the three trajectories.
For this evaluation, the same person (the thesis author) performed four iterations to train the model. For the RNN+FC iterative training procedure, 4 iterations are performed despite the stop criterion defined in 4.1.1 being reached after the first iteration. This is because we want to analyze if the model can produce some improvements and after how many iterations these improvements stabilize.

The LSTM model used is composed of 3 layers with 250 hidden nodes, and the FC is composed of two connected layers. Some of the parameters (optimizer, learning rate) of the RNN+FC model are obtained with Optuna [3]. The model is trained for 25 epochs with a size batch of 64 and a learning rate initially set at 0.001. The neural network used works with 125 times instant precedent step as input. The model predicts 50 instant steps ahead of the current state.

To evaluate the prediction model, we set the parameters with $\alpha = 0,8$ and $Rr = diag(0.0005)$

α, in this case, is chosen to allow sufficient assistance but also to allow the robot to recover the position of the robot set-point autonomously. The value of R_r, is set equal to the human's to mimic collaboration with another person.

The first dataset D_0 is collected with no prediction model loaded on the robot controller, assuming $x_{ref,h} = x_{ref,r}$. The robot moves along a path, and the human is grasping the handle at the end-effector. The human has to impose a force to deviate from the nominal path to avoid the obstacle. The robot doesn't know where the obstacle is and its dimension. For the user, the obstacle appears on the monitor along the path. It has the

same dimensions every trial but it randomly changes the position.

The training setup is visible in figure 5.6.

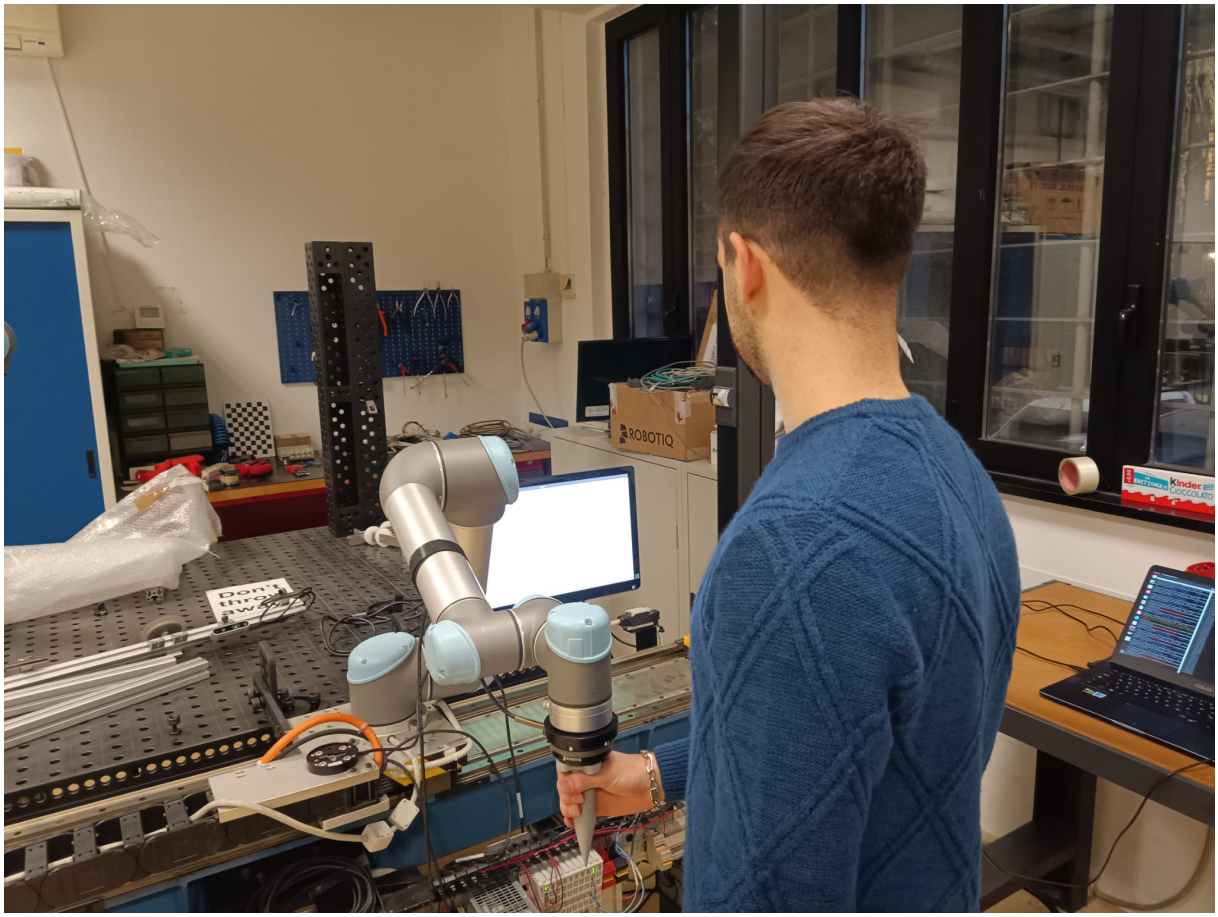


Figure 5.6: Setup during the data collecting phase

A ROS node collects the data during the trial and stops recording at the moment it arrives at the endpoint.

Figure 5.7 shows what is visible to the human during the execution of a task, regarding the linear trajectory, in this case. The human knows the current position, the nominal path, and the target pose. Moreover, human knows the obstacle and can avoid it.

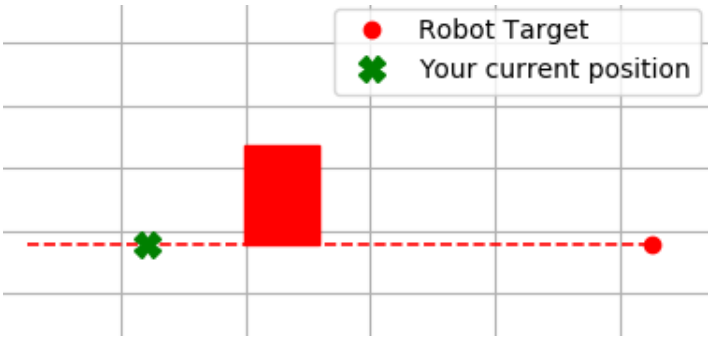


Figure 5.7: Monitor visualization of the trajectory to be followed with the obstacle and the targets, during the collecting phase when no model is loaded.

The data collected give information about the current position, current velocity, human wrench, and robot reference position. To be usable for our training, the data are rearranged. We consider only the columns of our interest, cutting the superfluous ones, and we align the data to set time equal for everyone. To make the training phase feasible, we pass the data into a function that normalizes them in a range between 1 and -1.

For the training phase, we divide the data set into two parts, so that one part is dedicated to testing and one to training.

We decided to dedicate 20% to the test part and the remaining 80% to the training part. The RNN+FC model is then trained and its parameters are updated. The model is then saved, indicating the specific iteration at which it was made.

For the first training, the weights of the neural network are initialized according to [25], while in the following iterations, the new model's parameters are initialized with the previous model ones, so it is sufficient to initialize a network with the previous parameters. This procedure is then repeated in succession several times to improve the model's capabilities and performance.

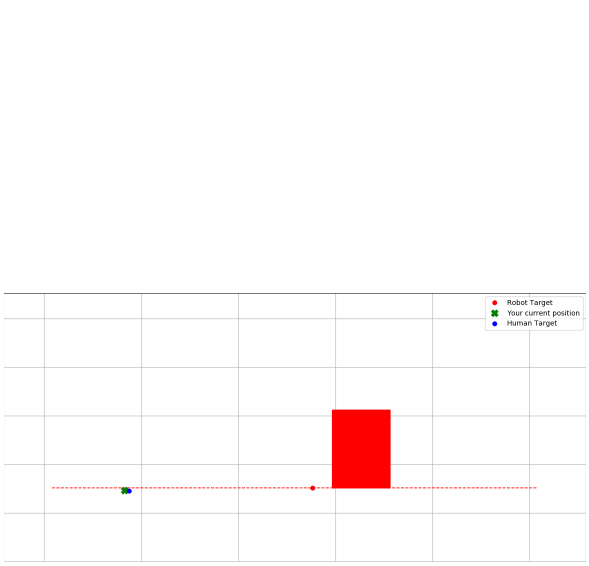
During the training phase we calculate, for each epoch, the loss function of the train and the test so that we can compare the performance of the two. The loss function is calculated as follows:

$$Loss = (y - f(x))^2 \tag{5.3}$$

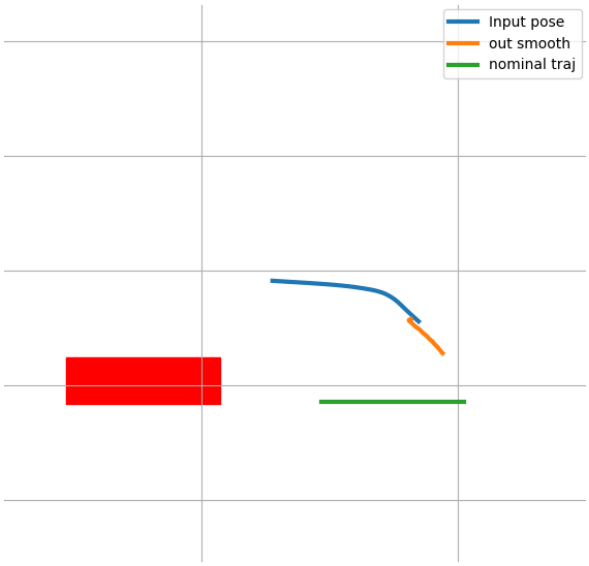
Where y is the nominal part and $f(x)$ is the predicted value.

Define the first model train as M_0, and with D_0 the dataset collected. With the subscript $_0$ we denote the model trained and the dataset collected with no model loaded. The following takes the name of the iteration we proceeded. The new dataset collects D_{N+1} are created with the model M_N loaded.

The collection procedure is done in the same way as described in the first case. Differently from the first data collection is that the assumption $x_ref = x_h$ does not hold anymore, but the RNN+FC model now predicts the x_h reference. In figure 5.8a is visible as a new target in blue but also in the live simulation graph, figure 5.8b, we can see the predicted horizon which represents each trajectory point's real-time prediction.



(a) Trajectory to be followed with the obstacle and the targets.



(b) Representation of the prediction horizon, shown in real time during the collection.

Figure 5.8: Monitor visualization during the collecting phase when the model is loaded.

From now the train is done based on the model created in the previous step.

At the end of the iterative training, we expect that the model has improved the prediction performance. Despite this, the iteration is done only by the same user and on three trajectories.

To adapt the model quickly, TL performances are evaluated. The procedure of TL is done on different users and adapts to a co-manipulated object and new trajectory.

The procedure to collect, clean the data, and train the model are the same. In all cases, the first data are collected based on the last model trained by the author, which here we called M_{prev}.

