THE UNIVERSITY
of EDINBURGH

A DYADIC COLLABORATIVE MANIPULATION FORMALISM FOR OPTIMIZING HUMAN-ROBOT TEAMING

THEODOROS STOURAITIS

Doctor of Philosophy
School of Informatics
University of Edinburgh

2021

Theodoros Stouraitis:
A Dyadic collaborative Manipulation formalism for Optimizing Human-Robot Teaming
Doctor of Philosophy, 2021

SUPERVISORS:
Prof. Sethu Vijayakumar, Ph.D., FRSE
Michael Gienger, Ph.D.

EXAMINERS:
Prof. Marc Toussaint, Ph.D.
Steve Tonneau, Ph.D.

ABSTRACT

Dyadic collaborative Manipulation (DcM) is a term we use to refer to a team of two individuals, the agent and the partner, jointly manipulating an object. The two individuals partner together to form a distributed system, augmenting their manipulation abilities. Effective collaboration between the two individuals during joint action depends on: (i) the breadth of the agent's action repertoire, (ii) the level of model acquaintance between the two individuals, (iii) the ability to adapt online of one's own actions to the actions of their partner, and (iv) the ability to estimate the partner's intentions and goals.

Key to the successful completion of co-manipulation tasks with changing goals is the agent's ability to change grasp-holds, especially in large object co-manipulation scenarios. Hence, in this work we developed a *Trajectory Optimization* (TO) method to enhance the repertoire of actions of robotic agents, by enabling them to plan and execute hybrid motions, i. e. motions that include discrete contact transitions, continuous trajectories and force profiles. The effectiveness of the TO method is investigated numerically and in simulation, in a number of manipulation scenarios with both a single and a bimanual robot.

In addition, it is worth noting that transitions from free motion to contact is a challenging problem in robotics, in part due to its hybrid nature. Additionally, disregarding the effects of impacts at the motion planning level often results in intractable impulsive contact forces. To address this challenge, we introduce an impact-aware multi-mode TO method that combines hybrid dynamics and hybrid control in a coherent fashion. A key concept in our approach is the incorporation of an explicit contact force transmission model into the TO method. This allows the simultaneous optimization of the contact forces, contact timings, continuous motion trajectories and compliance, while satisfying task constraints. To demonstrate the benefits of our method, we compared our method against standard compliance control and an impact-agnostic TO method in physical simulations. Also, we experimentally validated the proposed method with a robot manipulator on the task of halting a large-momentum object.

Further, we propose a principled formalism to address the joint planning problem in DcM scenarios and we solve the joint problem holistically via model-based optimization by representing the human's behavior as task space forces. The task of finding the partner-aware contact points, forces and the respective timing of grasp-hold changes are carried out by a TO method using non-linear programming. Using sim-

ulations, the capability of the optimization method is investigated in terms of robot policy changes (trajectories, timings, grasp-holds) to potential changes of the collaborative partner policies. We also realized, in hardware, effective co-manipulation of a large object by the human and the robot, including eminent grasp changes as well as optimal dyadic interactions to realize the joint task.

To address the online adaptation challenge of joint motion plans in dyads, we propose an efficient bilevel formulation which combines graph search methods with trajectory optimization, enabling robotic agents to adapt their policy on-the-fly in accordance to changes of the dyadic task. This method is the first to empower agents with the ability to plan online in hybrid spaces; optimizing over discrete contact locations, contact sequence patterns, continuous trajectories, and force profiles for co-manipulation tasks. This is particularly important in large object co-manipulation tasks that require on-the-fly plan adaptation. We demonstrate in simulation and with robot experiments the efficacy of the bilevel optimization by investigating the effect of robot policy changes in response to real-time alterations of the goal.

This thesis provides insight into joint manipulation setups performed by human-robot teams. In particular, it studies computational models of joint action and exploits the uncharted hybrid action space, that is especially relevant in general manipulation and co-manipulation tasks. It contributes towards developing a framework for DcM, capable of planning motions in the contact-force space, realizing these motions while considering impacts and joint action relations, as well as adapting on-the-fly these motion plans with respect to changes of the co-manipulation goals.

LAY SUMMARY

Human-Robot Interaction (HRI) is concerned with the relationship between humans and robots. Although, HRI has been considered to be a science fiction endeavour for many decades, recently more and more discussions about the role of robots in our society are in progress. This can be attributed to the developments on the field of artificial intelligence, such as natural language processing, and to the developments on the robot hardware, which enabled safe co-existence of humans and robots in the same physical space.

The amount of research topics considered in HRI is vast and the field is highly multidisciplinary, as it involves experts from human–computer interaction, artificial intelligence, robotics, natural language understanding, design, humanities, psychology, philosophy, social sciences, cognitive sciences, neuroscience, etc. As a result the field can be separated in a number of sub-fields, with one of them being the *physical Human-Robot Collaboration* (pHRC), which studies human-robot teaming in physical interaction scenarios. Research in this sub-field aims to develop an understanding and methodologies to enable robots assist humans in physical tasks. The main challenges can be summarized with the following three questions: (i) how should a robot act to assist a human partner? (ii) how can a robot coordinate its actions in space and time to collaborate with a human partner? (iii) how can a robot understand the needs and desires of a human partner?

This thesis proposes ways of improving the collaboration between a human and a robot in co-manipulation tasks. Two main avenues are explored, which correspond to the first and second question. In the first paradigm, it is proposed to enable robots to decide how and when to change the grasp-holds on the object during manipulation and co-manipulation tasks. For the human-robot team, the ability of a robot to change and select different grasp locations on the object, like the human does, empowers the team to complete tasks that where not possible before. In the second paradigm, a rough model that describes the behavior of the human partner is provided to a robot, along with the ability to adapt its behavior on-the-fly. This facilitates the co-manipulation of the object, grants a robot agent the ability to personalize and coordinate its motion plans to the human behavior and also, empowers a robot agent to adapt its motions to respond to changes that occur during the co-manipulation task. These two schemes make a step forward towards employing robots in real-world co-manipulation scenarios, where humans would enjoy an extra hand.

ACKNOWLEDGEMENTS

There are many individuals without whom I would not have been able to complete this research journey, and others whom I was fortunate to collaborate and deliberate with. I would like to express my sincere gratitude to all those who supported, helped and advised me.

First and foremost, I would like to profoundly thank my supervisors Prof. Sethu Vijayakumar and Dr. Michael Gienger for giving me the freedom to pursue my own ideas, enlightening me with excellent academic guidance and personal advice. Sethu unveiled to me the core principles of scientific exploration, as well as supported and trusted me from the very early steps of this endeavor. Michael supported me in every single step of the journey and brought me down to earth at the most critical points of it, which is imperative to the advancement of any scientific aspiration. I am also grateful to Prof. Subramanian Ramamoorthy for the insightful discussions and advice during my annual review meetings. Also, without the support and advice of my mentor and friend Dr. Neal Lii, I would not have embarked this wonderful journey in the first place and for that I am truly grateful. I also thank my thesis examiners, Prof. Marc Toussaint and Dr. Steve Tonneau, for providing me with valuable feedback and for helping me improve this dissertation.

I would also like to express my utmost thanks to my family, George, and Elpida for their everlasting love and encouragement, Maria for the unconditional support and mentorship, and Kostas for his advice and the tacit incitement to explore the broad world of sciences. There are no words that can describe; without you I would not have been here and I would not be who I am. Also, I would like to thank Stamatina for her love and kindness, which helped endure the toughest moments!

I would also like to express my gratitude to my friends and fellow PhD students, Jiayi, João, Iordanis, Evripidis, Agamemnon, Lei and Chris for all the inspiring discussions, all the help and support. Especially, Evripidi for proofreading parts of this thesis and Agamemnon for proofreading the full document and offering me great advice from the beginning to the end of my PhD journey. I wish also to especially thank Dr. Yan Lei, João Moura and Iordanis Chatzinikolaidis for their help and contribution in Chapter 3 of the thesis.

I further thank SLMC research group and the robotics research group at Honda Research Institute Europe for the passionate debates, conversations, help and for all the productive collaborations. In particular, I would like to thank Dr. Vladimir Ivan, Wolf-

gang Merkt, Daniel Gordon, Bence Varadi, Boon Han Charayaphan Nakorn, Henrique Ferrolho, Dr. Dirk Ruiken, Fabio Muratore, Tamas Bates, Dr. Simon Manschitz and Dr. Manuel Muehlig. Last, but not least, I would like to thank Stergios, Callum and all my flatmates for making my social life enjoyable in cold Scotland, as well as Dr. Nikolaos Bademis for the help, advice and hospitality offered to me during my visits to Germany, and all my friends around the world for keeping in touch with me.

My doctoral studies were fully funded by a scholarship from the Honda Research Institute Europe, which allowed me to focus most of my time on my research, and for that I am grateful. I am also grateful for being part of the IPAB community, which has been a stimulating working environment.

PUBLICATIONS

Parts of the research leading to this thesis has previously appeared in the following peer-reviewed publications. Some passages have been quoted verbatim from the respective sources.

JOURNAL ARTICLES

- T. Stouraitis, I. Chatzinikolaidis, M. Gienger, and S. Vijayakumar (2020). "Online Hybrid Motion Planning for Dyadic Collaborative Manipulation via Bilevel Optimization." In: *IEEE Transactions on Robotics* 36.5, pp. 1452–1471 (**Invited**) (Chapter 2 and Chapter 5).

CONFERENCE ARTICLES

- T. Stouraitis, L. Yan, J. Moura, M. Gienger, and S. Vijayakumar (2020). "Multimode Trajectory Optimization for Impact-aware Manipulation." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE (Chapter 3).

- T. Stouraitis, I. Chatzinikolaidis, M. Gienger, and S. Vijayakumar (2018). "Dyadic collaborative Manipulation through Hybrid Trajectory Optimization." In: *Conference on Robot Learning (CoRL)*, pp. 869–878 (**Nominated for the Best System Paper Award**) (Chapter 3 and Chapter 4).

- I. Chatzinikolaidis, T. Stouraitis, S. Vijayakumar, and Z. Li (2018). "Nonlinear Optimization using Discrete Variational Mechanics for Dynamic Maneuvers of a 3D One-Leg Hopper." In: *International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 1–9 .

- W. Merkt, Y. Yang, T. Stouraitis, C. E. Mower, M. Fallon, and S. Vijayakumar (2017). "Robust shared autonomy for mobile manipulation with continuous scene monitoring." In: *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, pp. 130–137 (**Awarded First Prize for Greatest Potential for Positive Impact**)(Appendix A).

DECLARATION

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Edinburgh, 2021

Theodoros Stouraitis,
May 17, 2021

Dedicated to my unfulfilled football dream
that fuels the pursuit of my aspirations ever since.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CoM	Center of Mass
DcM	Dyadic collaborative Manipulation
DoF	Degree of Freedom
GMM	Gaussian Mixture Model
GMR	Gaussian Mixture Regression
GP	Gaussian Processes
GS	graph search
HHI	Human-Human Interaction
HOLM	Hybrid Optimization Lexicon for Manipulation
HRC	Human-Robot Collaboration
HRI	Human-Robot Interaction
IK	Inverse Kinematics
MPC	Model Predictive Control
NLP	Non Linear Programming
OC	Optimal Control
pHRC	physical Human-Robot Collaboration
pHRI	physical Human-Robot Interaction
PP	Parametric Programming
Rcs	Robot control software
RL	Reinforcement Learning
TO	Trajectory Optimization

LIST OF SYMBOLS

π^a	Agent's policy
π^p	Partner's policy
θ	Model parameters

Scalars

$n \in \mathbb{N}$	Dimensionality of the agent's configuration space
$\rho \in \mathbb{N}$	Dimensionality of the partner's configuration space
$\nu \in \mathbb{N}$	Dimensionality of manipulation task
$\eta \in \mathbb{N}$	Dimensionality of the state of a system
$K \in \mathbb{N}$	Total number of agent's end-effectors
$T \in \mathbb{R}_{>0}$	Total motion duration (final time)

System

$\boldsymbol{u} \in \mathbb{R}^{\eta}$	Control vector
$\boldsymbol{x} \in \mathbb{R}^{\eta}$	State vector

Agent (Robot)

$\mathbf{q}^a \in \mathbb{R}^n$	Agent's configuration
$\mathbf{f}^{\kappa} \in \mathbb{R}^{\nu}$	Forces applied by agent's κ_{th} end-effector
$\mathbf{c}^{\kappa} \in \mathbb{R}^{\nu}$	Agent's κ_{th} end-effector position
$\mathbf{K}^a, \mathbf{D}^a \in \mathbb{R}^{\nu}$	Cartesian stiffness and damping of the agent

Object

$\mathbf{y}_{t:T} \in \mathbb{R}^{\nu \times N}$	Pose trajectory of the object

Partner (Human)

$\mathbf{q}^p \in \mathbb{R}^{\rho}$	Partner's configuration
$\mathbf{K}^p, \mathbf{D}^p \in \mathbb{R}^{\nu}$	Cartesian stiffness and damping of the partner

1

INTRODUCTION

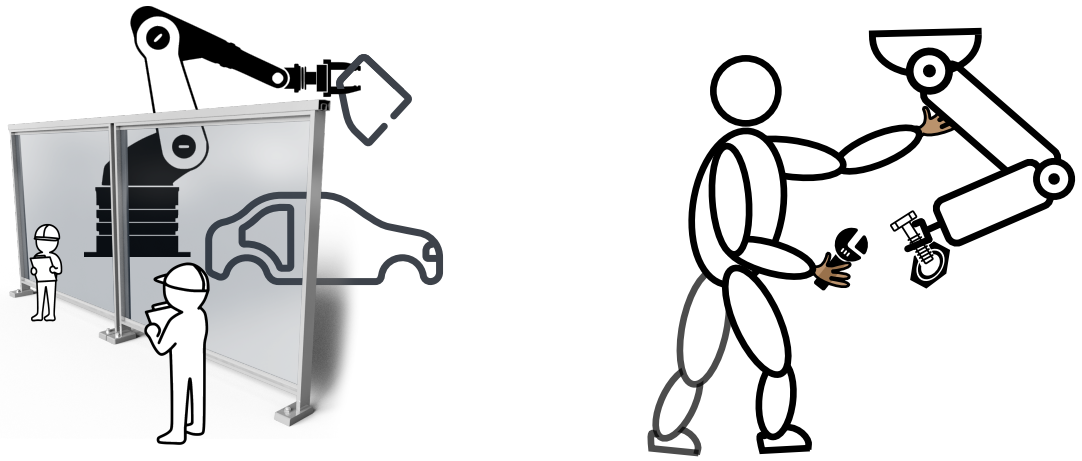
> Ask yourself this question: 'Will this matter a year from now?'
>
> Richard Carlson

Robots and automation technology are nowadays a key part of the well-functioning manufacturing and retail industries, while it is envisioned that robots can become a key part of our society in general. Broader use of robotic devices will become essential sooner rather than later, especially given the shrinking workforce and the aging population of our society. The pre-COVID-19 forecasts of labour deficit worldwide is approximately 10 million workers over the next fifteen years in Germany and 35 million workers over the next thirty years in US (Bonkenburg, 2016), while the recent pressure of social distancing rules only aggravate this problem.

Traditionally, robots found in industrial setups are spatially (fenced) or temporally separated from humans (Ermolov, 2020), to ensure humans' safety, as illustrated in Fig. 1.1a. Hardware advancements have enabled the development of smaller and more lightweight robots, called *cobots* (J. E. Colgate and M. A. Peshkin, 1996; Edward, Wannasuphoprasit, and M. Peshkin, 1999; M. A. Peshkin et al., 2001), that are safe (E. Colgate et al., 2008) to operate side-by-side with their human partners, as displayed with Fig. 1.1b. Although these robots are safe for humans, as soon as they get in contact with a human they are automatically immobilised, which can many times impede the workflow rather than promote it. Further, the unpredictability and variability of humans' actions generate scenarios with considerable uncertainty to the extend that robots might even fail to successfully collaborative with their human partners. Thus, the field of *Human-Robot Interaction* (HRI) has seen extensive and fast-paced research towards achieving the growing needs of our society.

Cobots are robots made for collaboration with humans.

HRI has already several sub-domains, which can be grouped into two main families with respect to the level of proximity between the human and the robot. These are *remote interaction* and *physical interaction* (Goodrich and A. C. Schultz, 2008). The former considers teleoperation and shared autonomy setups, e. g. controlling a robot in space, while the latter considers scenarios where the human and the robot co-exist. Yet, the ultimate goal is to enable robots not only to co-exist with humans, but also collaborate with them (Sendhoff and Wersing, 2020).



(a) (b)

Figure 1.1: Paradigm swift in terms of the interaction between humans and robots. (a) Industrial scenario, where the humans and the robot co-workers are spatially separated. (b) A human and a cobot working side-by-side to complete a task.

The main focus of this thesis is *physical Human-Robot Collaboration* (pHRC), which falls within the broader scope of *physical Human-Robot Interaction* (pHRI). In pHRC setups, a robot aims to form a team and physically collaborate with a human (De Santis et al., 2008; Alami, Albu-Schäffer, et al., 2006). In an ideal physical collaboration setup both individuals can observe, understand and complement their partners towards improving the performance of the dyad. Humans often engage in physical collaborative activities, such as sawing, rehabilitation, dancing, carrying a table together, etc. Yet, our understanding of how humans collaborate in such activities is subject to research (Jarrassé, Sanguineti, and Burdet, 2013), which makes pHRC even more challenging.

A *dyad* represents the unification of two individuals into a pair.

The diversity of approaches in pHRC is vast, ranging from robot and human perception interfaces to stability and transparency of the physical human-robot teams (Ajoudani et al., 2017). A few key factors that are crucial towards fluent collaboration are; (i) communication channels, which serve as the medium for understanding intentions (Goodrich and A. C. Schultz, 2008; Bauer, Wollherr, and Buss, 2008; Ajoudani et al., 2017), (ii) structure of the dyad, which establishes the relations between the members of the dyad (Goodrich and A. C. Schultz, 2008; Jarrassé, Sanguineti, and Burdet, 2013), (iii) robot's agency, which includes aspects of both motion generation and control design (Goodrich and A. C. Schultz, 2008; Bauer, Wollherr, and Buss, 2008; Ajoudani et al., 2017), and (iv) behavior modelling, learning and adaptation of the dyad's members during the interaction (Goodrich and A. C. Schultz, 2008; Bauer, Wollherr, and Buss, 2008).

Agency denotes the capacity of an agent to act in an environment.

Communication, which can be either implicit or explicit, is typically used to transmit task-related information between the members of the dyad (Bauer, Wollherr, and

Buss, 2008) and it has been shown to enhance the performance of the team (Reed and M. A. Peshkin, 2008). The primary matter of study in the implicit communication is intention estimation (Koppula and Saxena, 2016; Nikolaidis, Kuznetsov, et al., 2016), while *legible* motions (Dragan, K. C. Lee, and Srinivasa, 2013) have also attracted interest in the community. On the other hand, explicit communication is involved with the development of multi-modal interfaces for broadcasting and acquiring information, such as visual (Perzanowski, A. C. Schultz, and Adams, 1998), force (Kosuge and Kazamura, 1997), physiological (Glassmire et al., 2004), verbal (Pecchinenda, 1996) and various combinations thereof (C. Yang, Liang, et al., 2016).

The dyad's structure has been described with taxonomies based on the field of human–computer interaction (HCI) (Yanco and J. L. Drury, 2002; Yanco and J. Drury, 2004), and considers a broader notion of the configuration of the interaction, such as separation of the task in sub-tasks or dimensions (J. E. Colgate, M. Peshkin, and Klostermeyer, 2003; M. A. Peshkin et al., 2001). Another major focus point of researchers has been the role assignment of the members. The most common role assignment is the *master-slave* scheme, where the robot is the slave and the human is the master, e. g. in a table co-manipulation task the human guides the table's motion and the robot supports its weight (Stückler and Behnke, 2011). In recent works mixed-initiative systems have been considered, where role-switching occurs (Mörtl et al., 2012). In these systems, symmetric relationships can be attained, for example when the members of the team switch between speaker and listener roles (Losey et al., 2020). Further, a game-theoretic approach to role adaptation has been proposed (Y. Li et al., 2015).

The robot's agency investigates its capacity to act and alter its environment. In pHRC, this environment includes both the objects of interest and the human teammates. Hence, the action space of the robot needs to be rich, flexible and able to maintain a stable interaction with the dynamic environment. Y. Maeda, Hara, and Arai, 2001; Mainprice, Hayne, and Berenson, 2015 proposed to regulate the motion of the robot given the motion of the human hand. Gribovskaya, Kheddar, and Billard, 2011; Noohi, Žefran, and Patton, 2016 proposed to modulate the impedance characteristics of the robot to accommodate the partner's actions. Also, transparent interaction and stability in interaction, based in the concept of passivity, have been studied (Lamy et al., 2009; Albu-Schäffer, Ott, and Hirzinger, 2007).

Behavior modelling and learning arises from the need to acquaint both humans and robots with models of their partners (Sheridan, 1997). Behavior adaptation is essential in pHRC setups, simply because the dyadic system is frequently adjusted. Learning typically happens in an offline phase, where demonstration of the desired dyadic behavior is recorded. Then, the learned model is utilized in an online phase

to guide the behavior of the robot. Depending on the learned model, adaptation is encoded or can be inserted via an auxiliary module. In hand-over tasks, it has been shown that learned models (C.-M. Huang, Cakmak, and Mutlu, 2015; Vogt, Stepputtis, Jung, et al., 2018; G. Maeda et al., 2017) adapt to different conditions. In co-manipulation tasks, the learned models typically utilize an additional adaptation scheme to cope with alternations during the execution (Agravante et al., 2014; Nemec et al., 2018). Recently, for reaching and hand-over tasks, a method to obtain personalized models with iterative learning and continuous adaptation has been proposed (Munzer, Toussaint, and Lopes, 2018).

1.1 SCOPE

The *aim* of this thesis is to advance the state-of-the-art in DcM. With DcM we refer to a team of two individuals jointly manipulating an object, as shown in Fig. 1.2. The two individuals partner together to form a distributed system, augmenting their manipulation abilities. Such individuals can be either humans or robots. In scenarios where both individuals are humans, the collaboration is natural as we humans are adept at co-manipulation. One key element is our ability to understand our partner's intentions and adapt our actions accordingly. A second central skill is our ability to generate sequential manipulation plans. Nevertheless, our understanding of the mechanisms of joint action (Obhi and Sebanz, 2011) and sequential decision making (Dayan and Daw, 2008) are still subject of research.

DcM setups are distributed decision-making systems, also referred as *mixed initiative systems*. In these scenarios, the two individuals are able to coordinate one's own actions with those of others in time and space, which means that the range of the tentative action space of both individuals dramatically increases (Clark, 1996). The framework of *joint action* has been developed to analyze such dyadic systems, and the primary investigation axes are concerned: (i) with the representations used by the individuals to model their partners, (ii) with the means used to predict the actions of their partners, and (iii) with the mechanisms used to integrate the actions of their partners into one's own action plans (Sebanz, Bekkering, and Knoblich, 2006; Sebanz and Knoblich, 2009). The term *joint action* simply encapsulates more than merely reacting to the environment, for which the appropriate term is *interaction*. Without joint action, we humans could never realize smooth and fast coordination with our partner, which is needed to pass a ball or jointly lift an object (Sebanz and Knoblich, 2009). Interestingly enough, the task where an agent carries a heavy object with a partner is regarded as a prototypical joint action scenario (Allport, 1924).

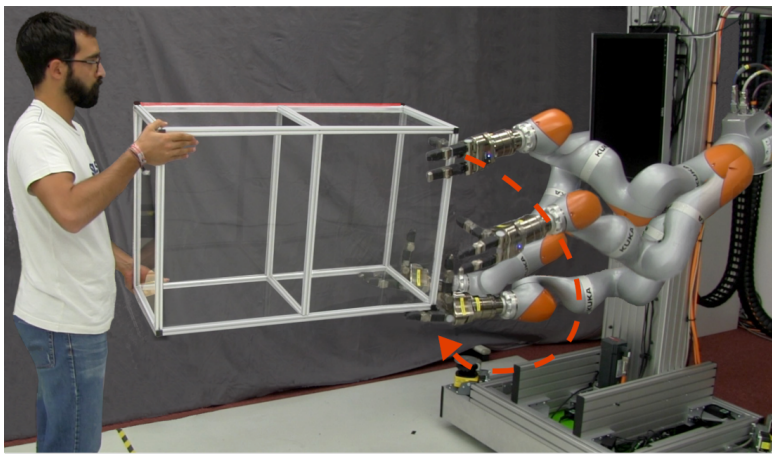


Figure 1.2: DcM scenario during change of contact.

DcM scenarios also demand a broad range of manipulation skills from both participants. Typically, manipulation tasks require making and breaking contact with objects. This results in challenges in motion planning and control due to, among other factors, (i) the hybrid nature of the problem (Mason, 1986) and (ii) the uncertainties that arises due to contact dynamics (Bauza and Rodriguez, 2018). It is within the scope of this thesis to study co-manipulation scenarios that involve multiple changes of contact, which is the crux of sequential manipulation (Dafle et al., 2014).

The secret behind humans' remarkable manipulation skills is our competence in control and prediction of *contact events* (Flanagan, Bowman, and Johansson, 2006). Strikingly, humans are not simply competent in object manipulation, but prefer to make contact with non-zero velocities, as it enables us to achieve the task faster and smoother (Bennett and Castiello, 1994). The two key enablers to realise this are the ability to skillfully switch between free-motion and contact (Flanagan, Bowman, and Johansson, 2006), and the capacity to shift between a variety of control mechanisms depending on the stage of the motion and their associated uncertainties (Johansson and Cole, 1992).

Joint action in DcM scenarios is extremely challenging and requires solution of the following five complex problems.

1.1.1 *Challenges*

1. *Agent's hybrid policy*: the agent's repertoire of actions needs to be sufficiently rich to participate in DcM tasks. Such actions belong to a hybrid space of both continuous and discrete quantities, such as forces and contact changes.

2. *Physically viable agent's action plans*: practical utility and tracking of hybrid action plans cannot be assumed. This is due to particularities of the motion, such as impacts experienced at the contact making event. Hence, the robot's motion planner and controller need to be in accordance with each other, i. e. the agent's actions need to be attainable by the controllers of the robot.

3. *Joint action space planning*: as the two individuals act upon the same object, their actions need to be coordinated with respect to the critical aspects of the task, e. g. balancing the object in collaboration with the partner.

4. *Online motion plan generation*: since the human partner's behavior is changing—i. e. non-stationary—a collaborative agent needs to update its own action plans on-the-fly according to the current goal and state of the interaction.

5. *Partner's intention estimation*: an agent can only contribute to the performance of the dyad if an estimation of the partner's intention can be obtained.

1.1.2 *Problem Statement*

We have so far presented the key factors investigated in the pHRC domain and the distinct properties of DcM setups, which lead to the identification of the main challenges in DcM setups. Corresponding to these challenges, next we frame in simple words the main questions studied in this thesis.

1. How can we formally describe a DcM setup, its components and the relations between them?

2. How can an agent plan and act in the hybrid space to manipulate an object, even in dynamic scenarios?

3. How can an agent represent and integrate the partner's actions to form joint action in DcM setups?

4. How can an agent's policy be adapted on-the-fly to deal with changing behaviors of the partner in DcM setups?

1.1.3 *Approach*

To achieve multi-contact manipulation, one could consider both prehensile and non-prehensile manipulation. Prehensile manipulation is performed by grasping or attaching the object on the agent's end-effector with a gripper or a suction-cup. Non-prehensile manipulation (Mason, 1999) is performed without attaching the object

on the agent's end-effector, such that the relative pose of the object and the robots' end-effectors is free to change throughout the interaction. To accomplish DcM tasks a series of manipulation actions, prehensile or not, is required. Thus, a key challenge in these setups is sequential decision making, which involves decisions on where and when to grasp or contact an object with respect to task, reachability and environment constraints. In this thesis, we primarily investigate the utility of non-prehensile manipulation for DcM setups. This comes with an additional key challenge, that is under-actuation[1], due to the nature of contact in non-prehensile manipulation tasks. To address these two challenges, the robot needs to consider the future effects (time horizon) of its actions to plan and control its current actions accordingly. Thus, our approach is based on *Optimal Control* (OC) (Dyer and S. McReynolds, 1968) and on its variants, e. g. *Trajectory Optimization* (TO). Further, this work delves beyond simply planning dynamic motions and also investigates how to obtain motion plans that directly translate into the real hardware.

In this thesis, we intend to realize DcM scenarios between a human and a robot, and our main inspiration are human-human teams that perform DcM tasks. Cognitive and physical interactions between humans are modulated via the expectations of the partner's behavior, which implies that the human partner is likely to expect a particular behavior (probably human-like) from the robot (Carruthers and Smith, 1996; Wolpert, Doya, and Kawato, 2003; Ganesh, Takagi, et al., 2014). Also, robot's motions that are legible to humans have been shown to benefit the interaction (Dragan, K. C. Lee, and Srinivasa, 2013). Although in our approach we do not explicitly strive to generate human-like collaborative behaviors, we do align our work with the joint action framework and its core principles. Thus, large part of our motivation and rationale is based on studies of human-human collaborative tasks and the joint action framework.

1.2 MAJOR CONTRIBUTIONS

The contributions of this thesis can be summarized as:

1.2.1 *Partner-aware dyadic action formalism*

We propose a dyadic action formalism that considers non-stationary[2] partner behaviors. Using this, the problem of finding the appropriate actions to co-manipulate the object can be addressed given an estimate of the partner's variable intentions. This

1 In under-actuation the actuated degrees of freedom are less that the ones of the system.
2 A behavior that is non-stationary is changing over time.

formalism serves as a principled basis for the development of technical aspects of the thesis.

1.2.2 *Multi-mode optimal control for multi-contact action generation*

We present a parametric programming technique to encode both hybrid dynamics and hybrid control in a single TO method. The proposed holistic model-based optimization method allows robotic agents to treat concurrently (i) forces, (ii) contact locations, (iii) stiffness profiles, (iv) actions timings, (v) object trajectory, and (vi) contact sequence pattern, towards obtaining an optimal solution for multi-contact tasks.

1.2.3 *Bilevel computational formulation*

Our bilevel optimization formulation enables the combination of graph search methods with trajectory optimization methods in one framework. The former provides a coarse solution which is refined by the latter. This combination allows us to efficiently explore the discrete modes of a problem, e. g. in our case, the contact state of an end-effector, and holistically reason about geometric and dynamic properties, e. g. in our case, contact locations, forces and timings. We further introduce a set of hybrid motion primitives to enable our method to generate hybrid plans on-the-fly without a pre-specified contact pattern.

1.2.4 *Partner and impact models*

We propose a simple but ample partner model based on a spring-damper system, which enables us to computationally integrate the behavior of the human partner into the TO method and realize joint action.

Further, we develop a generic force transmission model based on a second-order critically damped system and on a spring-damper model to concurrently optimize stiffness and generate smooth contact forces at impact.

1.2.5 *Online dyadic planning*

By combining the bilevel TO method with the proposed partner model, the hybrid motion primitives and an informed search planner, we realize a computationally efficient optimization method for DcM setups. The method generates hybrid motion

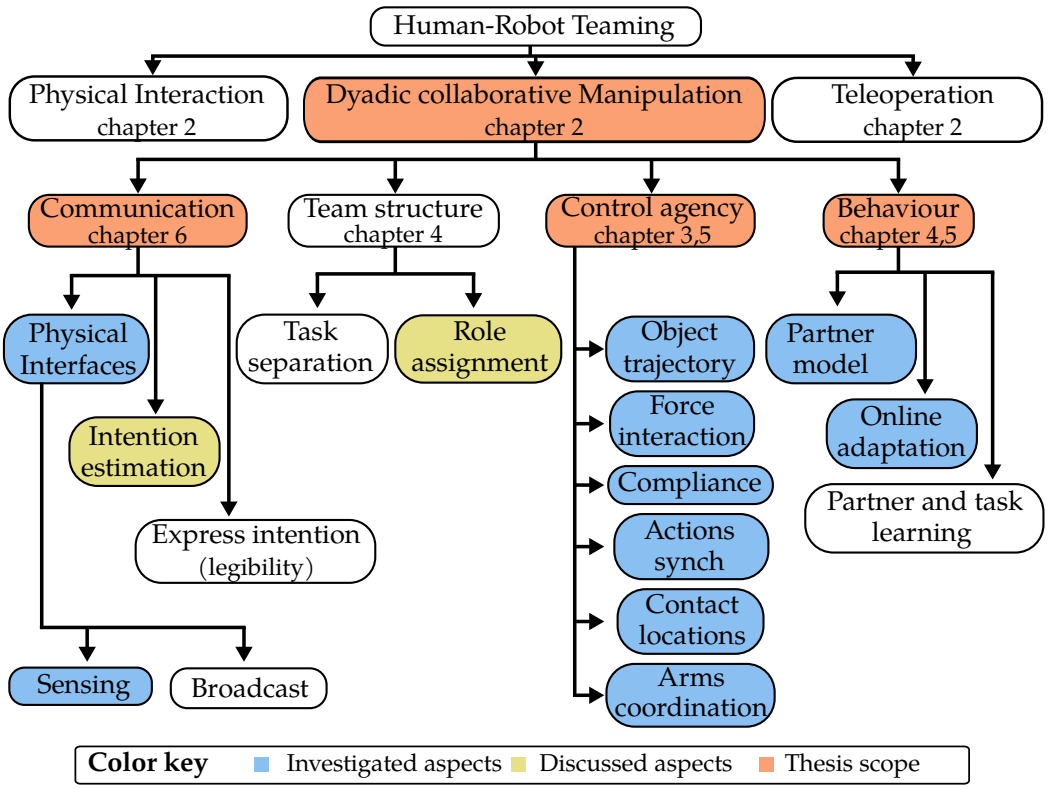


Figure 1.3: Aspects of a DcM setup.

plans online that can be adapted on-the-fly with respect to changes during the task, such as goal changes of the dyad.

1.3 THESIS OUTLINE

The remainder of this thesis is organised as follows:

- Chapter 2 presents an overview of Human-Robot teaming research via a description of the main schemes used in HRI. Next, we introduce a formal description of the investigated DcM system and we outline its components. Investigates question 1 and corresponds to contribution 1.2.1.
- Chapter 3 proposes and evaluates a computational method that enables robotic agents to plan and act in hybrid action space (contact and force) in order to manipulate objects—even for tasks with contacts at speed. Addresses challenge 1 and 2, investigates question 2 and corresponds to contributions 1.2.2 and 1.2.4.
- Chapter 4 introduces a model to represent the policy of the human partner and extends the hybrid TO method proposed in Chapter 3 into dyadic setups. Addresses challenge 3, investigates question 3 and corresponds to contributions 1.2.4 and 1.2.5.

- Chapter 5 is concerned with online adaptation of the agent's behavior. To realize this, we proposes a bilevel optimization method able to achieve fast planning with short computation times. Further, we demonstrate on-the-fly adaptation of the robot hybrid plans with respect to changes in the dyadic setup. Addresses challenge 4, investigates question 4 and corresponds to contributions 1.2.3 and 1.2.5.
- Chapter 6 summarises this thesis, discusses its limitations, outlines future work proposals, and concludes the thesis.

The schematic diagram of Fig. 1.3 displays the investigated and discussed aspects in this thesis within the general context of DcM scenarios. Regarding the reading of the thesis, it is recommended to follow the order of the chapters. All chapters include a brief introduction and discussion, such that the main idea can be understood without delving into the details of each chapter.

2

FROM INTERACTION TO JOINT-ACTION IN HUMAN-ROBOT DYADS

> Coming together is a beginning,
> staying together is progress, and
> working together is success.
>
> Henry Ford

In this chapter, we present an overview of Human-Robot teaming research with a particular focus on Human-Robot dyads and their modelling principles. To acknowledge some of the earliest forms of Human-Robot architectures, we first present the main schemes used to enable team synthesis between a human and a robot, and discuss their usefulness as well as their conceptual limitations. Next, we introduce a formal description of the investigated Human-Robot system and we delineate its components. This description establishes the basic structure of our work, which allows us to put in context the individual pieces of work, and serves as a guide to the following chapters. Finally, we discuss the utility and challenges of the proposed rationale.

2.1 SCHEMES OF HUMAN-ROBOT DYADS

As it has been discussed in the previous chapter, in this thesis we are interested in forming a pair between a human and robot that can work together towards a common manipulation goal. In comparison to stand-alone robotic systems in which a robot executes its own action without further being influenced by other individuals, Human-Robot systems consider the exchange of information and/or commands between a human and a robot. There is a number of different schemes that allow this synthesis and can be described by the following aspects: (i) the flow of information between the entities, (ii) whether distance, physical barriers or time delays exist between the actions of the entities, (iii) each entity's level of control authority on the environment, and (iv) the role of each entity within the dyad. According to these properties next we present the main categories of these setups along with a brief description of each one of them. Further, in Table 2.1 we provide a concise overview of the different schemes—shown in Figs. 2.1 to 2.4—and their respective properties.

Table 2.1: Overview of Human-Robot dyadic setups and their four fundamental properties.

Scheme \ Property	Information flow	Physical Proximity	Agency on environment	Role assignment
Teleoperation	Uni-directional	Barrier	Robot	Master-Slave
Physical Interaction	Bidirectional	Shared workspace	Robot	Master-Slave
Cooperation	Bidirectional	Minimal	Divided	Mixed-Initiative
Collaboration	Bidirectional	In-contact	Shared	Equal

Although these schemes have not been developed and used in a strict chronological order, in the description that follows we adopt an incremental perspective.

2.1.1 *Teleoperation*

Teleoperation of a robotic system means operating a robot at a distance. The term "tele" ("$\tau\eta\lambda\epsilon$") is of Greek origin and lends the context of far distance to the word operation. In robotics, teleoperation typically implies the existence of a spatial and/or temporal barrier between the human operator and the robot. To surpass this barrier an interface between the human operator and the robot is used, as shown in Fig. 2.1. The human transmits cognitive decisions and plans to the robot through the interface, while the robot acts on the remote environment.

Teleoperation setups were the first *human-in-the-loop* approaches in robotics. One of the earliest pieces of work dates back in 1950s (Goertz, Grimson, and Kohut, 1961). Since then, this approach has been used in a number of different domains, such as robotic surgery (Sung and I. S. Gill, 2001), nuclear waste disposal (Abi-Farraj, Pedemonte, and Giordano, 2016), space robotics (Lii et al., 2010), assembly (Sagardia et al., 2016), and subsea (Murphy et al., 2011). Robotic teleoperation remains a very active area of research with many branches, such as direct control, shared autonomy and supervisory control (Chapter 43 in Siciliano and Khatib, 2016). The main attributes of this approach are: (i) the flow of information is usually unidirectional, but can be also bidirectional, (ii) the spatial separation (barrier) excludes any direct physical link between the human, the robot and its environment, (iii) the state of the environment is not directly altered by the human, as the actions of the user are filtered by the interface and the robot, and (iv) the human commands the robot, so these systems are referred to as master–slave systems. The prime challenge is the potential

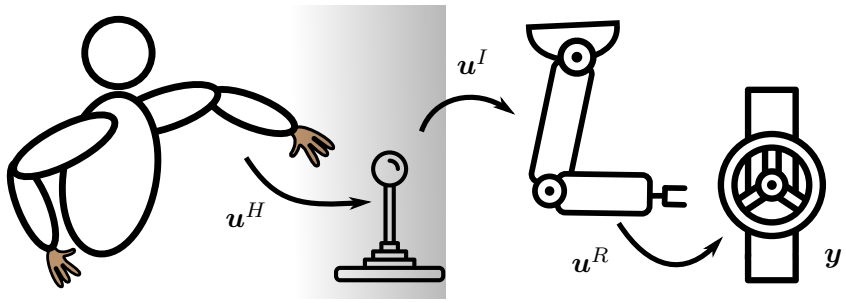


Figure 2.1: Pictorial illustration of the teleoperation scheme. The shaded region illustrates the barrier between the user and the robot, while only the robot can act on the environment. $\boldsymbol{u}^H$ are the control commands of the human. $\boldsymbol{u}^I$ are the control commands send from the interface to the robot. $\boldsymbol{u}^R$ are the controls of the robot applied on the environment and $\boldsymbol{y}$ denotes the state of the environment.

time delay of the information and control flow along the chain of commands from the human to the robot and eventually their effects on the environment. Other challenges include the selection of an appropriate interface, as well as the accurate distribution of autonomy between the operator and the robot (Merkt, Y. Yang, et al., 2017). More details on a shared autonomy system for remote operations, which we developed, can be found in Appendix A.

Teleoperation systems require commands from the human and can communicate information back to the human. Depending on the the existence of feedback, the flow of information can be either unidirectional or bidirectional. If the user interface is bidirectional—e. g. force feedback (Hirzinger et al., 2005)—then such teleoperation systems enter the realm of human–robot interaction as the human (master) is influenced by the sensed state of the robot (slave) and perceived state of the environment.

2.1.2 *Physical Interaction*

Until recently, most robots were position-controlled rigid platforms that were possibly dangerous for any human entering their workspace. To avoid human injuries physical barriers—similar to the one described in the teleoperation scheme—are placed in industrial and automation applications (see Chapter 1). Nevertheless, developments such as compliant robots with series elastic actuators or with variable impedance interfaces have enabled robot systems to safely act in shared workspaces with humans. These advancements have led to the realization of the physical interaction scheme referred as *physical Human-Robot Interaction* (pHRI), which is depicted in Fig. 2.2.

In elementary pHRI setups the robot acts on the environment while being compliant to the human's actions. These action may indicate the objectives of the hu-

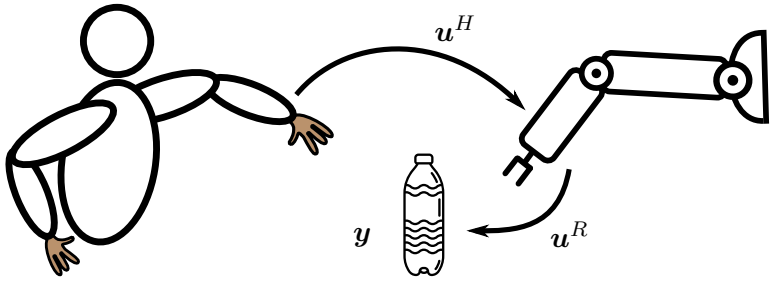


Figure 2.2: Pictorial description of a physical interaction scenario between a human and a robot (inspired by (Dragan, 2017)). In this approach only the robot agent acts upon the environment, while the human can physically interact with the robot to guide it to the goal.

man, referred as corrective actions, or they may occur as disturbances due to variations of the human's behavior. A classical example of pHRI are robotic exoskeletons (Riener, 2013). An exoskeleton is controlled by the high-level commands of a human operator, while the robot is responsible to assist through low-level actuation. One could perceive exoskeletons as an instantiation of a teleoperation setup without a barrier, as there is physical interaction between the human and the robot. Other examples include robots working side-by-side with humans in the factory floors (Beetz et al., 2015), homecare robots physically supporting humans (Nokata, Ikuta, and Ishii, 2002) and providing elderly care (Ikuta, Ishii, and Nokata, 2003). In all these examples, the robot performs tasks for humans under their supervision and in physical proximity to them.