The difference is that only 15 trials are performed in total, 5 for each trajectory. The training phase is done following the procedure defined in 4.1.2 and modifying only the weight that corresponds to the FC layers. After that, we collect some new data to compare the prediction of the model before and after TL.

First of all, we evaluate the TL improvements for a new trajectory.
The data are collected like before, with the handle mounted at the end effector. We collect data on 15 trials done on the new trajectory represented in the figure 5.9.

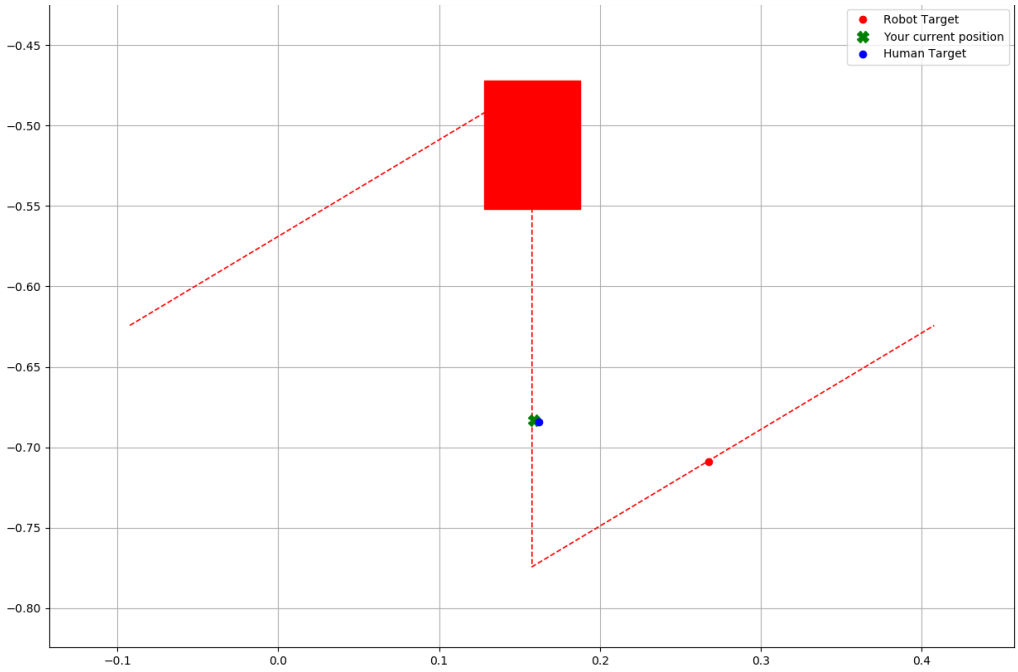


Figure 5.9: Z trajectory used in the application of TL.

After that, we tested the TL on 5 different users. The experiments are the same but taken on the first three trajectories. In particular, we collected data from 5 trials for each trajectory, and then we trained the model.

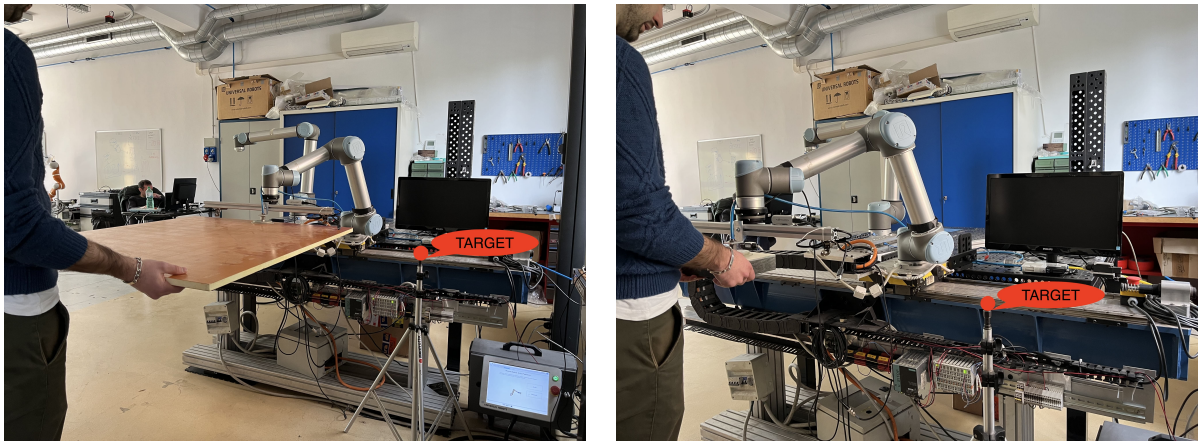
To expand the case of studies of the TL, we use a co-manipulated object, the wooden board. In this case, the experiments take a different setup as we substituted the gripper with the vacuum gripper able to support the object. In this case, we have some differences, as the force given by the human is not more centered on the end effector, and we have some additional Inertia. Also, here, we collect 15 trials divided into five each for the 3 trajectories. The collection procedure and setup are shown in the figure 5.10.

Figure 5.10: Data Collection using the co-manipulated object

5.2.3. Application scenario with large/heavy objects co-manipulation

After validating the model, we applied it to a real case application. The application consists of bringing two different objects to a target point. For this application, we use a wooden board of 106 x 82 x 2 cm in length, and a lumped wights object of 18 x 15.5 x 5 cm in length.

The setup is represented in figures 5.11a and 5.11b. In this case, the target is not represented in a monitor but is present in the real setup.



(a) Setup with the wooden board. (b) Setup with the lumped object.

Figure 5.11: The two setups used for the application scenario with the target point.

We define a start point and an end one, using the vacuum gripper to hold the object and we keep the object on one side to try to reach the final point in a more precise way.

It is interesting to analyze both as the first has a bigger dimension but is lighter, while the second one is heavier but more compact. The experiments are done avoiding a real obstacle put between the two target points. The task is represented as a real case as the movement of heavy/large objects to be positioned precisely at the target point. The co-manipulation of heavy/bulky objects is selected to simulate industrial tasks that may require co-manipulation.

The experiment was done by collecting at first 15 trials that are useful for the TL, and after it, the model is adapted to the new object. The gripper used is the same one used for the application of TL with an object, so a vacuum caption was applied at the end effector. As the objects are very different in terms of weight and dimension, two different TL training are done to better adapt the model to the object.

To see that the object co-transportation is better we compare the controller we have implemented with the other two standard control.

The difference in these two control takes into consideration the equation in (3.1). In particular, in Impedance Control (IMP) we take the equation and remove the robot assistance as u_r resulting in:

$$M_i\, a(t) + D_i\, v(t) + K_i\, \Delta x(t) = u_h(t) \tag{5.4}$$

In the MG from the equation (3.1) we don't have the u_r and K_i and formula result as:

$$M_i\, a(t) + D_i\, v(t) = u_h(t) \tag{5.5}$$

We will see the force applied during these two application types and the precision to reach the final point. The experiments done are exactly the same, both using the vacuum gripper.

5.2.4. Performance Indexes

To evaluate the performances, some indices are defined. To evaluate the method during the Iteration done by the author the e_{RMS}(Root Mean Square Error) and e_{MAX}(Maximum Expected Error) should be considered.

The e_{RMS} is calculated as follows:

$$e_{RMS} = \frac{1}{L}\sum_{T=1}^{L}\sqrt{\frac{1}{N}\sum_{K=T}^{T+N}(\|\hat{x}_{ref,h} - x_k\|^2)} \tag{5.6}$$

where $\hat{x}_{ref,h}$ is the predicted human intention, x_k the measured poses, L is the length of the trajectory, and N is the prediction horizon.

The e_{MAX}, instead, is calculated as finding, first, the maximum value of the prediction horizon in one point as:

$$e_{MAX,i} = \max_{i \in L}\{\|\hat{x}_{ref,h} - x_k\|\} \tag{5.7}$$

where $\hat{x}_{ref,h}$ is the predicted human intention and x_k the measured poses. L is the length of the trajectory. Then, the final value e_{MAX} is obtained by finding the maximum from all the values obtained before.

$$e_{MAX} = \sum_{i=1}^{L} e_{MAX,i} \tag{5.8}$$

with L being the length of the trajectory

The evaluation of the TL is done with the same index of the iterative comparison. First, we evaluate the model's capability to predict human intent with a new trajectory. After that, we propose TL to let the model learn the new trajectory and compare the errors. As for the previous case, different prediction horizons are evaluated.

Finally, the time required for each iteration's dataset collection and model training is compared with the time needed for the TL.

An interesting measure of the assistance provided is related to the force exchanged by the human. To assist humans, the robot should allow reduced interaction forces while collaborating. The indices f_{RMS} is based on the force root means square, defined as:

$$f_{RMS} = \sqrt{(f_x^2 + f_y^2)} \tag{5.9}$$

where f_x and f_y are the force applied in each direction during all trials. The value of the index is a mean value taken from three different trials.

Moreover, as the goal is to bring a co-manipulated object to a target point, it is interesting to check the precision of reaching the target. We measure the current position with the target point as:

$$\sigma_i = \|x_{current} - x_{final}\| \quad i \in L \tag{5.10}$$

with σ that gives the deviation from the point, $x_{current}$ the current value, x_{final} the final point, and L the length of the trajectory from starting point to the endpoint.

The final value we use to compare is calculated by taking the value of σ when it is in around $\pm\ \varepsilon$, arbitrarily decided and equal to 0.025cm, and finding the mean value over τ:

$$\sigma = \frac{\sum_{i=1}^{\tau} \sigma_i}{\tau} \tag{5.11}$$

with τ equal to 3 sec.

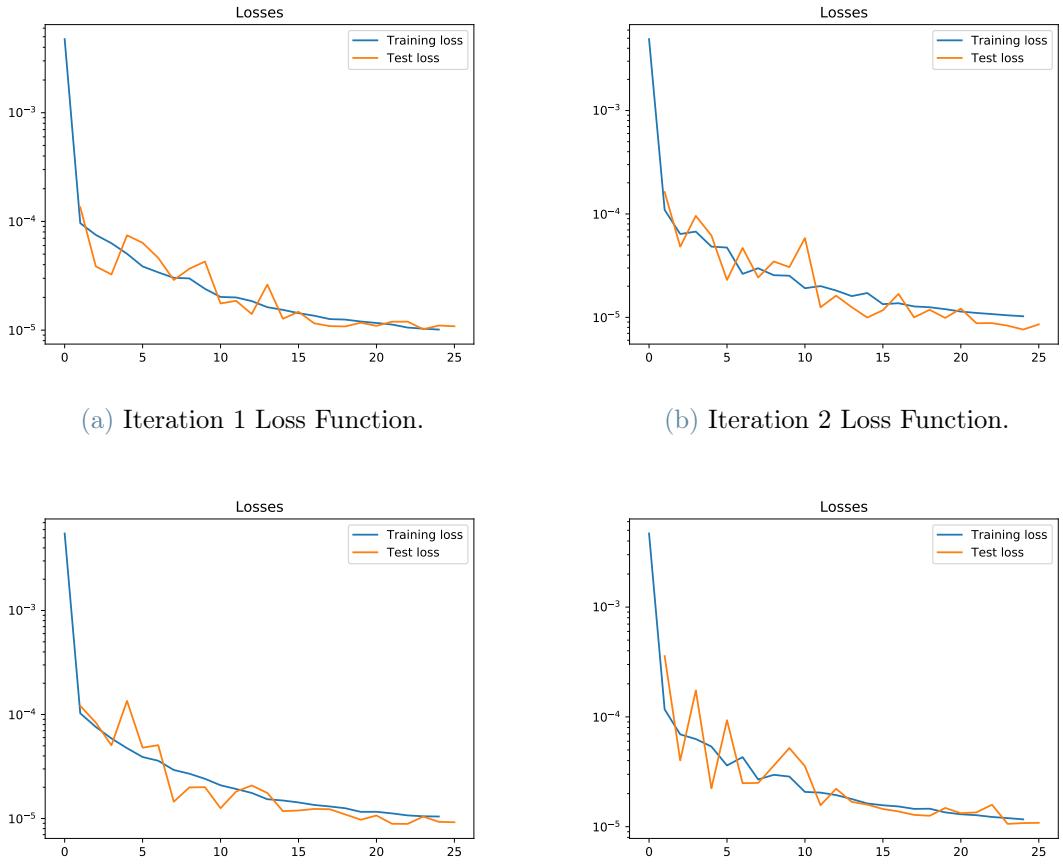
This index represents the distance variation in around ε of reaching and maintaining the final point.

5.3. Results

This chapter analyzes and presents results regarding the performances of the prediction model for the iterative process and the use of transfer learning. The performances are measured by the indices presented in chapter 5.2.4. Finally, results regarding the application scenario are presented with a comparison with two standard controllers for pHRI.

5.3.1. Model Evaluation

As during the training phase we have calculated the loss function of the test and of the train, it is interesting to see the behavior of this function. In figure 5.12 it is present the loss during each epoch for every 4 iterations. On the x-axis, we have the number of the epoch while on the y-axis the value of the loss function.



(a) Iteration 1 Loss Function.

(b) Iteration 2 Loss Function.

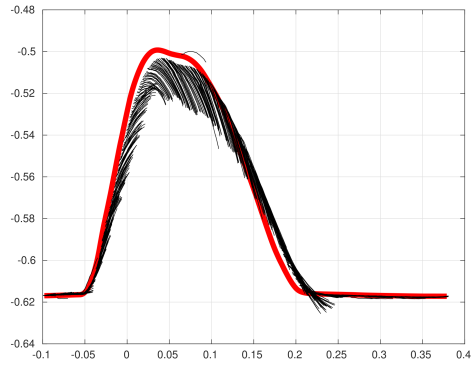
(c) Iteration 3 Loss Function.

(d) Iteration 4 Loss Function.

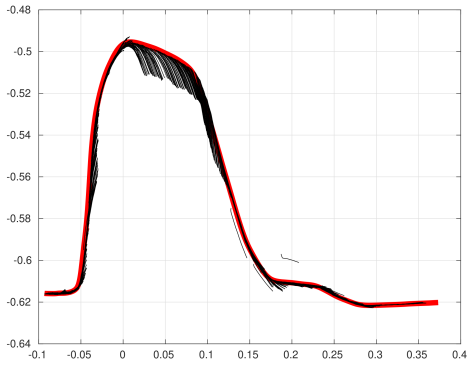
Figure 5.12: Train and Test Loss Function

As we can see, the loss passes from a value of about 10^{-3} to a value of about 10^{-5}. In fact, with 25 epochs we can say that they are enough to lower the loss function and attended an acceptable value that indicates to us the achievement of a good level of training. Furthermore, we can notice that after about 20 epochs the value of the loss function begins to stabilize. So for this case, we can assume that 25/30 epochs are enough for training the model However, the value of the single loss function does not allow a comparison between individual iterations. It tells us nothing about the quality of the time predictions but only about the goodness of training.

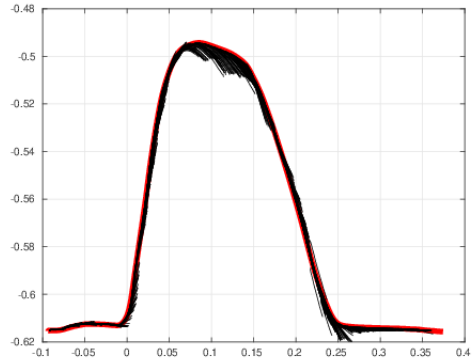
To analyze the improvements in the prediction during iteration we can start with a graphical comparison. We show in the next figure the trajectory that the robot has manipulated by the human and we represent for each point the prediction of the trajectory of the human 0.4 seconds ahead.



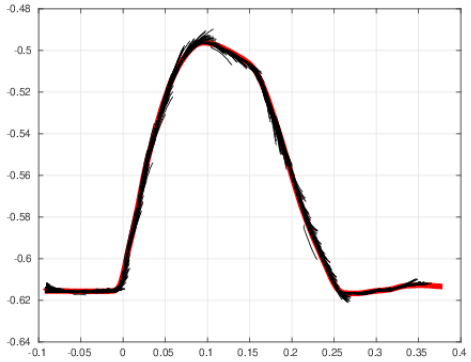
(a) Prediction with the model M0.



(b) Prediction with the model M1



(c) Prediction with the model M2

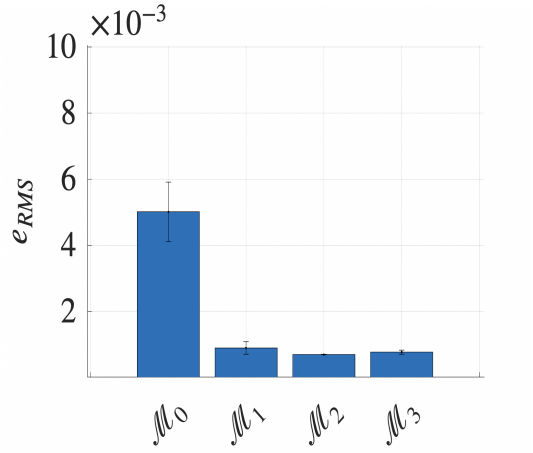


(d) Prediction with the model M3.

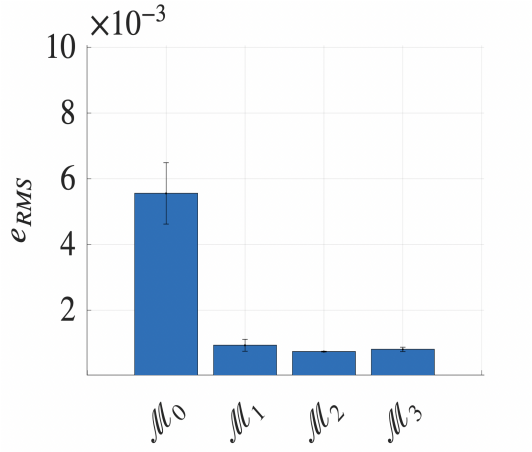
Figure 5.13: Comparison between the prediction at the various training iterations, with the maximum prediction horizon considered (0.4sec)

It is easy to see that the first (fig 5.13a), shows a very pronounced profile, a clear sign of the poor accuracy of the prediction. During the subsequent iterations, on the other hand, the profile becomes much less pronounced, showing us how the prediction follows the desired trend much more closely. Very clear in next as 5.13b, 5.13c and 5.13d

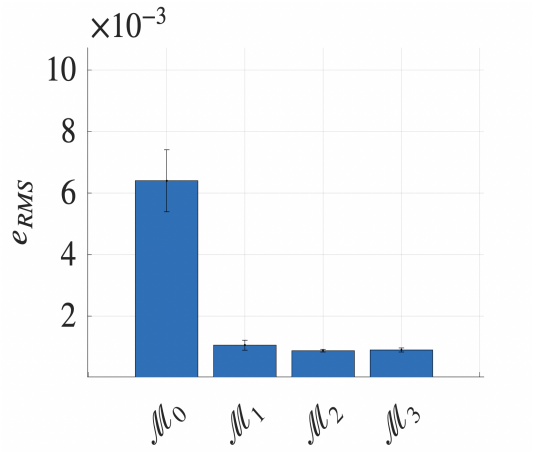
To give a value to this improvement we have introduced some indices in the section in 5.2.4. We start with e_{RMS}, and we evaluated this value comparing it with different time horizons to see its dependency.



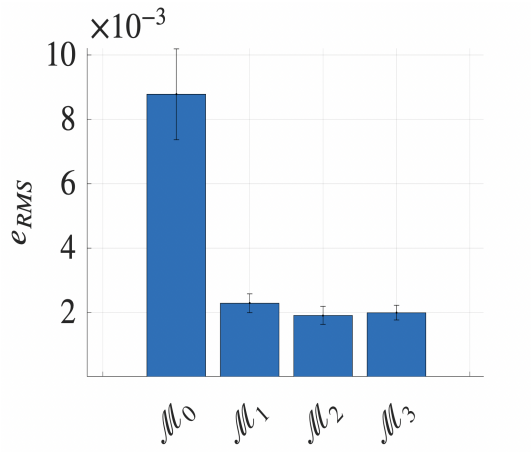
(a) The four models evaluation with prediction horizon of 0.04s.



(b) The four models evaluation with prediction horizon of 0.08



(c) The four models evaluation with prediction horizon of 0.16



(d) The four models evaluation with prediction horizon of 0.4

Figure 5.14: Comparison of the e_{RMS} for the four iterations with different prediction horizons.

The improvement is clear between the first iteration M_0 and the subsequent one. In particular between the first and the second, the difference is huge while from the third, and fourth, it's stabilized. The fact that the third iteration and fourth one are minimal indicates that another training iteration is not necessary, and the prediction model converges quickly. Regarding the different time horizons, we can see that it is more complex to predict long prediction horizons as the value of the error increases when we augment the prediction horizon. This is mainly because it is very complex to predict human deviations from the nominal trajectory in advance.