The key features of this scheme are: (i) there is direct access to the environment's state by both individuals (human and robot), yet the flow of commands is unidirectional from the human to the robot, (ii) the human can physically interact with the robot, (iii) the human expects from the robot to perform the task correctly and thus, the human agency on the environment is realized through the robot, and (iv) the robot complies to the physical interventions of the human, so its slave role remains in this scheme too. One of the key challenges is safety of the human, which is being addressed by preventing and mitigating the effect of unexpected contacts or collisions (Haddadin, Parusel, et al., 2013; Haddadin, Albu-Schäffer, and Hirzinger, 2009). Another core development was the design of human-friendly robot (Zinn et al., 2004) that can minimize tentative human injuries (Haddadin, Haddadin, et al., 2012). A comprehensive literature review on pHRI can be found in (De Santis et al., 2008). Given these advances, recent research focuses on how a human-friendly robot can revise its own behavior according to the physical interaction with the human. A number of works treat the physical input of the human as guidance to learn new behaviors, e. g. a humanoid standing-up (Ikemoto et al., 2009), to estimated the user's internal

goals and align them with those of the robot (Bajcsy et al., 2017) and to increase the performance of an assembly task correcting the trajectories of the robot (Likar et al., 2015).

Physical interaction between the human and the robot does not only lead to the pHRI setups described above, but also pave the way to the two schemes described next. In the following two categories, the agency is incrementally distributed equally between the two individuals.

2.1.3 *Cooperation* (*assistive action*)

Cooperation between a human and a robot arises when the two individuals work together in support of one another's goals. In *Human-Robot Interaction* (HRI) setups the robot assists the human to complete the task in hand and fulfill the person's goals. The term "cooperation" ("cooperari") is of Latin origin and means work *together*. The prefix "co-" adds the notion of bringing *together* several bodies and the word "operari" means work.

In the *Human-Robot cooperation* scheme—illustrated in Fig. 2.3—physical interaction is materialized in a transactional fashion, where the two individuals need to coordinate their actions in space and time to achieve the desired task. One of the most evident applications is flexible automation, where the robots can help humans assemble parts in both small and medium size enterprises (SMEs) (Krüger, Lien, and Verl, 2009; Nottensteiner et al., 2016). In these setups contact between the two individuals is infrequent. Earlier research was about robot motion planning methods aiming to avoid contact with the human partner while completing the task (Kulić and E. A. Croft, 2005; Kulić and E. Croft, 2007; Ebert and Henrich, 2002), as well as safe robot motion planning in case unexpected contact with the human happens (Haddadin, Albu-Schaffer, et al., 2008). In this approach the primary task originates from the human's objectives and is typically separated into a set of distinct and complementary subtasks. Each subtask is assigned either to the human or the robot, thus the interaction consists mostly of tool handovers, reciprocal action in turn-taking fashion with the same part or tool, manipulation of separate parts/tools, etc (Chapter 69 in Siciliano and Khatib, 2016).

The key properties of this scheme are: (i) both individuals have direct access to the environment's state, as a result individuals' actions can trigger bidirectional adaptation of their behaviors, (ii) the human and the robot have a common shared workspace and they occasionally interact physically, (iii) the two co-actors (human and robot) have shared agency on the environment that is divided into separate parts, and (iv) the robot has the slave role with regards to the overall task, but it acts autonomously

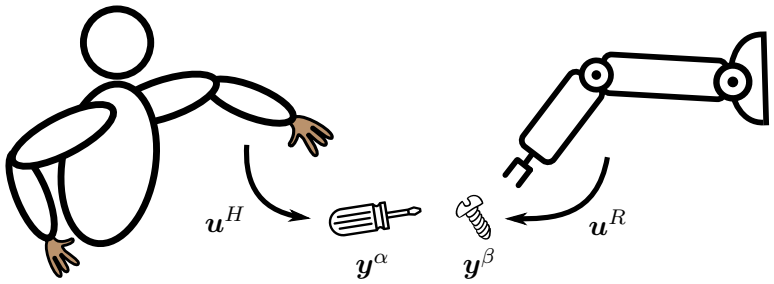


Figure 2.3: Pictorial demonstration of an assistive action setup, where the robot cooperates with the human towards the completion of the human's goal. The two individuals share the same workspace, yet the interaction is transactional in nature and each individual acts upon a separate item. $\boldsymbol{y}^\alpha$ and $\boldsymbol{y}^\beta$ indicate the different parts (items) of the environment's state.

within the subtasks, while it can initiate/lead a handover. Major challenges of this scheme revolve around the cognitive and decisional levels of the interaction. Specifically, the robot needs to predict the human's intentions and adapt its behavior such that the physical interference between the two is minimized. This led to research on the topics of estimating human intention to anticipate object-related human activities (Koppula and Saxena, 2016), human preferences-based assistance depending on the pHRI task (Grigore et al., 2018), coordination models between individuals for autonomous driving, where different manned or unmanned cars need to coordinate to merge their paths on a highway or to avoid getting stuck on a crossroad (Sadigh et al., 2016), and safe concurrent reaching motion planning through human motion prediction and robot trajectory planning modification (Mainprice and Berenson, 2013).

In all these examples prediction of the human's goals and motions along with adaptation to the human's behavior is crucial. The development of such capabilities for the robots aims to enable them to operate next to humans in an autonomous manner and proactively adapt their actions to their partner's behavior. Yet, the two individuals are not physically coupled and typically engage with different subtasks or parts of the environment. These two specific features of the cooperation scheme result in setups that first, do not require high frequency level alternations of ones own actions and second, potential conflicts can be simply resolved by freezing the motion of the robot. Similarly in our own work (Merkt, Y. Yang, et al., 2017), when tentative collision between moving objects (humans) and the robot are detected, the motion of the robot is immediately halted.

2.1.4 *Collaboration* (*joint action*)

Collaboration between a human and a robot occurs when the two of them work together towards a shared objective. This particular scheme of pHRI is named *physical Human-Robot Collaboration* (pHRC). The term “collaboration” ("collaborare") is of Latin origin and means to work *with* someone. The prefix "co-" is adjusted to "col-" which adds the notion of acting *with* someone and the word "labor" means work.

The most commonly mentioned example of pHRC is when two individuals jointly move an object, e. g. a box or a furniture, as shown in Fig. 2.4. This example manifests the defining difference between pHRC and the schemes described above. Due to the physical coupling, there is immediate effect of ones actions on the behavior of the corresponding partner, the two co-actors need to align their actions in a temporal and spatial manner as well as to coordinate the force interactions to achieve the desired task (Sebanz, Bekkering, and Knoblich, 2006). In other words, both individuals must continuously adjust their own actions with respect to the intentions, plans and motions of their partner (Clark, 1996). Instances of this scheme have been applied on collaborative object lifting (Evrard, Gribovskaya, et al., 2009), transporting (Mörtl et al., 2012) and on joint motion tracking (Y. Li et al., 2015). In these works the robot is tightly coupled with the human and can alternate between the leading and the following role depending on the state of the task and the magnitude of the interaction forces. A load sharing framework with predefined sharing modes was presented by Lawitzky, Mörtl, and Hirche, 2010 to allow one or more robots to carry a load with a human. In general, shared manipulation research is concerned with methods able to select suitable robot action in every time step based on the state of the human partner and the environment, while the majority of these works utilize some form of impedance control (N. Hogan, 1985; N. Hogan and Buerger, 2018) to achieve safe and compliant behavior during the interaction.

A closely related area of research is *Cooperative manipulation*, which is concerned with the topic of manipulating a common object by means of two or more robotic arms (Chapter 39 in Siciliano and Khatib, 2016). The specific topics of research are motion and force interaction control between the manipulators. The main difference between *cooperative manipulation* and pHRC is that the former requires *intrapersonal* coordination (Billard and Kragic, 2019; Gams et al., 2014), which sets a simplified problem as a robotic arm has full knowledge of the state and plans of any other cooperative manipulator. On the other hand, the latter requires *interpersonal* coordination (R. Schmidt et al., 1998), which poses challenging problems due to lack of information regarding the partner’s state, actions, plans and intentions (S. V. Albrecht and Stone, 2018). The key characteristics of this scheme are: (i) similar to the coop-

Intrapersonal refers to processes that occur within one person.

Interpersonal refers to processes that occur between people.

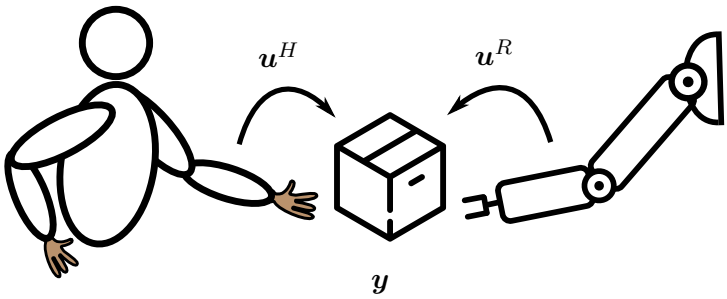


Figure 2.4: Pictorial portrayal of a joint action scenario, where a human and a robot collaboratively manipulate a common object. The two individuals share the same workspace, act upon the same part of the environment and the actions of the two need to be in harmony, due to the physical coupling.

eration scheme, the human and the robot can sense the environment directly, which allows them to re-plan and react depending on the sensed state of the world, (ii) the human and the robot simultaneously manipulate a common object and form a tightly coupled system through the physical medium, (iii) any change in the environment results from the blended actions of the individuals, thus the agency on the environment is shared between the two partners, (iv) the roles of both co-actors can change in an arbitrary fashion from master to slave and vice versa; in addition, both individuals may hold similar roles and contribute equally to the task.

The important challenges of this scheme include open questions mentioned in the two previous schemes. *Human intention estimation* is one of them, with canonical examples being the works of Madan et al., 2015 and Lanini et al., 2018 where multi-class classifiers were utilized to recognise human partner's commands through force interaction in co-manipulation tasks. Also, conditional random fields were adopted in Hoare and Parker, 2010 to infer the human's intended goal during box co-pushing tasks. Another subject of research is robot *motion adaptation* to the partner's action. Evrard and Kheddar, 2009 developed a robot controller based on homotopy to switch continuously between two distinct extreme behaviors (leader and follower), while Al-Jarrah and Zheng, 1997 proposed a reflexive motion controller to share a load between the human and the robot to reduce the strain on the human side. A turn-taking coordination concept was extended by Peternel, Tsagarakis, and Ajoudani, 2017 to allow collaboration between a human and a robot for a joint sawing task. Nikolaidis, Kuznetsov, et al., 2016 presented a human-robot mutual adaptation study, where the human and the robot of the dyad can select left or right rotational actions in a virtual table transportation task. Despite the progress made in *human intention estimation* and robot *motion adaptation*, both areas of research attract a lot of interest as their integration in pHRC is considered crucial (Ajoudani et al., 2017).

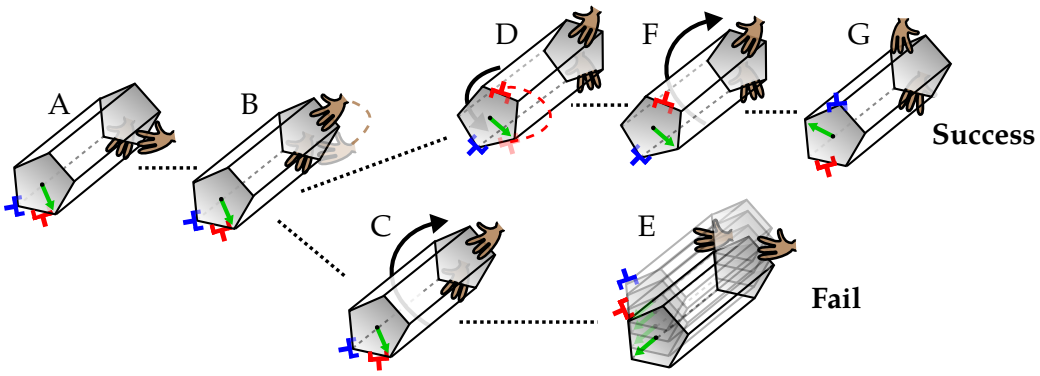


Figure 2.5: Pictorial illustration of different action paths to collaboratively manipulate the object from the initial pose (A) to the goal pose (G). The tree illustration has two branches. The lower branch (A-B-C-E), which does not involve any change of grasp from the robot side and results, in an object drop. The upper branch (A-B-D-F-G), where the robot changes grasp-hold at (D) after the human partner changed grasp-hold at (B), results in successful joint rotation of the object towards the goal (G).

Yet another challenge was introduced by the early work of Sheridan, 1997, where he identified the need to acquaint both co-actors (humans and robots) with models of their partners. This led to the subject of *co-representation* in *Human-Human Interaction* (HHI) (Wenke et al., 2011). In pHRC, Y. Maeda, Hara, and Arai, 2001 used a polynomial to model the human motion, while Gribovskaya, Kheddar, and Billard, 2011; Ghadirzadeh et al., 2016 introduced a task model that is learned offline to guide the interaction at the reproduction phase. However, the understanding and development of such models is still at its infancy. Last but not least, each individual's *agency in joint action* is another topic of research (Pacherie, 2011). Most pHRC approaches focused on the force interaction capabilities of the robots, with force-torque regularization (Lin et al., 2018; Otani, Bouyarmane, and Ivaldi, 2018) and impedance modulation methods (Gribovskaya, Kheddar, and Billard, 2011; Noohi, Žefran, and Patton, 2016) being the most commonly used. The controlled motion attributes of the robot enable the artificial agent to fulfill the task and shape the actions of the human partner, thus the selection of the appropriate robot action space remains an open question.

This thesis investigates a number of aspects related to the typical joint action example (see Fig. 2.4), where two individuals jointly manipulate an object. We refer to these tasks as *Dyadic collaborative Manipulation* (DcM) scenarios, as introduced in Chapter 1. In particular, we are interested in scenarios where a robot transports a large object with a human. During the task execution shown in Fig. 2.5, there are instances in which the current contact configuration is not sufficient for the continuation of the task, e. g. rotating the object upside-down (Fig. 2.5 A to Fig. 2.5 G). To avoid such deadlocks, both the human and the robot should predict the future state

of the object and change their contact locations accordingly, as shown in Fig. 2.5 B and Fig. 2.5 D. As illustrated in Fig. 2.5 E and Fig. 2.5 G, contact adjustments are crucial, as they result in failure or success of the task. Further, all actions like contact changes must comply with the partner's actions to jointly balance the object. Next, we provide a formal description of DcM systems.

2.2 FORMALIZING DYADIC COLLABORATIVE MANIPULATION

In this section, we provide a partitioned description of a DcM system, which enables us to describe each of its components formally and obtain a detailed view on the structure of such setups. Fig. 2.6 provides a graphical representation of DcM as a system, which is separated into three components: (i) the partner's policy π^p, (ii) the dyadic interaction, and (iii) the agent's policy π^a. With policy we refer to the function that maps the state of a system and its environment to actions, as defined by Dyer and S. R. McReynolds, 1970. This definition is sufficient to describe the behavior of an isolated individual in the world (human or robot), however it does not capture the interactive aspects of a DcM setup. Thus, key in this formulation is the dyadic interaction, which is used to capture the binding between the two individuals, both in physical and in mental terms. The physical pairing arises due to the object, whose state results from the joint action and acts as the physical medium for exchanging information, while the intentions of the individuals are naturally correlated due to the common task of the dyad. Next, we provide the formal description of these three components.

2.2.1 *Partner's policy*

Given the fact that the human's internal state and planned actions cannot be directly observed, in the block diagram of Fig. 2.6, the policy of the partner is depicted as an estimation block. An estimate of the partner's policy—denoted with $\hat{\pi}^p$—can be obtained based on a set of sensory measurements, an intention prediction process and a parametric model of the policy. In the proposed DcM formulation, the parametric model of the partner's policy depends on the state of the object $\mathbf{x}_t = [\mathbf{y}_t{}^T \ \dot{\mathbf{y}}_t^T]^T$, where $\mathbf{y}_t \in \mathbb{R}^\nu$ is the object pose at timestep t, and can be described by a set of parameters θ^p, formally written as

$$\boldsymbol{\lambda} = \hat{\pi}^p(\mathbf{x}_t; \theta^p), \tag{2.1}$$

$\boldsymbol{\lambda} \in \mathbb{R}^\nu$ is the contribution of the partner's policy with respect to the task, e. g. forces applied on the manipulated object. Further, to comply with the sequential nature of

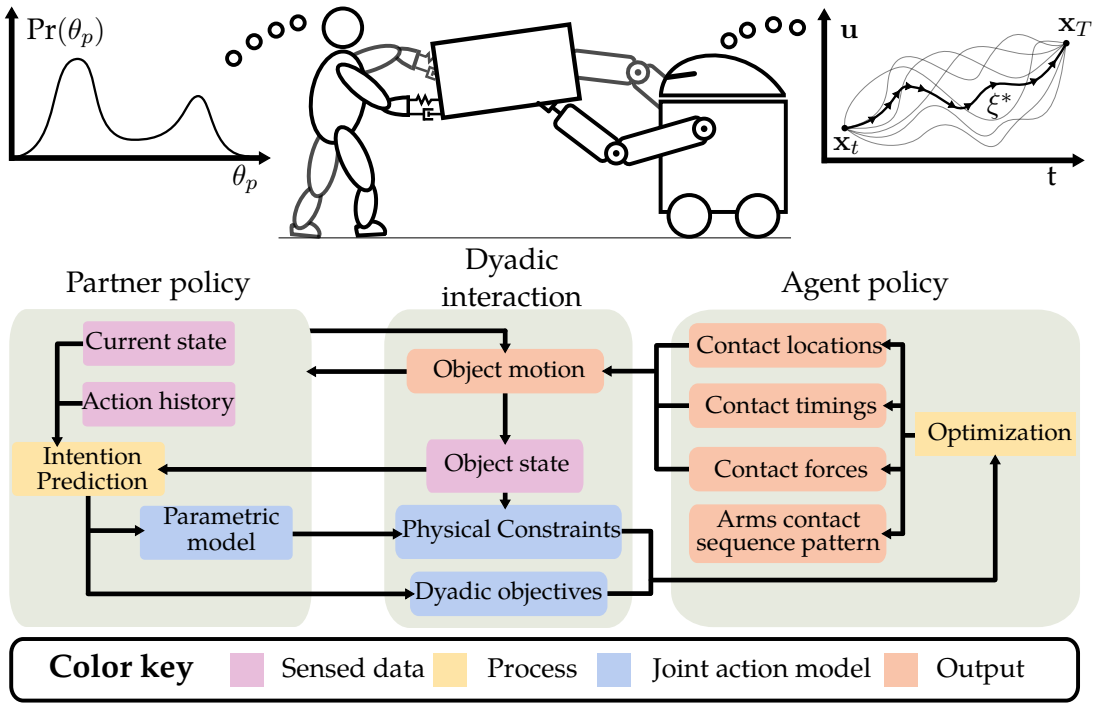


Figure 2.6: A typical DcM scenario along with a modular description of DcM as a system. On top left a multi-modal distribution describes the policy of the partner and top right illustrates the optimal trajectory ξ^* computed from the policy of the agent along with few other feasible trajectories. $\mathbf{u}$ represents the action space of the agent, $\mathbf{x}_t$ and $\mathbf{x}_T$ represent the task through the initial and final state of the manipulated object.

DcM tasks—illustrated in Fig. 2.5—the model of the partner's policy should be non-stationary. This can be represented with a multi-modal probability distribution

$$Pr(\theta^p|\mathbf{x}_t, \mathbf{q}_t^p, H^p) \tag{2.2}$$

over parameters θ^p, given the sensed data $\mathbf{x}_t$, the partner's configuration $\mathbf{q}_t^p \in \mathbb{R}^\rho$ and a history of partner's configurations $H^p \in \mathbb{R}^{0:t\times\rho}$. In every time instance of the dyadic interaction, the partner's policy is described by one of the modes of the distribution as shown in top left of Fig. 2.6. For instance, one mode can represent the attitude of the partner when rotating the object, while another mode can describe the partner's behavior when translating the object. The identification of the current partner's behavioral mode can be obtained by an intention estimation process, which informs the parametric model and the dyadic interaction module.

2.2.2 *Dyadic action*

The joint task shapes the dyadic action (joint action) of the two co-actors. In DcM scenarios the joint task—i. e. the object motion—is a function of the two individuals' policies described by the following formula

$$\mathbf{x}_{t:T} = f(\pi^a, \pi^p). \tag{2.3}$$

This pairing is formed through the physical constraints seen in Fig. 2.6. The exact configuration of this pairing—e. g. role allocation within the dyad—can be regulated through the dyadic objectives. Thus, the policies of the two individuals are coupled, forming the dyadic action. We represent this relationship with

$$\xi = \pi^a(\mathbf{x}_t, \hat{\pi}^p, \theta^{\mathcal{D}}, \theta^{\mathcal{M}}), \tag{2.4}$$

ξ denotes a trajectory defined in Section 2.2.3.

which indicates the dependency of the agent's policy to the state of the object $\mathbf{x}_t$, the estimated policy of the partner $\hat{\pi}^p$, the parameters of both the dyadic setup $\theta^{\mathcal{D}}$ and the manipulation task $\theta^{\mathcal{M}}$. The details of the agent's policy are given next.

2.2.3 *Agent's policy*

A trajectory ξ is a time-indexed sequence of configurations and actions that guides the object to the goal state $\mathbf{x}_T$ given its current state $\mathbf{x}_t$. In the top right of Fig. 2.6, we illustrate a few trajectories (grey) from all feasible ones that satisfy the task, as well as the optimal one (black) with ξ^*.

We define the full control policy π^a of an agent participating in DcM tasks as a function $\pi^a(\cdot) \rightarrow \xi$, where ξ in the most generic form can be used to represent a trajectory with several components as

$$\xi = \begin{bmatrix} \mathbf{f}_t^{\kappa\,T} & \mathbf{c}_t^{\kappa\,T} & \mathbf{\Delta T}_t{}^T & \mathbf{K}_t^{a\,T} & \mathbf{D}_t^{a\,T} & \mathbf{q}_t^{a\,T} \end{bmatrix}^T \quad \forall i \in \mathbb{N}, \tag{2.5}$$

where $\mathbf{f}_t^{\kappa} \in \mathbb{R}^{\nu}$ and $\mathbf{c}_t^{\kappa} \in \mathbb{R}^{\nu}$ denote the forces applied and contact locations used by agent at timestep t, κ is used to index to the κ_{th} end-effector of the agent, $\mathbf{\Delta T}_t$ is the duration of timestep t, $\mathbf{K}^a, \mathbf{D}^a \in \mathbb{R}^{\nu}$ are the cartesian stiffness and damping of the agent at timestep t and $\mathbf{q}^a \in \mathbb{R}^n$ is the agent's configuration at timestep t. Note that our work is motivated by the illustration provided in Fig. 2.5 and aims to generate hybrid motion plans. With hybrid motions we refer to motion plans that belong in the force-contact space, which is modelled with variables $\mathbf{f}_t^{\kappa}$ and $\mathbf{c}_t^{\kappa}$ to consider continuous regulation of forces and discrete contact changes.

The force-contact space is a hybrid space of both continuous and discrete variables.

In DcM scenarios the agent's policy depends on the parameters of both the dyadic setup and the manipulation task as well as on the current estimate of partner's policy. To obtain the optimal policy of the agent, we express partner-aware dyadic planning with

$$\xi^* = \min_{\pi^a} \int_0^T c(\pi^a, \hat{\pi}^p, \theta^{\mathcal{D}}, \theta^{\mathcal{M}})\, dt \quad \text{s.t.} \quad \boldsymbol{g}(\pi^a, \mathbf{x}_t, \hat{\pi}^p, \theta^{\mathcal{M}}) \leq 0, \tag{2.6}$$

by introducing the idea of considering the partner's actions into the motion plans of the agent through the constraint functions $\boldsymbol{g}(\cdot)$. As the partner's behavior is non-stationary, the parameters θ^p of the partner's policy π_p need to be estimated repeatedly during the dyadic action to provide $\hat{\pi}^p$, which will trigger an update of π^a. Further, the configuration of the dyad defined via $\theta^{\mathcal{D}}$ is met through the cost function $c(\cdot)$, where $\theta^{\mathcal{D}}$ can represent e. g. the role assignment within the dyad. Additionally, T is the total time duration of the trajectory and the task specifications can be satisfied either through the cost function $c(\cdot)$ or the constraints $\boldsymbol{g}(\cdot)$, where $\theta^{\mathcal{M}}$ may define e. g. the final pose of the object or a constant linear/angular velocity of the object.

2.3 DISCUSSION

This chapter presents a formalization towards addressing DcM scenarios. In our partner-aware dyadic formulation presented above we treat the two co-actors and the dyadic action separately (see Fig. 2.6). We believe that this design choice is of core importance. This allows the methods described in the following chapters to generalize over different tasks and partner behaviors, while the resulting action plans satisfy the joint action conditions.

Similar treatment was also used by Takagi, Ganesh, et al., 2017 to analyze joint action in Human–Human dyads, where a model for interpersonal goal integration was proposed. The authors show evidence of humans utilizing an estimate of their partner's goal to update their own task's goal and improve the performance in terms of the joint task. Further, in the fields of psychology, cognition and behavioral analysis, it is widely accepted that humans possess separate mental models for the task and for their partners (Mathieu et al., 2000). Also, it has been shown that planning and adjusting of one's own actions to the actions of the partner is central to joint action (Meyer, Wel, and Hunnius, 2016). This is achieved by merging their partner's task-based action representation into their own task-based action planning (Atmaca et al., 2008; Bruijn, Miedl, and Bekkering, 2011; Wenke et al., 2011). Sebanz, Knoblich, and Prinz, 2003 provided evidence of individuals representing their own and their

partner's actions in a functionally equivalent manner. In the proposed DcM formulation we aim to achieve analogous behaviors to the ones described above using the dyadic action module along with the parametric model of the partner.

Nevertheless, in the field of robotics, interaction models for dyads are separated in three prevailing schools of thought, each one of them with its own merits.

Control focused: Agravante et al., 2014 used impedance control in combination with a vision-based controller to accommodate partner's actions and collaboratively carry a table with a human, while balancing a free-rolling ball on it. Lawitzky, Mörtl, and Hirche, 2010 presented a load sharing framework based on force space redundancy, where the load sharing mode is predefined. In both cases, the human partner is treated as an external disturbance to the system, which hinders joint action as the robot agent can only react.

Coupled-policies focused: G. Maeda et al., 2017 proposed a method to transfer adaptive hand-overs to robots from kinesthetic demonstrations. Similarly, Rozo Castañeda et al., 2013 proposed to employ force and vision information to commence the appropriate learned impedance behaviors depending on the task phase. Data-driven extraction of interaction constraints during hand-over tasks was proposed by Vogt, Stepputtis, Jung, et al., 2018. The extracted constraints were then used to form online robot responses. These methods couple together the policies of the robot agent, the human partner, and the task evolution to learn a direct mapping towards generating online adaptive robot responses. Thus, their generalization capabilities are limited to the demonstration set.

Partner-model focused: Y. Maeda, Hara, and Arai, 2001 used a polynomial model to predict human motion and accordingly update the robot's goal towards minimizing the energy transferred from the human leader to the robot follower. Gribovskaya, Kheddar, and Billard, 2011 proposed a method, which learns the task model offline in the form of a dynamical system. At the reproduction phase, an adaptive impedance law is adopted to compensate for the unmodelled parts of the interaction. Similarly, Ghadirzadeh et al., 2016 used Gaussian Processes to learn the task model and Reinforcement Learning to train the robot policy upon the learned model. Such methods are elegant, as each entity (human partner, robot agent, task) of the dyadic action are modelled separately, actions of the two co-actors can be appropriately reasoned upon and allow to obtain generalizable robot behaviors. The DcM formulation, presented here, follows the same principle.

3

MANIPULATION PLANNING IN HYBRID SPACES

> The future depends on what we do in the present.
>
> Mahatma Gandhi

The previous chapter sets out the formal framework used to address the core challenges in Human-Robot dyads. One of the main components of the framework is the agent's policy. In Chapter 1, we point out that the agent's repertoire of actions needs to be sufficiently rich to participate in general manipulation and DcM tasks. In this chapter, we present the computational models developed to enrich robotic agents' policies with a comprehensive set of actions that span across the hybrid space.

Hybrid is the space of both continuous and discrete variables.

Humans organize their manipulation actions by combining force control with sequential planning. Yet, the secret behind humans' remarkable manipulation skills, is our competence in control and prediction of *contact events* (Flanagan, Bowman, and Johansson, 2006)—which are the crux of sequential manipulation (Dafle et al., 2014). Nevertheless, our understanding of the mechanisms of sequential decision making (Dayan and Daw, 2008) is still subject of research. To understand the type of tasks investigated here, let us consider actions like placing a box on a warehouse rack. The higher the shelf is on the rack, where the box should be placed, the lower are the grasp-holds favored by the human placing the box. With analogous experimental scenarios, Meyer, Wel, and Hunnius, 2013 examined whether humans perform higher-order planning[1] for action sequences—both when acting individually or jointly with another human partner. Their results suggest a similarity in the underlying mechanisms used during individual and joint action sequence planning for manipulation. Furthermore, it was shown that participants carried over their experience gained as individuals to the dyadic tasks.

Similarly, in this chapter we focus on a single robot agent, while extensions into dyadic setups are investigated in the following chapters. In order to familiarize the reader with the concepts presented here, first the background theory is provided both on the optimization based motion planning and on hybrid systems. Next, the relevant robotics literature is reviewed and the details of the proposed hybrid motion planning method are described. To validate the method, two different experimental

1 Higher-order planning is a process, according to which an agent's immediate action depends on one or several future actions (definition was instigated by (Meyer, Wel, and Hunnius, 2013)).

studies are considered: the first investigates the problem of object manipulation with multiple contact switches, and the second examines a scenario where contact with the object is realized at speed. Finally, we analyze the advantages and disadvantages of the proposed approach and discuss promising research directions.

3.1 BACKGROUND

With *motion planning* in robotics (LaValle, 2006), we refer to the process that enables robots to find a sequence of actions that lead to an *ego-motion*, which fulfils their goals, e. g. drive the robot from a start to a goal pose. In *Manipulation planning* we are not only interested in the *ego-motion*, but also in actions that can bring about the desired change to the environment, e. g. move an object to a particular location. Thus, with *manipulation planning* (Alami, Simeon, and Laumond, 1990; Alami, Laumond, and Siméon, 1997) we refer to the process that enables robots to find a sequence of ego-actions and environment interactions to accomplish their goals.

Ego-motion is motion of one's own.

Generally speaking, any type of robot planning process in continuous space belongs into one of two families of approaches: (i) sampling-based and (ii) optimization-based. Each category prioritizes a different property of the planning process, the first concentrating on finding a feasible plan that satisfies the task, and the second is focusing on generating plans that satisfy the task and optimize a goodness metric (cost function). The former promises to be asymptotically complete,[2] even in very complex motion planning problems (Kavraki et al., 1996; Choset et al., 2005; Lavalle, 1998; LaValle and Kuffner Jr, 2001; Kuffner and LaValle, 2000). The latter can attain both optimality and completeness in the convex domain, but its pledge for optimality does not admit any guarantees of finding a solution in non-convex problems. Yet, the recent developments in the field of numerical optimization and nonlinear programming (Betts, 2010; Nocedal and Wright, 2006; P. E. Gill, Murray, and Saunders, 2005; Wächter and Biegler, 2006) have enabled the use of these methods towards addressing challenging non-convex robotics problems (Diehl et al., 2006; Ratliff et al., 2009; Zucker et al., 2013; Schulman et al., 2014). Motivated by these advancements, in this chapter we explore optimization-based planning methods and for this reason, we next provide a compact overview of the underlying theory.

2 Completeness (Latombe, 2012) reflects the ability of the method to find a solution plan. A planner is complete when it can find a solution plan in a finite amount of time, given that a solution exists.

3.1.1 *Optimal motion generation*

In this approach, the motion of the system (robot) is formulated as a mathematical optimization problem where an objective function is optimized subject to constraints that take the form of differential equations, algebraic equations, and boundary conditions, with decision variables being the control and state of the system. This particular type of problems is referred as *Optimal Control* (OC) problems (Bertsekas, 1995) and are formally described as

$$\min_{\boldsymbol{x}(t),\boldsymbol{u}(t)} \int_0^T c\left(\boldsymbol{x}(t),\boldsymbol{u}(t)\right) dt + c_f\left(\boldsymbol{x}(T)\right) \tag{3.1a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t),\boldsymbol{u}(t)\right) \tag{3.1b}$$

$$\boldsymbol{x}(0) \in \mathbb{X}_0 \quad , \tag{3.1c}$$

$$\boldsymbol{x}(t_f) \in \mathbb{X}_f \tag{3.1d}$$

$$\boldsymbol{g}\left(\boldsymbol{x}(t),\boldsymbol{u}(t)\right) \in \mathbb{Z} \tag{3.1e}$$

$$t \in [0, T] \tag{3.1f}$$

where $\boldsymbol{x} \in \mathbb{R}^\eta$ is the model's state vector, $\eta \in \mathbb{R}$ is the dimensionality of a system's state, $\boldsymbol{u} \in \mathbb{R}^\iota$ is the model's control vector, $\iota \in \mathbb{R}$ is the dimensionality of a system's control, $c(\cdot), c_f(\cdot) \in \mathbb{R}$ in (3.1a) are the running and final cost functions, $\boldsymbol{f}(\cdot) \in \mathbb{R}^\eta$ in (3.1b) describes the system's dynamics, $\boldsymbol{g}(\cdot) \in \mathbb{R}^\zeta$ in (3.1e) describes the equality and inequality constraints of the system, $\zeta \in \mathbb{R}$ is the total number of constraints and (3.1c) to (3.1f) describe bounds on the initial state, final state, path constraints and motion duration, respectively.

Description (3.1) belongs to a general class of optimization problems—termed Infinite Programming problems—where we seek to find a set of continuous functions that fulfill a set of continuous constraints. A way to address a sub-class of these problems—where a cost function (3.1a) is minimized subject to the dynamics of the system (3.1b)—entails finding the solution of the Hamilton–Jacobi–Bellman equations (Kirk, 2004; Todorov, 2006). This leads to a feedback control law (closed-loop solution) for the complete state-space of the system based on the dynamic programming approach (Hillier and Lieberman, 2001; Bertsekas, 1995) and the notion of *value function*[3] (Bellman, 1966). Yet, for many problems a feedback control law is very unlikely to be obtained, e. g. in high dimensional systems (Todorov, 2006) due to the *curse of dimensionality*[4] or problems with general path constraints (3.1e) (Tous-

3 *The value function* maps a state to the total cost for completing the task, starting from this state. Also, referred as *cost-to-go*.

4 *Curse of dimensionality* refers to the exponential growth of the discretized state-space size with respect to the number of dimensions of the state-space (Bellman, 1966).

saint, 2017). Hence, for these problems the focus is turned towards obtaining a local open-loop solution (trajectory), where an optimal trajectory locally approximates the optimal *value function* up to a time horizon. Computing these trajectories can iteratively drive the system to the goal in a closed-loop fashion, through consecutive local approximations of the *value function*. The latter schema is called Model Predictive Control (MPC) (Rawlings, Mayne, and Diehl, 2017). The process involved with finding these local open-loop solutions is named *Trajectory Optimization* (TO) and is discussed next in further detail.

3.1.2 *Trajectory optimization (TO)*

This technique addresses the problem of finding locally optimal trajectories for dynamical systems with path constraints (Kelly, 2017; Betts, 2010) as the one described in (3.1) (Infinite Programming problem). To make such problems computationally tractable, the usual approach is to parameterize the problem using a finite number of decision variables, i.e. express the problem as a constrained parameter optimization problem.

The main two categories of methods are the *indirect* and *direct* approaches. *Indirect methods* are based on the calculus of variations or Pontryagin's Maximum Principle (Bryson, 2018), which also yields the same solutions with the dynamic programming approach. To compute the optimal controls they use necessary conditions that hold over a particular trajectory and then discretize the resulting equations to obtain the optimal solution. These conditions are usually boundary value problems in ordinary differential equations (Hamiltonian dynamical system). Methods based on Pontryagin's Maximum Principle can generalize to OC problems that include terminal state and control constraints. Yet, for the problems investigated in this thesis, multiple general path constraints (3.1e) have to be considered in the problem formulation, which is not straightforward when using this method.

Direct methods for TO first discretize (3.1) and then use standard nonlinear optimization techniques to solve the resulting parameter optimization problem (Betts, 2010). Since standard optimization techniques are used, general path constraints are easily incorporated. This comes with costs regarding the accuracy of the obtained solution (in terms of integration), for which the required level is always application dependent, while the resulting problems are easier to pose and solve. The methods developed in this thesis fall into the direct TO category and there are three main *direct* TO methods: shooting, transcription, and collocation (Rawlings, Mayne, and Diehl, 2017).

In *direct shooting methods*, an integration scheme[5] (e. g. an ODE numerical solver) is used to eliminate state trajectory variables from the problem. As a result, problems in this class require only discretization of the control. To compute the state trajectory, calls to an embedded integrator are needed, which first requires the integrator to provide sensitivities and also can be quite problematic for unstable or stiff systems. To mitigate this, *direct multiple shooting methods* perform both a state and control discretization, while calls to an integrator are still used, albeit for a shorter horizon and multiple separate segments of the trajectory (Diehl et al., 2006). This leads to larger but structured nonlinear problems.

Direct transcription methods do not require calls to an embedded integrator; the discrete system dynamics are enforced as constraints of the nonlinear problem. This is achieved by discretizing both the controls and the states in a grid as well as the objective integral, where the grid points are called knots. These knots are the discretization points of the transcribed continuous problem. Using direct transcription, the optimization problem (3.1) can be expressed as

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=0}^{N-1} c_i\left(\boldsymbol{x}_i, \boldsymbol{u}_i\right) + c_N\left(\boldsymbol{x}_N\right) \tag{3.2a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{i+1} = \boldsymbol{f}\left(\boldsymbol{x}_i, \boldsymbol{u}_i\right) \tag{3.2b}$$

$$\boldsymbol{x}_0 \in \mathbb{X}_0 \quad , \tag{3.2c}$$

$$\boldsymbol{x}_N \in \mathbb{X}_N \tag{3.2d}$$

$$\boldsymbol{g}\left(\boldsymbol{x}_i, \boldsymbol{u}_i\right) \leq 0 \tag{3.2e}$$

$$i \in \{0, N\} \tag{3.2f}$$

where the notation is the same as in (3.1), with the only difference that is written in its discrete form. In the discrete form we replace the subscript t for time with i that indicates corresponding knot. These optimization problems are typically large and sparse, and nonlinear solvers which exploit sparsity (`SNOPT` (P. E. Gill, Murray, and Saunders, 2005) or `IPOPT` (Wächter and Biegler, 2006)) can be used. Direct transcription methods have similar convergence characteristics with direct multiple shooting methods and are preferred for problems with challenging path constraints. Thus, in this thesis we express the motion generation problems utilizing *direct transcription*. Yet, these methods have the drawback that the time discretization and the integration scheme should be carefully selected, since there is a trade-off between accuracy of the solution and computation cost.

5 Integration methods are used to find numerical approximations to the solutions of differential equations (Butcher, 2016).

Finally, in *direct collocation methods* both the control and the state are parameterized by piecewise polynomial functions (splines). Using these polynomial functions, the values of the state and control are computed outside of the knot points (typically at the midpoint of each segment). These are referred as collocation points and are used to enforce the dynamics constraints. Most commonly, first-order polynomials are used for the control and third-order for the state. Defining the collocation points at the midpoints of the spline allows the practitioner to compute the state and control values at the collocation points without computing the spline coefficients (Hargraves and Paris, 1987).

Core to TO, and optimal motion generation schemes in general, is the model used to describe the behavior of the robot as a function of time. These models typically take the form of differential equations and are called dynamical systems. Next, we introduce the main features of hybrid manipulation and briefly sketch the underlying theory of modelling hybrid phenomena commonly observed during object manipulation.

3.1.3 *Hybrid manipulation preliminaries*

In manipulation, hybrid action trajectories guide the object from the current state to the goal. We illustrate one such trajectory in Fig. 3.1, where the object pose $\mathbf{y}_t$, the end-effectors positions $\mathbf{c}^k$ and the contact force $\mathbf{f}^l$ of the left end-effector are visualized. Such trajectories have hybrid nature due to contact change. The elements we would like to highlight in Fig. 3.1 are: (i) critical transition instances exist within the trajectory, where *discontinuities* occur, e. g. the force at T_1 and T_3, (ii) according to these time-instances, the motion can be separated in modes—called *contact-invariant* modes, e. g. *in-contact* and *free-motion* mode, and (iii) the sequential arrangement of these modes defines the outline of the trajectory—which we refer to as *structure of the motion* and we denote with $\mathbf{H} \in \{0,1\}^{K\times N}$. In manipulation setups the *structure of the motion* specifies the arms contact sequence pattern, i. e. the order with which the arms change contacts.

3.1.4 *Modelling hybrid phenomena*

In classical robot motion planning, we often consider continuous motions, from a start configuration to the goal and thus, the investigated dynamical models are continuous. However, in manipulation all motions involve *contact events* (Flanagan, Bowman, and Johansson, 2006), which are treated as discrete mechanical events. Manipulation tasks are not the only ones that are hybrid in nature; others include: loco-

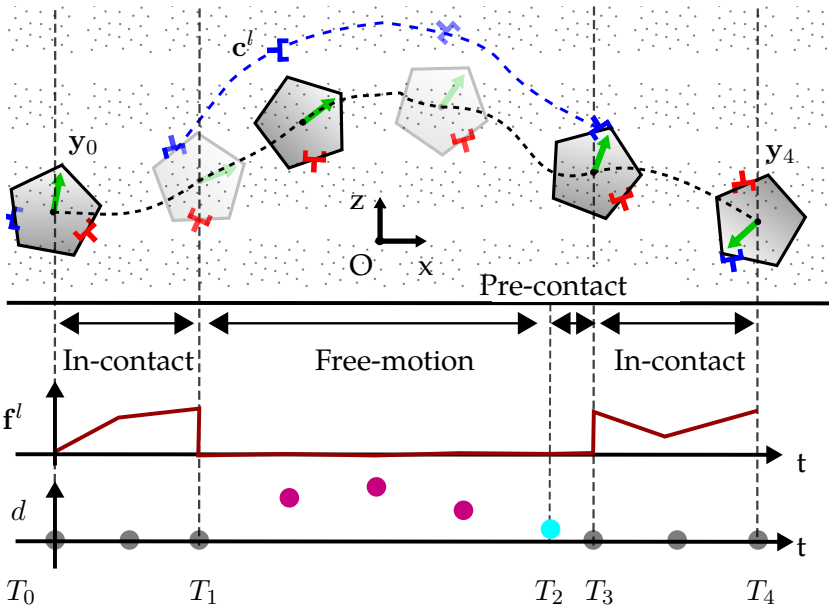


Figure 3.1: Hybrid motion plan with one grasp-hold change, separated into modes. The grey dotted area on top illustrates the physical space (x,z,ϕ). Orientation ϕ is illustrated with green arrow on the object. The force $\mathbf{f}^l$ applied by the left (blue) end-effector is shown in the middle plot. The knots of the trajectory with resolution 3 are shown in the bottom graph along with the contact distance d of the left end-effector from the object surface. The *in-contact* knots are shown in grey, the *free-motion* knots in pink, and the pre-contact knot in cyan. It is worth noting that all quantities shown here are optimized.

motion behaviors (G. Schultz and Mombaur, 2009), synchronization in groups of biological oscillators observed in swarms of fireflies (Buck, 1988) and neuronal oscillators (Hansel and Sompolinsky, 1992), switching electrical circuits (Gyugyi, 1979) and logic-based control algorithms for disk drivers (Goh et al., 2001). Thus, before we dive into the details on how to achieve hybrid manipulation planning, we would like to first review the definition of hybrid dynamical systems and hybrid control.

A general form used to represent hybrid systems is the following:

$$\dot{\boldsymbol{s}}(t) \in \boldsymbol{F}(\boldsymbol{s}(t)), \text{ if } \boldsymbol{s}(t) \in C, \tag{3.3}$$

$$\boldsymbol{s}^+ \in \boldsymbol{G}(\boldsymbol{s}), \text{ if } \boldsymbol{s} \in D, \tag{3.4}$$

where $\boldsymbol{F}(\cdot)$ and $\boldsymbol{G}(\cdot)$ are set-valued mappings[6], which can be understood as piecewise functions, e. g. $\boldsymbol{F}(\cdot)$ can be described as a set of $\boldsymbol{f}_k(\cdot)$ functions for $k \in \{0, 1, ...K\}$; $\boldsymbol{s} \in \mathbb{R}^n$ denotes a generalized notion of the state of the system, which may include both state and controls. C and D are subsets of $\mathbb{R}^n$ and define the domains of $\boldsymbol{F}(\cdot)$ and $\boldsymbol{G}(\cdot)$, respectively. $\dot{\boldsymbol{s}}(t)$ designates the derivative of the state and $\boldsymbol{s}^+$ denotes the con-

6 Set-valued mappings are functions that may associate several values to each input. When such functions are used in differential or difference form as in (3.3) and (3.4), they are referred to as differential and difference inclusion, respectively. Differential inclusion can be conceived as a generalization of differential equations.

secutive value of the state after an instantaneous change. Equation (3.3) indicates that the state of the hybrid system can evolve according to $\boldsymbol{F}(\cdot)$ when the state belongs to the set C. This simply describes the dynamics of the system: (i) under state constraints denoted by $\boldsymbol{s}(t) \in C$, and (ii) in cases where the behavior of the system depends on the state and belongs to a set rather than a single value or a single point in $\mathbb{R}^n$. Equation (3.4) suggests that the next immediate value of the state is given by $\boldsymbol{G}(\cdot)$ and depends on the current state when it belongs to the set D. Following the terminology used in (Goebel, Sanfelice, and Teel, 2009), we refer to (3.3) as the continuous-time dynamics of the hybrid system—also called flow map—and to (3.4) as the discrete-time dynamics of the hybrid system—also called jump map. To highlight the difference between continuous-time and discrete-time dynamics, we abuse the notation in (3.3) and (3.4), using $\boldsymbol{s}(t)$ and $\boldsymbol{s}$, respectively.

When considering a system, the hybrid phenomena can be grouped into two categories. First, hybrid behaviors that arise due to state related quantities, e. g. when the height of a bouncing ball is zero, the ball makes contact with the floor and will transit from falling to rising instantaneously. Second, hybrid behaviors that arise due to discontinuous controls used, e. g. on/off control of a room heater, i. e. thermostat; the evolution of the room temperature will change discontinuously when the heater switches from on to off and vice versa. These two categories are formally described below and are related to phenomena that occur in hybrid manipulation setups.

3.1.4.1 *Hybrid state dynamics*

As we mentioned at the beginning of this subsection, motion planning concepts for manipulation are often based on trajectories that guide an object to its desired state, with the most apparent characteristic being contact making and breaking. Trajectories of systems that evolve through contact events include discontinuous transitions between different contact states, such as *free-motion* or *in-contact*. Similar to (Toussaint, Allen, et al., 2018; Marcucci et al., 2017), we describe systems with switching contact using the hybrid system formulation as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_k\left(\boldsymbol{x}(t)\right), \text{ if } \boldsymbol{x}(t) \in \mathcal{C}_k, \tag{3.5}$$

$$\boldsymbol{x}^+ = \boldsymbol{g}_k\left(\boldsymbol{x}\right), \text{ if } \boldsymbol{x} \in \mathcal{D}_k, \tag{3.6}$$

where the notation is the same as in (3.1) in terms of variables, and as in (3.3) in terms of functions and their domain. Here, the generalized state $\boldsymbol{s}$ in (3.3) is replaced with the state of the system $\boldsymbol{x}$ and each k represents a distinct contact mode, i. e. *free-motion* or *in-contact*. In (3.5) each contact mode defines a specific manifold $\mathcal{C}_k \subset \mathbb{R}^\eta$ for the state of the system and corresponds to a specific differential equation $\boldsymbol{f}_k(\cdot)$. Similarly,

in (3.6) each contact mode defines a manifold $\mathcal{D}_k \subset \mathbb{R}^\eta$, where instantaneous jumps from *in-contact* to *free-motion* and vice-versa occur. Note that in robotics systems (3.5) defines both the dynamics of both the robot and the environment. Using this notation, the contact based hybrid dynamics of the system can be expressed in a compact form. In the case of manipulation, we use the above to describe the behavior of the system, i. e. robot, and the environment (object), when the two are *in-contact* and when they are apart (*free-motion*), as well as the transitions from one configuration to the other.

3.1.4.2 *Hybrid control*

Hybrid behavior in systems can also emerge due to the use of discontinuous controls, even if the system dynamics are continuous. These scenarios can arise in an number of setups, one such setup considers closed-loop systems that use variable structure control (J. Y. Hung, Gao, and J. C. Hung, 1993), e. g. hierarchies of local and global controllers (Antsaklis, Stiver, and Lemmon, 1993); another setup involves state-dependent reset controllers that can perform instantaneous control changes contingent to the state of the system (Beker et al., 2004). In robotics and nonprehensile manipulation[7] in particular, we are interested in planning and controlling the interaction forces (control) exerted on an object or on the environment from an end-effector at a contact point. Hybrid control can arise in cases of, (i) switching *in-contact* modes (F. R. Hogan, Grau, and Rodriguez, 2018), i. e. sticking and sliding[8], and (ii) hybrid force control (Ott, Mukherjee, and Nakamura, 2010), i. e. switching between impedance and admittance controllers. Inspired by (Goebel, Sanfelice, and Teel, 2009; Utkin, 2013; F. R. Hogan, Grau, and Rodriguez, 2018), we describe systems with hybrid control using the hybrid system formulation as

$$\dot{\boldsymbol{u}}(t) = \boldsymbol{f}_l\left(\boldsymbol{u}(t)\right), \text{ if } \boldsymbol{u}(t) \in \mathcal{C}_l, \tag{3.7}$$

$$\boldsymbol{u}^+ = \boldsymbol{g}_l\left(\boldsymbol{u}\right), \text{ if } \boldsymbol{u} \in \mathcal{D}_l, \tag{3.8}$$

which is the control equivalent of (3.5) and (3.6). $\boldsymbol{u}$ is the control and each l represents a distinct control mode, i. e. sticking or sliding or impedance or admittance. In (3.7) each control mode defines a specific manifold $\mathcal{C}_l \subset \mathbb{R}^\iota$ for the control of the system and corresponds to a specific differential equation $\boldsymbol{f}_l(\cdot)$. Similarly, in (3.8) each control mode defines a manifold $\mathcal{D}_l \subset \mathbb{R}^\iota$, where instantaneous jumps from between control modes occur. Note that (3.7) defines both the control of the robot and control

7 Nonprehensile manipulation is manipulation without grasping the object (definition was instigated by (Mason, 1999)).

8 Interaction forces depend on the *in-contact* mode. In sticking mode, the force lies within the friction cone, and in sliding mode, the force lies along one of the extreme rays of the friction cone (A. L. Ruina and Pratap, 2002).

input applied on the environment. Using this notation, the mode-dependent hybrid control of the system can be expressed in a compact form.

3.2 RELATED WORK

Next, we focus on the state-of-the-art multi-contact methods used for generating hybrid motions for manipulation and locomotion.

3.2.1 *Multi-contact planar manipulation*

Mason introduced the problem of planar non-prehensile manipulation, the *motion cone* concept, and the voting theorem (Mason, 1986). The *limit surface* concept was introduced in (Goyal, A. Ruina, and Papadopoulos, 1991) and used in (Lynch, Maekawa, and Tanie, 1992) to model the dynamics of planar pushing. These concepts map the motion of the contact point to the motion of the object, and have been used in (Zhou, Hou, and Mason, 2019; F. R. Hogan, Bauza, and Rodriguez, 2018; Zhou, Paolini, et al., 2017; F. R. Hogan, Grau, and Rodriguez, 2018) to address planning and control for planar pushing. In the work of (F. R. Hogan, Grau, and Rodriguez, 2018) an OC method was proposed to push objects to goal locations along a planar surface without making and breaking of contact. Recently, these geometrical constructs were generalized to a broader set of planar tasks (Chavan-Dafle, Holladay, and Rodriguez, 2018). Yet, the different contact modes are typically explored with offline sampling, and the quasi-static environment assumption limits their applicability to tabletop 2D pushing.

3.2.2 *Hybrid planning and control*

According to an important duality between manipulation and locomotion, the latter is an instantiation of non-prehensile manipulation (Mason, 2018). Our work is inspired by model-based TO methods (Winkler, Bellicoso, et al., 2018; Mordatch, Todorov, and Popović, 2012; Posa, Cantu, and Tedrake, 2014; Dai, A. Valenzuela, and Tedrake, 2014), which are not restricted by a quasi-static stability assumption. Next, we describe continuous methods for hybrid motion generation approaches.

CONTINUOUS PROGRAMMING These methods solve the problem using continuous optimization. For robot manipulation with contact changes, a number of TO methods (Toussaint, Allen, et al., 2018; Önol, Long, and Padır, 2019; Sleiman et al., 2019)

have been proposed. The underlying formulations have been borrowed from the locomotion domain (Posa, Cantu, and Tedrake, 2014; Mordatch, Todorov, and Popović, 2012; Winkler, Bellicoso, et al., 2018) and can be separated into two classes: *contact-implicit* (Posa, Cantu, and Tedrake, 2014; Sleiman et al., 2019; Patel et al., 2019) and *multi-phase* (Winkler, Bellicoso, et al., 2018; Stouraitis, Chatzinikolaidis, et al., 2018) or *multi-modal* (Toussaint, Allen, et al., 2018) approaches. The former is based on the work of Posa, Cantu, and Tedrake, 2014, where a mathematical problem with complementarity constraints was formulated in the presence of complex contact phenomena. These approaches require special attention in the relaxation of the problem to avoid spurious local minima (Nurkanovic, S. Albrecht, and Diehl, 2020). The latter includes the work of Mordatch, Todorov, and Popović, 2012; Winkler, Bellicoso, et al., 2018, where smooth nonlinear optimization problems were formulated based on a key observation: *motions through contacts have phases, while the contact set remains invariant within each phase*. This enables us to obtain a smooth *Non Linear Programming* (NLP) (Nurkanovic, S. Albrecht, and Diehl, 2020) given a mode sequence, which can be obtained from an outer-level process (Toussaint, Allen, et al., 2018). In our work, we adopt the latter paradigm as it admits a general notion of hybrid modes (Toussaint, Allen, et al., 2018), while the sequence of phases (or modes) can be also pre-defined.

3.3 MULTI-MODE PROBLEM FORMULATION

The problems addressed in this chapter are the generation of multi-contact trajectories for manipulation—by solving OC problems of the form Eq. (3.1)—and *"how can these trajectories consider a variety of different controllers and contact states?"*, as the selection of controllers and contact states can alter the behavior of the system. We refer to a single combination of a contact state and a controller as a mode of the system. The proposed notion for *contact-control* modes is similar to the notion of physical interaction modes introduced in (Toussaint, Allen, et al., 2018). In this work, two sets of modes are introduced: *contact mode* where contact states are constant and *stable mode* where the relative transformation between two objects is constant. Here, we only consider a limited number of contact states as physical interaction modes, but we extend the notion of mode by considering a variety of different controllers.

Contact-control modes form the *structure of the motion*.

The sequential arrangement of these modes $z_j = \{(k_j, l_j)\}$— also called *structure of the motion* (see Section 3.1.3)—defines the outline of the trajectory, while for each different sequence of *contact-control* modes $\boldsymbol{z} : \{z_0, z_1, ...z_J\}$ there is a different optimal solution of state ${}^*\boldsymbol{x}(t)$ and control ${}^*\boldsymbol{u}(t)$ trajectories (see Eq. (3.1)). $J \in \mathbb{Z}^+$ describes the total number of modes of the trajectory.

Given a mode sequence, we utilize the hybrid system models, mentioned in Section 3.1.4, to explicitly express multi-mode trajectories as a function of the initial state and the planned action sequence. Inspired by (Toussaint, Allen, et al., 2018; F. R. Hogan, Grau, and Rodriguez, 2018; Marcucci et al., 2017; Borrelli, Bemporad, and Morari, 2017), we think of manipulation planning as a special form of *Parametric Programming* (PP) (Narciso, Faísca, and Dua, 2011), where the sequence of modes $\boldsymbol{z}$ is encoded in the problem as

$$\min_{\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{v}(t)} \quad \boldsymbol{c}(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{v}(t),\boldsymbol{z}) \tag{3.9a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{v}(t),\boldsymbol{z}\right) \tag{3.9b}$$

$$\dot{\boldsymbol{v}}(t) = \tilde{\boldsymbol{h}}\left(\boldsymbol{u}(t),\boldsymbol{z}\right) \quad . \tag{3.9c}$$

$$\boldsymbol{g}(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{v}(t),\boldsymbol{z}) \leq 0 \tag{3.9d}$$

Equations (3.9a) - (3.9d) are piecewise functions from which the appropriate piece (interval) can be selected based on $\boldsymbol{z}$. Equation (3.9a) defines the objective function, (3.9b) and (3.9c) correspond to (3.5) and (3.7), respectively and $\boldsymbol{g}(\cdot)$ in (3.9d) represents both the equality and the inequality constraints of the system—and corresponds to (3.6) and (3.8) . It is worth pointing out that OC problems with hybrid dynamics are usually written as in (3.9), excluding (3.9c) , while OC problems with hybrid control are usually written as in (3.9), excluding (3.9b). The formulation above defines an OC problem where both dynamics and control can be hybrid. Next, we consider two instantiations of such a problem that are: (i) multi-contact object manipulation and (ii) impact-aware object manipulation.

3.4 MULTI-CONTACT OBJECT MANIPULATION

In Section 2.2.3, we referred to the full policy of an agent defined according to (2.5). Here, we aim to generate hybrid motion plans that belong in the force-contact space, thus, the output of the agent's policy can be simplified to

$$\xi = \begin{bmatrix} \mathbf{f}_i^{\kappa\,T} & \mathbf{c}_i^{\kappa\,T} & \boldsymbol{\Delta}\mathbf{T}_i{}^T \end{bmatrix}^T \quad \forall i \in \mathbb{N}, \tag{3.10}$$

where $\mathbf{f}_i^{\kappa}, \mathbf{c}_i^{\kappa}$ and $\boldsymbol{\Delta}\mathbf{T}_i$ correspond to the discrete form of the variables defined in (2.5). Here, we describe the details of the multi-mode TO formulation, which considers only different contact states, i. e. *contact-invariant* modes. This was developed to generate multi-contact manipulation plans, similar to the one shown in Fig. 3.2 and we will refer to it as hybrid TO.

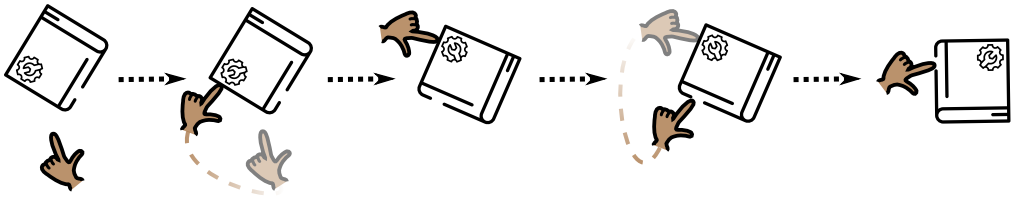


Figure 3.2: Illustration of a simple planar manipulation task that exploits multiple contact locations. First, the finger is not in contact with the book and the book is upside-down. Second, contact with the book is established. Third, the finger pushes the book. Forth, a contact change is performed to improve the relative position of the hand to the book. Finally, the book is pushed to the goal (upright) configuration.

3.4.1 *Hybrid Trajectory Optimization*

To generate multi-contact manipulation plans, a continuous optimization problem, in the form of Eq. (3.9), is formulated and solved. The formulation of the optimization problem entails the hybrid nature of the multi-contact manipulation setup. We perform direct transcription, as explained in Section 3.1.2, to obtain the discretized version of the problem, as in (3.2). This involves discretizing the trajectories of the following decision variables. For each i_{th} knot, the quantities of interest (see Section 2.2.3 and (3.10)) are: (i) the pose of the object $\mathbf{y}_i$, (ii) the velocity of the object $\dot{\mathbf{y}}_i \in \mathbb{R}^\nu$, (iii) action timings $\mathbf{\Delta T}_i$, (iv) the contact locations $\mathbf{c}_i^\kappa$, and (v) the contact forces $\mathbf{f}_i^\kappa$. We group [9] these quantities in two vectors

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{y}_i{}^T & \dot{\mathbf{y}}_i{}^T \end{bmatrix}^T \text{ and } \mathbf{u}_i = \begin{bmatrix} \mathbf{f}_i^\kappa{}^T & \mathbf{c}_i^\kappa{}^T & \dot{\mathbf{c}}_i^\kappa{}^T & \mathbf{\Delta T}_i{}^T \end{bmatrix}^T. \tag{3.11}$$

$\forall i \in \mathbb{N}$ the trajectories of $\mathbf{x}_i$ and $\mathbf{u}_i$ describe a hybrid motion (described in Section 3.1.3). TO problems with intermittent contacts can be expressed using complementarity constraints, yet in practice convergence of these problems is difficult (see Appendix B.1). In this chapter, the *structure of the motion* is fixed and is specified through variable $\boldsymbol{z}$, which allows us to customize our transcription (see Section 3.3), and separate the motion in modes with different constraints—the *contact-invariant* modes mentioned in Section 3.1.3. Next, we present the mode-free and the mode-conditioned constraints.

3.4.1.1 ***Mode-free constraints***

We introduce here constraints that are applied to all the knots of the trajectory regardless of the particular mode. In other words, constraints that are free of parameter set $\boldsymbol{z}$. We note that the object dynamics $\boldsymbol{f}_o(\cdot) \in \mathbb{R}^{2\nu}$ and the end-effectors motion $\boldsymbol{f}_e(\cdot) \in \mathbb{R}^\nu$ are integrated using trapezoidal quadrature. Also, $\psi_c \in \mathbb{R}^{2\nu}$ defines the reachable area of the agent's end-effectors, referred as arms workspace.

9 Regarding the grouping of the quantities; as we abstract away the robot kinematics we treat the contact $\mathbf{c}_i^\kappa$ as part of the robot actions, hence both $\mathbf{f}_i^\kappa$ and $\mathbf{c}_i^\kappa$ can be considered part of the control vector $\mathbf{u}_i$.

- Dynamics of the object (discussed in detail in Section 3.4.1.3)

$$\begin{bmatrix} \mathbf{y}_{i+1}{}^T & \dot{\mathbf{y}}_{i+1}{}^T \end{bmatrix}^T = \boldsymbol{f}_o(\mathbf{y}_i, \dot{\mathbf{y}}_i, \mathbf{f}_i^\kappa, \mathbf{c}_i^\kappa, \boldsymbol{\Delta}\mathbf{T}_i). \tag{3.12}$$

- Initial state of the object

$$\mathbf{y}_0 = \mathbf{y}_0^* \text{ and } \dot{\mathbf{y}}_0 = \dot{\mathbf{y}}_0^*. \tag{3.13}$$

- Desired final state of the object

$$\mathbf{y}_N = \mathbf{y}_N^* \text{ and } \dot{\mathbf{y}}_N = \dot{\mathbf{y}}_N^*. \tag{3.14}$$

- Kinematic limits of the agent's end-effectors

$$\mathbf{c}_i^\kappa \in \psi_c. \tag{3.15}$$

  We use box bounds to approximate them.

- Upper bound on the total time of the motion

$$\sum_{i=1}^{N} \boldsymbol{\Delta}\mathbf{T}_i \leq T. \tag{3.16}$$

3.4.1.2 *Mode-conditioned constraints*

The transcription of our hybrid TO problem follows the mode-based parameterization (Toussaint, Allen, et al., 2018), also called phase-based parameterization (see Fig. 3.1), introduced in (Mordatch, Todorov, and Popović, 2012) and used in (Winkler, Bellicoso, et al., 2018). We extend this by considering the three possible collision states between two rigid bodies as described in (Featherstone, 2014), and we split the knots in three sets according to the *contact-invariant* modes: the *in-contact*, *free-motion*, and *pre-contact* sets, as shown in Fig. 3.1. At each discretization point (knot) a constant subset of constraints needs to be satisfied, which is specified by the respective *contact-invariant* mode. Most of the mode-specific constraints are time independent, which allows us to optimize the duration of each mode and satisfy the constraints of each mode simultaneously. Each mode is characterized by a distinct set of decision variables that allows us to enforce a number of constraints implicitly and reduce the number of decision variables. A list of the constraints categorized according to the mode of the motion follows.

i) *Free-motion* mode ($k = 0$):

- End-effectors motion (the representation is described in Section 3.4.1.5):

$$\mathbf{c}_{i+1}^{\kappa} = \boldsymbol{f}_e(\mathbf{c}_i^{\kappa}, \dot{\mathbf{c}}_i^{\kappa}, \boldsymbol{\Delta}\mathbf{T}_i). \tag{3.17}$$

- End-effectors swing motion away from object:

$$d(\mathbf{c}_i^{\kappa}, \mathcal{S}_{obj}(\mathbf{y}_i, \mathbf{c}_i^{\kappa})) > 0. \tag{3.18}$$

- No force (*implicit constraint*):

$$\mathbf{f}_i^{\kappa} = 0. \tag{3.19}$$

ii) *In-Contact* mode ($k = 1$):

- Permissible contact forces (discussed in detail in Section 3.4.1.4):

$$\psi(\mathbf{f}_i^{\kappa}) \geq 0. \tag{3.20}$$

- No contact point slipping (*implicit constraint*):

$$\dot{\mathbf{c}}_i^{\kappa} = 0. \tag{3.21}$$

- End-effectors in contact with the object (*implicit constraint*):

$$d(\mathbf{c}_i^{\kappa}, \mathcal{S}_{obj}(\mathbf{y}_i, \mathbf{c}_i^{\kappa})) = 0, \tag{3.22}$$

where $d(\cdot) \in \mathbb{R}$ is the signed distance between end-effector and object. $\mathcal{S}_{obj} : (\mathbf{y}, \mathbf{c}^{\kappa}) \rightarrow \mathbb{R}^{\nu}$ computes the closest point on the object surface to the end-effectors location and emphasizes the importance of object shape representation described below.

iii) *Pre-contact* mode ($k = 2$):

- End-effectors touching the object:

$$d(\mathbf{c}_i^{\kappa}, \mathcal{S}_{obj}(\mathbf{y}_i, \mathbf{c}_i^{\kappa})) = 0. \tag{3.23}$$

- No force (*implicit constraint*):

$$\mathbf{f}_i^{\kappa} = 0. \tag{3.24}$$

These three contact modes correspond to the three potential states of two rigid bodies (Featherstone, 2014). The *free-motion* mode corresponds to the state where no constraints exist on the velocities or accelerations of the bodies, and the contact force is zero. The *in-contact* mode corresponds to the prolonged contact state where bodies are in contact, and their separation velocity is zero, so they will remain in contact. The *pre-contact* mode corresponds to the state at the moment of collision. This state should cause an impulsive force between the two bodies immediately after. The first two are essential to model hybrid motions, while accurate modelling of contact transitions motivates the use of the third.

3.4.1.3 *Dynamics of the object*

The dynamics of the object are described by

$$\begin{bmatrix} M\boldsymbol{I} & 0 \\ 0 & \boldsymbol{J} \end{bmatrix} \ddot{\mathbf{y}}_i + \begin{bmatrix} M\boldsymbol{g} \\ \dot{\mathbf{y}}_i^{\omega} \times (\boldsymbol{J}\dot{\mathbf{y}}_i^{\omega}) \end{bmatrix} = \sum_{\kappa}^{K} \begin{bmatrix} \boldsymbol{I} \\ \hat{\mathbf{c}}_i^{\kappa} \end{bmatrix} \mathbf{f}_i^{\kappa}, \tag{3.25}$$

where $M \in \mathbb{R}$ and $\boldsymbol{J} \in \mathbb{R}^{\nu \times \nu}_{\geq 0}$ are the mass and inertia of the object, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{g}$ is the acceleration due to gravity, $\dot{\mathbf{y}}_i^{\omega}$ is the angular velocity of the object, and we refer to the cross product matrix formed by the input vector with $\hat{(\cdot)}$.

3.4.1.4 *Permissible contact forces*

With (3.20) we denote the allowable contact forces exerted by the end-effectors to the object. These forces should satisfy the constraints[10]

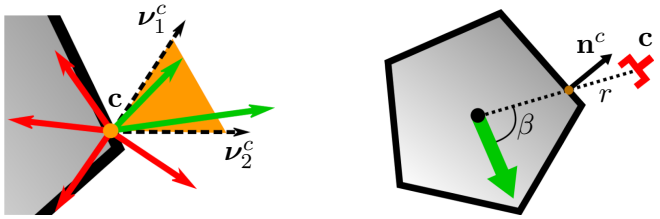
$$\mathbf{f}^T \mathbf{n}^c \geq 0, \tag{3.26a}$$

$$|\mathbf{f}^T \mathbf{t}^c| \leq \mu \mathbf{f}^T \mathbf{n}^c, \tag{3.26b}$$

where $\mathbf{f}$ is the force vector, $\mathbf{n}^c \in \mathbb{R}^{\nu}$ is the normal, $\mathbf{t}^c \in \mathbb{R}^{\nu}$ is the tangent vector at the contact point on the object surface, and $\mu \in \mathbb{R}$ is the friction coefficient. Here, (3.26a) is the unilateral contact constraint and (3.26b) is the friction cone constraint[11]. In (3.26) the constraints are denoted using the halfspace representation. Alternatively, the force constraints can be enforced using the vertex representation, also used in (Chatzinikolaidis et al., 2018)

$$\mathbf{f} = \sum_{\ell=1}^{R} \alpha_{\ell} \boldsymbol{\nu}_{\ell}^c, \tag{3.27}$$

10 For readability, we drop the indices with respect to end-effectors and knots of the trajectory.
11 In 3D, the linearized friction cone form is used.



(a) (b)

Figure 3.3: (a) Illustration of an end-effector in contact with the object. Valid 2D contact forces (shown in green) are generated by the conical combination of the rays $\boldsymbol{\nu}_1^c$ and $\boldsymbol{\nu}_2^c$. This form can be preferred for interior-point methods that traverse the interior of the feasible region and avoid unnecessary considerations of invalid contact forces (shown in red). (b) The position of the end-effector described in 2D polar coordinates $\mathbf{c} = [\beta, r]^T$, along with normal vector $\mathbf{n}^c$ at the imminent contact location are depicted.

where $\boldsymbol{\nu}_\ell^c = \mathbf{n}_\ell^c + \mu \mathbf{t}_\ell^c$ are the extreme rays of the friction cone, $\alpha_\ell \geq 0$ are weighting coefficients and $R \in \mathbb{R}$ is the number of rays used. Normals, tangents, and extreme rays are functions of the contact location and are obtained from $\mathcal{S}_{obj}(\cdot)$ according to the object shape representation. A 2D graphical illustration and intuitive comparison between the halfspace and the vertex forms is given in Fig. 3.3a. We choose to enforce constraint (3.20) with (3.27) as we have empirically noticed faster convergence.

3.4.1.5 *End-effectors position representation*

Equations (3.15), (3.17), (3.22), (3.18), (3.23), and (3.27) are realized given a specific representation of the end-effectors position $\mathbf{c}_i^\kappa \in \mathbb{R}^\nu$. Here, we choose to represent the end-effectors position relative to the object coordinate frame that is placed at the *Center of Mass* (CoM) of the object, graphically shown for the 2D case in Fig. 3.3b. We have experimented with both Cartesian coordinates as well as polar coordinates. We decided to use the latter as we empirically found that the optimizer convergence rate is improved in most cases.

3.4.1.6 *Object shape representation*

The surface of the object is represented with a closed cubic spline curve. The spline representation is a smooth description of the object surface from which all relevant properties along with their gradients can be extracted, like normal and tangent vectors. The use of continuous representations of the object surface is more generic than approaches like (Pajarinen et al., 2017) that rely on the convexification of the object shape, as scaling with respect to the number of edges/phases becomes cumbersome.

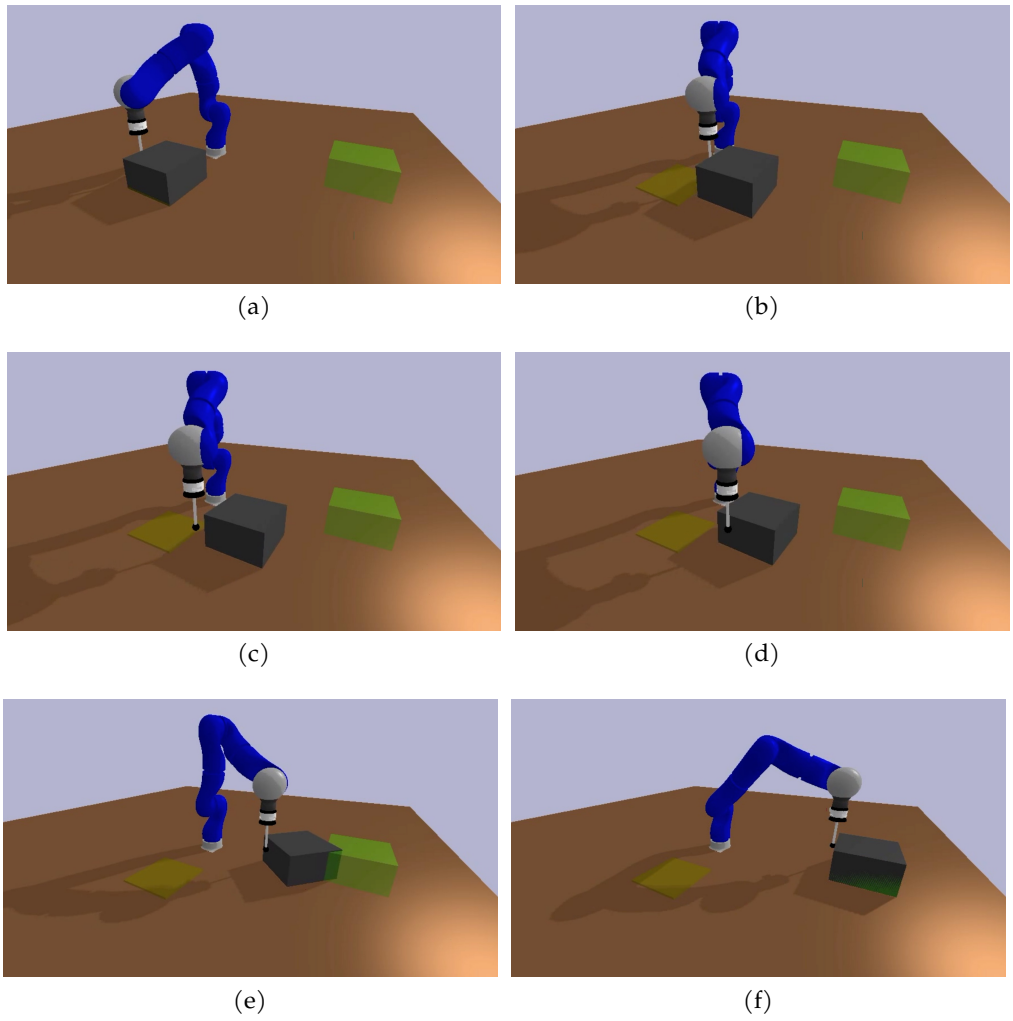


Figure 3.4: In these keyframes a single arm robot agent performs a pushing task that includes a change of contact, similar to Fig. 3.2. The yellow patch on the table denotes the initial location of the grey box and the transparent green box the goal location. The change of contact starts in (b), continues in (c) and finishes in (d).

3.4.1.7 *Input variables and hyper-parameters*

The only required input variable is the description of the manipulation task, $\theta^{\mathcal{M}}$. Here we use the start and goal state of the object, denoted as $\begin{bmatrix} \mathbf{y}_0^{*\,T} & \dot{\mathbf{y}}_0^{*\,T} \end{bmatrix}^T$ and $\begin{bmatrix} \mathbf{y}_N^{*\,T} & \dot{\mathbf{y}}_N^{*\,T} \end{bmatrix}^T$, respectively. Nonetheless, the specification of $\theta^{\mathcal{M}}$ can be as flexible as needed, and can be specified either via the cost function $c(\cdot)$, e. g. minimize acceleration of the object, or via the constraints $\boldsymbol{g}(\cdot)$, e. g. set upper velocity limits for the object or set forbidden regions of the workspace. To solve the TO problem two hyper-parameters need to be specified: (i) the resolution of the grid N (number of knots) shown in Fig. 3.1 and (ii) upper bound for the time of the motion T.

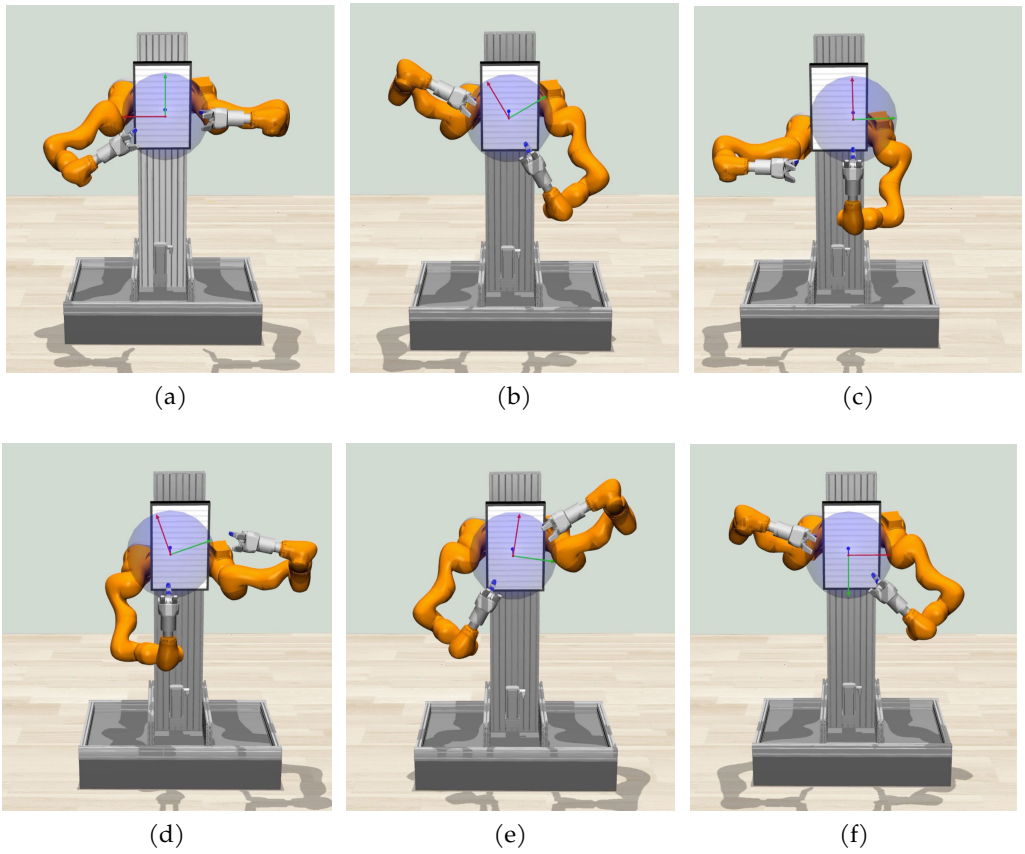


Figure 3.5: In these keyframes a single robot agent performs a multi-contact manipulation task, i. e. rotate a ball by changing twice the grasp-hold locations.

3.4.2 *Evaluations and simulations*

To validate the proposed method we perform a number of simulations both for a single-arm robot and for a bimanual robot. First, we present a number of different motion plans generated by the proposed method that demonstrate the capability to find multi-contact solutions. Second, to validate the physical correctness of the obtained motion plans, we evaluate the accuracy of those solutions computed by the hybrid TO method. Third, we evaluate the scalability of the proposed method in terms of the number of contact changes. In all the evaluations and demonstrations presented here, the state of the object is $\mathbf{y} = \begin{bmatrix} x & z & \phi \end{bmatrix}^T$ and the task dimension is $\nu = 2$. Similar to Fig. 3.2, a multi-contact motion plan is shown in Fig. 3.4, where a single arm robot performs a planar pushing task of approximately $0.7m$ translation and -90° rotation. Another multi-contact motion plan is shown in Fig. 3.5, where a bimanual robot performs in simulation the task of rotating a ball 180°. A more dynamic motion plan is illustrated in Fig. 3.6, where the challenging 90° rotation task that considers throwing and catching a ball is demonstrated. The main difference between the latter

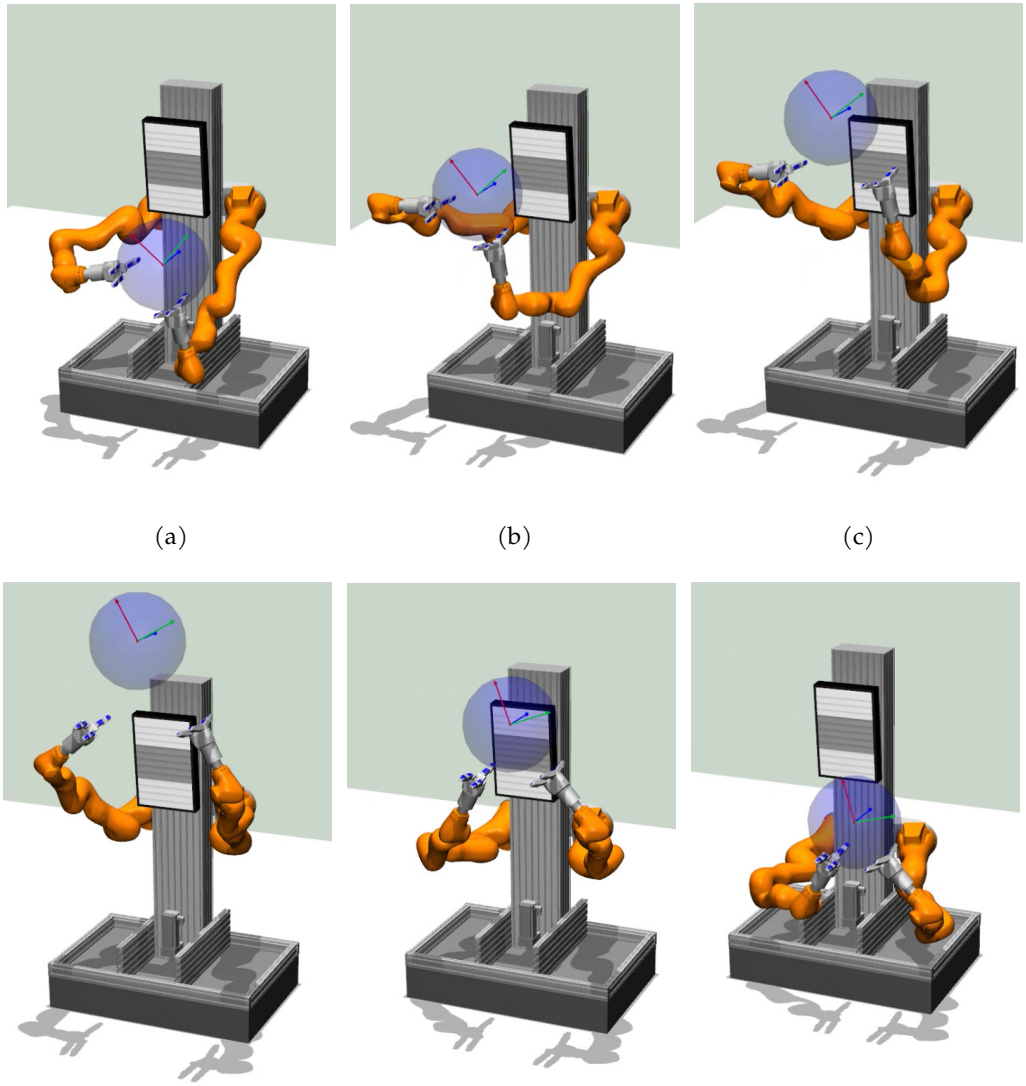


Figure 3.6: In these keyframes a single robot agent performs a dynamic hybrid manipulation task, i. e. rotate a ball by throwing it up in the air and catching it.

two motion plans lies in the pre-specified *structure of the motion*. In the former, the contact breaking and making events of the end-effectors happen in a sequence—first the right and then the left end-effector. In the latter, the contact making and breaking events of the end-effectors happen simultaneously, hence the solver finds a solution that involves throwing and catching the ball.