It is interesting to see also the value that the error of RMS, has on a single point. We

analyze on the linear trajectory in one trial. In the next figure, we have the four iterations during the time of trial, while on the y-axis we have the punctual value of the e_{RMS}

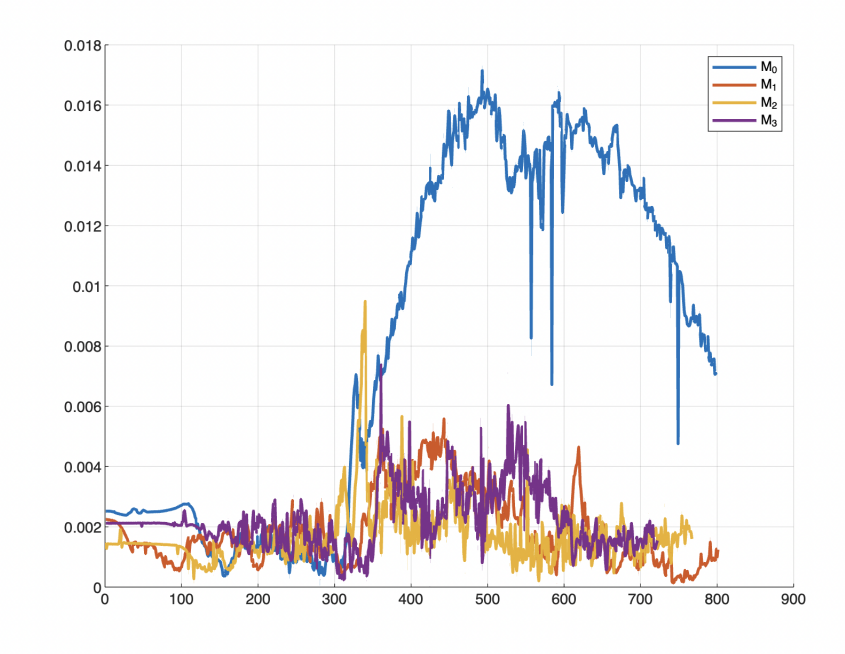
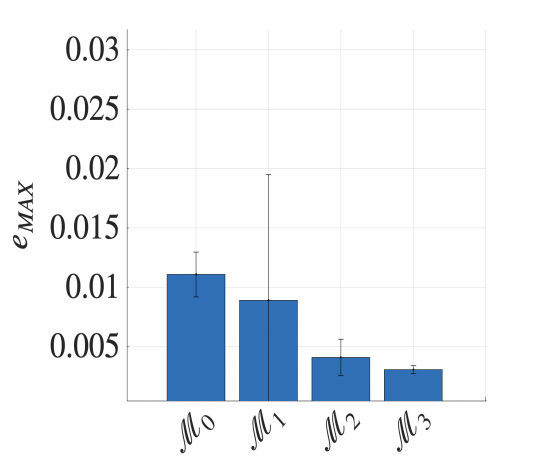


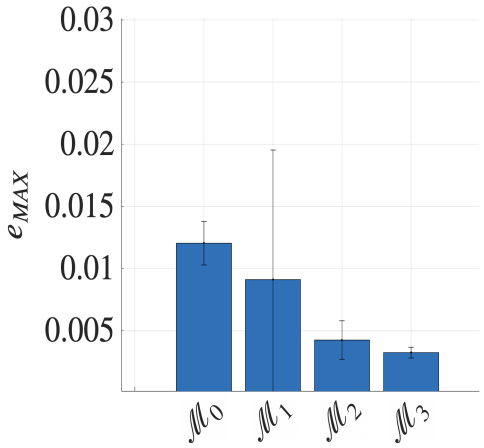
Figure 5.15: Punctual prediction error for the four iterations on the linear trajectory.

As confirmed by the indices that calculate the sum of the values taken for each point on the trajectory, we can see the improvement over the course of iterations, especially in the first one. In particular, we can see that the highest values are found just before the human starts the deviation from the nominal trajectory, and the robot cannot know it in advance (around timestep 300 on the x-axis).

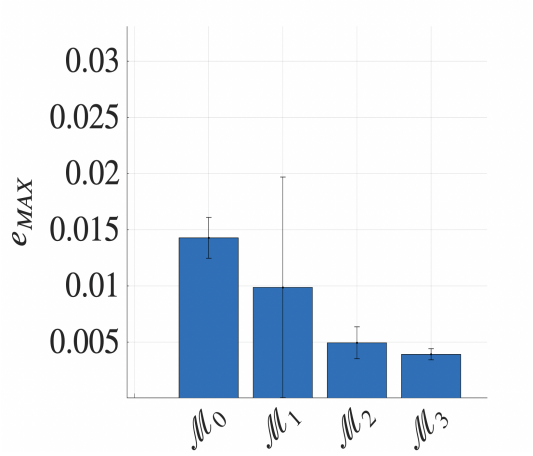
We also analyze the e_{MAX} for the four iterations, as in figure 5.16, based on the different prediction horizons.



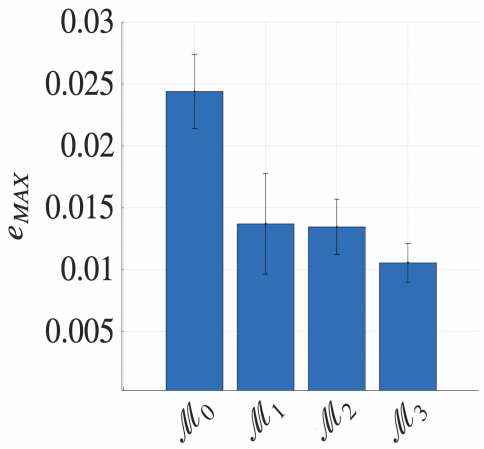
(a) The four models evaluation with prediction horizon of 0.04s.



(b) The four models evaluation with prediction horizon of 0.08



(c) The four models evaluation with prediction horizon of 0.16



(d) The four models evaluation with prediction horizon of 0.4

Figure 5.16: Comparison of the e_{MAX} for the four iterations with different prediction horizons.

The results, although with different values, confirm the improvement noted with the RMS index. The e_{MAX}, indeed, gives us some more information on the process. We can notice that it reduces only after 3/4 iteration and not only just after one. So, this value said that iterating the process multiple times can improve the model.

Also for this, we can see the value of each point on the linear trajectory:

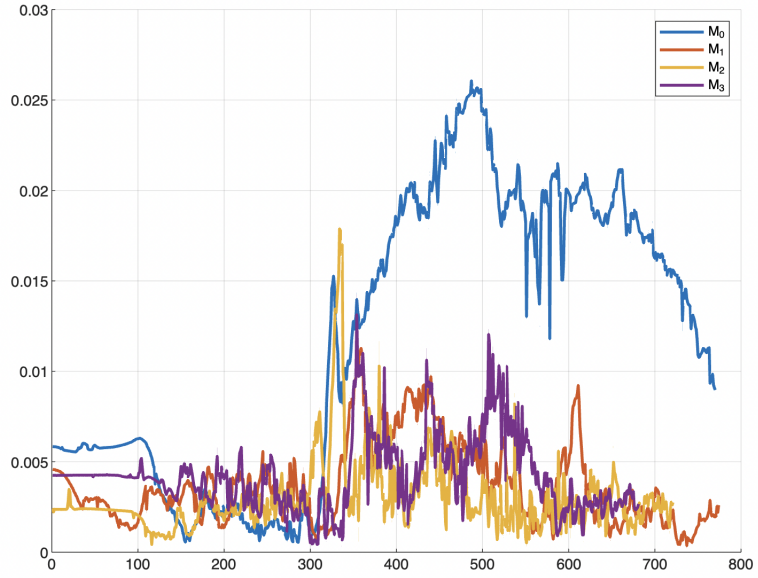


Figure 5.17: Punctual e_{MAX} for the four iterations on the linear trajectory.

Also here, the value follows the same considerations given by the e_{RMS}.

It is interesting to compare the time needed for all the processes, starting from the collecting data and the training of the model. Such a comparison is shown in table 5.1

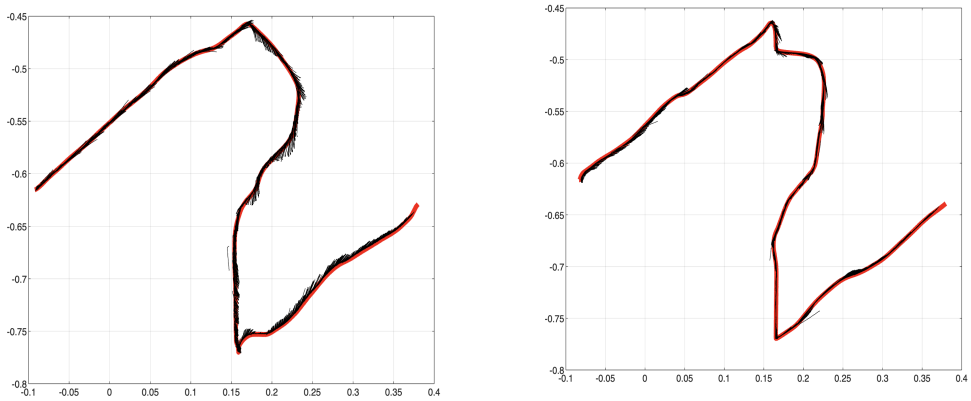
	Collecting Data	**Train the model**
Iteration 1	50 ± 10 min	50 ± 5 min
Iteration 2	60 ± 10 min	40 ± 5 min
Iteration 3	60 ± 10 min	40 ± 5 min
Iteration 4	60 ± 10 min	40 ± 5 min

Table 5.1: Comparison between the different times it takes to collect data during the training phase on each iteration.

5.3.2. TL Evaluation

The same indices are used to analyze the improvements done using transfer learning.

The first study was on the new trajectory, represent in figure 5.9, always done by the author. As before we start with a graphical comparison as in figure 5.18



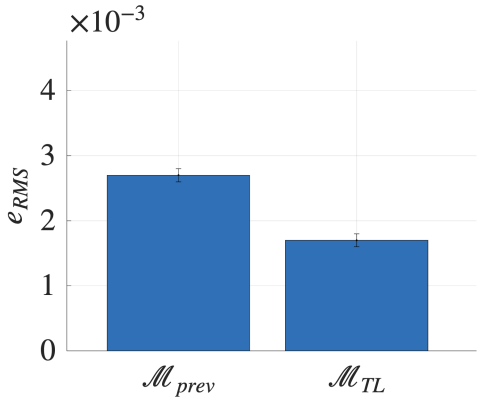
(a) Prediction with the model M_3.

(b) Prediction with the model M_{TL}.

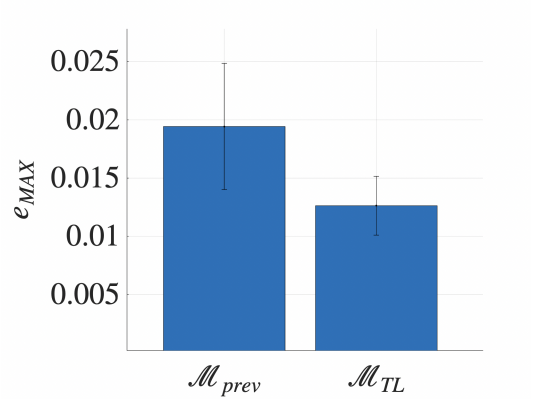
Figure 5.18: Comparison between the prediction at the two training iterations, of the z trajectory, with prediction horizon as 0.4sec

The difference is less pronounced but still, an improvement in the prediction is visible.

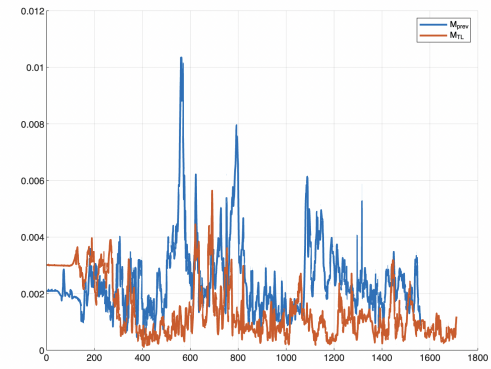
To explain better this concept we can observe the e_{RMS} and e_{MAX}. Let us first look at the values of the indices they take and also graph the trends of both errors along the single trajectory. In this case, the trajectory is only the new one, and so the example shown in figure 5.19, compares the trajectory on two iterations.



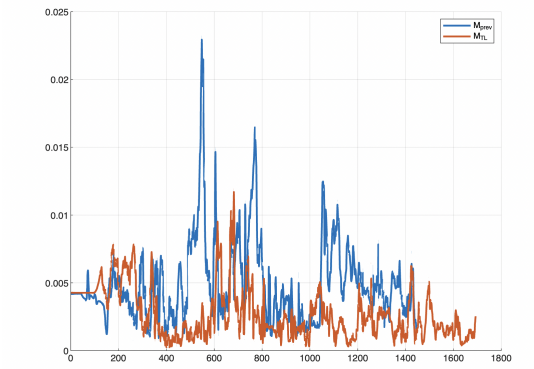
(a) The two model evaluation based on e_{RMS} error.



(b) The two model evaluation based on e_{MAX} error.



(c) Punctual e_{RMS} for the two iterations on the fourth trajectory.

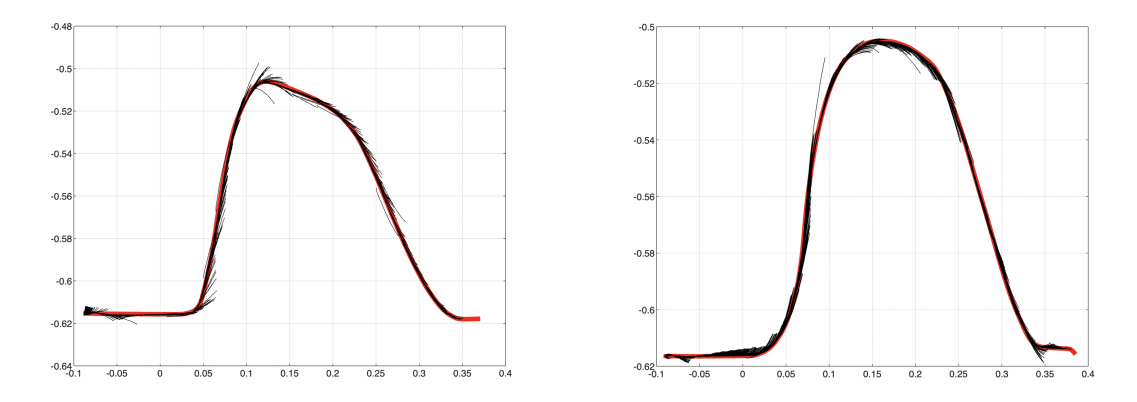


(d) Punctual e_{MAX} for the two iterations on the fourth trajectory.

Figure 5.19: Comparison done applying the TL approach on the new trajectory, analyzed on e_{RMS} and e_{MAX}.

We also observe an improvement here. The value of the first iteration represented is increased when compared with the last iteration collected during the training phase. This is because the model faces for the first time a trajectory it has never seen. Despite this, we can see that by using transfer learning we only need one iteration to reach performances comparable to the results obtained using the full iterative training procedure.

It is interesting to evaluate the TL approach on different users, we used 5 different people. The 5.30 figure shows the comparison of a single user trajectory to give an idea of the improvement that has occurred. The example trajectory is the one depicted in fig 5.5ccorresponding to the linear case.



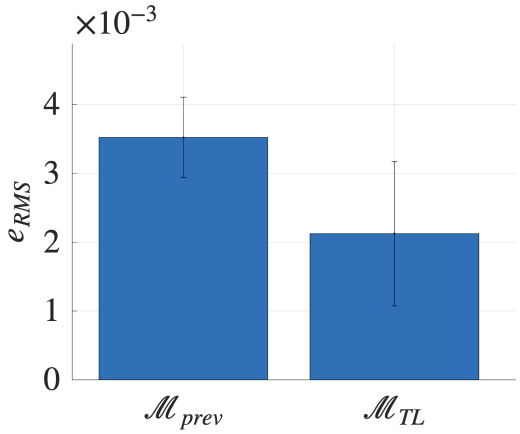
(a) Prediction with the model M_3.

(b) Prediction with the model M_{TL}.

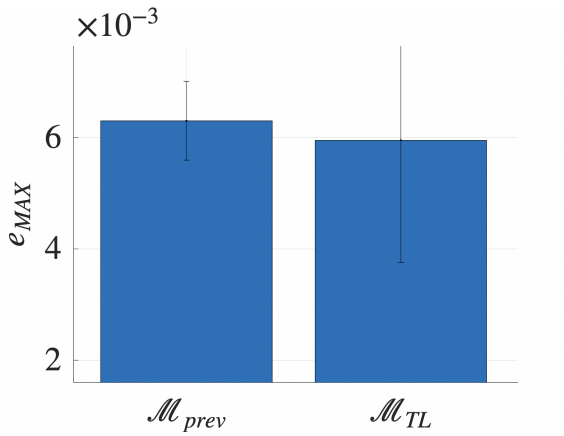
Figure 5.20: Comparison between the prediction at the two training iterations, of one different user, with prediction horizon as 0.4sec

While in the next e_{RMS} and e_{MAX} min indices are analyzed. In this case, having been tested on 5 different people, the values were obtained by averaging across all the values obtained from each person and represented in 5.21a and 5.21b

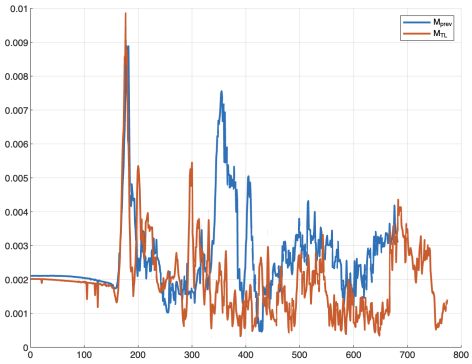
Indeed, for the cases of figures 5.21c and 5.21d, only one person along the linear trajectory was taken.



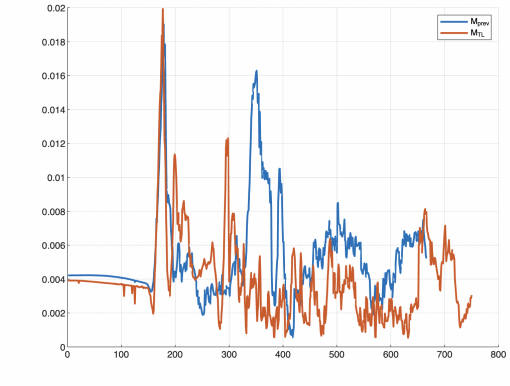
(a) The two model evaluation based on e_{RMS} error.



(b) The two model evaluation based on e_{MAX} error.



(c) Punctual e_{RMS} for the two iterations on the linear trajectory.

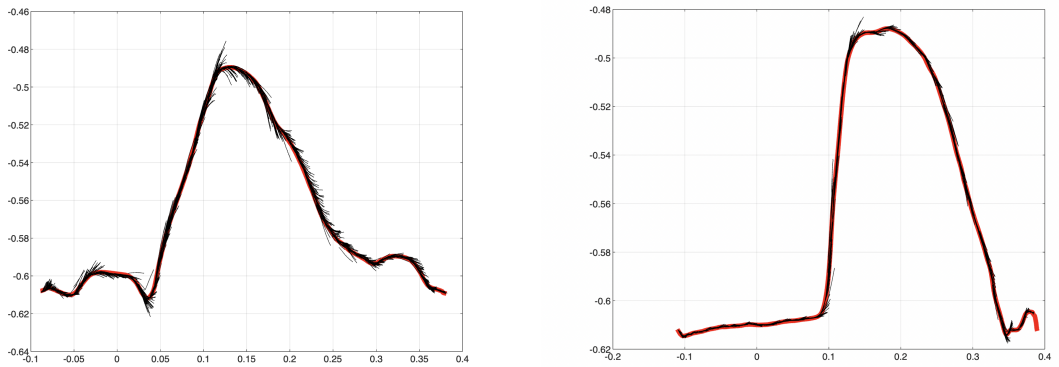


(d) Punctual e_{MAX} for the two iterations on the linear trajectory.

Figure 5.21: Comparison done applying the TL approach on the different users, analyzed on e_{RMS} and e_{MAX}.

The result obtained demonstrates the improvement of the model, even on different users from the author. It is the result based on one user. The test is all done on 5 different users and in every case, the error is better.

Finally, we compare the result collected using the coo-manipulated object *i. e.* the wooden board. Also in figure 5.22, we present first a graphical comparison based on the linear trajectory.

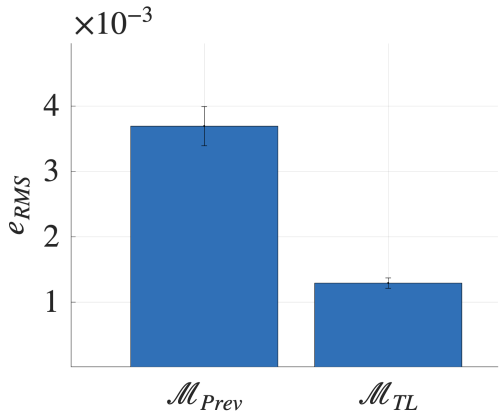


(a) Prediction with the model M_3.

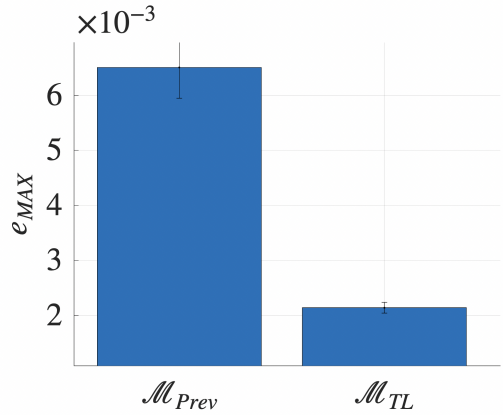
(b) Prediction with the model M_{TL}.