Regarding the implementation details, we use `CasADi` (Andersson et al., 2018) to realize the hybrid TO method. Each hybrid problem is a large and sparse nonlinear optimization problem which is solved using `IPOPT` (Wächter and Biegler, 2006), while the automatic differentiation capabilities of `CasADi` allow us to provide exact gradient and Hessian information. Also, no special care is taken regarding the initialization

Res	X(m)	Y(m)	ϕ(deg)	Time(s) per iter
2	0.034	0.125	8.42	0.0025
4	0.027	0.013	0.92	0.0047
8	0.023	0.013	1.32	0.0087

Table 3.1: RMSE values against three different mesh resolutions, for $0.5m$ translation and $90°$ rotation task including a single contact change per arm.

of the optimizer. The lower-level control aspects of the robot, e. g. *Inverse Kinematics* (IK), are implemented in the `Robot Control Software (Rcs)` framework[12] or in the `Bullet physics Engine`[13]. All experiments are conducted on a 64-bit Intel Quad-Core i7 3.40 GHz workstation with 16GB RAM. A video showing the simulations and experiments performed is provided (see SV1 in Appendix E).

Next, we consider the integration accuracy and the scalability of the method in a prototypical task of rotating a box $90°$ and translating it $0.5m$ diagonally.

Evaluation of solution accuracy: We compare the trajectory of the object generated by the proposed method and the trajectory generated by a Simulink-based dynamic simulation after feed-forward streaming of the planned forces onto it. We treat the dynamic simulation as ground truth and we compute the root mean square error (RMSE). Table 3.1 shows the RMSE between the planned and the ground-truth trajectories along each dimension of the motion, as well as the required computation time per iteration of the optimizer. For a resolution of two, the simulated trajectory diverges from the planned, revealing the need for higher resolution. However, by comparing resolutions four and eight, we can observe that it is not always the case that the higher the resolution the better the accuracy, although the computation time per iteration increases. Thus, sensible resolution must be chosen for each setup.

# Contacts	Vars	Time(s) per iter
1	510	0.0047
2	760	0.0066
3	1010	0.0091
4	1260	0.0117

Table 3.2: Number of decision variables and computation times per iteration with respect to the number of contact changes per arm. The task is 0.5 m translation and $90°$ rotation.

12 Information about `Rcs` can be found in: https://github.com/HRI-EU/Rcs
13 Information about `Bullet` can be found at: https://pybullet.org/

Evaluation of the method scalability: Toward evaluating the scalability of the proposed method with respect to the number of contact changes, the same task is solved with one, two, three and four contact changes per arm. The resolution used is four as it was the most prominent for the task. Table 3.2 provides the respective quantities of interest. Computation times scale linearly as the number of contact changes increases, however large number of unnecessary contact changes hinders the performance of the method.

3.5 IMPACT-AWARE OBJECT MANIPULATION

In the previous section, we investigated a methodology for generating hybrid motion plans including both forces and contact location, as specified in (3.10). In this section we get a step closer to the full policy of an agent, as defined in (2.5). Here, in addition to the hybrid motion plans, we also consider the compliance characteristics of the end-effector, thus, the output of the agent's policy can be written as

$$\xi = \begin{bmatrix} \mathbf{f}_i^{\kappa\,T} & \mathbf{c}_i^{\kappa\,T} & \mathbf{\Delta T}_i^{T} & \mathbf{K}_i^{a\,T} & \mathbf{D}_i^{a\,T} \end{bmatrix}^T \quad \forall i \in \mathbb{N}. \tag{3.28}$$

Next, we describe the details of the multi-mode TO formulation (see (3.9)), which considers both different contact states and controllers, i. e. a set of *contact-control* modes. As shown in Fig. 3.7, with multi-mode TO we try to address the problem of establishing contact with an object at speed, i. e. moving objects, at the level of 'impact-aware' manipulation planning. We ask ourselves: *it "How could we plan hybrid motions, such that they are easily tractable by out-of-the-box controllers?"*, which can be re-framed as a planning problem such that contact can be maintained during and after impact.

To realize contact at speed, a core insight that we exploit is the duality between the impact model used and the capabilities of compliance controllers available in nowadays collaborative robots (cobots). As it will be described in the following sections, by modulating the robot end-effector compliance, we can emulate a number of different types of collisions ranging from elastic to in-elastic, and deduce the optimal force transmission model given the limitations of the system, e. g. workspace limits.

The prime motivation for tasks which involve making contact at speed arises from human-human co-manipulation scenarios. Consider situations where the manipulated object is out-of-balance and is about to fall. In such a scenario, a reactive and very fast change of contact is performed, by at least one of the individuals, to stop the falling object and restore it back in-balance. Before we lay out the technical details of

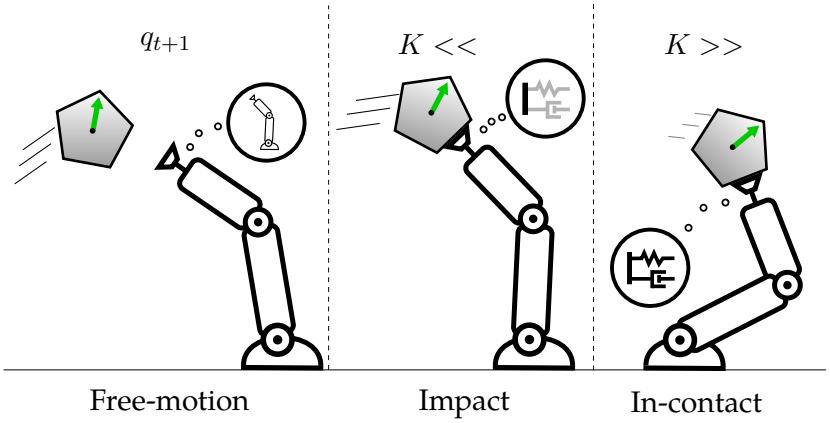


Figure 3.7: Pictorial description of the multi-mode TO for the task of halting a moving object.

the proposed method, we provide here a brief and focused review on relevant works, which further motivates the investigated setup[14].

3.5.1 *The significance of impact*

Recent hybrid TO methods in robotics (Toussaint, Allen, et al., 2018; F. R. Hogan, Grau, and Rodriguez, 2018; Stouraitis, Chatzinikolaidis, et al., 2018) have demonstrated efficient methods for multi-contact manipulation planning. Yet, it is not trivial to transfer these behaviors robustly onto the hardware due to the challenge of regulating the transitions between *free-motion* and motion *in-contact*, as well as dealing with imprecise timing of the transition in the reference motions.

To address this, a number of hybrid control (Rijnen, Saccon, and Nijmeijer, 2015; Rijnen, Mooij, et al., 2017) and compliance control (Roveda et al., 2015; E. Lee et al., 2003) methods have been proposed. However, given the inherent limitations of the hardware (Haddadin, Albu-Schäffer, and Hirzinger, 2009), the impacts that a stand-alone controller is capable of dealing with, are limited. On the planning side, TO methods for multi-contact manipulation planning transcribe the state and the control input of the system, which entails that forces are optimization variables too. Although Patel et al., 2019 pointed out the importance of an accurate force model during contact transitions, most previous works neglected this aspect of the problem. As a result, a number of assumptions were made to transfer the motion plans to the robots. In (Sleiman et al., 2019) purely inelastic collision was assumed to impose no-rebound condition, while in (Onol, Long, and Padlr, 2018; Önol, Long, and Padır, 2019) a variable smooth contact model was used, which allows virtual forces to be spawned from distance.

14 A video presentation of this piece of work is provided (see SV5 in Appendix E).

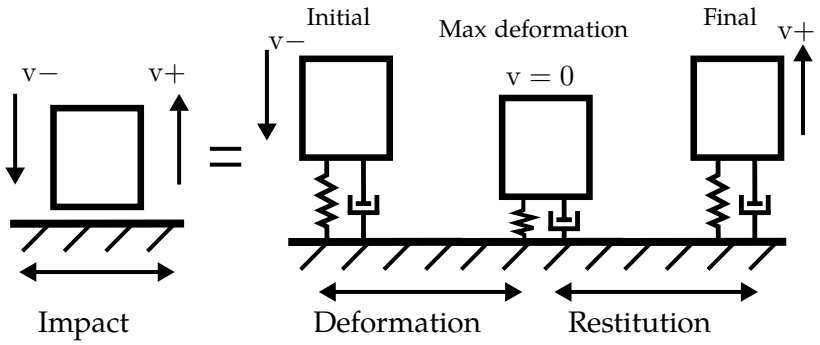


Figure 3.8: Correspondence between Newton's restitution model and the mass-spring-damper system.

Thus, a natural design question arises regarding the choice of the contact force transmission model. Such a model can be used to constrain the control inputs and could also be conditioned on the current *contact-control* mode of the system. For a task like the one shown in Fig. 3.7 the robot has to be initially soft to absorb the impact and then stiff to manipulate the object. Thus, next we provide a description of an impact model.

3.5.2 *An impact model*

In scenarios where two objects collide (contact transition) with non-zero relative velocity, an impulsive force is caused. The velocity discontinuity between pre-impact and post-impact is described with the following relationship

$$M\left(\mathrm{v}^{+}-\mathrm{v}^{-}\right)=\Lambda\delta t, \tag{3.29}$$

where M is the mass of the system, v^{-} and v^{+} are the pre-impact and post-impact relative velocities, respectively. Λ is the impact force and $\delta t \simeq 0$ is the impact duration.

For a moving object that experiences an impact, the Newton's coefficient of restitution c_r is calculated by

$$\epsilon_r=\left|\frac{v_{+}}{v_{-}}\right|=\sqrt{\frac{\frac{1}{2}M(\mathrm{v}^{+})^2}{\frac{1}{2}M(\mathrm{v}^{-})^2}}, \tag{3.30}$$

where $\epsilon_r = 1$ represents a perfectly elastic collision, and $0 \leq \epsilon_r \leq 1$ represents a real-world inelastic collision. $\epsilon_r = 0$ represents a perfectly inelastic collision, where all the kinetic energy is converted to deformation, and is often used to impose a no-rebound condition. The dissipated energy—due to the impulsive force—during the collision is

$$E_{\Lambda}=\frac{1}{2}M(\mathrm{v}^{-})^2-\frac{1}{2}M(\mathrm{v}^{+})^2. \tag{3.31}$$

Here, we adopt the mass-spring-damper system to model real-world collisions proposed by Nagurka and S. Huang, 2004. The equation of motion for a mass-spring-damper system shown in Fig. 3.8 is written as

$$M\ddot{x} + D\dot{x} + Kx = -Mg, \tag{3.32}$$

where K, D, M are the stiffness, damping and mass respectively; g is gravity and x is the state of the system. The energy dissipation of such a system is caused by the damper and can be calculated as follows

$$E_p = \int_0^{\delta t} D\dot{x}dx = \int_0^{\delta t} D\dot{x}^2 dt. \tag{3.33}$$

Thus, by equating (3.31) with (3.33), we can model the energy loss during impact with a spring-damper system (Nagurka and S. Huang, 2004), where the dissipated energy during deformation and restitution stages (see Fig. 3.8) is related to both stiffness and damping.

In addition, based on (Nagurka and S. Huang, 2004; Zhu, Zwiebel, and Bernhardt, 1999), the characteristics of the physical system, such as duration of impact and restitution coefficient, can be related to the mass, damping and stiffness parameters of the mechanical system. We utilize this observation to accurately emulate the physical interaction through the impedance controller of the manipulator.

As shown in Fig. 3.8, the negative contact is defined as the deformation stage during which the contact force for making a stable contact is generated. The positive contact is defined as the restitution stage that generates the contact force for the manipulation tasks, such as pushing an object far away. We encode these two stages of the contact as

$$l = \begin{cases} -1, & \mathrm{v}^- \to 0 \\ 1, & 0 \to \mathrm{v}^+ \end{cases}. \tag{3.34}$$

In terms of impact-aware manipulation, the stiffness should be minimized during negative contact, while it should be maximized during positive contact to achieve accurate manipulation. These contact stages are encoded in (3.9) in the form of controllers according to (3.7) and (3.8). In this way, the controller parameters (stiffness) can be optimized to conform with the different stages of the contact.

3.5.3 *A contact force transmission model*

For a smooth transition from *free-motion* to contact, the contact duration and contact force profile should obey the impact model shown in Fig. 3.8. Thus, (3.32) becomes

$$M\Delta\ddot{c} + D\Delta\dot{c} + K\Delta c = \mathrm{f}_d, \tag{3.35}$$

where f_d is the desired contact force, Δc, $\Delta\dot{c}$, and $\Delta\ddot{c}$ are the deformed position, velocity and acceleration, respectively.

In order to plan smooth contact force without oscillations, we model the force transmission as a second-order critically damped dynamical system (cd-DS). A cd-DS (D. Braun, Howard, and Vijayakumar, 2012; D. J. Braun et al., 2013) was first used to guarantee that the motor position is tractable, and it was further used to provide constraint consistent output for any admissible input. Here, we formulate a cd-DS for contact force transmission as

$$\ddot{\mathrm{f}}(t) + 2\alpha\dot{\mathrm{f}}(t) + \alpha^2\mathrm{f}(t) = \alpha^2\mathrm{f}_d, \tag{3.36}$$

where the contact force $\mathrm{f}(t)$ satisfies $\mathrm{f}(t) \in [0, \mathrm{f}_d]$, while $\dot{\mathrm{f}}(t)$ and $\ddot{\mathrm{f}}(t)$ are its first and second derivatives. For any $\alpha > 0$, the contact force $\mathrm{f}(t)$ is critically damped.

Additionally, we enforce the second-order contact force transmission model (3.36) to have the same characteristics as the impact model (3.35), i.e. the same natural frequency ω_n and damping ratio ζ. Given that $K\Delta\boldsymbol{c} = \mathrm{f}, \Delta\dot{c} \approx 0$ and $\Delta\ddot{c} \approx 0$ at the maximum deformation point of (3.35) and the Laplace transform of (3.35) and (3.36),

$$\Delta c = \frac{1}{Ms^2 + Ds + K}\mathrm{f}_d = \frac{\frac{K}{M}}{s^2 + \frac{D}{M}s + \frac{K}{M}}\frac{\mathrm{f}_d}{K}, \tag{3.37}$$

$$\mathrm{f}(t) = \frac{\alpha^2}{s^2 + 2\alpha s + \alpha^2}\mathrm{f}_d, \tag{3.38}$$

we can obtain the following relationship between parameter α and the parameters of the impact model (3.35)

$$\alpha = \sqrt{\frac{K}{M}}, \tag{3.39}$$

$$D = 2\sqrt{MK}. \tag{3.40}$$

Thus, given the mass of the object and the stiffness parameter we can obtain the cd-DS parameter α. Further, for the differential equation (3.36) with output $\mathrm{f}(t)$, input f_d and damping factor $\zeta = 1$, we can obtain the relationship between α and settling time t_s (within 5%, $t_s \approx \frac{3.0}{\omega_n\zeta} = \frac{3.0}{\alpha}$). This reveals that α is coupled to the contact duration. Also, the feasible contact duration is related to the velocity of the object

and the workspace of the robot, thus, α should be limited according to the attributes of the physical system. In multi-mode TO (see next section), both α—i. e. stiffness—and contact duration are optimized to satisfy the workspace limits of the robot.

Simultaneously, the mass-spring-damper system that establishes the relationship between force and position is also adopted in the impedance control to regulate the operational force of the manipulator. To achieve fast tracking performance without oscillation with such a controller, the mass, damping and stiffness parameters of (3.35) need to form a second-order critically damped system. Thus, based on the optimized parameter α, the optimal inertia, damping and stiffness of the impedance controller are obtained from (3.39) and (3.40), which satisfy the critically damped constraint.

3.5.4 *Multi-mode Trajectory Optimization*

Similar to Section 3.4.1, we solve the continuous optimization problem in (3.9) by discretizing the trajectory according to direct transcription (Rawlings, Mayne, and Diehl, 2017). The transcription of the parametric optimization problem considers for each knot i the following decision variables: (i) the pose of the object $\mathbf{y}_i$, (ii) the velocity of the object $\dot{\mathbf{y}}_i \in \mathbb{R}^\nu$, (iii) action timings $\Delta\mathbf{T}_i$, (iv) the end-effector position $\mathbf{c}_i$, (v) the contact force $\mathbf{f}_i$ and the cd-DS parameter $\boldsymbol{\alpha}$. We group [15] these quantities in three vectors

$$\boldsymbol{x}_i = \begin{bmatrix} \mathbf{y}_i^T & \dot{\mathbf{y}}_i^T & \mathbf{c}_i^T & \dot{\mathbf{c}}_i^T & \ddot{\mathbf{c}}_i^T \end{bmatrix}^T, \tag{3.41a}$$

$$\boldsymbol{v}_i = \begin{bmatrix} \mathbf{f}_i^T & \dot{\mathbf{f}}_i^T & \ddot{\mathbf{f}}_i^T \end{bmatrix}^T, \tag{3.41b}$$

$$\boldsymbol{u}_i = \begin{bmatrix} \boldsymbol{\alpha}_i^T & \Delta\mathbf{T}_i^T \end{bmatrix}^T, \tag{3.41c}$$

$\forall i \in \mathbb{N}$ the trajectories of $\boldsymbol{x}_i$, $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ describe a multi-mode motion. In addition to the decision variables, through the mode sequence $\boldsymbol{z}$ the transcription of the continuous problem can be customized (see Section 3.3). This results in a TO problem that is separated into modes with different constraints, as in Section 3.4.1. It is worth pointing out that the cd-DS parameter $\boldsymbol{\alpha}$ in (3.41c) is related to the compliance characteristics $\mathbf{K}_i^a$ and $\mathbf{D}_i^a$ mentioned in (3.28) according to (3.39) and (3.40). Also, (3.36) specifies the transformation from the control actions of the robot (see (3.41c)) to the control inputs of the environment (see (3.41b)).

15 This grouping of the quantities is selected in order to emphasize both the contact and the control modes of the problem. We treat $\mathbf{c}_i$ as part of the state vector $\boldsymbol{x}_i$ to model on and off contact states, while we consider $\boldsymbol{\alpha}_i$ part of the control vector $\boldsymbol{u}_i$ to model soft and stiff control behavior.

Next, we present only the mode-conditioned constraints, as the mode-free constraints are the same with the ones presented in Section 3.4.1 excluding the dynamics constraint (3.12), which is reformulated here.

Mode-conditioned constraints

Here, we organize the description of the constraints into time-dependent and time-independent constraints, respectively.

TIME-DEPENDENT: Equation (3.9b) describes the dynamics of the system, which includes both the dynamics of the object and the motion of the end-effector. We note that the object dynamics $\boldsymbol{f_o}(\cdot) \in \mathbb{R}^{2\nu}$ and the end-effector motion $\boldsymbol{f_e}(\cdot) \in \mathbb{R}^{2\nu}$ are integrated using trapezoidal quadrature. Additionally, (3.9b) depends on the current mode and therefore is written in a piecewise form.

OBJECT DYNAMICS: As discussed in Section 3.1.4.1, for $k = 0$ the object is in *free-motion*, whereas for $k = 1$ the object is *in-contact*. According to the above, the dynamics of the object are described by

$$\begin{bmatrix} \mathbf{y}_{i+1} \\ \dot{\mathbf{y}}_{i+1} \end{bmatrix} = \begin{cases} \boldsymbol{f}_o(\mathbf{y}_i, \dot{\mathbf{y}}_i, \mathbf{c}_i, \Delta\mathbf{T}_i), & \text{if } k_i = 0, \\ \boldsymbol{f}_o(\mathbf{y}_i, \dot{\mathbf{y}}_i, \mathbf{c}_i, \Delta\mathbf{T}_i, \mathbf{f}_i), & \text{if } k_i = 1, \end{cases} , \tag{3.42}$$

while (3.42) for $k = 1$ is given in more detail in (3.25). The hybrid nature of the system dynamics is evident from (3.25) (see Section 3.4.1.3). For $k = 1$, the RHS of (3.25) remains, while for $k = 0$ the RHS of (3.25) disappears.

END-EFFECTOR MOTION: When planning motions with impacts, particular care needs to be taken while enforcing the integration constraints of the motion (Goebel, Sanfelice, and Teel, 2009). The motion of the end-effector is described with the following function

$$\begin{bmatrix} \mathbf{c}_{i+1} \\ \dot{\mathbf{c}}_{i+1} \end{bmatrix} = \begin{cases} f_e(\mathbf{c}_i, \dot{\mathbf{c}}_i, \ddot{\mathbf{c}}_i, \Delta\mathbf{T}_i), \text{ if } k_i = 0 \text{ or } 1, \\ f_e(\mathbf{c}_i, \dot{\mathbf{c}}_i, \Delta\mathbf{T}_i), \text{ if } k_i = 0,\ k_{i+1} = 1, \\ f_e(\mathbf{c}_i, \dot{\mathbf{c}}_i, \Delta\mathbf{T}_i), \text{ if } k_i = 1,\ k_{i+1} = 0, \end{cases} , \tag{3.43}$$

where time integration from accelerations to velocities needs to be skipped at specific mode transitions, similar to (Rijnen, Mooij, et al., 2017). These transitions are the making and the breaking of contact, which are liable to impact and are noted by $k_i = 0, k_{i+1} = 1$ and $k_i = 1, k_{i+1} = 0$, respectively. By omitting $\ddot{\mathbf{c}}$ at these transitions ve-

locity jumps are possible. Hence, the space of solutions of the mathematical program includes state jumps, typically described with jump maps (Goebel, Sanfelice, and Teel, 2009). Note that we need to omit the time integration of the robot, but not that of the object. The forces exerted by the robot in (3.42) are unbounded (can trigger velocity jumps), while the accelerations of the robot arm need to be bounded—which constrains velocity changes (curb velocity jumps)—according to the capabilities of the robot.

CONTACT FORCE TRANSMISSION: Based on the impact model and the contact force transmission model the forces of the TO are generated according to the following differential equation

$$\begin{bmatrix} \dot{\mathbf{f}}_\mathbf{i} \\ \ddot{\mathbf{f}}_\mathbf{i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\boldsymbol{\alpha}_l{}^2 & -2\boldsymbol{\alpha}_l \end{bmatrix} \begin{bmatrix} \mathbf{f}_\mathbf{i} \\ \dot{\mathbf{f}}_\mathbf{i} \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{\alpha}_l^2 \end{bmatrix} \mathbf{f}_d. \tag{3.44}$$

According to (3.44) the contact force transmission model is parameterized by only one parameter $\boldsymbol{\alpha}_l$. Hybrid control (see Section 3.1.4.2) is realized with two controllers (i) $l = -1$ and (ii) $l = 1$, one for each contact stage defined in (3.34). For each controller, $\boldsymbol{\alpha}_l$ is optimized to modulate the contact time and contact force profile. Furthermore, based on the relationship between α and K in (3.39), the stiffness characteristics are also optimized in a coherent way through $\boldsymbol{\alpha}_l$, without separating the contact scheduling from stiffness modulation into two levels, as in (Nakanishi, Radulescu, and Vijayakumar, 2013).

TIME-INDEPENDENT: These constraints are grouped with respect to the modes of the trajectory and they are a subset of the constraints presented in Section 3.4.1.

Free-motion mode ($k = 0$):

- End-effector away from object: $d(\mathbf{c}_i, \mathbf{y}_i) > 0$, where $d(\cdot) \in \mathbb{R}$ is the signed distance between the end-effector and the object.

In-contact mode ($k = 1$):

- Permissible contact forces: $\psi(\mathbf{f}_i) \geq 0$.
- End-effector at the contact point: $d(\mathbf{c}_i, g_{pt}) = 0$, where $g_{pt} \in \mathbb{R}^\nu$ is the defined desired contact point on the object surface.

In the next section we provide further design choices for this TO problem.

3.5.4.1 *Representations*

For this task, the end-effector position is represented relative to the world in Cartesian coordinates. The object surface is represented with a closed cubic spline as in Section 3.4.1.

3.5.5 *Evaluations and experiments*

In this section, we first perform a simulation ablation study on methods that could halt a moving object. We investigate whether standard compliance controllers can reduce the contact force by tuning their compliance parameters. We also study the effects of realizing the motion plans obtained by TO methods in physical simulation (see Fig. 3.9). Specifically, we compare an impact-agnostic method against the impact-aware method. Next, we report computational results in which contact force profiles are optimized to satisfy both workspace limits of the robot and the task of manipulating a large-momentum object. Last, we evaluate the proposed method with real-world experiments, as shown in Fig. 3.10 and compare against standard compliance control.

IMPLEMENTATION SETUP We use `CasADi` (Andersson et al., 2018) and its automatic differentiation capabilities to realize the multi-mode TO method. Motion planning is done in the task space and the motions are projected into the configuration space of the robot with IK. All simulations are conducted on a 64-bit Intel Quad-Core i9 3.60GHz computer with 64GB RAM and are realized with the Bullet physics simulation library. A video showing the simulations and experiments performed is provided (see SV2 in Appendix E).

SIMULATION SETUP The task is to halt a moving object, as shown in Fig. 3.9. The mass of the object is 20 kg and its velocity is $[0.0\ 0.65\ 0.0]^T\ m/s$. There is no sliding friction between the object and the table while the contact is rigid.

3.5.5.1 *Standard compliance control*

In the first ablation scenario, the manipulator attempts to halt the moving object only utilizing a compliant controller. The desired contact force is $0\ N$. This means that the robot will remain still until contact occurs and then, it will give in according to the selected compliance characteristics. We explore a set of different mass-spring-damper parameters, yet independently of the parameters—as there is no pre-contact motion by the robot, the speed of the object at impact is $\mathrm{v}^- = 0.65\ m/s$—the resulting impact

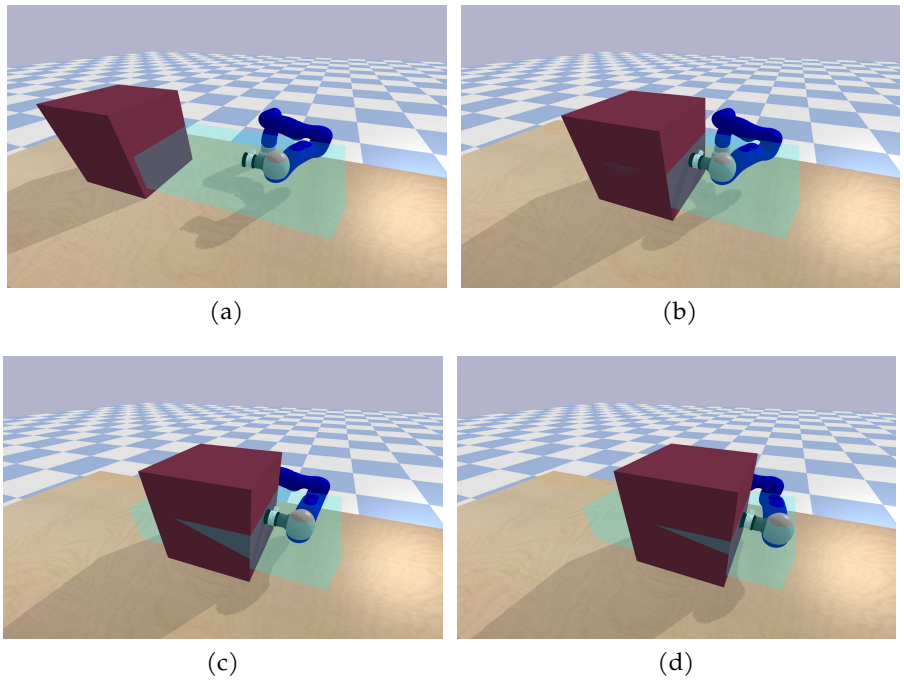


Figure 3.9: In these keyframes the robot halts a moving object travelling with 0.65 m/s. The cyan transparent box is an illustration of the workspace of the robot.

force is $\approx 680N$ and the force profile is similar to ones shown in Fig. 3.16. Such impact force can be catastrophic for the real robot and the object, while many times lead to the object moving away.

3.5.5.2 *Impact-agnostic vs Impact-aware methods*

Impact-agnostic is a term we use to refer to TO methods that plan the contact force solely on the complementarity condition (Stouraitis, Chatzinikolaidis, et al., 2018; Önol, Long, and Padır, 2019; Sleiman et al., 2019), while the impact-aware TO is realized with the multi-mode TO that utilizes the proposed contact force trasmission model (see Section 3.5.3). In the second ablation study, we compare these two methods on the same halting task. The computation time for the impact-agnostic method is on average 63 ms and for the impact-aware 141 ms. In Fig. 3.11a we show the relative distance between object and the end-effector. For the impact-agnostic method, the contact transition is abrupt and drives the object away from the manipulator ($t \approx 1.2s$). For the proposed method, the contact transition is smooth and results in stable contact. The reason behind the two different outcomes is the planned force profiles displayed in Figs. 3.11b and 3.11c. The impact-agnostic method plans a single impulsive force with duration $\delta t \approx 0.15s$—similar to Sleiman et al., 2019—which stops the object abruptly. The planned force cannot be tracked accurately, which results in an even higher impulsive force. In contrast, due to the enlarged duration

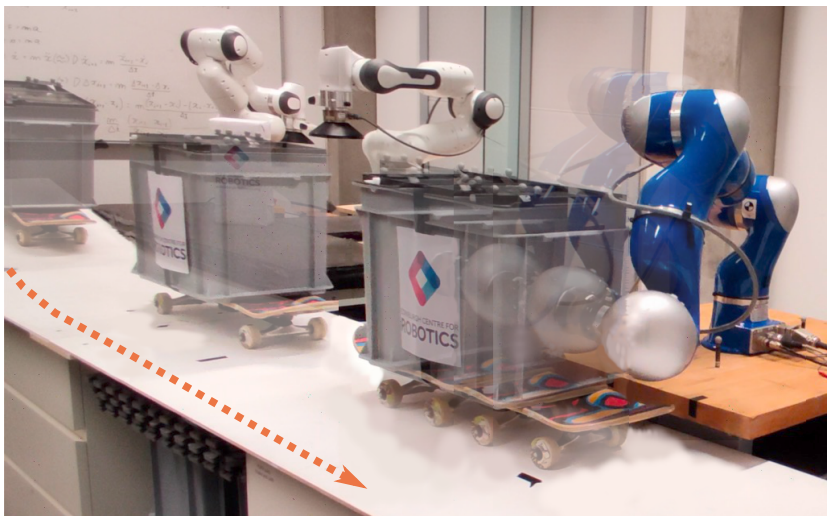


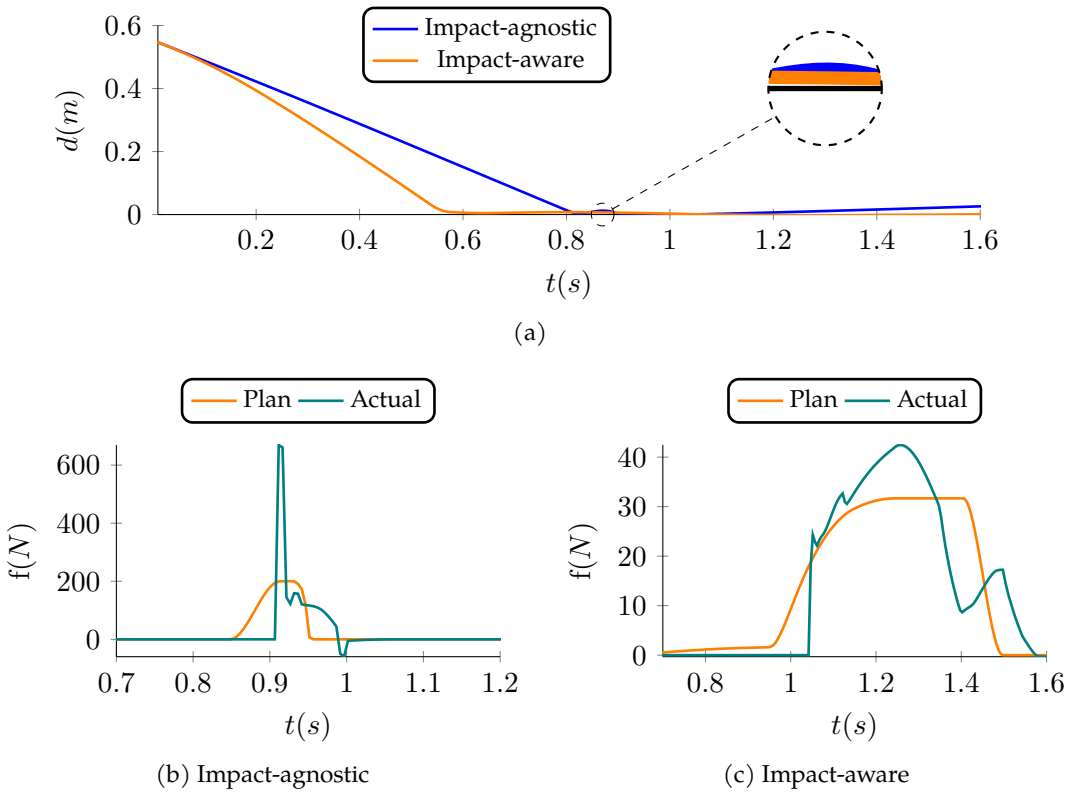
Figure 3.10: Experimental setup where the robot halts an object with a mass of 20 kg travelling at speed of 0.88 m/s.

($\delta t \approx 0.9s$) of the planned force, the proposed method generates a tractable force profile that smoothly halts the object. It is worth noticing the scale difference of the force (f) axis in Figs. 3.11b and 3.11c.

3.5.5.3 *Halt-Push the object with multi-mode TO*

Here, the task is altered. In addition to the task of halting the object, the manipulator needs to push it back to a desired location. The initial object velocity is $[0.0\ 0.4\ 0.0]^T\ m/s$. This task emphasizes the stiffness regulation capability of the proposed method. We report the optimized $\boldsymbol{\alpha}_l$ and contact force profile that simultaneously satisfy the task and the workspace limits of the robot. The computation times for these evaluation are between 96 ms and 512 ms.

In Fig. 3.12 we show the planned contact force profile for a number of different workspace limits (box bounds) and a number of different desired positions for the object. For the same desired position 0.8 m, if the workspace limit of the manipulator is increased, the optimized α_{-1} (halting stage) gradually decreases from 7.72 to 2.23, while α_1 (pushing stage) remains at the maximum value of 20. Also, the contact duration is increased and the maximum contact force is decreased. On the other hand, by reducing the workspace limit towards zero the planned force becomes an impulsive force–similar to impact-agnostic methods. For a fixed workspace limit 0.5 m, while the desired position of the object is varied from 0.8 m to 0.4 m, the maximum contact force decreases, α_{-1} (halting stage) gradually decreases from 2.23 to 1.75 and α_1 (pushing stage) remains at the upper bound value 20. In these cases, the contact duration is similar and only the force magnitude is adapted.



(a)

(b) Impact-agnostic

(c) Impact-aware

Figure 3.11: Impact-agnostic versus impact-aware. Relative distance between object and the end-effector (a) for both methods. (b) and (c) show planned and measured in simulation contact forces for each of the two methods.

Furthermore, due to the individual stiffness regulation for each contact stage based on α_l, the planned contact force increases slowly to refrain from impacts during the halting stage (l = -1), while in the pushing stage (l = 1) the planned contact force increases rapidly in order to push the object to the desired position.

3.5.5.4 *Robot experiments*

We validate our approach in a real setting with the KUKA LWR arm and the `VICON` motion capture system (see Appendix D), where the latter is used to measure the position of the object in real time. The experimental setup is shown in Figs. 3.10 and 3.13. The object is 20 kg and its initial position on a slope accelerates it to an initial velocity. Once the object arrives in a predefined position range—based on measurements of its position—we estimate its velocity and acceleration on-the-fly. These are estimated by fitting a rolling friction model with the *coefficient of determination* being in the range $0.5 \leq r^2 \leq 0.8$, which implies that there is uncertainty in the motion estimation. These estimated values are then passed onto the impact-aware method, which predicts (by integration) the future motion of the object, and computes an optimal motion plan in less than 150 ms to halt the object within the workspace limits. The position and

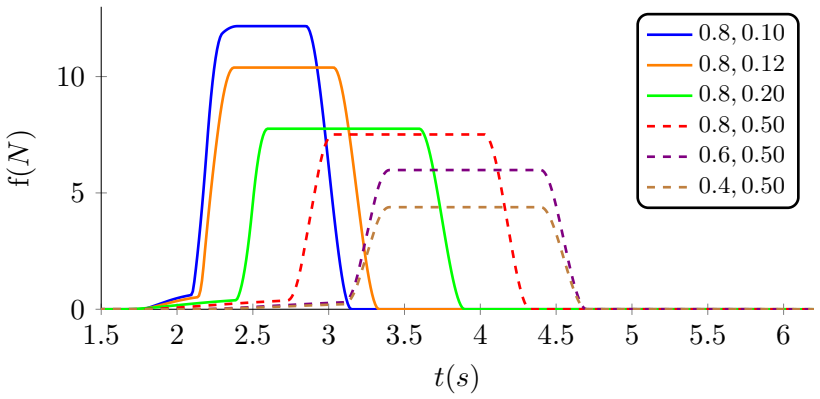


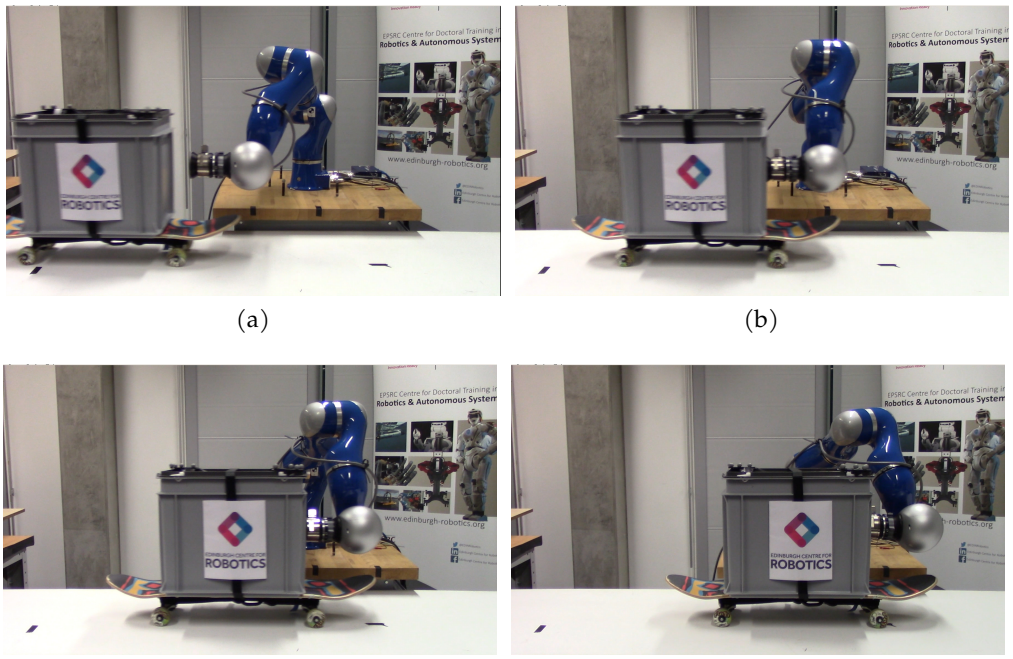
Figure 3.12: Contact force profiles with different desired positions for the object (left column) and workspace limits of the robot (right column).

stiffness profiles of the motion plan are streamed to the robot, such that the joint position with Cartesian stiffness control mode of the KUKA LWR arm tracks the optimal motion[16]. The impact-aware position trajectory is optimized in the task space and is realized on the robot in an open-loop fashion after being mapped to the configuration space using IK, along with the task space stiffness trajectory. The manipulator is controlled at 200 Hz, while the Vicon system runs at 100 Hz. The material of the end-effector is stiff to have rigid contact.

In Fig. 3.15 we report the measured contact force obtained by an ATI F/T sensor at 100 Hz. The object travels with a speed of 0.66 m/s. The proposed method halts its motion with the maximum force being less than 20 N when the full workspace is available, as shown in Fig. 3.13, and less than 30 N (see video) in a reduced workspace scenario, as shown in Fig. 3.14. As a baseline—similarly to Section 3.5.5.1—we report the measured force with a very soft configuration ($K = 10$ and $\lambda = 1$) of the LWR arm compliance controller. In this case, the maximum impact force is 199.47 N (see Fig. 3.16), which is 10 times larger than the one shown in Fig. 3.15.

Furthermore, to emphasize the capabilities of the method we consider the same object with a speed of 0.88 m/s. The positions of the object and the end-effector are shown in Fig. 3.17 to display their alignment, while the contact force is spread-out as shown in Fig. 3.15. For these initial conditions the contact force remains smaller than 55 N, while the baseline is omitted in order to avoid stressing the hardware.

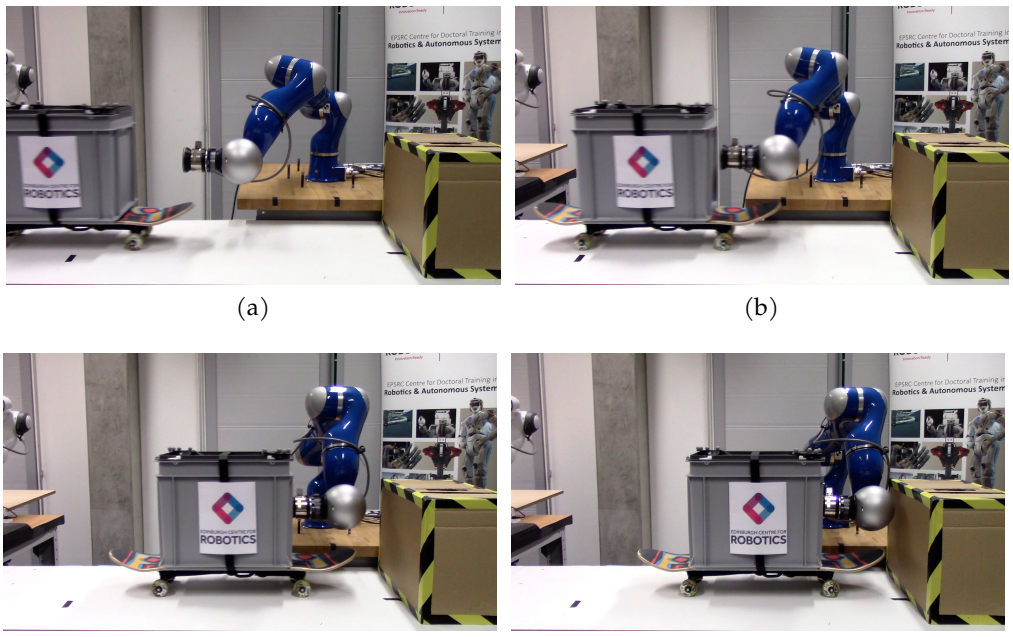
16 The Cartesian stiffness control mode of the KUKA LWR arm is based on impedance control (Albu-Schaffer et al., 2003), while an alternative option to compliance could be admittance control, for which more details can be found in Appendix B.2.



(a) (b)

(c) (d)

Figure 3.13: Keyframes of the experiment where the robot halts a moving object with speed of 0.66m/s.



(a) (b)

(c) (d)

Figure 3.14: Keyframes of the experiment where the robot halts a moving object with speed of 0.66m/s. The workspace of the robot is 20cm smaller, due to the presence of an obstacle.