Figure 5.22: Comparison between the prediction at the two training iterations, of the coo-manipulated object, with prediction horizon as 0.4sec

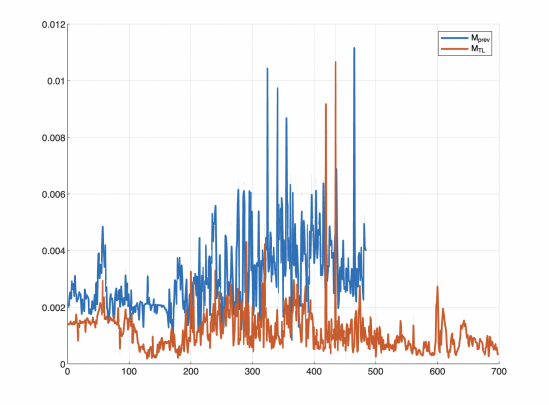
And then we check the indices of e_{RMS} and e_{MAX} The punctual error is still represented with the linear trajectory



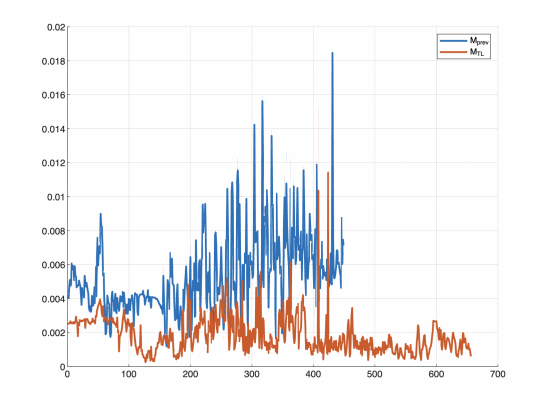
(a) The two model evaluation based on e_{RMS} error.



(b) The two model evaluation based on e_{MAX} error.



(c) Punctual e_{RMS} for the two iterations on the linear trajectory.



(d) Punctual e_{MAX} for the two iterations on the linear trajectory.

Figure 5.23: Comparison done applying the TL approach on the coo-manipulated object, analyzed on e_{RMS} and e_{MAX}.

The improvement here is very visible as the presence of the object changes many parameters of the learning model. That is why in the case of the old model, the value is very high but thanks to the TL we get very good results.

One of the main advantages of using the TL is saving time. The tables in 5.2, explain this by comparing the time needed to collect data e train the model between the TL approach and training the model for the first time. This allows us to use our model more easily on different people and procedures.

	Iterations	TL Trajectory	TL subject	TL Object
dataset collection	60 ± 10 min	5 ± 2 min	5 ± 2 min	5 ± 2 min
training	45 ± 5 min	4 ± 1 min	4 ± 1 min	4 ± 1min

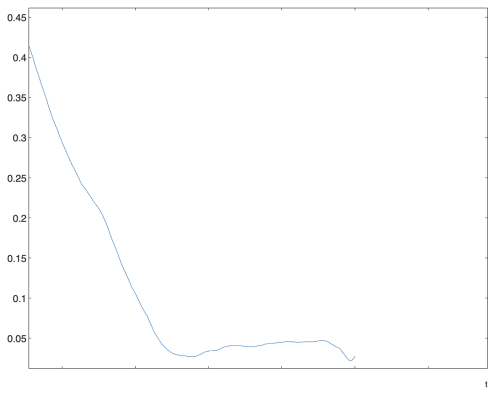
Table 5.2: The time required for the data collection and model training at the various steps.

As we can see the time for adapting the model to new users/objects dramatically decrease compare to the iterative training phase.

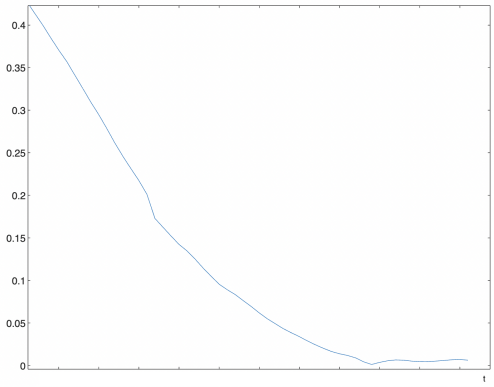
5.3.3. Human-Robot co-manipulation of large/heavy components

We now analyze the result of the application scenarios. We will analyze the improvement of the application of the TL through the precision of reaching a final point, and then we compare this result with Manual Guidance and Impedance control. The analysis was conducted taking into consideration the parameter defined in 5.2.4.

The first is the with a wooden board. As we already evaluated that the model improved its assistance, in this application approach, we test the precision in arriving at the endpoint and the interaction force required. To do this, the figure 5.24a represents the distance taken for equal time from when the robot started moving to 3 seconds after the endpoint is reached. This is because we want to analyze if the distance keeps its endpoint and how difficult it is to stay there. The comparison here is just graphical as we want to see the improvements of reaching the endpoint. On the x-axis, we see the time, while on the y-axis, the distance value decrease approaching the endpoint. The test is conducted on a single trial.



(a) Trial before applied M_{TL}.



(b) Trial with M_{TL}.

Figure 5.24: Comparison of the precision in arriving at an endpoint using the σ index

The improvements are significant as we can see how the model assists better after the application of the TL. The fact that the distance is lower tells us that the variations of distance, at the endpoint in particular, are less than the first case also in this task.

Then, we compare it with a different controller, as Manual Guidance and Impedance Control.

First, we see the force involved:

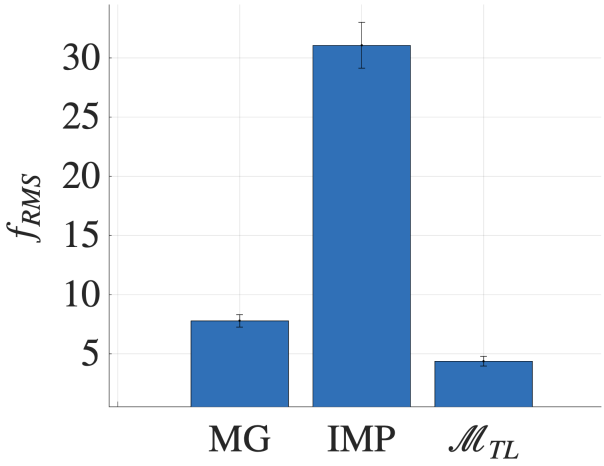
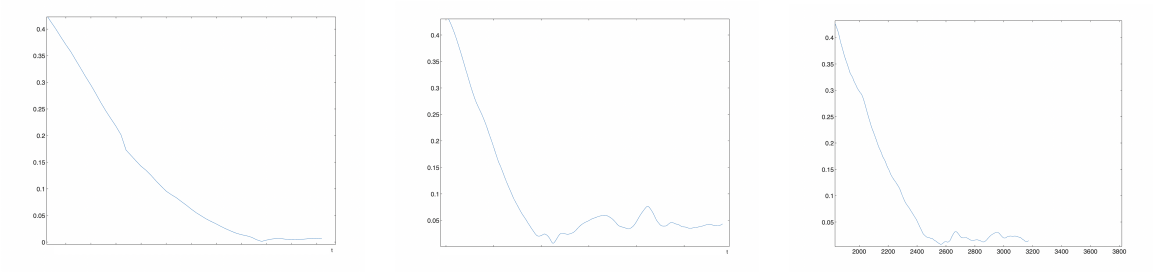


Figure 5.25: Force comparison between MG, IMP and M_{TL} using the wooden board.

We can see that the model with the M_{TL} is better and the Impedance is the worst.

Then we analyze the precision to arrive in the case of MG and assistive controller after TL using the graph of distance.



(a) Trial with the model after M_{TL}.

(b) Trial using MG.

(c) Trial using IMP.

Figure 5.26: σ_i representation of a single trial from the initial point to endpoint comparing the model with TL, IMP and MG control on a wooden board.

To have a comparison we see the index described before in 5.2.4, between MG and M_{TL}.

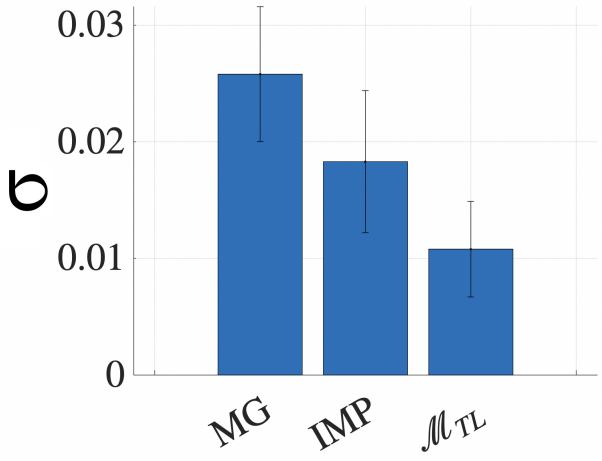
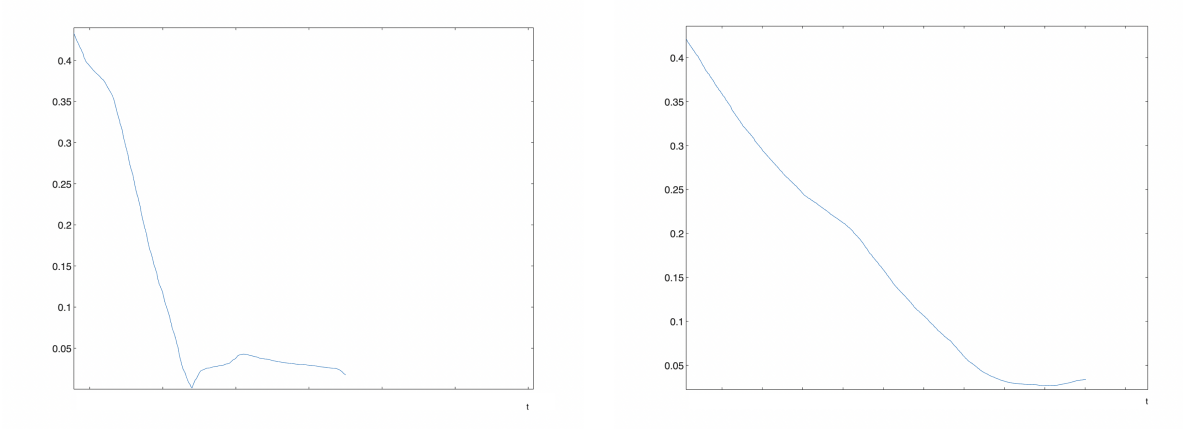


Figure 5.27: Precision index σ comparison between MG, IMP e M_{TL} using the wooden board

As we can see from the figure in 5.26 and 5.27 the precision is better when we have the assistance of the controller we have studied and the error in reaching this point.

The application was done also with the lumped object.
The analysis is the same as with wooden board.
First, we check the precision distance between the model before the application of the TL approach and after. The result is in fig 5.28.



(a) Trial before applied M_{TL}. (b) Trial with the model after M_{TL}.