3.6 DISCUSSION

In this chapter we have presented a multi-mode Trajectory Optimization method that encodes both hybrid dynamics and hybrid control in a single formulation. Our ap-

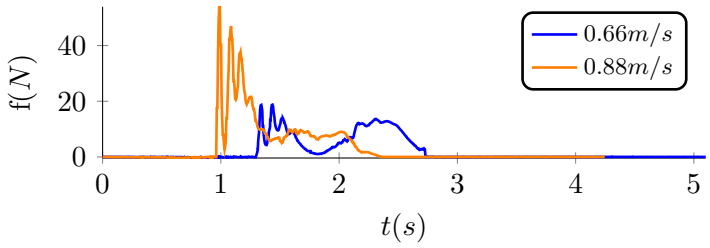


Figure 3.15: Experimental result of contact force during halting motion.

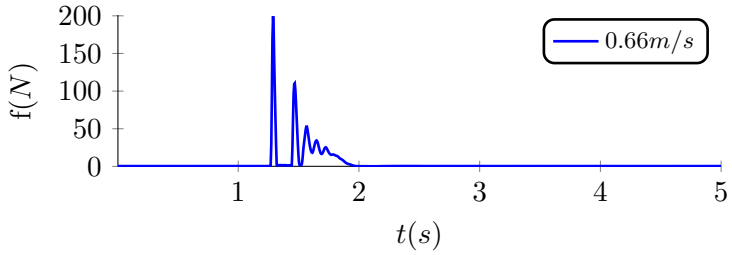


Figure 3.16: Experimental result of impact between the object and the end-effector during halting an object with speed of 0.66 m/s.

proach computes the optimal multi-contact plans for the robot to complete manipulation tasks given a parametric representation of the object. The method computes a dynamically consistent and optimal hybrid solution for the (i) trajectory of the object, (ii) agent's forces, (iii) agent's stiffness profiles, (iv) agent's contact locations and (v) respective timings of these actions.

In our multi-mode TO, we decided to incorporate contact transitions and controller switches into the TO problem by pre-defining the *structure of the motion* (see Section 3.1.3). The *structure of the motion* sets the sequence of *contact-control* modes (see Section 3.3) and is a generalization of the observation (Winkler, Bellicoso, et al., 2018; Mordatch, Todorov, and Popović, 2012) that motions can be separated into *contact-invariant* phases. At a high level, two of the supporting arguments for this design choice are: (i) the *structure of the motion* can be efficiently planned at a higher-level of abstraction, as it has been shown by Toussaint, Allen, et al., 2018, and which we will also present in the following chapter; and (ii) to the extend of our knowledge, in scenarios where a trajectory includes a number of different modes of motion or control, such as switching controllers or multi-physics motions (Toussaint, Jung-Su, and Danny, 2020), these need to be explicitly specified in the TO problem.

On the other hand, if we only consider multi-contact problems like the one investigated in Section 3.4, there are a number of approaches that follow variations of the *contact-implicit* formulation (Posa, Cantu, and Tedrake, 2014). Thus, next we discuss

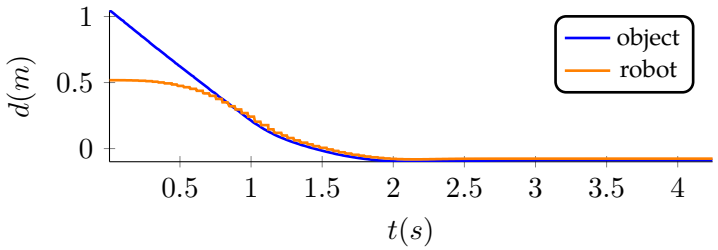


Figure 3.17: Experimental result of the position of the object and the end-effector during halting an object with speed of 0.88 m/s.

the advantages and disadvantages of *multi-phase* methods[17] against *contact-implicit* methods.

The first attribute we consider is the solution space explored. Both types of methods can explore an arbitrarily shaped surface to discover the best suited contact locations. Similarly, for both approaches there are no restrictions in terms of force profiles obtained. In terms of the contact schedule—also called *structure of motion*—*multi-phase* (Winkler, Bellicoso, et al., 2018) methods require the maximum number of contact switches to be given in advance. In theory, the duration of each phase can be reduced to zero, thus the contact schedule discovered can have fewer contact switches, if needed. In practice, due to numerical instabilities that might arise in the bi-linear product in (3.12), when a phase duration goes to zero, phase elimination is difficult. This typically results in modest adaptation (Winkler, Bellicoso, et al., 2018) of the contact schedule or exact match thereof (Stouraitis, Chatzinikolaidis, et al., 2018). Yet, phases of the motion are stretched and squeezed in time to comply with the physics of the motion. On the other hand, *contact-implicit* (Posa, Cantu, and Tedrake, 2014) methods can alter the contact schedule as needed. Further, the switching contact model used in *multi-phase* approaches, enables three distinct following contact conditions: (i) *free-motion*, (ii) *in-contact*, and (iii) *pre-contact*. This essentially models free motion and sticking contact[18]. In *contact-implicit* methods the switching contact model enables the consideration of the three distinct following contact conditions: (i) *free-motion*, (ii) *in-contact*, and (iii) *sliding-in-contact* (Toussaint, Jung-Su, and Danny, 2020). This in turn models free motion, sticking contact and sliding contact.

The second attribute we consider is the convergence properties of each method. Both methods are based on non-linear programming and as both are non-convex and non-linear, the solutions depend on the initial seed provided and are local minima. In terms of the optimization landscape, Nurkanovic, S. Albrecht, and Diehl, 2020

17 In this part of the discussion, we inspect only multi-contact problems, where the notion of modes is not required. Thus, we use the term *multi-phase* methods, which is consistent with the relevant literature.

18 We empirically observed that the use of the *pre-contact* condition does not significantly improve either the accuracy or the real-world realization of the hybrid motion plans. Hence, it can be neglected, as we also did at the impact-aware manipulation instantiation of the problem (see Section 3.5).

indicate that *contact-implicit* methods require special attention in the relaxation of the problem to avoid spurious local minima, while *multi-phase* approaches provide a smooth non-linear program. Further, the computation times of *multi-phase* methods (Winkler, Bellicoso, et al., 2018) is typically smaller that the ones of the *contact-implicit* methods (Posa, Cantu, and Tedrake, 2014). Recently, a preliminary study that directly compared the two approaches (Turski, Norby, and Johnson, 2020), presented results that further support the two previous statements. These being; *multi-phase* methods converge faster and with higher success rate than *contact-implicit* methods, yet *contact-implicit* methods can attain more optimal solutions in a number of cases. In Chapter 5 we use the *multi-phase* method presented here and we propose an approach to simultaneously assemble the *structure of the motion* on-the-fly.

In terms of the more general multi-mode TO framework presented in Section 3.5, we suggest that simply planning contact forces based only on the contact state does not suffice towards making stable contact. Thus, an impact model is embedded in the TO in the form of a controller. Here, we remind the reader that the TO problem is solved with non-linear programming methods and yields optimal stiffness profiles—suited to absorb impacts—for an impedance controller. In parallel, a number of works have used variants of Differential dynamic programming (DDP) (W. Li and Todorov, 2004) to plan multi-contact motions both for manipulation (Önol, Long, and Padır, 2019) and locomotion (Neunert, Farshidian, et al., 2017; Neunert, Stäuble, et al., 2018; Mastalli et al., 2020) tasks. The DDP-like approaches can efficiently compute motions and can by default provide continuous feedback gain profiles (D. Braun, Howard, and Vijayakumar, 2012; D. J. Braun et al., 2013), which correspond to stiffness profiles. Yet, these have not been employed in multi-contact problems (Önol, Long, and Padır, 2019; Neunert, Farshidian, et al., 2017) to cope with deviations and instabilities that might arise due to impacts.

4

DYADIC MANIPULATION PLANNING IN HYBRID SPACES

> Alone we can do so little, together we can do so much!
>
> Hellen Keller

Chapter 3 is focused on the agent's policy and in particular, on the mechanics of hybrid action planning using *Trajectory Optimization* (TO) methods. These developments enable single agent robots to plan and realise manipulation tasks with multiple changes of grasp-holds, while impacts during contact transitions can be mitigated. This chapter investigates extensions of the hybrid TO method into dyadic setups.

Currently, the contribution of robots in human-robot dyads is limited, as robots can only assist humans in a handful of scenarios, e. g. transactional tasks. One of the underlying reasons is that the robots' agency is inadequate, since one crucial capability of collaborative agents' is neglected; this is the ability to select and alter grasp-holds. In (Busch et al., 2017) task space attributes, e. g. the trajectory of the object, were optimized to facilitate human ergonomics. In (Peternel, Tsagarakis, and Ajoudani, 2017) adaptation during co-manipulation was realized through turn-taking collaboration, which boils down to planning actions in a timely manner. Further, a number of methods focus on the dynamic properties of the interaction. Inverse dynamics approaches concentrated on the torque and force regularization (Lin et al., 2018; Otani, Bouyarmane, and Ivaldi, 2018), while others adapted the impedance characteristics of the robot online to accommodate for partner's actions (Gribovskaya, Kheddar, and Billard, 2011; Noohi, Žefran, and Patton, 2016). However, a central aspect of manipulation is the selection of the appropriate contact locations on the object (Roa and Suárez, 2014). Also, as it has been shown in Fig. 2.5, grasp-hold changes can be critical with respect to the completion of the collaborative tasks. Accordingly, the exploitation of the contact space of the object by the individuals is vital in *Dyadic collaborative Manipulation* (DcM) scenarios. To the best of our knowledge, contact adaptation within collaborative manipulation scenarios has not been addressed prior to this work, although it is crucial for enabling robots to perform complex DcM tasks jointly with a human partner.

To extend the study on hybrid TO methods presented in Chapter 3 to dyadic hybrid TO, we utilize the formulation provided in Chapter 2 and in particular the schematic

shown in Fig. 2.6. We perceive dyadic setups from a modular point of view, hence we use the dyadic interaction component to model joint action. We shape the hybrid TO method (see Chapter 3) to match the dyadic coupling alluded in the schematic and formulation presented in Chapter 2.

One of the key discussion points in Chapter 2 is the separate treatment of the two individuals' policies and the dyadic interaction. Given that we follow this design choice to realize the dyadic hybrid TO, we further motivate this decision based on two works, one in psychology, cognition and behavior (Meyer, Wel, and Hunnius, 2013), and the other in human-motor control (Takagi, Ganesh, et al., 2017). In the former, participants were asked to complete tasks that require higher-order planning. The investigated setups included both single-agent tasks, as well as dyadic tasks. The results demonstrated similarity between the underlying planning process in both setups. Further, it was shown that individuals improve their task performance through learning in both setups and do transfer their task experience from single-agent setups to dyadic setups. In the latter, the behavior of humans participating in an interactive task was analyzed. The task's goal was to track a moving target for both individuals of the human dyad, while communication between the two could only be achieved through the haptic channel. The study showed that the two individuals model the task and control their behaviors individually. To coordinate their actions, they estimate their partner's movement goal and optimally blend it into to their own movement goal to improve their own task performance. Both these studies analyze collaborative dyadic setups, like DcM, and share the same principle; humans plan and control their movements as individuals, while to collaborate with others, they tailor their plans and controls to utilize their partner's contribution in the task and to match the dyadic constraints, for instance, partner's capabilities.

In this chapter, firstly we recapitulate the main proposition of joint action. Secondly, to further acquaint the reader with the state-of-the-art on robots acting jointly with humans, we review a number of related works from the technical perspective. Next, we describe a simple way of reshaping the single agent TO method into a partner-aware TO method, while simulation-based evaluations and real-world experiments between a robot and a human follow, and reflect the effectiveness of the partner-aware TO method. Last, we refer to an alternative approach used to replicate joint action and qualitative compare it to the proposed method, which leads to future work pointers.

4.1 BACKGROUND IN JOINT ACTION

We start with the definition of joint action according to Natalie Sebanz, which is "joint action is the deliberate attempt to coordinate one's actions with others' to bring about

a change in the environment" (Sebanz and Knoblich, 2009). In addition to the previous definition, Natalie Sebanz indicated three primary abilities required to establish joint action (Sebanz, Bekkering, and Knoblich, 2006). These are: (i) the ability to form representations for the shared task and the partner, (ii) the ability to predict actions of the partner, and (iii) the ability to integrate into the future effects of own and others' actions.

Representations are useful as they enable encoding of desired outcomes of joint action, partner's actions and the agent's own contribution in realizing these outcomes. This in turn allows individuals to plan their actions for longer temporal horizon and act in anticipation of their partner's actions rather than simply responding to them. According to Knoblich, Butterfill, and Sebanz, 2011, in scenarios where the agents' behavior is driven by these representations, the type of coordination between the two individuals can be distinguished as *planned coordination*.

Action predictions are separated into two sub-categories (Sebanz, Bekkering, and Knoblich, 2006; Knoblich, Butterfill, and Sebanz, 2011). In the first category, predictions are based on associations between certain events within the environment and the actions of the partner. Such predictions allow an individual to plan actions in response to events that will only occur a considerable time ahead, thus this skill is essential in *planned coordination*. In the second, predictions are based on action observation and thus are simple and immediate. This coordination occurs spontaneously, due to vision-motor couplings of the individuals and is independent of any joint plans. According to Knoblich, in such cases the type of coordination between the two individuals can be distinguished as *emergent coordination*.

Integration of actions' effects is the ultimate basis of action planning and essential skill of *planned coordination*. Whether these effects result from one's own, or the partner's actions or joint action, an individual—participating in a joint action task—needs to be able to predict them (Sebanz, Bekkering, and Knoblich, 2006; Sebanz and Knoblich, 2009).

Having briefly discussed the key aspects of joint action, we next state key insight of action representation and their principle modelling axes. A number of studies (Umilta et al., 2001; Kohler et al., 2002; Fogassi et al., 2005; Bekkering, Wohlschlager, and Gattis, 2000; Prinz, 1997) have shown that humans encode actions in terms of action goals and are able to predict both the intended future goals of others' and the means others intent to act to reach this goal (Verfaillie and Daems, 2002).

One of the modelling axes is considered with the '*what*' action, which refers to the kind of action an individual will perform and to the intention that drives this action (Sebanz and Knoblich, 2009). Essentially, the '*what*' action denotes the semantics, while the types of action in manipulation can be grasp-hold change, push, pull,

communicate, etc. Another modelling axis is involved with '*where*' the actions and its effects take place. Actors need to coordinate their actions, so they need to effectively distribute the common space. Such space can be the Cartesian space, the force space, etc. A way to obtain these spatial predictions about others is as a by-product of their goals (Erlhagen and Jancke, 2004). This has also been proposed by Sebanz, Knoblich, and Prinz, 2005, where task and environment conditions of an individual can be mapped into particular actions. Yet, merging one's own and partner's action plans to complete a joint task is not always sufficient. Thus, the third axis is concerned with the '*when*' an action happens. This is critical for joint action task with close temporal coordination and enables acting both synchronously or in turns (Sebanz and Knoblich, 2009).

Before laying out the previous work related to this chapter, we list two closing remarks borrowed from the work of Knoblich, Butterfill, and Sebanz, 2011, which summarizes findings in the fields of developmental psychology, cognitive psychology and cognitive neuroscience. (i) Joint tasks such as lifting and manipulating a heavy object with a partner require *planned coordination*. This is because agents have to plan their own future actions in relation to joint action goals and in relation to partner's actions. (ii) Although, such tasks require *planned coordination*, this does not preclude that *emergent coordination* can naturally arise. In fact, it is not unusual that both coordination types are observed in human dyads when performing such tasks.

4.2 Related work on human-robot collaborative motion planning

Here, we review the state-of-the-art *physical Human-Robot Collaboration* (pHRC) methods that aim to achieve collaborative tasks according to the joint action principles mentioned above. In comparison to our literature review in Chapter 2, here we examine previous work in further depth and we skip control-focused methods (Agravante et al., 2014; Lin et al., 2018), as they typically omit joint action models and focus more on stability and safety properties of the controllers.

4.2.1 *Planned coordination between a human and a robot*

The works mentioned next can be placed under the family of *planned coordination*. They all use a model for the task or the partner that enables them to anticipate the future and ultimately plan the robot's action accordingly.

In (Thobbi, Gu, and Sheng, 2011), a collaborative lifting task was studied. A confidence measure of the human's goal prediction was used to alter between reactive and proactive robot behaviors. One could perceive the uncertainty of the goal estima-

tion as an indication of whether the human partner clearly drives the object towards a particular goal or hesitates, which results in the robot assisting or waiting, respectively. In another collaborative manipulation scenario, Nikolaidis, Kuznetsov, et al., 2016 proposed a model for the human behavior that relates trust with willingness to comply with the motion plans of the robot. Based on this model, the human behavior can switch from leader-like to follower-like, and vice versa. In correspondence, the robot adapts its behavior to retain the trust of the human or to optimally guide the task. The representations used in these works addressed the '*what*' question of joint action, where the type of human action or behavior triggers a switch in the robot's plans.

In the early work of Y. Maeda, Hara, and Arai, 2001, a 1D joint transportation task is considered. The motion of the human partner is predicted according to a minimum jerk model (Flash and N. Hogan, 1985). Based on the human motion prediction, the robot's reference goal is updated and the motion is compliantly tracked with an impedance controller. Similarly, in (Mainprice and Berenson, 2013; Mainprice, Hayne, and Berenson, 2015), the human's occupied workspace of a reaching motion is predicted and used from the robot to plan collision-free kinematic reaching motions. In (Mainprice and Berenson, 2013), the human motion prediction is based on a learned motion library based on *Gaussian Mixture Model* (GMM) (C. M. Bishop, 2006). In (Mainprice, Hayne, and Berenson, 2015) the human motion is predicted with online motion planning. The cost function used was obtained with Inverse Optimal Control (or Inverse Reinforcement Learning (Ng, Russell, et al., 2000)) and with example trajectories collected from two individuals performing a simultaneous reaching task. Thus, the human motion prediction takes into consideration the likely adaptation with respect to the robot's motions. A common assumption of these works is the access to human's intended goal. These representations addressed the '*where*' question of joint action and are valid models for kinematic tasks, for example, reaching motion.

Hand-over tasks is another type of reaching motions that has attracted attention. G. Maeda et al., 2017 proposed a method to transfer adaptive hand-overs to robots from kinesthetic demonstrations using interaction probabilistic movement primitives. A key element of this approach is the robot's ability to estimate the phase of the human motion and predict the human trajectories under sparse observations. This work proposed a probabilistic representation to address both the '*where*' and '*when*' questions of joint action in kinematic tasks.

On a different line of work, collaborative manipulation of plank-like objects between a human and a robot has been studied. In (Ghadirzadeh et al., 2016), *Gaussian Processes* (GP) (Williams and Rasmussen, 2006) were used to learn a forward model

that includes task evolution and interaction forces generated by the human partner. The learned model was then used to train a robot policy with *Reinforcement Learning* (RL) (Sutton and Barto, 2018). Likewise, Gribovskaya, Kheddar, and Billard, 2011 proposed a method that offline learns the task model in the form of a dynamical system, which includes both a kinematic and a haptic information. On the robot side, an adaptive impedance law was adopted to track the reference motion and eventually support the human partner in an object co-manipulation task. Chen, Figueredo, and Dogar, 2018 considered a scenario where the robot assists the human in performing forceful tasks, such as drilling. The method assumes that the human selects the most comfortable body pose and proposes an optimization-based planning method for the robot configuration and the object pose, that promotes two metrics of human comfort while satisfying task constraints. In these works, the task goal was fixed and the representations also addressed the '*where*' question of joint action. However, these models are valid for tasks that also involve force interaction.

4.2.2 *Emergent coordination between a human and a robot*

Here, we review works that we place under the family of *emergent coordination*. All of them generate the robot's actions as a response to stimuli caused either by their partner's action or by the state of the task.

Hand-over tasks have also been pursued from a reactive point of view. Vogt, Stepputtis, Jung, et al., 2018 distilled interaction constraints during hand-over tasks using a data-driven method. These constraints specify the relationship between the state of the two individuals for the whole duration of the hand-over and they were used to form online robot responses. This method addressed the '*where*' question of joint action to demonstrate online kinematic robot responses.

Under the reactive regime, Peternel, Tsagarakis, and Ajoudani, 2017 realized a robot-assisted sawing task with a human partner. Sensorimotor information of the human's state was estimated using a combination of vision-based markers for arm force manipulability and electromyography (EMG) signals from muscles for arm stiffness estimation. These estimations were used by the impedance-controlled robot to coordinate along the time axis the push and pull sawing actions with the ones of the human partner. The proposed system addressed the '*when*' question of joint action in a reactive manner. However, making predictions about the future timing of events, such as stiffness changes, is yet to be addressed.

Rozo Castañeda et al., 2013 employed force and vision information to commence the appropriate learned impedance behaviors depending on the task phase. Task-parameterized GMMs were used to statistically capture the relation between force-

based impedance behaviors of the robot and the evolution of the task—both in terms of execution phase and continuous interaction. Execution phase defines the axis of compliance and continuous interaction the stiffness magnitude. Thus, the proposed method addressed both the '*what*' and '*where*' questions of joint action to create appropriate robot responses, for example, stiffness modulation.

4.3 PARTNER-AWARE TRAJECTORY OPTIMIZATION

In this section, we present the specifics of how to incorporate the partner's policy in the TO framework. To help the reader contextualize the details that follow, we start by briefly restating the dyadic problem formulation. We then introduce the details on how to integrate the partner's actions into our framework and the representation used to model the behavior of the partner. We conclude this section by categorizing our approach according to the propositions and modelling axes of joint action[1].

4.3.1 *Dyadic problem formulation*

We express dyadic planning based on the abstract formulation for partner-aware dyadic planning, introduced with (2.6). This formulation is now written in a form similar to (3.9) as

$$\begin{aligned} \min_{\boldsymbol{x}(t),\boldsymbol{u}(t)} \quad & \boldsymbol{c}(\boldsymbol{x}(t),\boldsymbol{u}(t),\hat{\pi}^p,\boldsymbol{z}) && \text{(4.1a)} \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t),\boldsymbol{u}(t),\hat{\pi}^p,\boldsymbol{z}\right) \quad , && \text{(4.1b)} \\ & \boldsymbol{g}(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{z}) \leq 0 && \text{(4.1c)} \end{aligned}$$

where all variables are defined as in Chapter 2 and Chapter 3. The key characteristic of (4.1) is the inclusion of the partner's policy estimate $\hat{\pi}^p$ in (4.1a) and (4.1b). The latter reflects (2.3) in the formulation, which indicates that the computed trajectory of the object results from both the agent's and the partner's policy. (4.1a) denotes that the final cost of the motion also depends on the partner's policy. To solve the optimization problem (4.1), we perform direct transcription, as explained in Section 3.1.2, and the discretized version of the problem takes the form of (3.2).

Next, the actual form of the dynamics constraint (4.1b) is detailed. As the cost function (4.1a) can take many different forms depending on the task, we prefer not to dictate a specific composition of it. The rest of the constraints remain the same as in Section 3.4.1.

1 A video presentation of this piece of work is provided (see SV4 in Appendix E).

4.3.2 *Integration of partner's actions*

In DcM scenarios, the object is jointly manipulated by both individuals by applying forces on it. Thus, we propose to incorporate the partner's policy in the TO framework through the *transcription* constraints defined in (3.12). In this case, however, the object dynamics—previously defined in (3.25)—are subject to the partner's wrenches too, and are described by

$$\begin{bmatrix} MI & 0 \\ 0 & J \end{bmatrix} \ddot{\mathbf{y}}_i + \begin{bmatrix} Mg \\ \dot{\mathbf{y}}_i^{\omega} \times (J\dot{\mathbf{y}}_i^{\omega}) \end{bmatrix} = \sum_{\kappa}^{K} \begin{bmatrix} I \\ \hat{\mathbf{c}}_i^{\kappa} \end{bmatrix} \mathbf{f}_i^{\kappa} + \boldsymbol{\lambda}, \tag{4.2}$$

where $M \in \mathbb{R}$ and $J \in \mathbb{R}^{\nu \times \nu}_{\geq 0}$ are the mass and inertia of the object, I is the identity matrix with size $\nu \times \nu$, g is the gravitational acceleration, $\dot{\mathbf{y}}_i^{\omega}$ is the angular velocity of the object, and with $\hat{(\cdot)}$ we refer to the cross product matrix formed by the input vector. By realizing (3.12) according to the augmented dynamics—where $\boldsymbol{\lambda}$ represents the partner's contribution—the trajectory optimization generates plans in accordance to the partner's policy, referred as *partner-aware*. This is illustrated in Fig. 2.6 with the physical constraints block. Note, that our method requires only an estimate of partner's policy—represented with $\boldsymbol{\lambda}$—in contrast to (Toussaint and Lopes, 2017), where the method assumes full control authority over the partner's actions.

4.3.3 *A representation of the partner's policy*

To incorporate partner's actions into the dyadic planning framework, we first need to obtain an estimate of the partner's policy. An essential step towards this goal is to identify the appropriate function space (representation) in which the partner's policy lies. This is shown in Fig. 2.6 with the *parametric model* block. We use here a simple but ample parametric model for the partner's policy,

$$\boldsymbol{\lambda}_i = \mathbf{K}^p(\mathbf{y}_N^* - \mathbf{y}_i) + \mathbf{D}^p(\dot{\mathbf{y}}_N^* - \dot{\mathbf{y}}_i). \tag{4.3}$$

The parameters $\mathbf{K}^p$ and $\mathbf{D}^p$ denote a spring-damper behavior of the partner towards the goal $\begin{bmatrix} \mathbf{y}_N^{*\,T} & \dot{\mathbf{y}}_N^{*\,T} \end{bmatrix}^T$ of the co-manipulation task. $\mathbf{K}^p$ can be interpreted as the parameter that can shape whether the partner acts as a leader (i. e. $\mathbf{K}^p \gg 0$) or as a follower (i. e. $\mathbf{K}^p = 0$), along with all the intermediate behaviors. The goal $\begin{bmatrix} \mathbf{y}_N^{*\,T} & \dot{\mathbf{y}}_N^{*\,T} \end{bmatrix}^T$ captures the partner's intentions relative to the task. This model has been used in human motor control research (Takagi, Usai, et al., 2018), as it can capture essential aspects of the partner's policy.

Partner's policy oracle function: For the following analysis and experiments, we assume that an oracle function exists. The oracle function can predict the parameters $\theta^p = (\mathbf{y}_N^*, \dot{\mathbf{y}}_N^*, \mathbf{K}^p, \mathbf{D}^p)$ that describe the current mode of the partner's policy (see Section 2.2.1). This in turn enables the use of (2.2) without the need to compute $Pr(\theta^p|\mathbf{x}_t, \mathbf{q}_t^p, H^p)$. In practice these parameters are manually provided, while the realization of the oracle function as a probability distribution is discussed in the future work section.

Our dyadic planning TO is a *planned coordination* approach to joint action, due to its ability to integrate into the future the actions of the partner. The representation used for the partner's policy addresses the '*where*' question of joint action, with hyperparameters on the goal and the impedance characteristic of the human behavior. As has been mentioned above, stiffness can be related to the role of the human partner (leader or follower), thus this representation could also be used to address the '*what*' question of joint action.

4.4 EVALUATION AND EXPERIMENTS

To evaluate the proposed partner-aware TO method we perform a number of simulations. The main focus is on investigating the ability of the method to adapt various aspects of the hybrid motion—such as discrete grasp-hold locations, continuous trajectory of the object, etc—with respect to the partner's policy. A dyadic and dynamic hybrid motion is illustrated in Fig. 4.1(a-f), where a robot assist a human avatar in simulation to complete the challenging task of rotating $180°$ a box. Next, we validate our method with real world experiments on a human-robot dyad, where the human and the bimanual robot carry out a collaborative manipulation task that requires change of grasp-holds. In all evaluations and demonstrations presented here, the state of the object is $\mathbf{y} = \begin{bmatrix} x & z & \phi \end{bmatrix}^T$, the task dimension $\nu = 2$ and the *structure of the motion* is pre-specified(see Section 3.1.3). A video showing the simulations and experiments performed is provided (see SV1 in Appendix E).

4.4.1 *Simulation evaluations*

We present here a number of scenarios where different parameters of the partner's policy model (see Section 4.3.3) are altered. Each scenario instructs for a distinct solution that is tailored to the specific dyadic setup.

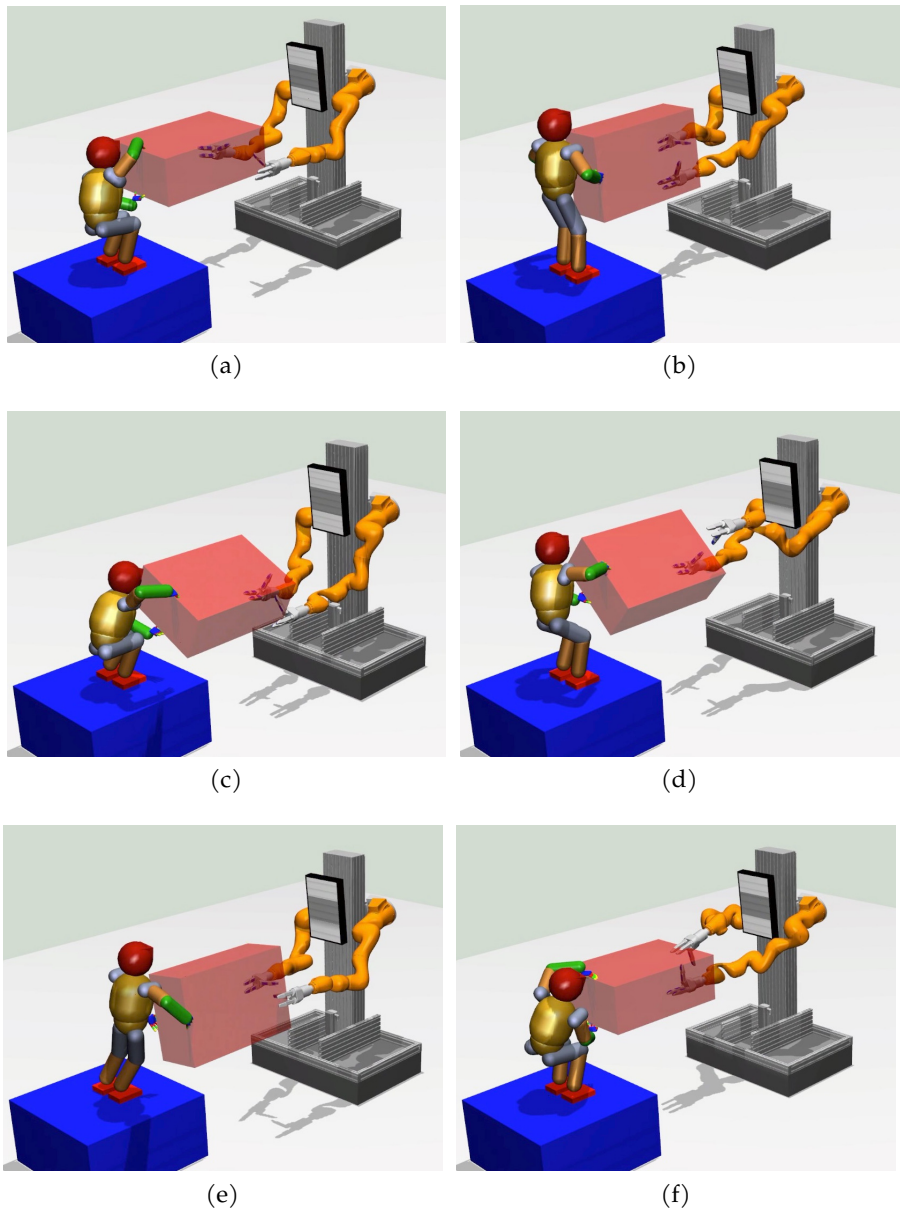
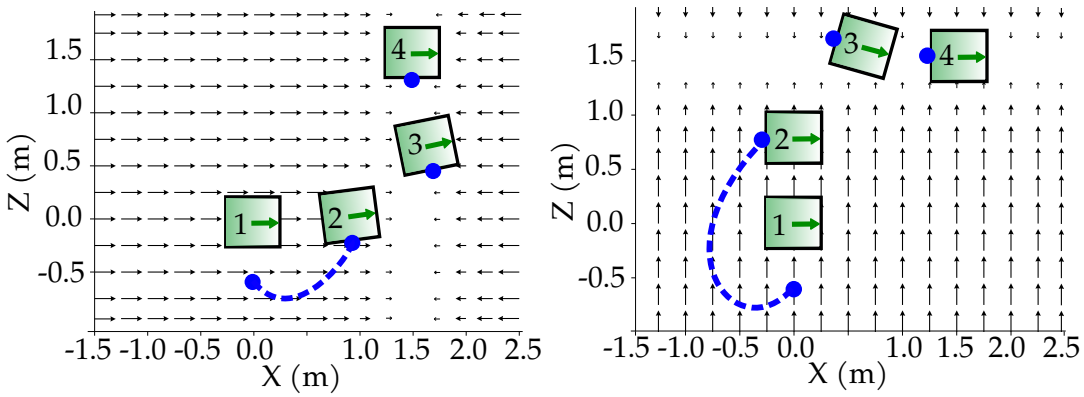


Figure 4.1: Keyframes of a $180°$ box rotation DcM scenario. The human avatar and the robot jointly complete the task. (a) Initial configuration. (b) Contact change by the right arm has been completed. (c) Orientation of the object is adjusted such that a contact change of the left arm can be realized. (d) Contact change by the left arm. (e) The left arm has changed contact and the object is rotated upside-down. (f) The task is completed.

4.4.1.1 *Partner-aware solutions*

First, we alter the partner's policy parameters $\mathbf{K}^p$ and $\mathbf{D}^p$ in (4.3). Each partner's policy is expressed as a force field along one axis: in $\pi^p 1$ along X, in $\pi^p 2$ along Z, in $\pi^p 3$ along the main diagonal, and in $\pi^p 4$ along ϕ axis. In Fig. 4.2a and Fig. 4.2b, the agent has one end-effector and jointly completes with the partner a $2.12m$ translation task



(a) Partner's policy 1: $\pi^p 1$ (b) Partner's policy 2: $\pi^p 2$

Figure 4.2: In (a) and (b) we illustrate generated motion plans in response to two different partner policies. The green rectangle is the manipulated object, the blue dot is the left end-effector of the agent. The start pose of the object is annotated with 1 and the goal with 4. The arrow field illustrates the forces applied by the partner. (a) and (b) emphasize the dependency of chosen contact location to the partner's policy.

in a zero gravity (table-top) scenario. Fig. 4.3a and Fig. 4.3b illustrate solutions for a $0.98m$ translation and a $-90°$ rotation task, generated as responses to two different partner policies in a scenario with gravity along the z axis. The former task highlights the effect of the partner's policy on the selected contact location. The variation of the computed solutions is evident in the latter task in Figs. 4.3c, 4.3d and 4.4, where we present trajectories for four distinct partner's policies.

Second, we adjust the partner's goal $\begin{bmatrix} \mathbf{y}_N^{*\,T} & \dot{\mathbf{y}}_N^{*\,T} \end{bmatrix}^T$ in (4.3). Fig. 4.5 shows the optimized contact locations and swing motions for three goals. These experiments demonstrate the capability of our method to adapt trajectories, contact locations, and action timings in response to different partner policies.

4.4.2 *DcM experiments*

We validate our approach in a real setting, where a human partner jointly manipulates an object with a bi-manual, i. e. $\kappa \in \{1, 2\}$, and $n = 32$ DoF robot. The robot moves on the horizontal plane in a omni-directional fashion—due to its mobile base—and utilizes its two Kuka LBR iiwa 820 arms along with two Schunk dexterous 3-finger hands for manipulation and DcM tasks. A linear joint allows the arm base to be translated along the vertical axis. We use two large boxes (36cm $\times$ 64cm $\times$ 100cm), so that a human cannot perform the task alone. The hybrid motion plans are optimized in the task space and are realized on the robot in a open-loop fashion, after being mapped onto the configuration space using *Inverse Kinematics* (IK). A detailed

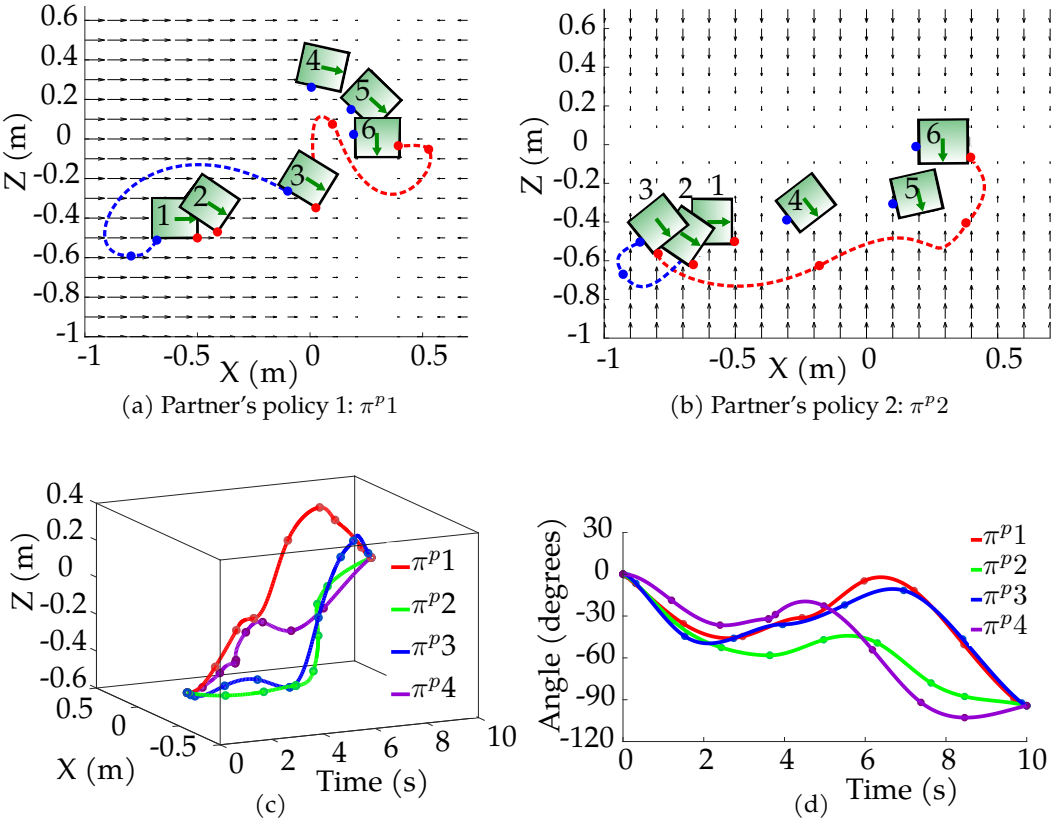


Figure 4.3: Similar to Fig. 4.2a and Fig. 4.2b, (a) and (b) depict the resulting trajectories in response to two partner policies. The red and blue dots are the right and left end-effectors of the agent. The start pose is annotated with 1 and the goal with 6. Most of the object's trajectory is planned at the active regions of the partner's force field, indicating that the robot utilizes the partner's contribution to the task accordingly. Trajectories are displayed separately for four distinct partner's policies: (c) in x and z dimensions, (d) in ϕ dimension.

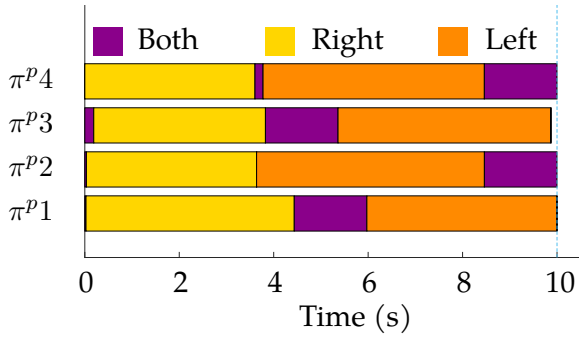
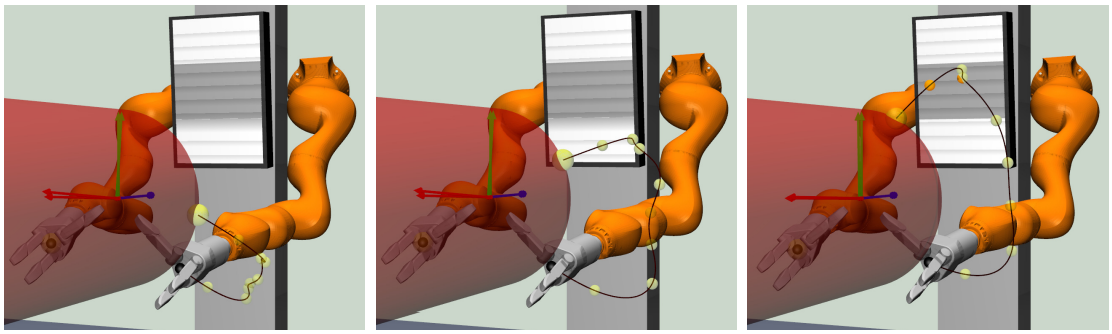


Figure 4.4: The arms' contact sequence pattern for the four distinct partner's policies shown in Figs. 4.3c and 4.3d. The colors indicate which arm is in contact with the object.

description of the physical system can be found in (Gienger, Ruiken, et al., 2018). The robot utilizes surface contacts at the planned contact locations as a form of mechan-



(a) (b) (c)

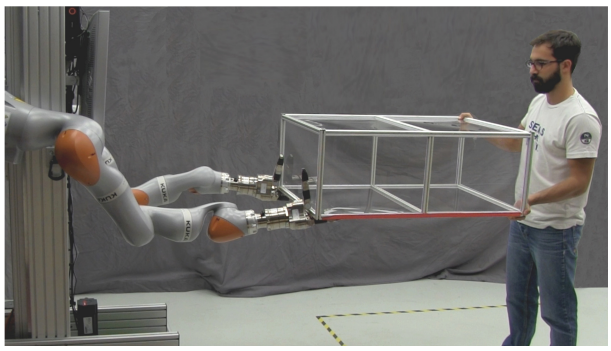
Figure 4.5: The agent's left end-effector performs a swing motion, while the partner supports the object from the opposite side. Depending on the partner's goal, the contact locations change. The small yellow spheres denote the knots of the trajectory and the largest one the anticipated contact location. The black curve is the interpolated trajectory. The partner's intended object orientations are (a) 30°, (b) 60° and (c) 90°.

ical feedback. Further, it is worth noting that the joint range of the robot does not permit for the task to be completed without grasp-hold changes.

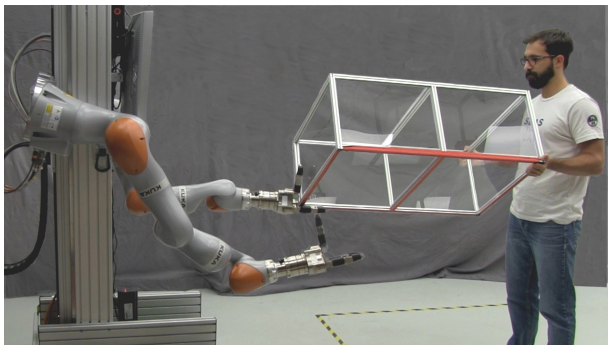
4.4.2.1 *Human-aware solutions*

In correspondence to the simulation experiments shown in Figs. 4.3 and 4.4, we demonstrate here a 90° rotation task of 10.0 kg and approximately 1.5 kg boxes, where one contact change per arm occurs. The key-frames of the DcM scenario are depicted in Fig. 4.6 and in Fig. 4.7. During these real-world evaluations, the human partner is rotating a box jointly with the robot. To emulate the human's policy, the parameters of the partner's policy model are selected such that the force field is uniform along all dimensions of the task. For the 10.0 kg box setup (see Fig. 4.6), the robot is supporting the weight of the object and passively assists towards the completion of the task, like a follower. For the 1.5kg box setup (see Fig. 4.7), the robot is actively contributing towards the completion of the task, like a leader and the human is mainly supporting the weight of the object.

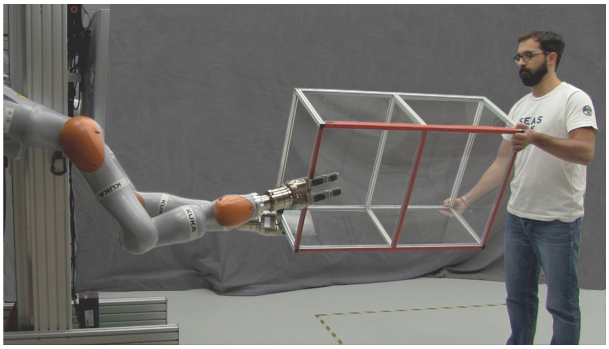
In terms of the motion plans generated for these two scenarios, the planned forces and the timings of the grasp changes differ. However, during the execution of these collaborative tasks such changes are subtle and they are generally difficult to identify. In order to assist the understanding of the reader, let us consider a handshake between two humans. For a third human observing the handshake it is generally difficult to identify who leads and who follows the handshake. This is also the case for collaborative tasks. Hence, according to our understanding and empirical experience we believe that the human policy model is adequate to represent a follower human partner. However, in scenarios that the human leads, although the robot policy is



(a)



(b)



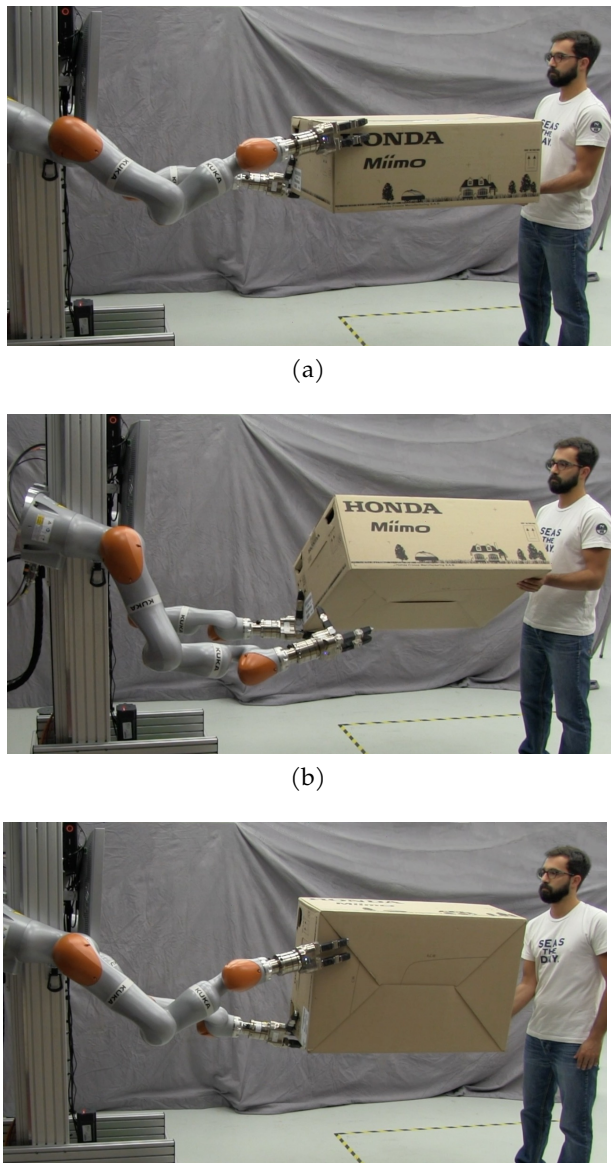
(c)

Figure 4.6: Keyframes of a 90° box (10 Kg) rotation DcM scenario. The human and the robot jointly complete the task. (a) Initial configuration. (b) Contact change by the right arm. (c) The left arm has changed contact and the task is completed.

accordingly tailored, the robot should also be able to react to any variations of the human's behavior in real-time.

4.5 DISCUSSION

In Chapter 4 we provide the details on how to realize joint action with a TO framework, based on the assumption that an estimate of the partner's policy exists. Our method computes the optimal hybrid policy for the robot to complete manipulation tasks as a member of a dyad. The concept only assumes roughly known model of the



(a)

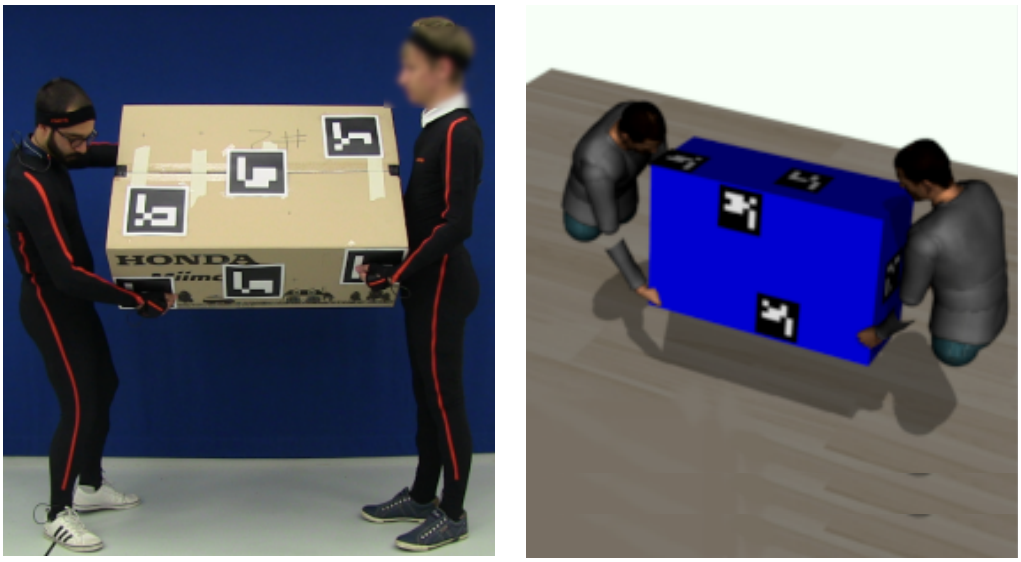
(b)

(c)

Figure 4.7: Keyframes of a 90° box (1.5 Kg) rotation DcM scenario. The human and the robot jointly complete the task. (a) Initial configuration. (b) Contact change by the right arm. (c) The left arm has changed contact and the task is completed.

partner's policy, a parametric representation of the the object and the pre-specified number of contact changes (*structure of the motion*). Our approach belongs to a family of methods that can generate *planned coordination* between the robot and the human partner. A characteristic of this family of methods is the use of a representation for the policy of the partner, which allows to anticipate the future effects of the partner's actions. To realize this in practice, the policies of the individuals are often treated separately.

Separate treatment has been extensively motivated in the introduction of this chapter and in the discussion section of Chapter 2. Yet, according to the work of Knoblich,



(a) (b)

Figure 4.8: Two humans manipulate a box. In (a) the data-recording is realized using two `Xsens` suits(see Appendix D) and the `ArUco` tracking library (Garrido-Jurado et al., 2014). (b) Replay of the collected data in a digital twin version of the real-world setup using `Robot Control Software (Rcs)`.

Butterfill, and Sebanz, 2011, there exists another type of coordination, namely *emergent coordination*. Hence, one should investigate methods towards *emergent coordination* too, within the scope of DcM scenarios.

To this front, we performed a preliminary exploratory study of a data-driven grasp-hold planner. This planner is essentially a reactive policy that maps the agent's actions to the actions of the partner. This study included: (i) data collection of a human dyad during a collaborative manipulation task, which necessitates grasp-hold changes from both individuals, e. g. box rotation as the one shown in Fig. 4.8a, (ii) processing the collected data to extract the relevant features of the task using a digital twin model in `Rcs` as shown in Fig. 4.8a and (iii) use of the regression version of a GMM, namely *Gaussian Mixture Regression* (GMR), to predict the contact locations of the agent given the observed contact location of the partner. This can be formally described as

$$p(\mathbf{c}^{L,a}, \mathbf{c}^{R,a} | \mathbf{c}^{L,p}, \mathbf{c}^{R,p}), \tag{4.4}$$

where each $\mathbf{c}^{\kappa,j} \in \mathbb{R}^3$ denotes the contact location used by an individual with $\kappa \in \{L, R\}$ and $j \in \{p, a\}$. More technical details and results of this preliminary study can be found in Appendix C.

During this elementary study, we identified a number of key challenges that could impede the applicability of such an approach into DcM setups. These are: (i) encoding physical conditions and constraints such that generalization to different configu-

rations is possible, e. g. encoding whether the environment is quasi-static or gravity is acting on the object, (ii) identification of an appropriate representation for the domain of the reactive policy, such that generalization between tasks, object, etc is attainable, (iii) collection of sufficient amount of demonstrations from human dyads that could represent a sufficient regime of emergent behaviors. These challenges could be approached; the first with a data-informed constrained optimization method (Vogt, Stepputtis, Jung, et al., 2018), the second with use of representations, such as the interaction primitives (G. Maeda et al., 2017; Campbell and Amor, 2017) and the third with incremental learning (Vijayakumar, D'souza, and Schaal, 2005). Yet, in non-periodic behaviors like manipulation, *emergent coordination* appears to be capable only of driving the system to a local equilibrium. In other words, it serves as reactive controller that can compensate for deviations around the nominal behavior. Thus, our primary focus was turned on developing a *planned coordination* that can anticipate the partner's behavior and jointly plan the actions of the robot.

Last but not least, during the robot experiments we identified the usefulness of adapting the hybrid motion plans on-the-fly. This capability is essential in Human-Robot DcM setups, as the human behavior is usually non-stationary. Furthermore, in the future we could integrate the proposed method (*planned coordination*) with a reactive feedback-based controller (*emergent coordination*), that would enable us to realize more robust behavior during the interaction.

5

ONLINE ADAPTATION DURING DYADIC COLLABORATIVE MANIPULATION

> It is not the strongest of the species that survives, nor the most intelligent. It is the one that is most adaptable to change.
>
> Charles Darwin

The two previous chapters form the basic components towards computing robot plans that are optimal, partner-aware and have broad action space, e. g. hybrid actions. Yet, another key attribute observed in collaborative setups, such as human dyads, is adaptation (Ajoudani et al., 2017). In this chapter, our focus is online motion adaptation of the robot agent's behavior. To achieve this, we build on the framework described in the two previous chapters. First, regarding the hybrid plan generation, we relax the condition on the fixed *structure of the motion* (see Chapter 3), which further broadens the repertoire of agent robot actions, and second, we achieve fast planning computation times that enable on-the-fly adaptation of the robot hybrid plans with respect to changes in the dyadic setup, e. g. task goal of the dyad.

The *structure of the motion* defines the mode sequence of the contact schedule.

Adaptation and awareness are aspects that affect humans social behavior and are part of our everyday life (Marsh, Richardson, and R. C. Schmidt, 2009). For example, this can be intuitively understood, if we consider that the presence of another person extends or restricts one's own action possibilities, and so one's own behavior adaptation is only just. Adaptation in human teams has been extensively studied and it has been shown to be one of the key success factors of collaboration (Converse, J. Cannon-Bowers, and Salas, 1993; J. A. Cannon-Bowers and Salas, 1998; Salas, J. A. Cannon-Bowers, and Blickensderfer, 1993). The shared mental models, i. e. models that represent the behavior of the partner (see Chapter 4), serve as the underlying mechanism of adaptability and can significantly improve team performance (Mathieu et al., 2000). J. Shah and Breazeal, 2010 investigated the way team members adapt their action plans, using both verbal and nonverbal cues. At the same time, the level of adaptation depends on a number of other factors (LePine, 2005), such as goal difficulty and members' willingness to change, etc. Moreover, competent human teams exploit awareness of the task and partner's state. Seamless coordination is achieved

by monitoring the partner's actions and adapting one's own actions based on these observations (Sebanz, Bekkering, and Knoblich, 2006).

Humans not only adapt their behavior when they partner with others, but they also adapt various aspects of their motion in single agent tasks. A number of human motor control works have studied different aspects of human motion under changing or uncertain environment conditions. Lackner and Dizio, 1994 and Franklin et al., 2007 analyzed adaptation of humans' arm trajectories and impedance characteristics to force perturbations. Tee et al., 2004 proposed a computational model that can describe the geometrical characteristics of human impedance and how these are changed during arm movements. In (Burdet et al., 2001), the authors investigated the way humans learn unstable dynamics (perturbations) and show that the adaptation of the Cartesian stiffness is directional. A computational model of the joint impedance modulation was introduced by (Mitrovic et al., 2010) based on stochastic optimal control and internal model uncertainty.

Naturally, adaptation has been studied in robotics too, both in tasks where the robot acts alone and in collaboration with a human. One of the main aims in the area of exoskeletons is adaptation. Typically, reference motion trajectories or postures of the human are optimized online based on real time interaction measurements, such as positions, forces and torques (Jezernik, Colombo, and Morari, 2004) and according to the principle of automatic reference motion adaptation (Riener, 2013). Other strategies found in the exoskeleton literature include Assitance-As-Needed (Wolbrecht et al., 2008), tunnel-controllers (Duschau-Wicke et al., 2010; Banala, Agrawal, and Scholz, 2007) and transparency (Kao and Ferris, 2005; Vallery, Duschau-Wicke, and Riener, 2009a; Vallery, Duschau-Wicke, and Riener, 2009b). Similarly, in more general *Human-Robot Interaction* (HRI) tasks adaptation plays a crucial role and has been one of the main areas of research (Goodrich and A. C. Schultz, 2008). For example, bounded memory models and game theoretic ideas have been utilized to demonstrate mutual-adaptation in *Human-Robot Collaboration* (HRC) tasks (Nikolaidis, Kuznetsov, et al., 2016; Nikolaidis, Nath, et al., 2017), such as table-carrying and table-cleaning; adaptive handover motions (C.-M. Huang, Cakmak, and Mutlu, 2015; Vogt, Stepputtis, Jung, et al., 2018); adaptive co-manipulation (Thobbi, Gu, and Sheng, 2011; Rozo Castañeda et al., 2013; Ghadirzadeh et al., 2016); and reaching motions (Mainprice and Berenson, 2013; Mainprice, Hayne, and Berenson, 2015). Furthermore, a number of ideas have been transferred from the human motor control domain into robotics. In (Ganesh, Albu-Schäffer, et al., 2010), the authors present an adaptive control architecture that exhibits in one *Degree of Freedom* (DoF) robot, force and impedance adaptation that is similar to a computational human motor control model. The au-

thors later extended their work to a 7-DoF arm, presenting human-like force and impedance adaptation characteristics (C. Yang, Ganesh, et al., 2011).

Our approach to achieve adaptation within a human-robot dyad is based on computing the robot motion plans in online rates. This in turn enables adaptation of the robot actions with respect to changes of the dyadic setup or the partner's behavior. The rest of the chapter is organized as follows. First, we briefly introduce the background formulation of discrete search methods and optimization motion planning. Second, we present work in this area. Right after that, we describe our formulation that combines *graph search* (GS) methods with continuous optimization motion planning methods. Next, we provide the details on the inner working of the method and present the evaluations and experimental results. Finally, we discuss promising research directions and conclude this chapter.

5.1 BACKGROUND

5.1.1 *On the multi-fidelity of motion generation*

Robotics literature is broadly inspired by nature, in part due to the fact that animals are capable of solving complex problems (Wimpenny et al., 2009). Although, we have not to date been able to fully uncover the crux of the problem solving process, there seems to be consensus that humans utilize different levels of abstraction (Mayer, 1992; Y. Wang and Chiew, 2010).

In our attempt to develop artificial agents that are capable of addressing complex problems, roboticists have often used the same rational in the core of motion generation methods. This involves some sort of structure that can be based on abstractions, hierarchies, decompositions, etc. Hierarchies have been explored in robotics to solve multiple kinematic (Slotine J J, 1991) and dynamic (Escande, Mansard, and Wieber, 2014) tasks and are still utilized in a number of different setups, such as motion generation for quadrotors (Liu et al., 2018) and legged robots (Carpentier et al., 2016), as well as in Hierarchical Reinforcement Learning (Barto and Mahadevan, 2003). Recently Konidaris, 2019 argued that an appropriate and problem-specific abstraction is required in order to solve many real world tasks and proposed an MDP-based approach to reason about abstractions.

In the meantime, there has been a large body of work combining task and motion planning (TAMP), either with problem-specific (Hauser, 2010) or more general implementations (Srivastava et al., 2014; Plaku and Hager, 2010). The general idea is based on the combination of symbolic planners with geometric planners. Yet, TAMP has also been approached with a hierarchical approach (Kaelbling and Lozano-Pérez,

2011) and as a constraint satisfaction problem (Lozano-Pérez and Kaelbling, 2014). In the latter, the notion of an action skeleton was proposed and was utilized in (Toussaint, 2015; Toussaint, Allen, et al., 2018) to describe a skeleton of physical interactions that forms a path optimization problem with differentiable constraints.

In this chapter, we are also bringing together two motion generation methods—GS and *Trajectory Optimization* (TO)—to solve the hybrid motion planning problem (see Chapter 3) in a *Dyadic collaborative Manipulation* (DcM) setup (see Chapter 4). This approach benefits from the best of both worlds, that is: (i) computational efficiency, (ii) exploitation of domain knowledge, (iii) high accuracy and (iv) physical reasoning. Next, we provide a brief background on GS methods and a reminder of TO theory.

5.1.2 *Graph search algorithms preliminaries*

Search algorithms are used to find paths within graphs (Hart, Nilsson, and Raphael, 1968). A graph $G = (V, E)$ is described with a set of nodes V and set of directed edges E. Each edge $e^{pq} \in E$ denotes a directed link between node $v^p \in V$ and $v^q \in V$, where v^q is a *successor* of v^p. In the context of search algorithms, each edge e^{pq} has an associated cost $c^{pq} > 0$, and the graph can be obtained given a set of initial nodes $\{v^\iota\} \subset V$ and a successor operator Γ. Γ is defined on the set V, and when applied on a node v^ι provides all its directed edges to *successor* nodes with the respective costs. Finally, given a node v^s and the successor operator Γ, a subgragh $G^s = (V^s, E^s)$ can be constructed such that all the nodes of G^s define the *accessible* set of nodes from node v^s.

Graph search algorithms address the problem of finding the optimal path[1] $\mathbf{v} = \{v_0, ..., v_\mathcal{N}\}$ of length $\mathcal{N} \in \mathbb{N}$ from a start node v^s to a set of goal nodes $\{v^g\}$, expressed as

$$\min_{\mathbf{v}} \quad \mathcal{C}(\mathbf{v}) \tag{5.1a}$$

$$\text{s.t.} \quad v_{j+1} = f_\gamma(v_j, e_j) \tag{5.1b}$$

$$e_j \in \Gamma(v_j) \quad , \tag{5.1c}$$

$$v_0 = v^s \tag{5.1d}$$

$$v_J \in \{v^g\} \tag{5.1e}$$

1 We use superscripts to index nodes and edges in a time agnostic fashion and subscripts to index the nodes in the optimal path sequence.

where $\mathcal{C}(\mathbf{v}) \in \mathbb{R}$ is the total cost along the path, (5.1c) indicates the set of all directed edges starting from a given node, $f_\gamma(\cdot)$ is the transition function responsible for computing the next node in the sequence, and (5.1d), (5.1e) specify the initial node and the set of final nodes, respectively (Hart, Nilsson, and Raphael, 1968). The specific characteristics of $\mathcal{C}$, Γ, f_γ in our problem and the representation of v are discussed in Section 5.3.1. These methods compute efficiently a sequence of transitions, i. e. the *structure of the motion*. Yet, they neglect details of the actual continuous motion through transitions.

5.1.3 *Trajectory optimization preliminaries*

Here, we reminder the reader the main formulation used in TO methods, to facilitate the direct relation of our method. This is the following:

$$\min_{\boldsymbol{x}(t),\boldsymbol{u}(t)} \int_0^T c\left(\boldsymbol{x}(t),\boldsymbol{u}(t)\right) dt + c_f\left(\boldsymbol{x}(T)\right) \tag{5.2a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t),\boldsymbol{u}(t)\right) \tag{5.2b}$$

$$\boldsymbol{x}(0) \in \mathbb{X}_0 \quad , \tag{5.2c}$$

$$\boldsymbol{x}(t_f) \in \mathbb{X}_f \tag{5.2d}$$

$$\boldsymbol{g}\left(\boldsymbol{x}(t),\boldsymbol{u}(t)\right) \in \mathbb{Z} \tag{5.2e}$$

$$t \in [0, T] \tag{5.2f}$$

where $\boldsymbol{x} \in \mathbb{R}^\eta$ is the model's state vector, η is the dimensionality of a system's state, $\boldsymbol{u} \in \mathbb{R}^\iota$ is the model's control vector, ι is the dimensionality of a system's control, $c(\cdot), c_f(\cdot) \in \mathbb{R}$ in (5.2a) are the running and final cost functions, $\boldsymbol{f}(\cdot) \in \mathbb{R}^{\eta+\iota}$ in (5.2b) describes the dynamics of the system, $\boldsymbol{g}(\cdot) \in \mathbb{R}^\zeta$ in (3.1e) describes the equality and inequality constraints of the system, $\zeta \in \mathbb{R}$ is the total number of constraints and (5.2c) to (5.2f) describe bounds on the initial state, final state, path constraints and motion duration, respectively. For more details on TO methods, the reader is referred to Section 3.1.1. These methods are used to compute efficiently continuous motion plans through *discontinuities*, and typically require a proper initial seed.

5.2 Related work on manipulation planning

5.2.1 *Sequential manipulation planning*

On a different line of work, Siméon et al., 2004 employed probabilistic roadmaps to produce motion plans with multiple grasp-hold changes. King et al., 2015 used

Monte Carlo Tree Search to plan sequences of discrete pushes and reason about object interactions. The A* algorithm was used by Gienger, Ruiken, et al., 2018, to demonstrate DcM scenarios with a human and a robot. Graph-based planning has also been proposed by Chen, Figueredo, and Dogar, 2020, to enable robots choose stable grasps on the object during forceful collaborative tasks, where a human drills or cuts an object held by a robot. These methods discretize the state space to employ search algorithms. Furthermore, combinations of these sampling or symbolic-based methods with motion planners have been realized (Srivastava et al., 2014). Nevertheless, to the extend of our knowledge, these methods are limited to kino-dynamic planning and have not been applied in hybrid problems, where the generation of dynamically feasible plans is of core importance.

5.2.2 *Hierarchical approaches*

These approaches address hybrid problems by decomposing them into action planning (Winkler, Mastalli, et al., 2015), contact planning (Tonneau et al., 2018), and motion control (Carpentier et al., 2016). Such hierarchies allow to exploit domain knowledge at the task planning level and have been used for online motion generation. Yet, as these elements are designed separately the final solution is often not optimal, or sometimes not even feasible.

5.2.3 *Mixed-integer programming*

This formulation explicitly models the hybrid nature of the problem and has been used by (Deits and Tedrake, 2014; A. K. Valenzuela, 2016; F. R. Hogan and Rodriguez, 2020) for both locomotion and planar manipulation. Yet, mixed-integer methods need to explore both the continuous and discrete parts of problems, while reasoning for the discrete part is done using general combinatorial optimization methods like *Branch and Bound*. This typically leads to large computation times that can be prohibitive for DcM needs.

5.2.4 *Logic geometric programming (LGP)*

LGP with physics synthesizes logical planning with optimal control to demonstrate a broad range of sequential robot manipulation planning capabilities. LGP has also been used for multi-agent cooperative manipulation tasks (Toussaint and Lopes, 2017), such as handovers. The cooperative aspects are limited to the kinematic domain,

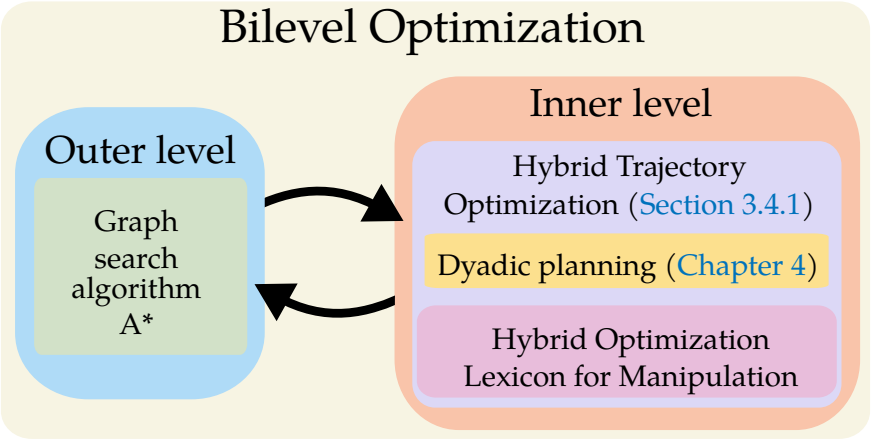


Figure 5.1: Overview of the methods@ optimized paths are obtained through an iterative execution of the outer (discrete) and inner (continuous) levels of the bilevel optimization.

where both actors act synchronously, but their actions are not physically coupled. In parallel to LGP with physics, the proposed method combines the benefits of informed GS algorithms and optimal control formulations. Informed search methods such as A* are a more efficient special case of the *Branch and Bound algorithm* (Nau, Kumar, and Kanal, 1984), which makes them well suited for online motion adaptation during DcM tasks. Toussaint's work (Toussaint, Allen, et al., 2018) exhibits creative solutions for manipulation puzzles in simulation, while our method considers dynamic aspects of the dyadic interaction and enables online re-planning.

5.3 ON-THE-FLY ADAPTATION WITH BILEVEL OPTIMIZATION

In this section, we provide the core computational formalism, that enables on-the-fly generation of hybrid motion plans, both for single agent manipulation planning and for joint manipulation planning in dyads. First, we describe how GS algorithms can be formally combined with TO methods. The former forms the outer level and the latter the inner level of the bilevel optimization. Next, we present the details of these outer and inner levels. The schematic shown in Fig. 5.1 illustrates the interplay between the outer and inner level, and reveals the nested structure of the inner level.

BILEVEL FORMULATION The aim of this formalism is to provide the means to generate hybrid trajectories, like the one shown in Fig. 3.1, without pre-specifying the *structure of the motion*. Motivated by the key observations mentioned in Section 5.1 and inspired by the bilevel method presented in (Nakanishi, Radulescu, D. J. Braun, et al., 2016) as well as the "Mixed-Logic Program" (Toussaint, Allen, et al., 2018),

we combine the two formulations presented in (5.1) and (5.2) into a single bilevel optimization formulation, as follows:

$$\min_{\mathbf{v}} \quad \mathcal{C}(\mathbf{v}) \tag{5.3}$$

$$\text{s.t.} \quad v_0 = v^s \tag{5.4}$$

$$\boldsymbol{x}(v_J) \in \mathbb{X}_N(\theta^p) \tag{5.5}$$

$$e_j \in \Gamma(v_j, \theta^p) \tag{5.6}$$

$$v_{j+1} \in \left\{ \begin{array}{rl} \underset{\boldsymbol{x}(t),\boldsymbol{u}(t)}{\arg\min} & \displaystyle\int_0^T c(\boldsymbol{x}, \boldsymbol{u}, e_j) dt \\ \text{s.t.} & \dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, e_j, \theta^p\right) \\ & \boldsymbol{x}_0 \in \mathbb{X}_0(v_j), \\ & \boldsymbol{x}(T) \in \mathbb{X}_f(e_j), \\ & \boldsymbol{g}\left(\boldsymbol{x}, \boldsymbol{u}\right) \in \mathbb{Z}(e_j), \\ & t \in [0, T] \end{array} \right\}. \tag{5.7}$$

The outer level of the optimization is described with equations (5.3) to (5.6) and is responsible to construct the *structure of the motion*. This is achieved by performing a discrete search using the GS method, as shown in Fig. 5.1. The inner level of the optimization is described with the TO problem (5.7) and its role is to compute the continuous trajectories, such that the discrete transitions can be realized. Equations (5.3) and (5.4) are identical to (5.1a) and (5.1d), however the discrete transition function f_γ described in (5.1b) is now replaced by (5.7), which denotes a nonlinear continuous optimization problem of the form (5.2). Additionally, to account in the computational formalism for the dependency of the solution to the current mode of the partner's policy—as denoted by (2.2) and (2.4)—we modified (5.5), (5.6), and (5.7) such that they depend on the parameters θ^p. More details on the dyadic planning can be found in Chapter 4.

OUTER-INNER LEVEL INTERPLAY First, the outer level computes a discrete sequence of states, which define the initial *structure of the motion* towards the goal. Second, each segment of the motion is passed on and is optimized by the inner level. One or more of these segment may be altered by the inner level, resulting in a modification of the initial *structure of the motion*. Consequently, the discrete sequence of states—subsequent to the modified segment—might become obsolete and therefore required to be re-computed by the outer level. This third step is closing the bilevel

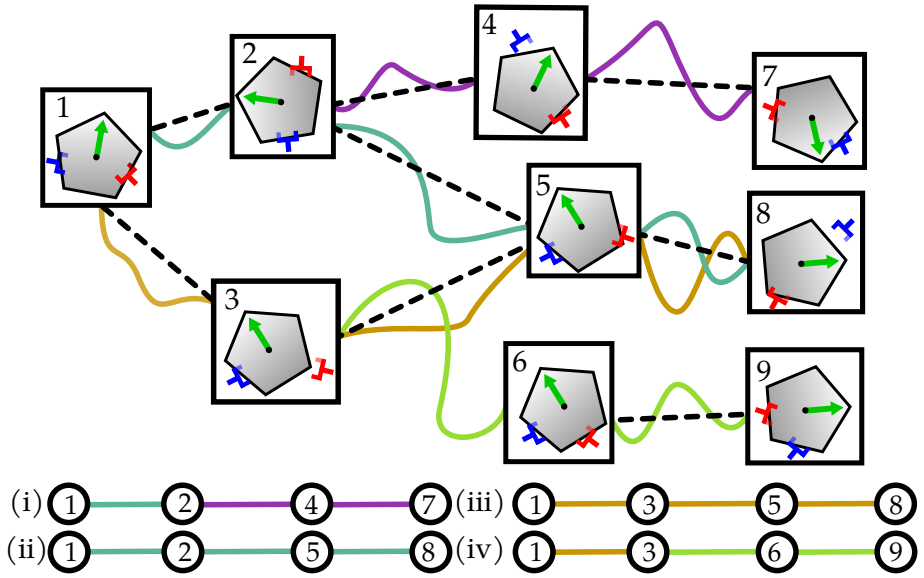


Figure 5.2: A representative illustration of the of four different solution paths (i) to (iv) obtained with the proposed bilevel optimization method. The dashed lines depict the discrete transition found from the outer (discrete) level of the optimization, whereas the full lines are the continuous segments obtained from the inner (hybrid) level of the optimization. All four paths start from the same initial node with index 1. Solution path (i) ends at node 7. Solution paths (ii) and (iii) both end at node 8 although they are different. In particular, path (ii) will be generated when experiencing a change of goal from final node 7 to node 8. Similarly, paths (iii) and (iv) end at different nodes that are identical with respect to the task, if we only observe the state of the object. An interesting point is the alternation of the transition from $e^{3,5}$ to $e^{3,6}$ by the inner (hybrid) level optimization, which results in a new path from node 6 to the goal.

loop. By running these three steps iteratively, the bilevel optimization converges to the goal in a sliding window fashion. In Fig. 5.2, we illustrate the bilevel nature of the method with four examples.

5.3.1 *Outer optimization level*

For the outer optimization level, we aim for a fast GS method to compute the *structure of the motion*, e. g. a sequence of contact changes and object motions. Given a discrete state representation, the state-space can be encoded into a graph, with each discrete state being a node v on the graph as described in Section 5.1.2. We use a coarse state representation that includes a discrete description of the object state y and contact locations of agent's end-effectors c^k. A node v on the graph corresponds to the tuple (y, c^l, c^r), where $y, c^l, c^r \in \mathbb{N}$ and v is defined as an index to the tuple with $v \in \mathbb{N}$. Fig. 5.3 depicts a viable 2D state discretization.

A key element of the GS algorithms is the *successor* operator Γ, defined in Section 5.1.2. Γ allows us to attain a low branching factor and perform graph expansion

more efficiently than brute-force node insertion (S. Wang and Hauser, 2018). We realize Γ for multi-contact manipulation and DcM scenarios specifically. We construct a simplified and intuitive physics model of the object-hand interactions based on the following rules (Gienger, Ruiken, et al., 2018).
Feasible states:

1. Left end-effector must always be on the left of the right end-effector.
2. A minimum distance between end-effectors is defined.
3. Applied forces have to be permissible given the contact location, as described in Section 3.4.1.4.
4. When both end-effectors are in contact, they must quasi-statically counteract gravity effects on the object CoM.
5. The pivoting torque spawned in scenarios with single contact must not violate a given threshold (DcM-specific).

Feasible transitions (*task-dependent*):

i) Both end-effectors must be in contact to rotate the object.

ii) A single or both end-effectors can change contact within one transition.

Rules 1, 2 and i are realized based on a mapping from the discrete state to the continuous Cartesian space of end-effectors. Rules 3 and 4 are computed based on quasi-static principles which are configuration dependent, typically used in grasping literature (Roa and Suárez, 2014). Further, rule 5 reserves as an implicit upper bound on the required torque the partner has to apply, counteracting the pivoting torque applied by the agent as states with high torques are not allowed.

Regarding the particular choice of GS method we use a heuristic A* algorithm for the following two reasons. First, the A* algorithm is considered a special case of Dynamic Programming (DP) (Bertsekas, 1995; LaValle, 2006). Thus, the solution of our overall problem is obtained by a bilevel optimization process. Second, A* is known for its computational efficiency, as it exploits heuristics to achieve a very low effective branching factor.

A* constructs an optimal sequence of states in terms of the evaluation function $c(\cdot) = g(\cdot) + h(\cdot)$. The cost term $g(\cdot) \in \mathbb{R}$ is obtained from edges costs (see Section 5.1.2), while the heuristic term $h(\cdot) \in \mathbb{R}$ needs to be admissible (always underestimate the actual cost) and monotonic to ensure that the solution path is optimal (Hart, Nilsson, and Raphael, 1968). To facilitate optimal composition of the solution paths shown in Fig. 5.2, the heuristic term $h(\cdot)$ needs to be designed in accordance

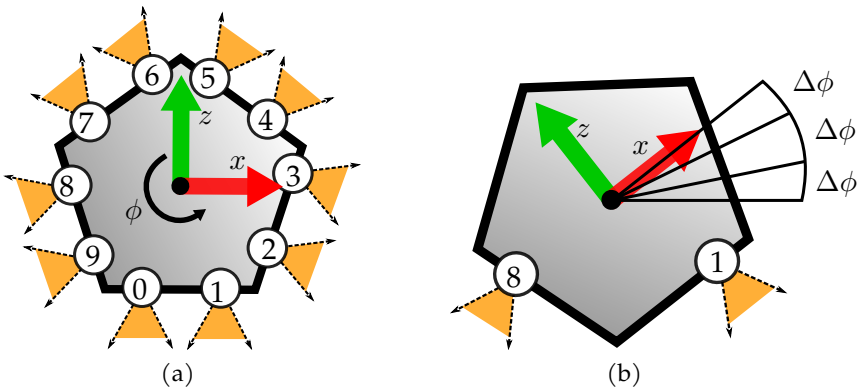


Figure 5.3: 2D illustration of an example state space representation of the outer (discrete) level. The numbers in circles denote contact points. (a) Discretization of the contact space. (b) Discretization of the object orientation (the translational part can be discretized with a checkerboard-like grid).

with the cost function $c(\cdot)$ of the inner optimization level in (5.7). The details of the heuristic and cost terms are given in Section 5.4.

The outer level provides the optimal *structure of the motion* efficiently, as well as an initial guess for the inner optimization level by converting the optimal sequence of states to continuous trajectories using fifth-order-polynomials (Gienger, Ruiken, et al., 2018).

5.3.2 *Inner optimization level*

The inner level (5.7) is responsible for optimizing the hybrid path (see Fig. 3.1), given the *structure of the motion*. Here all motion-relevant quantities (see Section 3.1.3) are optimized within their continuous manifold, while the discretized description of the quantities (see Section 5.3.1) serves as a basis for the initial seed of the continuous problem. For example, contacts $\mathbf{c}_i^\kappa$ are optimally selected from the entire object surface, not only from the discrete contact locations shown in Fig. 5.3.

As it has been reported in (Stouraitis, Chatzinikolaidis, et al., 2018; Toussaint, Allen, et al., 2018; Winkler, Bellicoso, et al., 2018; Mordatch, Todorov, and Popović, 2012; Posa, Cantu, and Tedrake, 2014; Dai, A. Valenzuela, and Tedrake, 2014), the computational times of optimizing the full path at once are extensively large for any type of online motion adaptation. Furthermore, the non-convex nature of the problem gives no global optimality guarantees. Thus, to address the computational efficiency challenge, we propose to optimize each segment of the motion separately. To realize this, we introduce a decomposition of general hybrid motion into a set of hybrid motion primitives, referred as *Hybrid Optimization Lexicon for Manipulation* (HOLM). Fig. 5.4a

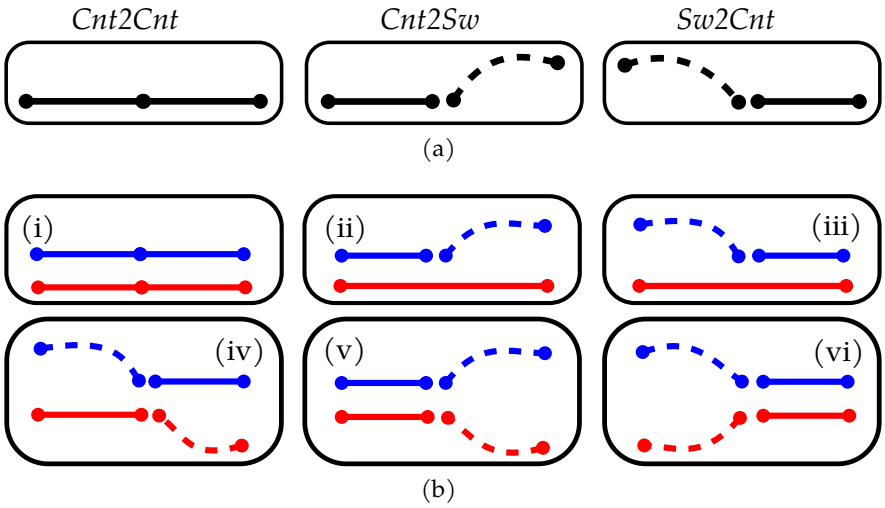


Figure 5.4: The set of primitives referred as HOLM. Dashed lines denote free-motion phase and full lines denote in-contact phase. (a) Three primitives for a single end-effector. (b) Six bimanual primitives, where the left end-effector is colored blue and the right is red. For primitives (ii) and (iii) end-effectors can be switched, such that the left (blue) remains in contact and the red performs a change of grasp-hold. Similarly, (iv) can be symmetrically switched.

shows the primitives for a single end-effector and Fig. 5.4b illustrates a few combinations of HOLM primitives for bimanual agents.

In contrast to (Al Borno, de Lasa, and Hertzmann, 2013), where the hybrid motion is chopped into spacetime windows with fixed contact configuration and time duration, we choose to build each primitive as a sequence of two *contact-invariant phases* (see in Section 3.1.3) of variable time duration. The primitive *Cnt2Cnt* has two consecutive contact phases, the *Cnt2Sw* has an *in-contact* phase followed by a *free-motion* phase—where the grasp-hold change starts—and the *Sw2Cnt* has a *free-motion* phase followed by an *in-contact* phase, where the grasp-hold change is completed. A single *free-motion* primitive does not contribute to the task, thus every *free-motion* phase is accompanied by an *in-contact* phase. Hence, each segment of the motion is optimized including the critical transitions of making and breaking contact (*discontinuities*) to anticipate the contact configuration of the next phase. The transition from one HOLM primitive to the next one does not require special treatment as the contact configuration is not altered.

Regarding the collection of primitives used, *Sw2Cnt* and *Cnt2Sw* form the the minimal set of making-breaking contact, while *Cnt2Cnt* is used to maintain contact. This is particularly useful when the robot agent rotates the object towards the goal. The use of the *Cnt2Cnt* primitive is encouraged with rule (a) described in Section 5.3.1. In general, this set of primitives allows to fine-tune the hybrid robot motions to be legible (Dragan, K. C. Lee, and Srinivasa, 2013).

The inner level accomplishes very fast optimal hybrid motion plan generation, given the *structure of the motion*. To the extend of our knowledge, this in turn empowers the bilevel optimization to be the first on-the-fly re-planning capable hybrid optimization method. This allow us to demonstrate online hybrid policy adaptation with respect to non-stationary dyadic interactions.

5.4 EVALUATION AND EXPERIMENTS

In this section, we first provide computational evaluations of the proposed method. We proceed with simulations on both a single agent and a dyadic setup. Last, we evaluate the proposed method with real-world DcM experiments. We refer the reader to the attached video material of the simulations and the human-robot experiments during DcM tasks. A video showing the simulations and experiments performed is provided (see SV3 in Appendix E).

The purpose of the computational study is to highlight the computational gains of HOLM, and emphasize the importance of specific algorithmic choices. The objective of the dyadic simulations present a multitude of situations where the resulting hybrid policy of the agent is conformed to the partner's policy. The experiments with a human-robot dyad demonstrate the viability of our method to synthesize on-the-fly under real-world conditions.