Figure 5.28: Comparison of the precision in arriving at an endpoint using distance graph and σ index

After that, we compare the same experiments with Manual Guidance and Impedance Control as before. First, we see the force involved in:

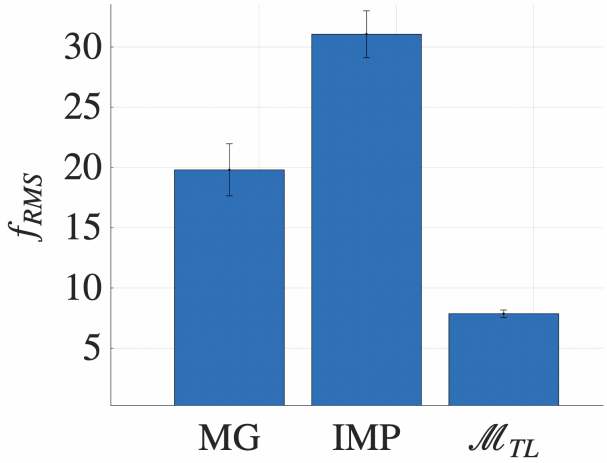
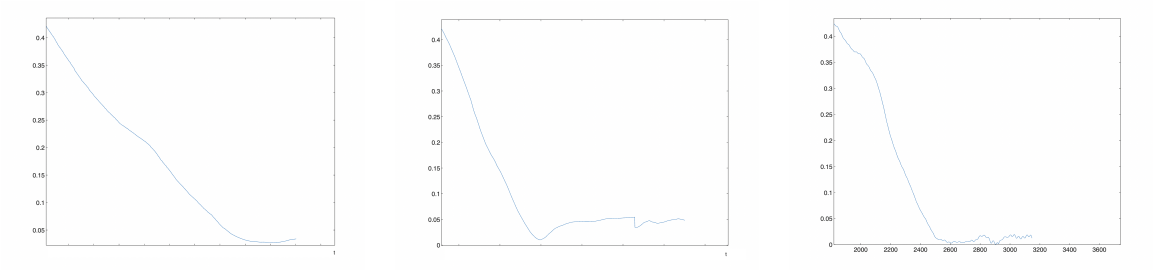


Figure 5.29: Force comparison between MG, IMP e M_{TL} using the lumped object.

Also here force is bigger when using the MG, and the IMP is always the biggest. The force involved is bigger than with the wooden board as the weight here is bigger.

Then we analyze the precision to arrive in the case of MG, IMP, and M_{TL}.



(a) Trial with the model after M_{TL}.

(b) Trial using MG.

(c) Trial using IMP.

Figure 5.30: Comparison of the precision in arriving at an endpoint using the σ index

To have a comparison we see the index described in 5.2.4 between MG, IMP and M_{TL}

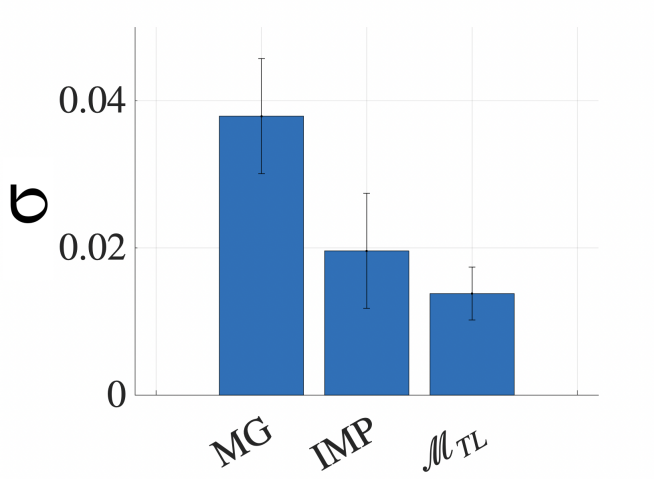


Figure 5.31: Precision index σ comparison between MG, IMP, and M_{TL} using the lumped object.

Also here, the precision is better when using the model trained with TL, and the precision index is better.

The final considerations are that the force in the Impedance control is bigger than the other because the spring implementation of the control tends to bring the robot to the nominal trajectory and not assist the human in the deviation, so the force is bigger. In the manual guidance, the situation is better but bigger than the M_{TL}, because the human has to put force in every moment that depends on damping and velocity. Then our controller is better because when it deviates from the trajectory doesn't bring back to them but assistive the human in the trajectory he wants to do.

In the case of reaching the target point, we have difficulty in MG because the robot is free to move but difficult to maintain in a single position, while impedance gives a force that is difficult to contrast. So the model we train is better also for this task.

6 | Conclusions and future developments

This work presents an assistive controller for pHRI. Two main components are investigated, namely the distributed Model Predictive Control (dMPC) formulation of the interaction, and a learning method for predicting the desired human trajectory over a finite time horizon. The interaction between humans and robots is described by Differential Cooperative Game Theory (DCGT). The dMPC control framework is simulated to analyze its behavior according to different tunable parameters, in order to define the values that better allow its applicability in pHRI.

The proposed approach is validated with real-world experiments done by the author on the UR5 robotic arm. An iterative training procedure is defined for model training, allowing the model's adaptation to the interaction with the human. Four iterations are performed even though it is observed that after just three, good results are obtained.

To make the model adaptive to new situations and users, Transfer Learning is applied. The results obtained are satisfactory as starting from a model trained on a single subject and on a specific task, it is possible to quickly adapt the model to new users and tasks with comparable performances. This method allows to dramatically reduces the time necessary for data collection and training the model compared to the iterative procedure.

Different prediction horizons are also evaluated to show the dependency of the error. On the one hand, the longer the prediction horizon is, the more the error increases. On the other hand, with longer prediction horizons the robot is able to better assist the human with the dMPC formulation.

Finally, the superiority of the assistive controller enhanced by the RNN+FC model, compared to standard controllers typically used in pHRI is shown by measuring the average interaction force and the precision to reach a final point.

Future works will focus on implementing the model not only with force sensors but also with cameras that detect the position of the human and impart a fictitious force to the

robot, allowing flexible material co-manipulation. Moreover, the update of the model will be implemented online in order to reduce even more the data collection and training time. The possibility of varying online the assistance level will be addressed by feeding the RNN+FC with this additional time-varying parameter. Finally, a model that considers the full 6 dofs will be investigated, with real applications involving co-manipulation of deformable objects to fabricate composite materials.

Bibliography

[1] C. C. Aggarwal. *Neural Networks and Deep Learning: A Textbook.* Springer International Publishing, 2018.

[2] N. Akhlaghi, C. Baker, M. Lahlou, H. Zafar, K. Murthy, H. Rangwala, J. Kosecka, W. Joiner, J. Pancrazio, , and S. Sikdar. Real-time classification of hand motions using ultrasound imaging of forearm muscles. *IEEE Trans. Biomed. Eng., 63(8*, pages 1687–1698, 2016.

[3] T. Akiba, S. Sano, T. Yanase, T. Ohta, , and M. Koyama. Optuna: A next-generation hyperparameter optimization framework,. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery data mining,*, pages 2623–2631, 2019.

[4] W. Bi, X. Wu, Y. Liu, and Z. Li. Role adaptation and force, impedance learning for physical human-robot interaction. *IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 111–117, 12 September 2019.

[5] H. C., G. Wasson, A. M., S. P., and L. A. Shared navigational control and user intent detection in an intelligent walker,. *IEEE/ASME Trans. Mechatronics, 18(1)*, pages 285—-296, 2005.

[6] D.A.Braun, P.A.Ortega, and D.M.Wolpert. Nash equilibria in multi-agent motor interactions. *PLoS Comput. Biol., vol. 5, no. 8*, pages 1–8, 2009.

[7] W. Dai, Q. Yang, G. Xue, and Y. Yu. Self-taught clustering. *Proc. 25th Int'l Conf. Machine Learning*, pages 200–207, July 2008.

[8] J. Dong, J. Xu, Q. Zhou, and S. Hu. Physical human–robot interaction force control method based on adaptive variable impedance. *J of the Franklin Institute*, 357(12): 7864–7878, 2020. ISSN 0016-0032.

[9] A. Duschau-Wicke, J. von Zitzewitz, A. Caprez, L. Lunenburger, and R. Riener. Path control: A method for patient-cooperative robot-aided gait rehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng., 18(1)*, pages 38–48, 2010.

[10] J. Engwerda. Lq dynamic optimization and differential games. *Sussex, U.K.: Wiley*, page ch.1, 2005.

[11] F. Ficuciello, L. Villani, and B. Siciliano. Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *EEE Transactions on Robotics, vol. 31, no. 4*, pages 850–863, 2015.

[12] P. Franceschi, N. Pedrocchi, and M. Beschi. Adaptive impedance controller for human-robot arbitration based on cooperative differential game theory. *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7888–7894, 2022.

[13] P. Franceschi, N. Pedrocchi, and M. Beschi. Inverse optimal control for the identification of human objective: a preparatory study for physical human-robot interaction,. *IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA),*, pages 1–6, 2022.

[14] S. S. Ge, Y. Li, and H. He. Neural-network-based human intention estimation for physical human-robot interaction. *8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 390–395, 2336 Nov 2011.

[15] A. Ghadirzadeh, J. B. tepage, A. Maki, D. Kragic, and M. Bj'orkman. A sensorimotor reinforcement learning framework for physical human-robot interaction. pages –.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *In CVPR*, pages 1–5, 2014.

[17] Y. Guo, H. Shi, A. Kumar†, K. Grauman, T. Rosing, and R. Feris. Spottune: Transfer learning through adaptive fine-tuning. *In CVPR*, pages 4805–4814, 2014.

[18] 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, VOL 17 NO 4*, pages 1937–1949, OCTOBER 2020.

[19] S. Hochreiter and J. Schmidhuber. Long short term memory. *Neural Computation*, pages 1–32, 1997.

[20] N. Hogan. Impedance control: An approach to manipulation:. *Journal of Dynamic Systems, Measurement, and Control- Part I: Theory*, Vol. 107:1–7, MARCH 1985.

[21] C. Huang, G. Wasson, M. Alwan, P. Sheth, and A. Ledoux. Shared navigational

control and user intent detection in an intelligent walker. *AAI Fall 2005 Symposium, Arlington, VA, Nov.*, pages 4–6, 2005.

[22] B. Hudgins, P. Parker, , and R. N. Scott. A new strategy for multifunction myoelectric control. *IEEE Trans. Biomed. Eng., 40(1)*, pages 82–94, 1993.

[23] Y. Huo, X. Li, X. Zhang, and D. Sun. Intention-driven variable impedance control for physical human-robot interaction. *in 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1220–1225, 2021.

[24] 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, vol. 78,*, pages 1–9, 2022.

[25] P. A. Josh. *Neural Network and PyTorch Basics. In: Hands-on Machine Learning with Python.* Apress, Berkeley, CA., 2022.

[26] S. Ko and R. Langari. Shared control between human driver and machine based on game theoretical model predictive control framework. *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Boston, USA*, pages 649–654, July 6-9, 2020.

[27] G. Kuhlmann and P. Stone. Graph-based domain mapping for transfer learning in general games. *Proc. 18th European Conf. Machine Learning*, pages 188–200, Sept. 2007.