PARAMETERS OF EXPERIMENTAL SETUP The state of the object is $\mathbf{y} = \begin{bmatrix} x & z & \phi \end{bmatrix}^T$ with task dimension $\nu = 2$, which is sufficient for the demonstrations; however, both levels (Section 5.3) can be realised in 3D space with $\nu = 3$, e. g. the inner level can be modified based on our work (Chatzinikolaidis et al., 2018). To obtain the discrete state (y, c^l, c^r) mentioned in the outer level (Section 5.3.1), we only need to consider ϕ and the contact locations, as the translational components of $\mathbf{y}$ do not affect the *structure of the motion*. ϕ is discretized with $30°$ resolution and for each of c^l and c^r we specify 16 contact locations. With these choices and with the rules defined in Section 5.3.1 the branching factor for a brute-force search method is $b \approx 23$. Yet, the A* algorithm uses heuristics to guide the search, thus the average effective branching factor for our setup is $b^* \approx 4$, which is the key for very efficient outer level computation times. Regarding the evaluation function C of A*, the heuristic term h models the angular difference between the current and goal rotation angles of the object, while the transition cost function g corresponds to the required movement length; shorter transitions in the continuous Cartesian space are cheaper. The cost function of the HOLM primitives similarly minimizes distance to goal and overall path length. With this setup—as discussed in Section 5.3.1—the resulting A* discrete solution sequence

Table 5.1: Average computation time of commonly used HOLM primitive types described in Fig. 5.4.

HOLM type	No. of variables	Exact Hessian	Limited-memory BFGS
Cnt2Cnt	134	34ms, 28 iter	75ms, 34 iter
Cnt2Sw	128	47ms, 38 iter	>6000ms, >3000 iter
Sw2Cnt	129	50ms, 29 iter	2800ms, 1535 iter
Cnt & Cnt	194	44ms, 25 iter	58ms, 26 iter
Cnt & Sw2Cnt	189	131ms, 70 iter	>8000ms, 3000 iter
Cnt & Cnt2Sw	188	49ms, 29 iter	2740ms, 1521 iter

is optimal and in accordance with the inner optimization level. The knot resolution used for the HOLM primitives is 6 knots per phase, and the friction cone is $\mu = 0.5$. For each HOLM primitive, we use an upper time bound of $T = 3.5s$ for in-contact phases and $T = 6.5s$ for free-motion phases. Once the hybrid motions are optimized in the task space, they are being mapped to the configuration space of the robot using *Inverse Kinematics* (IK) (Gienger, Toussaint, and Goerick, 2010).

Regarding the implementation details, we use `CasADi` (Andersson et al., 2018) to realize the HOLM primitives[2], where each primitive is a separate parameterizable hybrid problem. Each hybrid problem is a large and sparse nonlinear optimization problem which is solved using `IPOPT` (Wächter and Biegler, 2006), while the automatic differentiation capabilities of `CasADi` allow us to provide exact gradient and hessian information. The A* planner and the lower-level control aspects of the robot, e. g. IK, are implemented in the `Robot Control Software (Rcs)` framework[3]. All experiments are conducted on a 64-bit Intel Quad-Core i7 3.40GHz workstation with 16GB RAM.

5.4.1 *Computational evaluations*

5.4.1.1 ***HOLM computation times***

In Table 5.1, we present the average computation times for 15 runs of each HOLM primitive. Each primitive is evaluated on a variety of tasks, using three objects with different shape that is a sphere, a rectangular box, and a parallelogram box. The tasks involve translation from $0m - 1m$ and rotation from $0°$ - $180°$, similar to the actions shown in Figs. 4.2 and 4.3. The computational times reported are obtained with the

2 An open-source repository with our HOLM implementation can be found in: https://github.com/stoutheo/HybridManip/tree/HOLM-primitives

3 Information about `Rcs` can be found in: https://github.com/HRI-EU/Rcs

initial seed set to zero and they scale linearly with respect to the number of knots and the time horizon. These results reveal the computational benefits of HOLM.

5.4.1.2 *Bilevel optimization computation times*

In Table 5.2, we present the average computation times for the bilevel optimization. We group tasks in terms of angular distance from the initial state of the object to the goal, as this grouping nicely relates to the number of contact changes required to complete the task. As the number of contact changes depends on the initial contact configuration, a range of contact changes is given rather than an exact number (second column of Table 5.2). We also provide the approximate horizon of the resulting motion. These tasks are:

1. $0^\circ < \Delta\phi < 20^\circ$, with motion horizon $\sim$7 s,
2. $20^\circ < \Delta\phi < 120^\circ$, with motion horizon $\sim$20 s,
3. $120^\circ < \Delta\phi < 140^\circ$, with motion horizon $\sim$28 s,
4. $140^\circ < \Delta\phi < 200^\circ$, with motion horizon $\sim$71 s,
5. $200^\circ < \Delta\phi < 360^\circ$, with motion horizon $\sim$114 s.

We show the computation time required for the first segment of the motion, the average computation time for each of the consecutive segments (fifth and sixth column of Table 5.2). The former indicates the planning time until the receding horizon plan can be updated, while the latter specifies how fast the successive segments are computed. Computation times also include revising *structure of the motion*, and are proportional to the graph size displayed with the number of explored nodes (fourth column of Table 5.2). These evaluations showcase the online planning capabilities of the bilevel method.

5.4.1.3 *Discussion*

The main steps that allow us to improve the computation times from tens of seconds presented in our previous work (Stouraitis, Chatzinikolaidis, et al., 2018), to milliseconds for HOLM and few seconds for the bilevel optimization are: (i) decomposing the problem into HOLM primitives, which allows to keep the size of the hybrid problems small (second column of Table 5.1), (ii) exploring the hybrid structure of the problem with an efficient GS algorithm, (iii) formulating a sparse problem that can be efficiently solved[4], (iv) providing the exact Hessian using automatic differentia-

4 Interior-point methods are able to solve our specific problem more robustly than sequential quadratic programming methods.

Table 5.2: Computational evaluation of the bilevel optimization, and specifically the inner level, with respect to five different groups of tasks.

Task	No. of contact changes	No. of segments	Graph size	First segment time	Time per segment	Full path time (HOLM)	Full path time
i	0	1	1011	0.24s	0.24s	0.02s	0.02s
ii	1	3-4	11567	1.08s	0.45s	0.09s	0.30s
iii	1-2	4-6	18510	1.52s	0.53s	0.13s	0.75s
iv	2-3	6-8	47097	4.86s	1.07s	0.18s	1.30s
v	4-6	8-12	102329	8.23s	2.93s	0.25s	5.20s

tion, (third and fourth column of Table 5.1), and (v) selecting the end-effectors and permissible force representation discussed in Sections 3.4.1.4 and 3.4.1.5.

The seventh column of Table 5.2 shows the average computation times required to optimize the full continuous path using the HOLM primitives only for the inner optimisation level. First, as the HOLM primitives utilize the initial seed provided by the outer level (see Section 5.3.1), the computation times are much smaller than the ones in Table 5.1. Second, in the eighth column of Table 5.2 we provide the computation times (only inner level) needed to compute the full path using a hierarchical approach, as in (Liu et al., 2018). The comparison between the seventh and eighth column of Table 5.2 reveals the computational gain of using HOLM primitives with respect to the baseline approach (hierarchical)[5]. Further, by inspecting the relation between the second, the fifth and the sixth column of Table 5.2, we can observe that the computation times scale linearly with respect to the number of contact changes. This aspect of the method is particularly important in comparison to mixed-integer approaches (A. K. Valenzuela, 2016; F. R. Hogan and Rodriguez, 2020), which are prohibitively expensive as the number of discrete variables (contact changes) increases.

Finally, the success rate of the bilevel optimization depends on the selected discretization of the outer level. If a fine discretization is selected, an optimal solution is always found. However, this is achieved at the expense of computational efficiency. Therefore, we used a discretization of 30° that provides fast solutions and satisfying success rate. The inner optimization level has been empirically observed to provide robust solutions in terms of convergence, due to the appropriateness of the initial seed. This allows us to mitigate sensitivity issues with respect to the initial seed, which is a common drawback of continuous optimization methods. Furthermore, even if the

5 In addition to the computational gains, we also empirically observed that the solutions obtained when the full path is optimized are harder to regularize. This would hinder the realization of such solutions on real hardware. On the other hand, the solutions obtained with the HOLM primitives were easy to regularize, which is also reflected with the experiments presented in Section 5.4.3

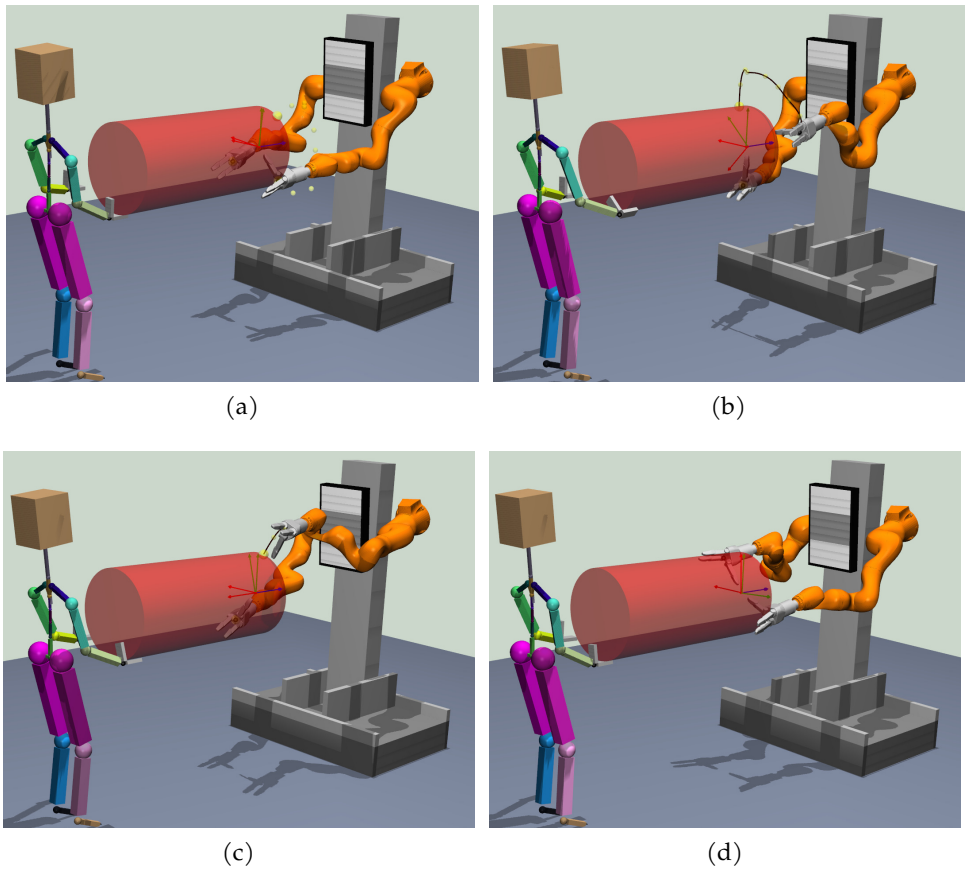


Figure 5.5: Keyframes of a rotational DcM task with $\mathbf{y}_N^* = 90°$ intended goal, where the partner is not properly supporting the object. (a) The left hand of the avatar (partner) is not in contact with the object. (b) The object is first rotated in the opposite direction to be properly supported by the agent's right hand. (c) The free-motion to change grasp-hold is performed. (d) The object is properly held and jointly rotated to the intended target.

inner level fails to converge, we can always use the interpolated trajectory obtained by the outer level.

The computation times presented demonstrate the online planning capabilities of our method. We gained approximately a $\times 10$ to $\times 50$ speedup in comparison to the results presented in our previous work (Stouraitis, Chatzinikolaidis, et al., 2018), while simultaneously the arms contact sequence pattern (*structure of the motion*) is automatically computed.

5.4.2 *Simulation experiments*

We present here a number of different motion plans generated by the proposed method that demonstrate the capability to find on-the-fly dynamic, and partner-aware solutions.

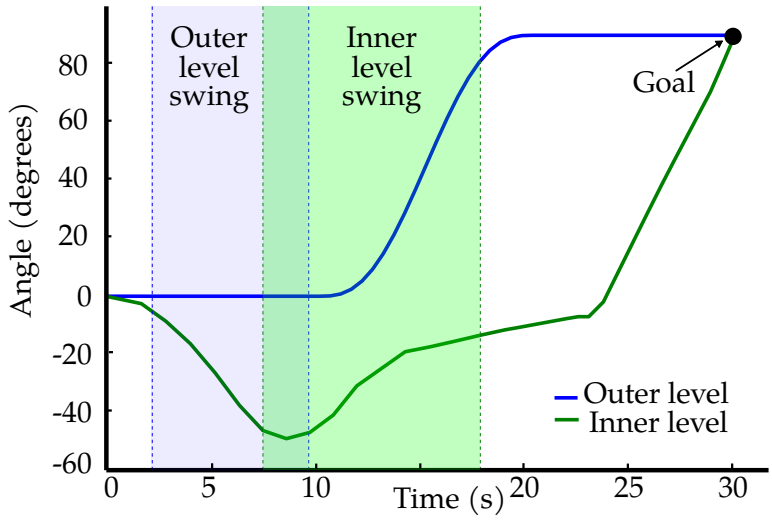


Figure 5.6: Evolution of the object orientation for the $90°$ DcM task shown in Fig. 5.5. The blue curve is the path computed from the outer level of the optimization, while the green is the final path optimized by the inner level. The shaded areas indicate the duration and temporal placement of the free-motion of the left end-effector. The inner level initially rotates the object opposite to the goal to satisfy the dynamic constraints of the task.

5.4.2.1 *Outer vs. inner level solutions*

With Fig. 5.6, we show the benefits of our method over solely search-based planning approaches (Gienger, Ruiken, et al., 2018). During this $90°$ object rotation DcM task, the human partner does not properly support the object, as shown in Fig. 5.5, where the avatar's (partner) left hand is not in contact with the object. In our partner model, this is represented through parameters $\mathbf{K}^{p,\phi} = 0$ and $\mathbf{D}^{p,\phi} = 0$ in (4.3). The search-based outer level provides a coarse solution (blue line in Fig. 5.6) that does not take into account the policy of the partner, while the inner level significantly alters the plan (green line in Fig. 5.6) to conform to the dynamic constraints of the task, i. e. jointly balance the object. This shows that the inner level significantly alters trajectories, durations, and action timings of the outer level solution, to respect dynamic aspects of the interaction.

5.4.2.2 *Online adaptation to alternations of the dyadic goal*

During this DcM scenario, the initial target orientation of $150°$ for the object changes to $-55°$, while the agent is not aware of this change in advance. The target of the object serves as a proxy to the partner's intention. This is realized by altering the goal $\left[\mathbf{y}_N^{*\,T} \quad \dot{\mathbf{y}}_N^{*\,T}\right]^T$ during the interaction shown in Fig. 5.7. In Figs. 5.8a to 5.8c we show the angular state evolution of the object and the two end-effectors. Once the change of dyadic goal occurs, replanning is completed in $0.95s$ for the first segment of the receding horizon plan. The consecutive segments are adapted in $1.13s$. This

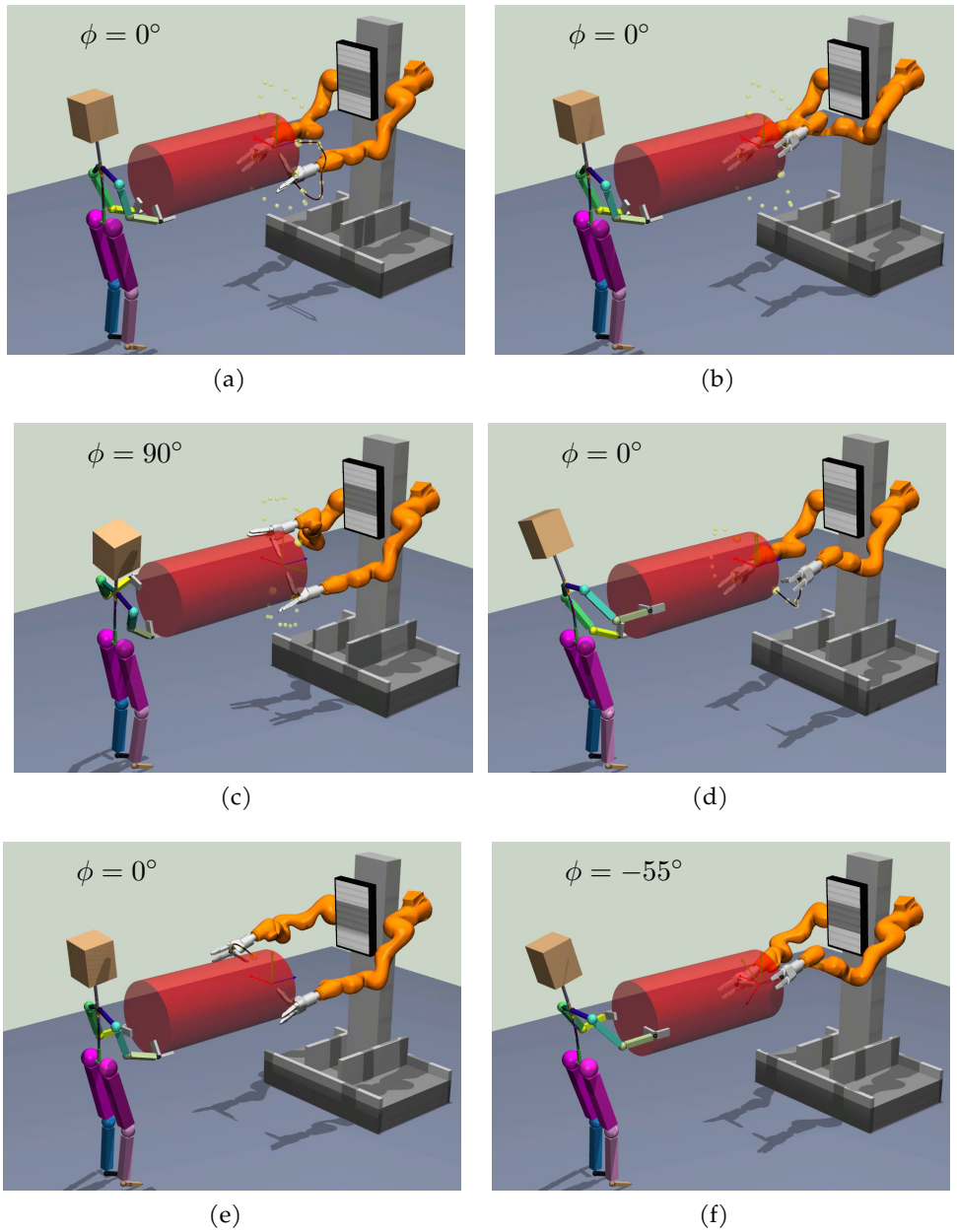


Figure 5.7: A sequence of frames of a non-stationary DcM scenario. The orientation of the object is given at the top left corner of every keyframe. The initial joint goal is to rotate the object to $150°$; keyframes (a) and (b) show the hybrid plan and the early execution steps for achieving this joint partner-agent goal. However, in between (b) to (c) the joint intended goal changes to rotate the object to $-55°$. This causes an on-the-fly adaptation to a new hybrid motion plan in (c). Keyframes during the execution of the adapted plan are shown in (d), (e) and (f).

illustrates that our method can adapt on-the-fly trajectories, action timings, durations, the *structure of the motion* and contact locations to respond to real-time changes of the dyadic task.

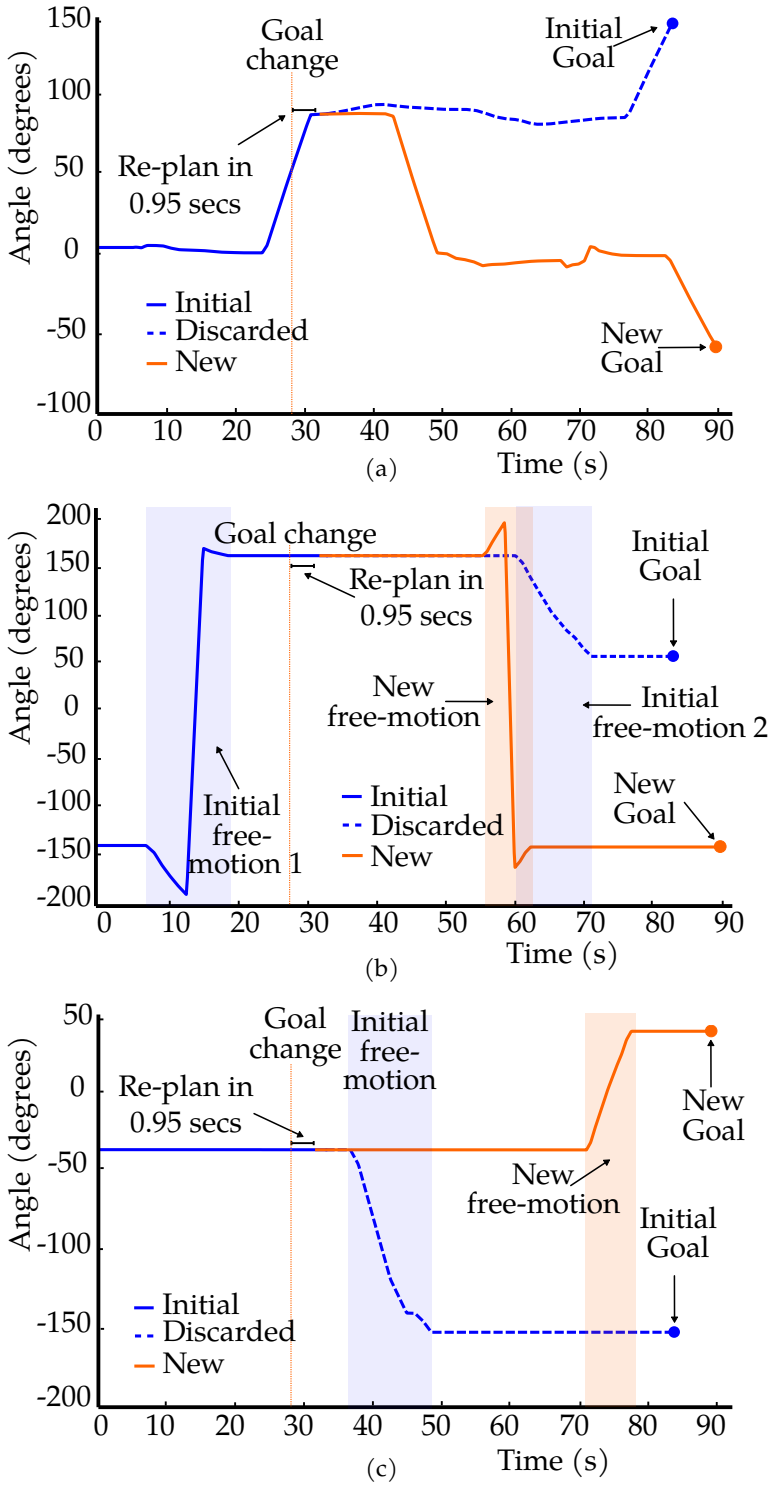
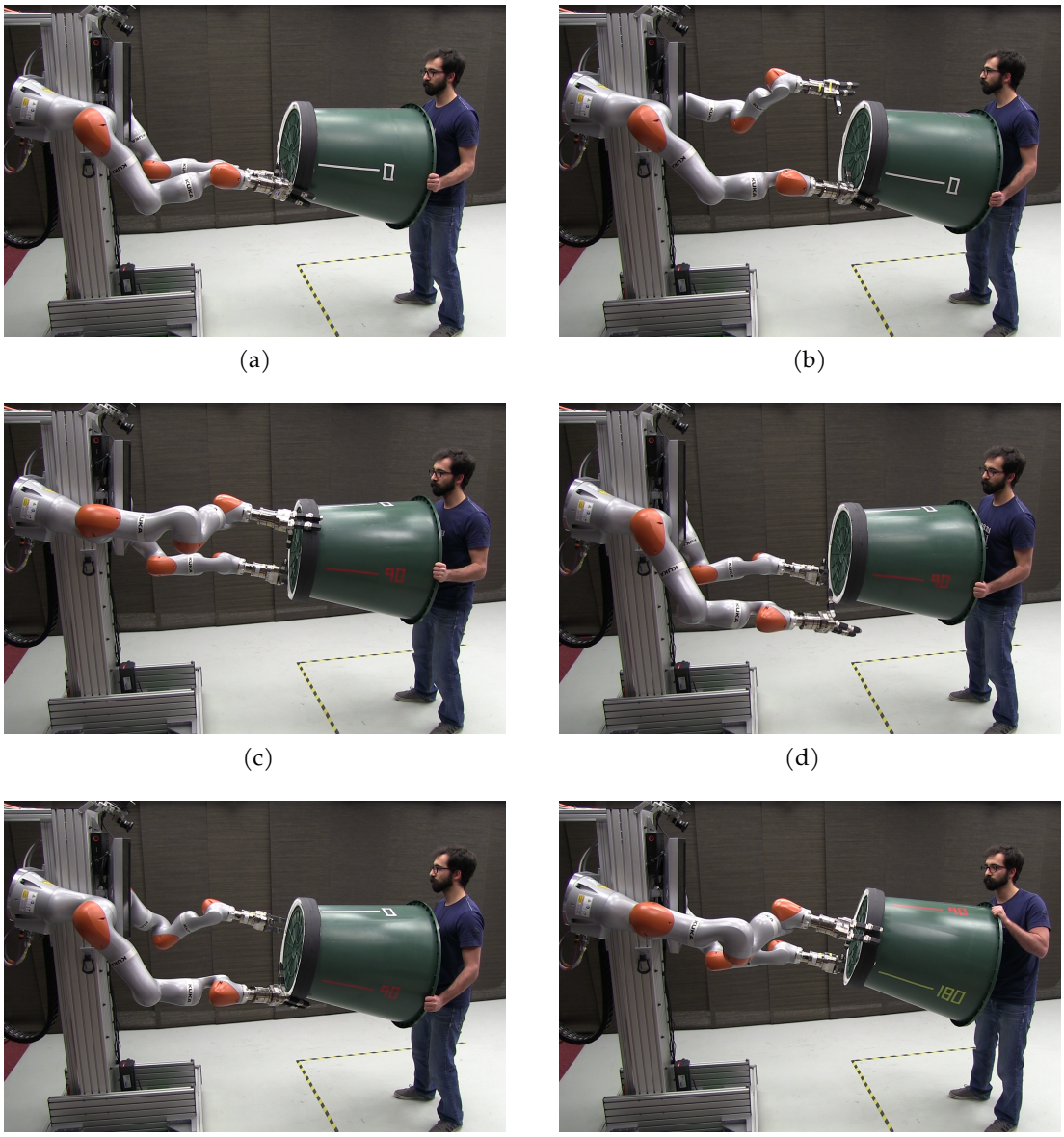


Figure 5.8: Evolution of (a) the absolute orientation of the object, and the relative to the object orientation of the (b) left end-effector and (c) right end-effector for a non-stationary task. The shaded areas indicate the duration and temporal placement of the end-effectors free-motion and its adaptation according to the switch of the joint goal. The vertical orange dotted line indicates the exact point in time where the change happens. The re-planning duration of $0.95s$ is shown with respect to the total motion duration of $88.27s$.



(a) (b) (c) (d) (e) (f)

Figure 5.9: Keyframes of a DcM task, where the human and the robot rotate a cylinder. (a) Initial state. (b) To realize the initial dyadic goal of orienting the cylinder at $90°$, the robot performs a left arm contact change. (c) The partner's goal changes to a $180°$ orientation for the cylinder. (d), (e) The robot performs the new contact changes in accordance to the adapted joint plan. (f) The updated plan is completed given the latest human goal.

5.4.3 *DcM experiments*

We validate our approach in a real setting, where a human partner jointly manipulates two different objects with a bi-manual, i. e. $\kappa \in \{1, 2\}$, and $n = 32$ DoF robot. The robot moves on the horizontal plane in a omni-directional fashion—due to its mobile base—and utilizes its two Kuka LBR iiwa 820 arms along with two Schunk dexterous

3-finger hands for manipulation and DcM tasks. A linear joint allows the arm base to be translated along the vertical axis. We use a cylindrical object which is bulky, so that a human cannot perform the task alone. The hybrid motion plans are optimized in the task space and are realized on the robot in a open-loop fashion, after being mapped to the configuration space using IK. A detailed description of the physical system can be found in (Gienger, Ruiken, et al., 2018). The robot utilizes surface contacts at the planned contact locations as a form of mechanical feedback. Further, it is worth noting that the joint-range of the robot only permits rotations of the object of about $90°$ before it reaches kinematic limits, thus grasp-hold changes are required.

5.4.3.1 *Online adaptation to human's goals*

We perform two experiments to demonstrate the on-the-fly adaptation to the human's real-time changing goals. Similar to the simulations shown in Figs. 5.7 and 5.8 these changes are unexpected.

In the first experiment, shown in Fig. 5.9, the initial goal of the human partner is to rotate the cylindrical object to $90°$. The full plan is computed by the bilevel optimization in $1.72s$ and the duration of the resulting hybrid motion is $27.31s$, with one contact change. During the experiment, the human partner decides that the preferred orientation of the object should be $180°$. Given the change of the human's goal, the robot agent computes the first segment of the adapted plan in $0.54s$ and the remaining segments of the hybrid plan are computed within $1.28s$, while the total duration of the updated plan is $51.63s$ with two contact changes.

In the second experiment, shown in Fig. 5.10, the initial goal of the human partner is to rotate the object to $-45°$. The hybrid motion plan includes a grasp-hold change of the right arm and has a total duration of $28.93s$, which is computed within $1.69s$. During execution, the human alters the intended dyadic goal and aims for a $90°$ desired object orientation. The first segment of the adapted motion (receding horizon) is computed in $2.60s$, while the remaining hybrid plan is computed in parallel with the execution of the first segment in $4.98s$. The total updated plan has a duration of $57.35s$ and includes two contact changes. Note that this experiment requires a complete reversal of the object orientation. The computed motion stops near $-45°$ (see attached video) and then an opposite rotation is initiated. The stop is due to the rotation reversal and not due to stretched computation time.

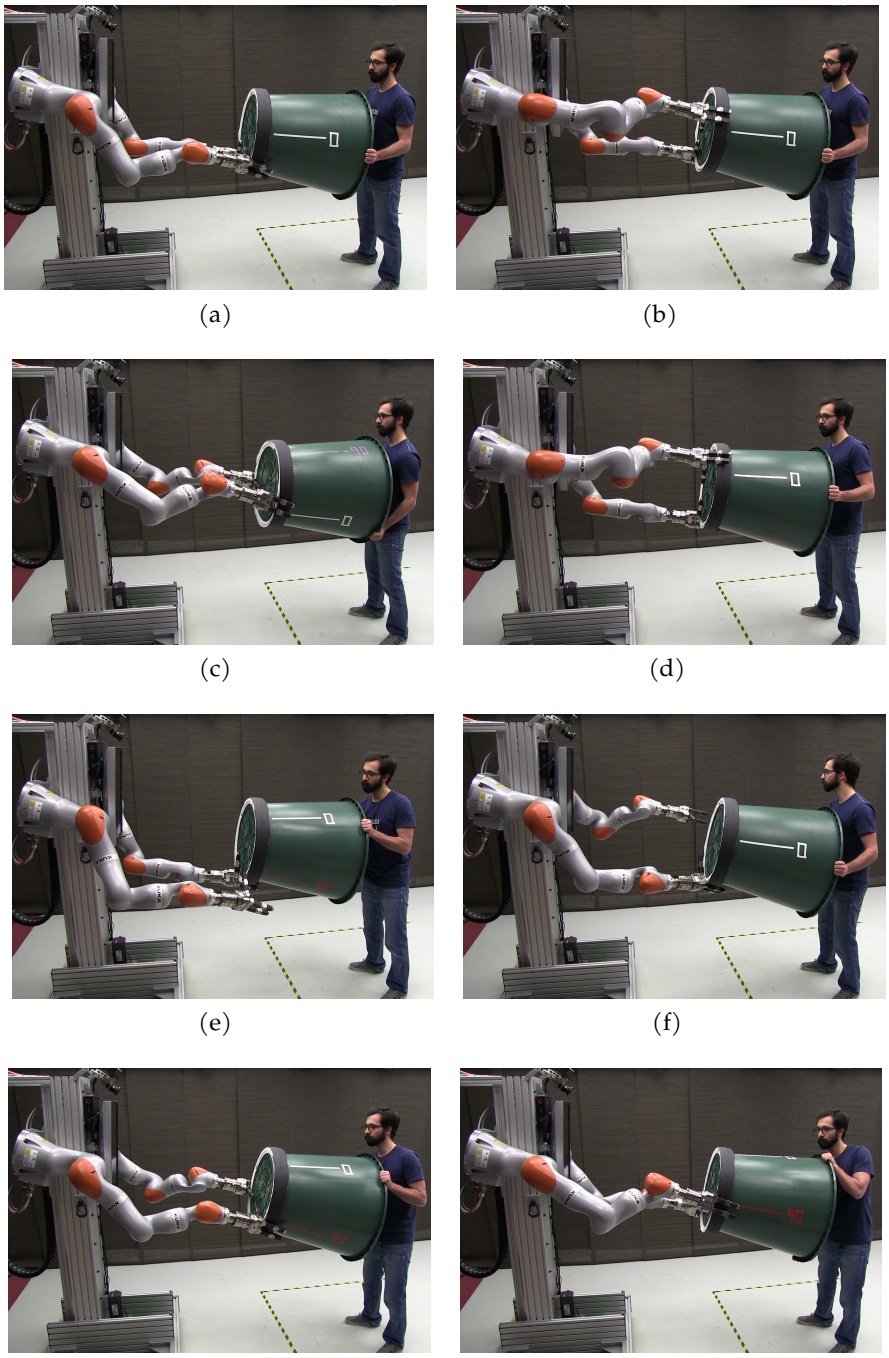


Figure 5.10: Frame sequence of a DcM task, where the human's initial goal is to orient the cylinder at $-45°$ and during the task execution his goal changes to $90°$. (a) Initial configuration. (b) Right arm during free-motion phase. (c) Right arm grasp-hold change has completed. (d) Human and robot jointly rotate the cylinder towards the original goal. (e) Given the human's goal change, the adapted hybrid motion plan is in progress. (f) Right arm contact location changes according to the updated plan. (g) Object weight is transferred to the right arm and the left arm changes grasp-hold. (h) All grasp-hold changes have finished and the final rotation of the object starts. (i) The dyad reaches the goal orientation for the object.

In this chapter, we have proposed a novel computational formalism to exploit the efficiency of informed GS methods in combination with the dynamic and geometric reasoning of optimal control methods. Our approach computes the optimal hybrid policy for the robot to complete manipulation tasks as a member of a dyad. Our bilevel optimization computes dynamically consistent and optimal hybrid paths for the (i) trajectory of the object, (ii) agent's forces, (iii) agent's contact locations, (iv) respective timings of these actions, and (v) arms contact sequence pattern. Due to the computational efficiency of the method, the optimal paths can be computed online, such that on-the-fly adaptation to real-time changes of the dyadic interaction can be realized. It is important to note here, that with this work we choose to have a loss in optimality of the motion plans to gain the ability to adapt the hybrid motion plans on-the-fly. This capability of the proposed method is particularly important for HRC scenarios, where typically the human partner alters intentions and behaviors multiple times throughout the interaction.

For the outer level described in Section 5.3.1, we used a specific discrete state representation and rules that do not model the partner explicitly. Nevertheless, our framework can be easily extended to enable multi-layered dyadic interaction modelling. The inner level (see Section 5.3.2) takes into account geometric (Toussaint and Lopes, 2017) and dynamic aspects of the interaction (see Section 5.4.2.1), while the outer level could incorporate logical interaction rules (der Spaa et al., 2020), e. g. if one of the partner's arms is in free-motion, the agent's end-effectors should remain in contact with the object.

In terms of the bilevel method described in Section 5.3, we used path (iv) shown in Fig. 5.2 to demonstrate the true bilevel nature of the method. This solution path represents a set of solution paths where the inner level radically alters a discrete transition computed by the the outer level. This corresponds to a new discrete solution path, which could not have been obtained by the outer level alone. Yet, to obtain such solution paths that arise from the interaction of both levels, both levels need to converge to a local solution at least. This has been empirically demonstrated with the simulations and experiments provided (see Section 5.4), however one cannot guarantee that, due to the non-linear and non-convex nature of the inner level. Ideally, in cases where the inner level fails to converge the outer level should compute an alternate discrete solution (*structure of the motion*) that could allow the inner level to converge as it has been shown in (Toussaint, Allen, et al., 2018) at the expense of computation time. In our bilevel method this is not the case, due to the A* choice for the outer level.

The outer level will always find the same optimal solution given the same initial state. Hence, in such cases the method will fail to converge.

To address this drawback induced by the A* choice, one could enhance the planning flexibility at the expense of optimality or computation time. This can be achieved by realizing the outer level with Anytime Repairing A* (Likhachev, Gordon, and Thrun, 2003), which computes multiple incrementally optimal solutions. An alternative for the outer level could be based on the multi-heuristic A* algorithm (Aine et al., 2016), which can utilize multiple inadmissible heuristic functions to guide the search. Using such an outer level method could allow for the activation and de-activation of different heuristic functions during the execution. This in turn will alter the solution obtained from the outer level (discrete solutions), in cases the inner level is infeasible.

Last, during the robot experiments, we identified the potential usefulness of micro-scale adaptation to task current state. Essentially, coping with arbitrary dyadic situations requires both our online planning adaptation method (long horizon) and closed-loop control (short horizon). To this front, one could investigate how to integrate together the proposed method with a hybrid controller, such as the one proposed by (F. R. Hogan, Grau, and Rodriguez, 2018). This would allow the robot to correct for small errors during the evolution of the task, e. g. close the loop with respect to the object state. Another option to close the loop could be a Hybrid *Model Predictive Control* (MPC) implementation based on the HOLM primitives. The HOLM computation times presented in Section 5.4.1.1 serve as a first promising step towards this direction.

6

CONCLUSION AND FUTURE WORK

> I may not be there yet, but I'm closer than I was yesterday.
>
> José N. Harris

This thesis studied manipulation and co-manipulation tasks in the context of *physical Human-Robot Collaboration* (pHRC). Our work proposes a variety of optimization-based motion synthesis methods to enable robotic agents manipulate static and dynamic objects, which are large, heavy and bulky—either alone or with a human partner. In particular, the investigated scenarios require appropriate contact selection and timely contact changes, which need to be inline with the behavior of the human partner. Further, due to variations of the human's behavior, online adaptation of the manipulation plans is essential.

The four key ideas introduced in this thesis are summarized below:

- A modular description of *Dyadic collaborative Manipulation* (DcM) setups in terms of computational processes and information flow between them, which allows to unscramble DcM into three separate and closely coupled modules; the human's policy, the robot's policy, and the task model.
- A multi-mode *Trajectory Optimization* (TO) formulation based on parametric programming to incorporate characteristics of the robot's controller into the planning layer, such that impacts are mitigated and dynamic multi-contact manipulation can be realized on hardware.
- A rough human model for the inclusion and consideration of the partner's policy in co-manipulation tasks with contact changes, which enables a robot to realize general object manipulation tasks jointly with a human.
- A bilevel optimization method that combines *graph search* (GS) methods with TO methods to achieve online multi-contact motion planning, by trading optimality for adaptation.

The following section briefly discusses these four key ideas presented in Chapters 2-5.

6.1.1 *Dyadic action formalism*

The aim of this thesis is to extent the robot motion capabilities for pHRC tasks. We first present a linear evolution from the *Human-Robot Interaction* (HRI) scheme to the pHRC scheme and eventually to the DcM setup. DcM setups are collaborative tasks, that exhibit two particular characteristics. First, the completion of DcM tasks, such as rotating an object upside-down, requires discrete grasp-hold changes. Second, in DcM setups the two individuals are coupled via the physical medium and their force interaction is continuous. The former differentiates DcM from co-manipulation tasks where the contact configuration is fixed and the latter diversifies DcM from instantaneous or intermittent force interaction tasks, such as the hand-over tasks.

The formalization for the DcM concept is provided next, where the two co-actors and the dyadic action are treated separately (see Fig. 2.6), the policy of the agent is hybrid to consider grasp-hold changes, and the behavior of the human partner is considered to be non-stationary—it can change over time. This enables the methods developed to generalize over different tasks and partner behaviors, while the joint action constraints are met.

6.1.2 *Multi-mode Trajectory Optimization for multi-contact and impact-aware manipulation*

Regarding the motion generation, we are interested in methods that have the following three properties. First, they can synthesize hybrid motions, i. e. motions that simultaneously consider discrete and continuous quantities. Second, due to the size of the objects, we decided to consider non-prehensile manipulation, which in turn leads to under-actuation. For that reason, when synthesizing such hybrid motions, actions need to be selected with respect to a future time horizon. Third, on top of planning hybrid motions we are also interested in obtaining motions that directly translate into the real hardware. Hence, we introduce the concept of *contact-control* modes and propose a multi-mode TO framework (see Eq. (3.9)). This encodes *free-motion* and *in-contact* modes, i. e. hybrid motion, as well as different compliance control modes, such as low and high stiffness of a compliance controller.

To achieve the above, we present a single optimization formulation that includes both hybrid dynamics and hybrid control. Our method computes an optimal hybrid policy for the robot to complete manipulation tasks, assuming that a model of the object and the contact schedule sequence, called *structure of the motion*, are provided.

With this information, our method computes a dynamically consistent and optimal hybrid solution for the (i) trajectory of the object, (ii) agent's forces, (iii) agent's contact locations, and (v) respective timings of these actions. In addition, we use an impact model to develop a contact force transmission model (see Eq. (3.44)), which in turn enables planning smooth transitions from *free-motion* to contact at speed. With this addition the multi-mode TO method can also compute the stiffness profiles simultaneously with an optimal hybrid motion.

We demonstrate hybrid motion plans in simulation on planar and dynamic balancing manipulation tasks. Also, we validate the method with real-world experiments, where a compliant robot halts a large mass and fast moving object. The results show that the proposed method can respect hardware limitations, such as force and workspace limits, while it also enables much lower contact transition forces than standard compliance controllers and impact-agnostic TO methods.

6.1.3 *Partner-aware Trajectory Optimization*

To enable a robot to realize joint manipulation with a human, we utilize the multi-mode TO method discussed above. Only in this case, the motion of the object is also subject to external forces that do not originate from the robot. These forces represent the contribution of the human partner in the dyadic task. We propose to model the collaborative manipulation behavior of the human partner with a simple but effective spring-damper model.

Our approach (see Eq. (3.9)) only assumes a roughly known pattern of the partner's policy, and it computes the optimal hybrid policy for the robot to complete the manipulation task as a member of a dyad. The generated hybrid motion plans consist of i) the trajectory of the object resulting from dyadic action, ii) robot's forces to jointly balance the object with the human partner, iii) robot's contact locations and iv) respective timings of these actions.