[28] T. Lenzi, S. M. M. D. Rossi, I. Nicola Vitiello, Member, and M. C. Carrozza. Intention-based emg control for powered exoskeletons. *EEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 59, NO. 8,*, pages 2180–2190, AUGUST 2012.

[29] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering. *Proc. 26th Int'l Conf. Machine Learning*, June 2009.

[30] S. Li, L. Zhang, and X. Diao1. Deep-learning-based human intention prediction using rgb images and optical flow. *Journal of Intelligent Robotic Systems (2020) 97:95–107*, pages 95–107, : 12 July 2019. URL `https://doi.org/10.1007/s10846-019-01049-3`.

[31] Y. Li and S. S. Ge. Human–robot collaboration based on motion intention estimation. *IEEE/ASME TRANSACTIONS ON MECHATRONICS, VOL. 19, NO. 3*, pages 1007–1014, JUNE 2014.

[32] 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 Tran on Robotics*, 31(3):672–681, 2015.

[33] Y. Li, S. S. Ge, and C. Yang. Learning impedance control for physical robot–environment interaction. *Int. J. Control*, vol. 85, no. 2:182–193, Feb 2012.

[34] Y. Li, K. P. Tee, W. L. Chan, M. Rui Yan, Y. Chua, and D. K. Limbu. Continuous role adaptation for human-robot shared control. *Science and Engineering Research Council (SERC), A*STAR, Singapore.*, pages 672–681, June 2015.

[35] Y. Li1, G. Carboni, F. Gonzalez, D. Campolo, and E. Burdet. Differential game theory for versatile physical human-robot interaction. *Nature Machine Intelligence, vol. 1, no. 1*, pages 36–43, Jan 2019.

[36] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *In ICML*, pages 2–5, 2015.

[37] D. P. Losey and M. K. O'Malley. Trajectory deformations from physical human–robot interaction. *IEEE Tran on Robotics*, 34(1):126–138, 2018.

[38] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malle. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction. *Applied Mechanics Reviews, 70(1),*, pages 1–19, 02 - 2018.

[39] 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.

[40] H. Maithani, J. A. Corrales, and Y. Mezouar. Predicting human intent for cooperative physical human-robot interaction tasks. *in 2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pages 1116–1121, 2019.

[41] G. McMullen, D. P.and Hotson, B. A. F. M. S. Katyal, K. D.and Wester, A. McGee T. G.and Harris, M. S. Johannes, A. D. Vogelstein, R. J.and Ravitz, T. Anderson, W. S., N. V., and N. E. Crone. Demonstration of a semi-autonomous hybrid brain–machine interface using human intracranial eeg, eye tracking, and computer vision to control a robotic upper limb prosthetic,. *IEEE Trans. Neural Syst. Rehabil. Eng.*, pages 784–796, 2014.

[42] P. Mishra. Cnn and rnn using pytorch. in: Pytorch recipes. *Apress, Berkeley, CA.*, 2023. URL `https://doi.org/10.1007/978-1-4842-8925-9_3`.

[43] H.-S. Moon and M. Jiwon Seo. Prediction of human trajectory following a haptic robotic guide using recurrent neural networks. *2019 IEEE World Haptics Conference (WHC) Tokyo, Japan*, pages 157–162, 9-12 July 2019.

[44] M. Morchid. Parsimonious memory unit for recurrent neural networks with application to natural language processing. *Neurocomputing 314*, pages 48–64, 2018.

[45] X. Na and D. J. Cole. Game-theoretic modeling of the steering interaction between a human driver and a vehicle collision avoidance controller. *IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, VOL. 45, NO. 1, FEBRUARY*, pages 25–38, 2015.

[46] X. Na and D. J. Cole. Modelling of a human driver's interaction with vehicle automated steering using cooperative game theory,. *IEEE-CAA J. Automatica Sin., vol. 6, no. 5*, pages 165–198, 2019.

[47] X. Na and D. J. Cole. Experimental evaluation of a game-theoretic human driver steering control model. *IEEE TRANSACTIONS ON CYBERNETICS*, pages 1–14, 25 January 2022.

[48] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, page VOL. 22, OCTOBER 2010.

[49] J. Park, M. B. Jun, and H. Yun. Development of robotic bin picking platform with cluttered objects using human guidance and convolutional neural network (cnn). *Journal of Manufacturing Systems 63 539–549*, pages 539–549, 2022.

[50] A. U. Pehlivan, D. P. Losey, and M. K. O'Malley. Minimal assist-as-needed controller for upper limb robotic rehabilitation. *IEEE TRANSACTIONS ON ROBOTICS, VOL. 32, NO. 1,*, pages 113–124, FEBRUARY 2016.

[51] 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, vol. 68*, pages 1–13, 2021.

[52] A. Radmand, E. Scheme, and K. Englehart. A characterization of the effect of limb position on emg features to guide the development of effective prosthetic control scheme. *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Chicago, IL*, pages 662–667, Aug. 26–30 2014.

[53] M. Rahman, R. Ikeura, and K. Mizutan. Investigation of the impedance characteristic of human arm for development of robots to cooperate with humans. *JSME Int. J. Series C, vol. 45, no. 2*, pages 510—-518, 2022.

[54] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: Transfer learning from unlabeled data. *Proc. 24th Int'l Conf. Machine Learning*, pages 759–766, June 2007.

[55] M. Rasouli, K. Chellamuthu, C. J.-J., and S. L. Kukreja. Towards enhanced control of upper prosthetic limbs: A force-myographic approach. *Sixth IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 232–236, Singapore, June 26–29.

[56] L. Roveda. A user-intention based adaptive manual guidance with force-tracking capabilities applied to walk-through programming for industrial robots. In *2018 15th International Conference on Ubiquitous Robots (UR)*, pages 369–376. IEEE, 2018.

[57] 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 Int Conf on Ubiquitous Robots (UR)*, pages 406–411, 2018.

[58] 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.

[59] 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. *J of Intelligent & Robotic Systems*, 100(2):417–433, Nov 2020. ISSN 1573-0409.

[60] 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. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2022.103771.

[61] A. D. Santis, B. Siciliano, A. D. Luca, and A. Bicchi. An atlas of physical human-robot interaction. *Science Direct, Mechanism and Machine Theory 43*, pages 254–268, 2008.

[62] B. Shen, J. Li, F. Bai, and C.-M. Chew. Motion intent recognition for control of a lower extremity assistive device (lead). *Department of Mechanical Engineering National University of Singapore, 21 Lower Kent Ridge Road, Singapore 119077*, pages 926–931, 2013.

[63] Y. Shoham and K. Leyton-Brown. Multiagent systems: Algorithmic, game-theoretic, and logical foundations. *Cambridge, U.K. Cambridge*, page ch.3, Univ. Press, 2009.

[64] B. Siciliano and L. Villani. Robot force control. *Springer Science Business Media,*, 1999.

[65] D. Sirintuna, I. Ozdamar, Y. Aydin, and C. Basdogan. Detecting human motion intention during phri using artificial neural networks trained by emg signals. *in 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1280–1287., 2020.

[66] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. URL `https://www.ros.org`.

[67] K. M. Tarwani and S. Edem. Survey on recurrent neural network in natural language processing. *International Journal of Engineering Trends and Technology (IJETT) – Volume 48 Number 6 June*, pages 301–304, 2017.

[68] A. Toedtheide, E. Shahriari, and S. Haddadin. Tank based unified torque/impedance control for a pneumatically actuated antagonistic robot joint. *in Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1255–1262, May 2017.

[69] L. Torrey and J. Shavlik. Transfer learning. *University of Wisconsin, Madison WI, USA*, pages 1–22.

[70] S. Veselic, C. Zito, and D. Farina. Human-robot interaction with robust prediction of movement intention surpasses manual control,. *Frontiers in Neurorobotics, vol. 15,*, 2021. URL `https://www.frontiersin.org/articles/10.3389/fnbot.2021.695022`.

[71] V. Villani, F. Pini, F. Leali, , and C. Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive t interfaces and applications. *Mechatronics, vol. 55,*, pages 248–266, 2018. URL `https://www.sciencedirect.com/science/article/pii/S0957415818300321`.

[72] K. Wakita, J. D. Huang, P., K. Sekiyama, and T. Fukuda. Human- walking-intention-based motion control of an omnidirectional-type cane robot. *IEEE/ASME Trans. Mechatronics, 18(1)*, pages 285—-296, 2013.

[73] S. Wang, X. Zhaiìo, and T. Yuan. Identification of driver braking intention based on long short-term memory (lstm) network. *School of Automobile, Chang'an University, Xi'an 710064, Chin VOLUME 8*, pages 1–11, 2020.

[74] Z. Wang, A. Peer, and M. Buss. An hmm approach to realistic haptic human-robot interaction. *Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems Salt Lake City, UT, USA*, pages 374–379, March 18-20, 2009.

[75] X.Zhao, S.Chumkamon, S.Duan, J.Rojas, and J.Pan. Collaborative human-robot motion generation using lstm-rnn. *in 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9, 2018.

[76] M. Y. and A. S. K. Design of a cable-driven arm exoskeleton (carex) for neural rehabilitation. *IEEE Trans. Rob., 28(4)*, pages 922–931, 2012.

[77] N. Yadav, A. Yadav, and M. Kuma. *An Introduction to Neural Network Methods for Differential Equations*. SpringerBriefs in Applied Sciences and Technology, 2015.

[78] B. Yao, Z. Zhou, L. Wang, W. Xu, Q. Liu, and A. Liu. Sensorless and adaptive admittance control of industrial robot in physical humanrobot interaction. *Science Direct*, pages 158–168, 2018. URL `www.elviser.com/locate/rcim`.

[79] J. Yosinski, J. Clune, Y. Bengio, , and H. Lipson. How transferable are features in deep neural networks? *In CVPR*, pages 1–9, 2014.

[80] J. Zhanga, H. Liub, Q. Changc, L. Wang, and R. X. Gao. Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. *CIRP Annals - Manufacturing Technology*, pages 9–12, 2020.

[81] R. Zou, Y. Liu, J. Zhao, and H. Cai. A framework for human-robot-human physical interaction based on n-player game theory. *Sensors, vol. 20, no. 17,*, 2020. URL `https://www.mdpi.com/1424-8220/20/17/5005`.

List of Figures

List of Tables

List of Algorithms

Acknowledgements

I would like to express my gratitude to my primary supervisor, Professor Paolo Rocco, who assisted and guided me throughout this project.

I would like to thank my supervisor Paolo Franceschi for all his help and advice with this thesis. In addition to guiding me in the writing of this work, he gave me the passion and enthusiasm necessary for the thesis to take shape day after day.

I wish to acknowledge the help provided by the staff of STIIMA with whom sharing this work has been very pleasant. Especially the workshop leader Nicola Pedrocchi who supervised and coordinated the work with great support.

I would also like to thank all my friends, who supported me and with whom this would have not been possible.

In the end, I want to dedicate all this work to my family and in particular my father, my mother, and my sister who have supported me and motivated me during the years of study.