The proposed concepts have been evaluated both in simulation and with an actual human-robot dyad. In simulation it is demonstrated that the resulting solutions can be personalized to the specific behavior of each partner. The robot experiments demonstrate that hybrid motions can be realized even with an uncertain model of the human partner.

6.1.4 *Online adaptation during dyadic manipulation*

In collaborative tasks adaptation of one's own actions to the actions of the partner is essential. To address this challenge, we propose a novel bilevel computational formal-

ism (see Eq. (5.3)) for online adaptive robot motion generation. The method exploits the efficiency of informed GS methods in combination with the dynamic and geometric reasoning of optimal control methods, e. g. TO methods. Our approach assumes a roughly estimated pattern of the partner's policy and a model of the object. With this information, our bilevel optimization computes dynamically consistent and optimal hybrid paths for the (i) trajectory of the object resulting from dyadic action, (ii) robot forces to jointly balance the object with the human partner, (iii) robot contact locations, (iv) respective timings of these actions, and (v) contact sequence pattern of the arms, called *structure of the motion*. The computational efficiency of the method permits online computation of hybrid motions. This enables on-the-fly adaptation to real-time changes of the dyadic collaboration, e. g. alternations of the human partner's intentions and behaviors.

In short, the proposed method is able to optimize over a variety of different modes spanning both

1. the hybrid action space that arises, due to the multi-contact nature of the task,
2. the multi-modal nature of joint-action planning, due to the non-stationary policy of the human partner.

A further pivotal aspect of the method is the use of an informed GS algorithm (see Section 5.3.1) in combination with the decomposition of the hybrid motion into the *Hybrid Optimization Lexicon for Manipulation* (HOLM) primitives (see Section 5.3.2). This enables to holistically optimize over such a complex and multi-modal space efficiently. We use a set of rules to explore only the useful part of the solution space and exploit a set of hybrid motion primitives (HOLM) to generate hybrid motion plans very efficiently.

We evaluated the method both in simulation and with an actual human-robot dyad. Both results demonstrate that the proposed method enables the robot agent to adapt its motion plans online, in response to real-time changes of the dyadic setup. These indicate the large potential of the method to be employed in general real-world co-manipulation scenarios.

6.2 LIMITATIONS AND DIRECTIONS FOR FUTURE WORK

This section outlines the limitations of the thesis and discusses potential interesting directions for future work.

In our dyadic action formulation we treat the two individuals and their dyadic coupling separately (see Fig. 2.6); such treatment was used to analyze Human–Human interactions (Takagi, Ganesh, et al., 2017). We believe that this design choice is of

core importance, as it allows our method to generalize over different tasks and partner behaviors, which has been demonstrated in a variety of scenarios in Chapter 4 and Chapter 5. Although this aids our understanding of such systems and enables great generalization capabilities, one could argue that in practice the behaviors (policies) of the individual members of a dyad are more intertwined and cannot be treated as separate policies that exchange information. A number of approaches (Vogt, Stepputtis, Jung, et al., 2018; G. Maeda et al., 2017; Campbell and Amor, 2017) follow a strategy towards directly capturing the joint behavior of the system. These methods in theory can reason based on the joint probability distribution of both individuals' policies and their coupling. Yet, in practice the vast variability of existing behaviors and interaction patterns leads to distributions with a very large support, which makes reasoning intractable. In contrast, imposing structure to such problems has been shown to be very beneficial (Unhelkar and J. A. Shah, 2019; Kaess, Ranganathan, and Dellaert, 2008). In conclusion, we consider that the potential benefit of adopting a unified approach is yet to be assessed.

Regarding the multi-mode TO method, we make two important assumptions. First, we assume that the *structure of the motion* is either provided in advance or from an outer-level process (Toussaint, Allen, et al., 2018) (see Chapter 5). This assumption imposes a particular contact schedule for the case of the multi-contact manipulation scenario, which is locally adjusted when solving the hybrid TO problem. As a result, the optimization problem is smooth (Nurkanovic, S. Albrecht, and Diehl, 2020) and can be solved efficiently. However, only part of the full hybrid solution space is considered, while there exist more flexible but less efficient TO methods (Posa, Cantu, and Tedrake, 2014). For the case of impact-aware manipulation, the same assumption also specifies the hybrid control structure. We are not aware of any work that could enable automatic switching and selection of controllers. Yet, for the specific problem studied, one could investigate addressing it with a differential dynamic programming (DDP) approach (D. J. Braun et al., 2013; Tassa, Mansard, and Todorov, 2014), which provides continuous stiffness profiles (Nakanishi, Radulescu, D. J. Braun, et al., 2016).

The second assumption concerns the a priori known model of the object. Our method assumes that a parametric differentiable representation of the object is provided, which limits the use of the method only to manipulation tasks with known objects. An interesting extension could utilize vision sensors to obtain a differentiable model of any object online (Wu et al., 2015; Qi et al., 2017), such that manipulation of unknown objects can also be realized.

Further, regarding the experimental validation of impact-aware manipulation, currently the robot computes the impact-aware trajectories online, but executes them in open-loop, thus requiring fairly accurate estimation of motion parameters such as ve-

locity and rolling friction. To realize arbitrary halting and catching tasks, one could explore extending the proposed framework to a fully closed loop MPC implementation, which can cope with even more uncertain estimation of the object motion.

In terms of the partner-aware TO discussed in Chapter 4, the main assumption is that the partner's behavior is stationary. During the robot experiments we identified the usefulness of non-stationary human policy models, especially for long horizon motions. This assumption was relaxed to a particular extent in Chapter 5. Also, to verify the validity of the proposed human partner model, one could perform a quantitative comparison between the actual behavior of the human partner and the behavior predicted according to the proposed spring-damper model. Such a comparison could be an extension of Takagi's work (Takagi, Ganesh, et al., 2017; Takagi, Usai, et al., 2018) which inspired us to explore such a model and it would be a very interesting human motor control study.

In Chapter 5 we consider online adaptation during dyadic manipulation. We relax the assumption on the stationarity of the human partner model. We allow arbitrary changes of the human's policy, which results in changes of the robot's policy via online adaptation of the motions. Yet, we assume that an estimate of the partner's policy exists. This estimate indicates the policy change and the timing of this change. To fully realize our vision presented in Fig. 2.6, our future work will focus on developing online methods for human intention estimation in the context of collaborative manipulation tasks and potentially explore the potential usefulness of more elaborate models for the human. The first step towards this goal includes the use of sensors to track the motion of the human partner and the object, e. g. a use case is described in Appendix D. We did perform a few exploratory developments in this direction fusing together information from the `Xsens` motion suit and the `VICON` motion capture system to exploit the advantages of these complementary sensing systems. More details can be found in Appendix D.2.

In terms of the bilevel optimization, we decided to trade optimality for adaptability. This decision stems from the online adaptation requirements of DcM setups. Also, in the outer level described in Section 5.3.1, we use a specific discrete state representation and rules that do not model the partner explicitly. Nevertheless, our framework can be easily extended to enable multi-layered dyadic interaction modelling. The inner level (see Section 5.3.2) takes into account geometric (Toussaint and Lopes, 2017) and dynamic aspects of the interaction (see Section 5.4.2.1), while the outer level could incorporate logical interaction rules, e. g. if one of the partner's arms is performing a change of contact the robot end-effectors should remain in contact with the object. Additionally due to the A* choice, the outer level finds only the optimal discrete solution. One could enhance the planning robustness at the expense of op-

timality at the discrete level or computation time, by realizing the outer level with Anytime Repairing A* (Likhachev, Gordon, and Thrun, 2003), which computes multiple incrementally optimal solutions.

In the inner level and in the multi-mode TO, we delegate the motion generation of the robot configuration to a standard *Inverse Kinematics* (IK) solver. One could also model the robot configuration within the TO method (Merkt, Ivan, and Vijayakumar, 2019) at the expense of computation time.

Last but not least, during the robot experiments, we identified the usefulness of micro-scale closed loop adaptation to current state of the task. Essentially, coping with arbitrary dyadic situations requires both our online planning adaptation method (long horizon) and closed-loop control (short horizon). To this front, a Hybrid *Model Predictive Control* (MPC) implementation based on the HOLM primitives would allow the robot to correct for small errors during the evolution of the task, e. g. close the loop with respect to the object state. The HOLM computation times presented in Section 5.4.1.1 serve as a first promising step, which would enable a tighter integration of feedback to realize more robust behaviors during the collaboration.

This work setups the basic formulation, which we can build upon towards obtaining collaborative robots in DcM tasks. Having mentioned that, little exploration has been done in terms of cost functions design to attain human legible and anticipatory motions during the manipulation task (Dragan, K. C. Lee, and Srinivasa, 2013), which could be another very interesting avenue for research.

6.3 EPILOGUE

This thesis proposed a formalism and a computational framework to address some of the challenges faced in object co-manipulation tasks, while there are further challenges that require further attention and study. Looking into the future, the direct impact of this research to society is to be found in daily scenarios, where robots will accommodate humans in pHRC tasks. We envision robots being equal partners to humans, which can learn from them, but also guide them through physical collaboration to more ergonomic postures.

APPENDIX

A

HUMAN-ROBOT INTERACTION VIA SHARED AUTONOMY

This appendix briefly introduces an *Human-Robot Interaction* (HRI) integration project we worked on. This project enabled us to obtain a more rounded understanding of the HRI domain, which in turn led to the development and investigation of the *Dyadic collaborative Manipulation* (DcM) setup. In particular, this project is an instantiation of the teleoperation scheme, described in Chapter 2. It is important to note here, that this work (Merkt, Y. Yang, et al., 2017) was led by Wolfgang Merkt and was achieved in collaboration with Wolfgang Merkt, Christopher Mower, Yiming Yang and myself.

Figure A.1: The bimanual mobile manipulator executing dual object pickup task, that has been commanded by a human operator. The system segments the target objects from the scene, automatically plans the placement of the mobile base and navigates to it.

A.1 OVERVIEW

This work studies a shared autonomy system, where the human operator provides high level commands to a remote robot. Given a high level objective, such as pick a specific object or two objects as shown in Fig. A.1, the appropriate sequence of action plans is composed by the robot. These high level motion plans can be verified by the human operator and are automatically executed by the robot. During execution the robot utilizes a continuous scene monitoring module to adapt its actions to dynamic changes of the environment and to inform the operators about the updated state of the scene. Further, the developed system is able to automatically recover from a va-

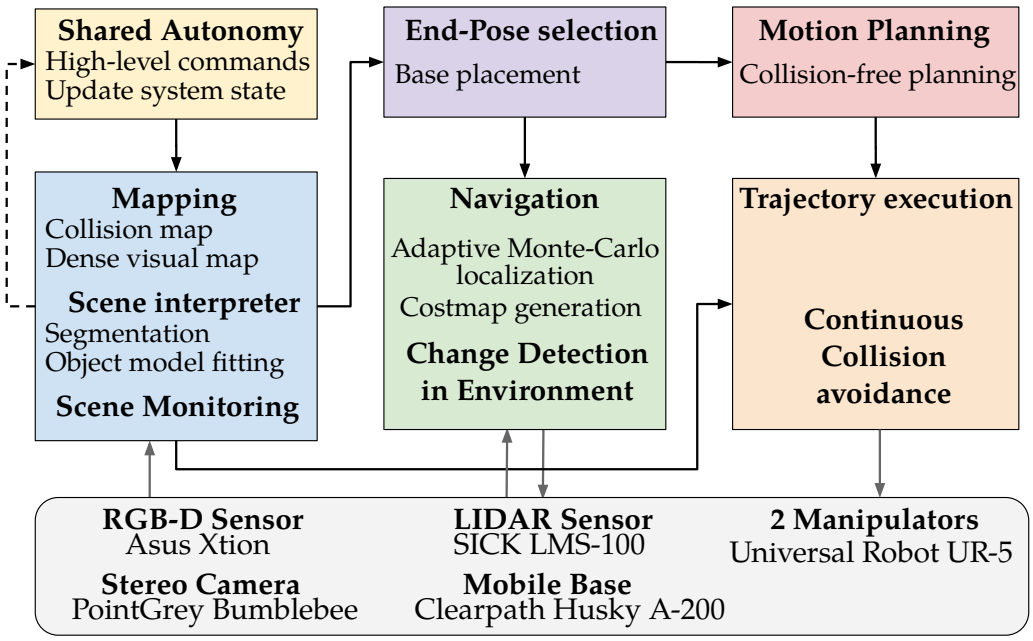


Figure A.2: Hardware and Software architecture of a shared autonomy system. All the hardware components of the dual arm Husky robot shown in the grey box are connected with different software components, indicated with various colors.

riety of deadlocks and fall back to the operator in case autonomous operation is not attainable. The system has been demonstrated in action with a bimanual mobile base robot in remote operation scenarios.

A.2 SYSTEM DESCRIPTION

Fig. A.2 depicts the architecture of the modular system. The modules of the system are: (i) a dense visual mapping module that captures and fuses multiple views and sensors into a dense, 3D representation of the workspace, (ii) a continuous scene monitoring module along with a change detection module, which are both used for failure recovery and adaptation to dynamic changes of the environment. (iii) a reliable collision-free motion planning module (Y. Yang, Ivan, et al., 2016), and (iv) an end-pose planning module (Y. Yang, Merkt, et al., 2017) used to select the optimal pose for the mobile base of the robot, such that reachability constraints can be satisfied.

The interaction between the human operator and the robot is realized via a shared autonomy user interface, shown in Fig. A.3. This user interface provides an abstracted action-perception layer that allows the user to provide high-level commands, and receive feedback in terms of the state of the environment. Utilizing this interface the

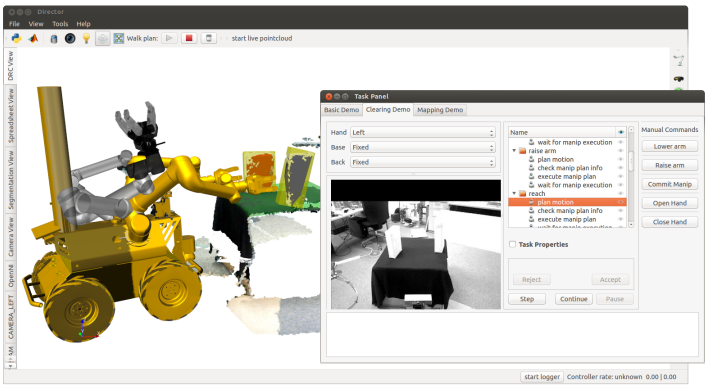


Figure A.3: The user interface for the bimanual mobile manipulator showing live perception data, segmented objects and fitted affordances as well as the candidate plan in gold.

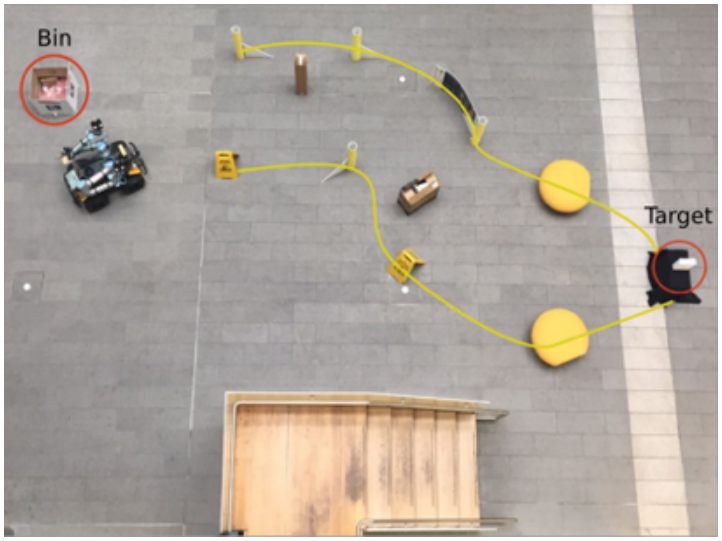


Figure A.4: Shared autonomy for navigation through narrow cluttered path and recovery task, while there is no line-of-sight and restricted communication between the operator and the robot.

operator can override, interrupt and resume the motion plans of the robot at any point in time.

A.3 DEPLOYMENT

We validated the shared autonomy system on hardware with a dual-arm mobile platform. In our experiments we demonstrate the utility and simplicity of the shared autonomy method. Further, we tested the system on a dismantle and dispose challenge at the International Robotics Challenge named "The Robots for Resilient Infrastructure Competition". The arena of the challenge was similar to the one shown in Fig. A.4 and the task involved navigating and clearing narrow pathways, picking up critical

packages and disposing them to a bin. Two videos showing the experimental validation of the system are provided (see OV1 and OV2 in Appendix E).

B

ALTERNATIVES ON INTERMITTENT CONTACT PLANNING AND COMPLIANCE CONTROL

This appendix briefly describes an alternative way to model hybrid systems that transit from *free-motion* mode to *in-contact* mode and vice versa. In addition and for completeness reason, here we briefly describe an alternative method to achieving compliance with position-controlled robots.

B.1 TRAJECTORY OPTIMIZATION THROUGH CONTACT

In trajectory optimization through contact, the hybrid nature of the intermittent contacts is usually expressed via a complementarity formulation defined as $0 \leq d \perp f \geq 0$, where d is a signed distance between the contacting objects and f is the constraint normal force between them. This states that only unilateral force can be exerted between the bodies, penetration is not allowable, and that situations involving no contact but contact force are excluded. Mathematical programs with complementarity constraints are in practise difficult to solve as they do not satisfy constraint qualifications and relaxations are usually needed (Posa, Cantu, and Tedrake, 2014).

B.2 TASK SPACE COMPLIANCE WITH POSITION-CONTROLLED ROBOTS

Here, we describe an admittance controller that utilizes force-torque sensors to perform interaction tasks with a position-controlled robot. Such tasks typically require force or compliance control (Siciliano and Khatib, 2016).

Given an error between the measured wrench[1] $\mathbf{W}^s \in \mathbb{R}^6$ and the desired wrench $\mathbf{W}^d \in \mathbb{R}^6$ at a specific task coordinate frame, we can regulate the robot compliance at that frame. This can be achieved with the following displacement law

$$\Delta \mathbf{c}_t = \chi \Delta \mathbf{c}_{t-1} + \mathbf{Q} \Delta \mathbf{W}, \tag{B.1}$$

where the $\Delta \mathbf{c}_t \in \mathbb{R}^\nu$ denotes the displacement at task coordinate frame, $\chi \in [0, 1]$ is a forgetting factor that phases out the displacement at the task coordinate frame, $\Delta \mathbf{W} = (\mathbf{W}^d - \mathbf{W}^s)$ is the wrench error and $\mathbf{Q} = \frac{1}{\mathbf{K}}$ is the compliance gain. Both χ

1 Wrench is a stacked vector of force and torque.

and $\mathbf{Q}$ can be fixed or can be updated at each time step according to an adaptation scheme (Roy and Whitcomb, 2002).

If the task coordinate frame matches the one of the force-torque sensor, it is straight forward to map the displacement $\Delta\mathbf{c}_t$ to the configuration space with the well known *Inverse Kinematics* (IK) equation as $\Delta q_t = \mathbf{J}^\dagger \Delta\mathbf{c}_t$, where $\mathbf{J}^\dagger \in \mathbb{R}^{n\times\nu}$ denotes the pseudo-inverse Jacobian of the manipulator. However, in many scenarios the coordinate frame of the force-torque sensor might differ to the one of the task coordinate frame. For example, consider a scenario where we aim to regulate the compliance at the coordinate frame matching the center of mass of the object, while the force-torque sensors are placed at the wrists of the robot arms. For these case, we describe next a task-generic mapping of the force-torque measurements obtained at the sensor coordinate frames to the task space coordinates of interest.

Similarly to the wrench error $\Delta\mathbf{W}$, we define $\Delta\mathbf{G} \in \mathbb{R}^6$ to represent the error wrench at the task coordinate frame. Using Eq. (B.1) and IK both errors can be mapped to displacements in the configuration space denoted as $\Delta\mathbf{q}(\Delta\mathbf{W})$ and $\Delta\mathbf{q}(\Delta\mathbf{G})$, respectively. By equating the two displacements, we obtain

$$\mathbf{q}(\Delta\mathbf{W}) = \Delta\mathbf{q}(\Delta\mathbf{G}), \tag{B.2}$$

$$\mathbf{J}_W^\dagger \Delta\mathbf{c}(\Delta\mathbf{W}) = \mathbf{J}_G^\dagger \Delta\mathbf{c}(\Delta\mathbf{G}), \tag{B.3}$$

where $\mathbf{J}_W^\dagger$ is the stacked pseudo-inverse Jacobian of the force-torque sensor frames and $\mathbf{J}_G^\dagger$ is the pseudo-inverse Jacobian of the task coordinate frame. Given that there might be S number of force-torque sensors, the expanded form of the above equation is

$$\Delta\mathbf{c}(\Delta\mathbf{G}) = \mathbf{J}_G \begin{bmatrix} \mathbf{J}_{W\_0} \\ \mathbf{J}_{W\_1} \\ . \\ . \\ \mathbf{J}_{W\_S} \end{bmatrix}^\dagger \begin{bmatrix} \Delta\mathbf{c}(\Delta\mathbf{W}_0) \\ \Delta\mathbf{c}(\Delta\mathbf{W}_1) \\ . \\ . \\ \Delta\mathbf{c}(\Delta\mathbf{W}_S) \end{bmatrix}, \tag{B.4}$$

and its solution provides the least-square displacement at the task coordinate frame with respect to all the force-torque sensor wrench errors.

This operational space compliance controller was tested in a physical simulation environment in *Robot control software* (Rcs) in a number of bimanual manipulation tasks similar to the one shown in Fig. B.1. The bimanual robot utilizes its force-torque sensors placed at its wrists and its joints are assumed to be position-controlled.

Although, the admittance controller described is a valid approach to compliance, it has been demonstrated that impedance controllers are more suitable for interaction tasks with rigid contacts (Ott, Mukherjee, and Nakamura, 2010). Furthermore, the

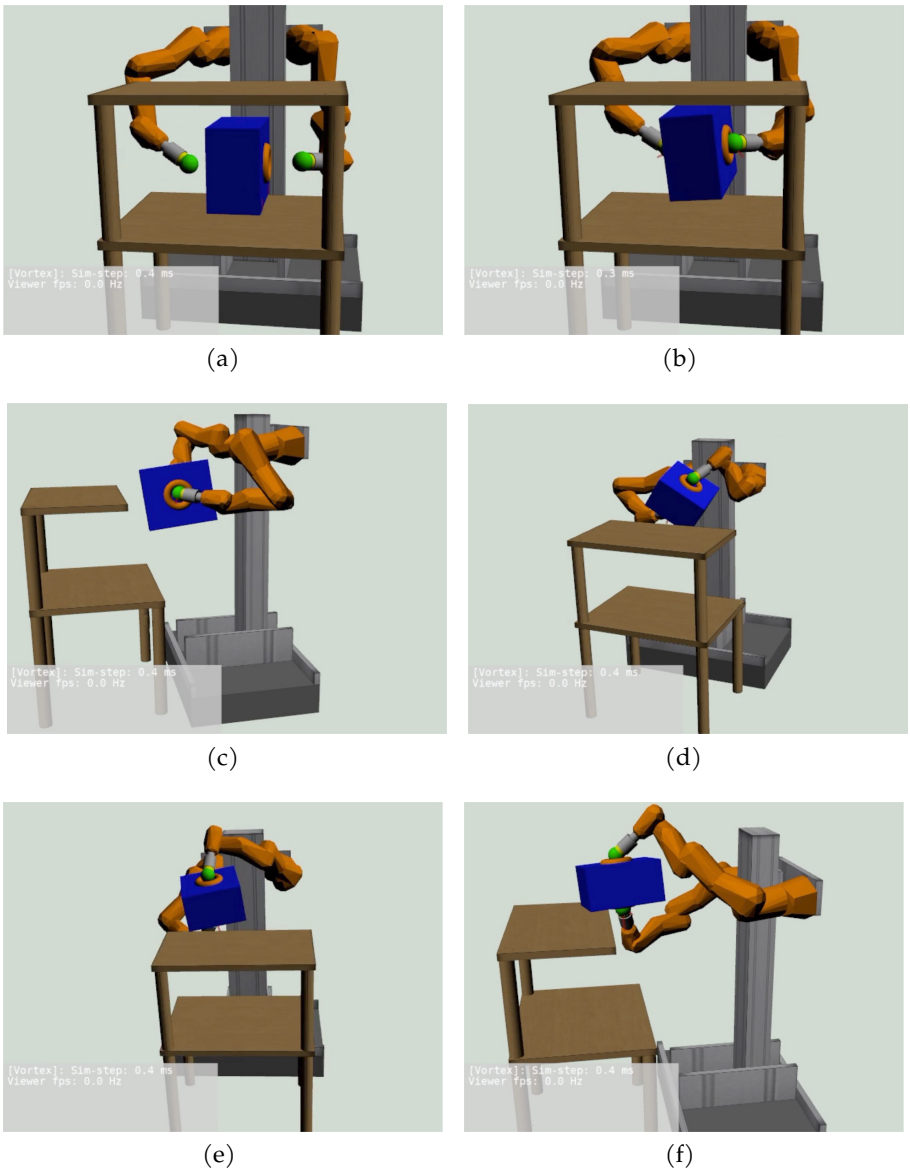


Figure B.1: A sequence of frames of a bimanual manipulation scenario, captured from different points of view. The robot utilizes the two force-torque sensors at its wrists, to establish stable contact with the object and to realize a bimanual grasp via concurrent compliance control at the center of mass of the object and at the coordinate frames of the two end-effectors (green spheres).

control frequency of admittance controllers is typically limited to approximately 100 -150 Hz, due to the use of external force-torque sensors, which in turn further limits the utility of such a controller in real-world agile manipulation tasks, like the one studied in Section 3.5.

C

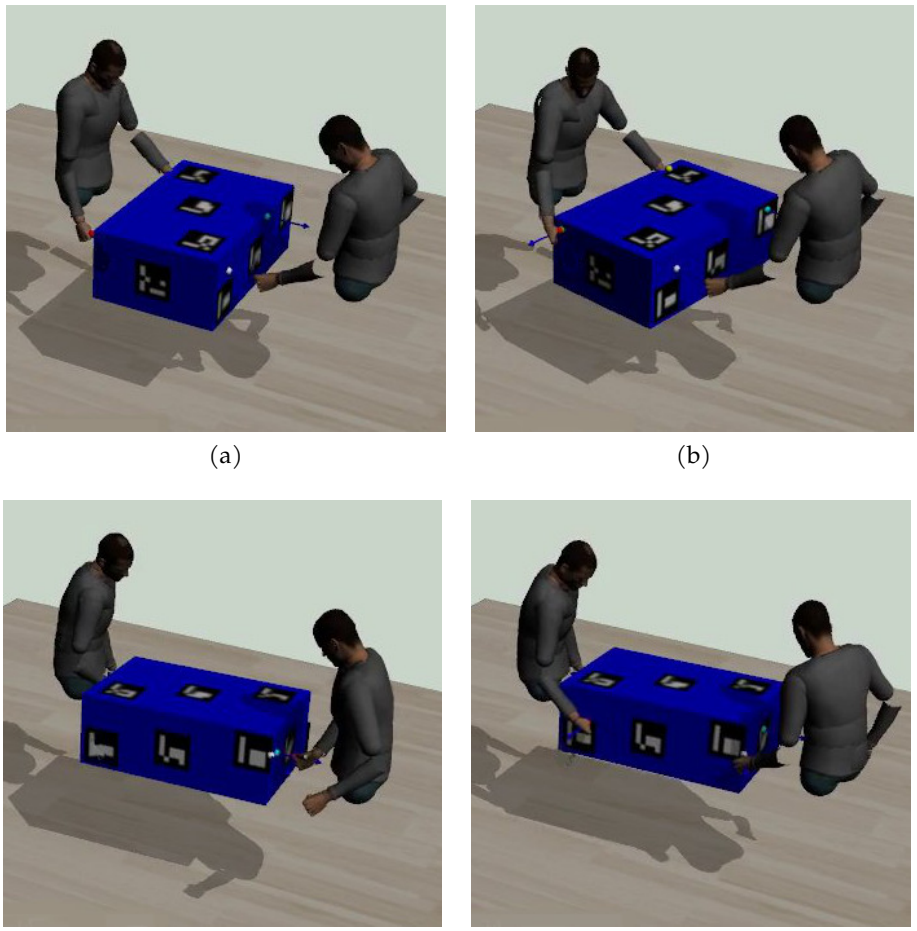
TOWARDS EMERGENT COORDINATION WITH A DATA-DRIVEN METHOD

In Chapter 4, we referred to the dyadic planning problem and into the two types of coordination; *planned coordination* and *emergent coordination*. Here, we describe a preliminary study in which we investigated a data-driven approach to the multi-contact dyadic planning problem. The method can be categorized as an *emergent coordination* approach and it aims to address the '*where*' question of joint action, as well as the '*when*' question in a reactive manner, similar to the work of Peternel, Tsagarakis, and Ajoudani, 2017. The first subsection reviews the data collection and processing steps and the second describes the undertaken method. It is important to note here, that the interface to parse the kinematic data from `Xsens` and `ArUco` to *Robot control software* (Rcs) was developed at the Honda Research Institute Europe (HRIeu) by Felix Treede, Mark Dunn and Michael Gienger.

C.1 DATA COLLECTION AND PROCESSING

Using two `Xsens` suits (see Appendix D) and the `ArUco` marker-based pose estimation library (Garrido-Jurado et al., 2014), we collected a number of demonstrations that consist of kinematic data from a human dyad completing a collaborative manipulation task with a box. The collected data was analyzed using Rcs, where the motion of the object and of the humans (avatars) can be visualized, as shown in Fig. C.1.
A list of the extracted attributes regarding the motion of the humans follows:

- Cartesian distance between two hands of the avatars, along each axes.
- Distance between two hands of the avatars.
- Parameters of the axis-angle representation of the relative orientation between two hands of the avatars.
- Chasle-point of the left hand (3D) of the avatars.
- Chasle-axis of the left hand (3D) of the avatars.
- Chasle-point of the right hand (3D) of the avatars.
- Chasle-axis of the right hand (3D) of the avatars.
- Linear-velocity of the left hand of the avatars.
- Linear-velocity of the right hand of the avatars.



(a) (b)

(c) (d)

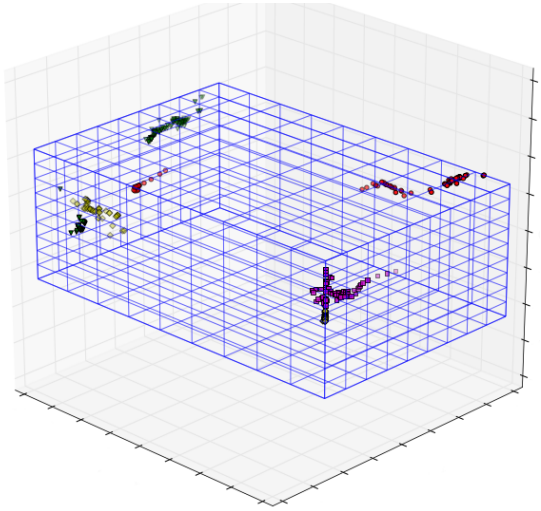
Figure C.1: A sequence of frames based on recorded data while two humans first rotate a box and then translate it.

Next, we list the respective attributed for the the motion of the object.

- Object pose.
- Object linear velocity.
- Object angular velocity.

Last, we list the extracted properties of the interaction between the humans and the object.

- Distance between the left hand and the object.
- Left hand contact point on the object.
- Left hand contact normal on the object surface.
- Distance between the right hand and the object.
- Right hand contact point on the object.
- Right hand contact normal on the object surface.



(a)

Figure C.2: Contact regions on the object surface, where the hands of the avatars make contact with the object. To designate the avatar and hand indexed contact regions, different colors and marker types were used for each of the avatars and each of their hands.

These properties were extracted using a collision detection algorithm. Each of the demonstrations includes changes of grasp-holds, which result into multiple distinct contact regions on the objects. Hence, for each demonstration we utilized a number of the properties listed above to obtain; the contact regions and the contact or non-contact instances. One such example of contact regions is illustrated in Fig. C.2.

C.2 LEARNING-BASED REACTIVE CONTACT PLANNER FOR DYADS

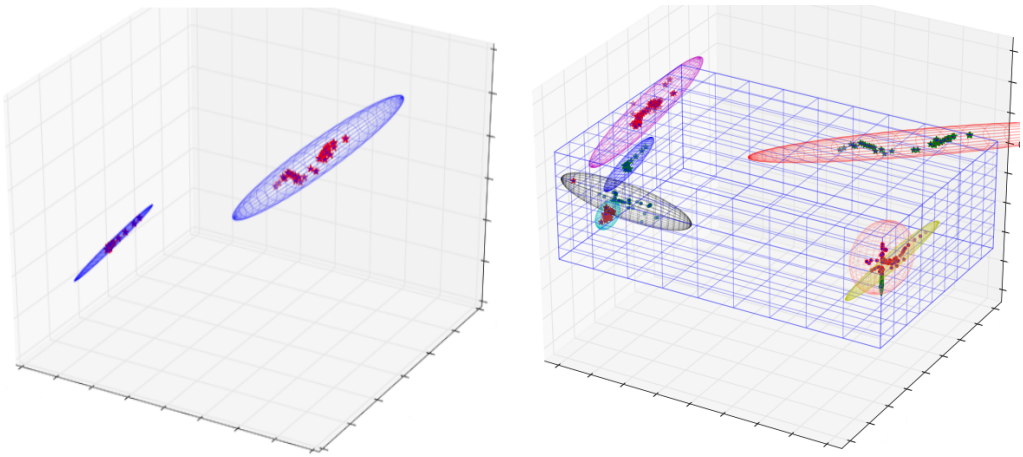
Here, we present the modelling approach used to learn how two humans coordinate their grasp-holds changes during *Dyadic collaborative Manipulation* (DcM) tasks.

C.2.1 *Contact regions as probability distributions*

The first step involved the representation of the contact regions shown in Fig. C.2 as probability distributions. The primary goal was to maintain the spatial separation of the regions as this is their key attribute. We used *Gaussian Mixture Model* (GMM) to encode the contact region information as

$$p(\mathbf{c}^{\kappa}) = \sum_{\iota=1}^{I} w_{\iota} \mathcal{N}(\mathbf{c}^{\kappa} | \mu_{\mathbf{c}^{\kappa}}, \Sigma_{\mathbf{c}^{\kappa}}), \tag{C.1}$$

where $\mathbf{c}^{\kappa} \in \mathbb{R}^3$ denotes the contact location relative to the object coordinate frame.



(a) (b)

Figure C.3: Gaussian components of the GMM visualized as ellipsoids. In (a) the contact points and the respective Gaussian components only of the one hand of one human are visualized, while the object is omitted for illustration reasons. (b) All the Gaussian components for each GMM are shown along with the contact points and the wire-frame of the box object.

The parameters $\mu_{\mathbf{c}^\kappa}, \Sigma_{\mathbf{c}^\kappa}$ describe the mean of the contact region in Cartesian space and the extent of the region in 3D. I is the total number of Gaussian components used to capture the regions and w_ι are weighting factors. We used one GMM for each of the four hands and the number of Gaussian components per hand was empirically set. To fit the GMM parameters based on the contact regions, we utilized the Expectation-Maximization algorithm (C. M. Bishop, 2006) from the scikit-learn library (Pedregosa et al., 2011).

A result of the fitting process is shown in Fig. C.3, where the Gaussians are vizualized in the form of ellipsoids. In Fig. C.3a, we show the fitted Gaussians only for the left hand of one of the humans. In Fig. C.3b, we illustrate all the fitted Gaussians for all the contact regions. This figure also demonstrates that given an appropriate number of Gaussian components, the GMM is able to accurately encode the spatial distribution of the contact regions. In this case, we selected the optimal number of Gaussian components empirically, yet one could also use a principled way that utilizes other methods, e. g.an iterative procedure based on k-Means clustering, or GMM fitting and Bayesian information criterion (BIC) (Calinon, Guenter, and Billard, 2007).

The current model is able to capture the spatial distribution of the contact data, yet we aim to have a model that encodes the change of grasp-holds. A straightforward approach would involve combining the GMM with a sequential model, such as *Hidden Markov Model* (HMM) (Calinon, Guenter, and Billard, 2007), which can provide a time indexed description of the contact regions (Gaussian components) for each hand. These models typically require additional temporal alignment algorithm—such

as *Dynamic Time Warping* (DTW) (Vogt, Stepputtis, Grehl, et al., 2017)—to scale time across demonstrations, which would further hinder the generalization capabilities of the approach. Thus, we decided to explore an instance-based alternative that is based on a regression method. This method generates the contact points of the agent as a response to the contact points of the partner.

C.2.2 *Agent's contacts as a response to partner's contacts*

In comparison to before, where we separated the contact data with respect to each avatar and its hands, here we describe a method that models the joint distribution of all four contact points. To do this, we group both contact points of both individuals with respect to the time instance they occur. In this way, we can learn a joint distribution that correlates the contact locations of the agent with the contact locations of the partner. This can be formally written in the GMM form as

$$p(\mathbf{c}^{\kappa,p}, \mathbf{c}^{\kappa,a}) = \sum_{\iota=1}^{I} w_\iota \mathcal{N}(\mathbf{c}^{\kappa,p}, \mathbf{c}^{\kappa,a} | \mu_{\mathbf{c}}, \Sigma_{\mathbf{c}}), \tag{C.2}$$

where each $\mathbf{c}^{\kappa,j} \in \mathbb{R}^3$ with $\kappa \in \{L, R\}$ and $j \in \{p, a\}$. As a reminder, a is agent, p is partner, L is left hand and R is right hand. The parameters $\mu_{\mathbf{c}}$ and $\Sigma_{\mathbf{c}}$ denote the mean and the covariance of a contact configuration instance. At this point, we can use *Gaussian Mixture Regression* (GMR), which utilizes conditional probabilities and linear combination of Gaussian distributions to factorize this joint distribution. According to this factorization the moments of the Gaussian components can be split into a part for the agent[1]

$$\mu^a = \begin{bmatrix} \mu^{L,a} \\ \mu^{R,a} \end{bmatrix} \quad , \quad \Sigma_a = \begin{bmatrix} \Sigma^{L,a}\Sigma^{L,a} & \Sigma^{R,a}\Sigma^{L,a} \\ \Sigma^{L,a}\Sigma^{R,a} & \Sigma^{R,a}\Sigma^{R,a} \end{bmatrix}, \tag{C.3}$$

and the respective part for the partner

$$\mu^p = \begin{bmatrix} \mu^a \\ \mu^p \end{bmatrix} \quad \text{and} \quad \Sigma^p = \begin{bmatrix} \Sigma^{p,p} & \Sigma^{p,a} \\ \Sigma^{a,p} & \Sigma^{a,a} \end{bmatrix}. \tag{C.4}$$

1 Note that we dropped the symbol $\mathbf{c}$, which denotes the contact location, not to overload the notation.

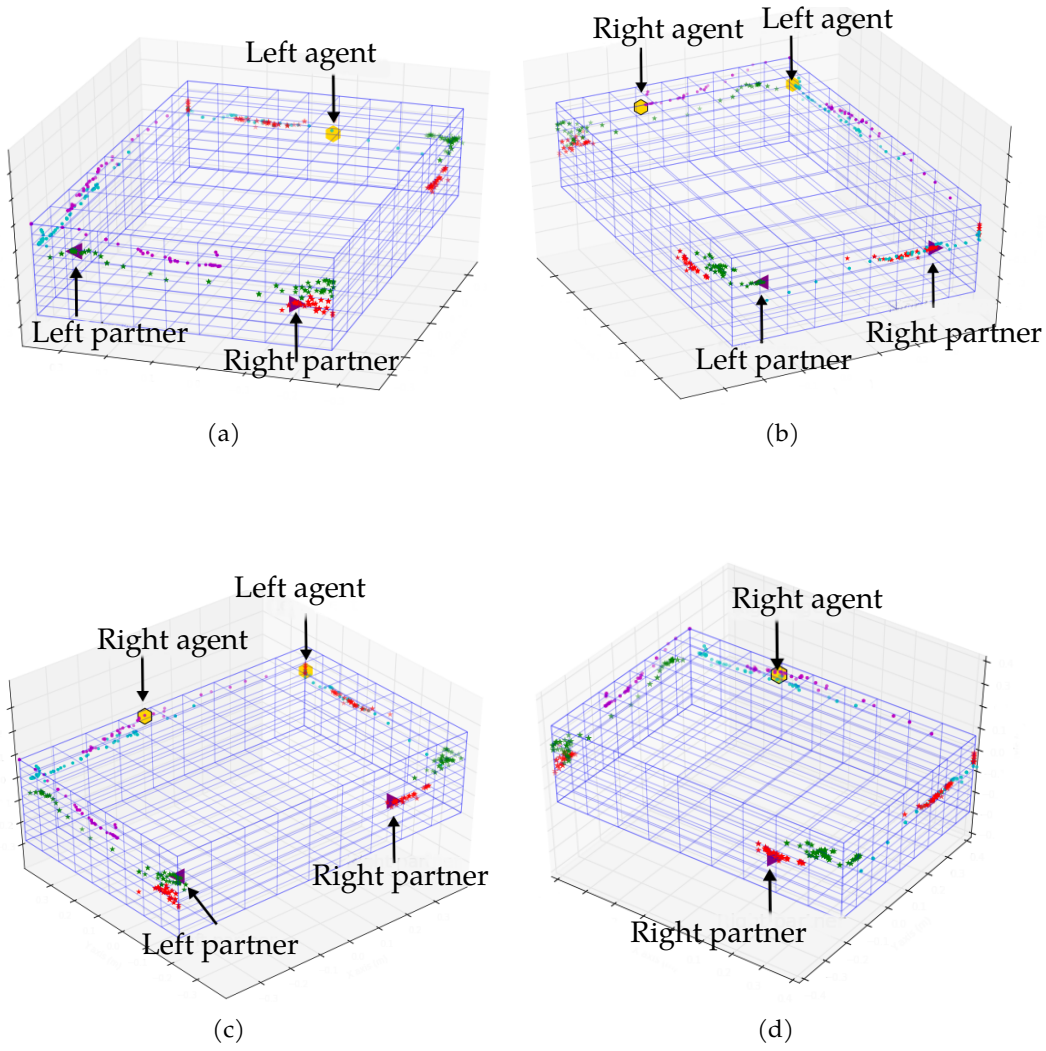


Figure C.4: Four different cases where the agent's contact locations are inferred. The contact regions are green and red for left and right hand of the partner, cyan and magenta for left and right hand of the agent. The test datapoints are annotated with purple triangles and gold hexagons, for the partner and the agent respectively.(a) It is a three-contact case. (b) It is a four-contact case. (c) Another instance of a four-contact case. (d) It is a two-contact case.

Next, given the contact points of the partner $\mathbf{c}^p$, we can predict the respective contact points for the agent with

$$\begin{aligned} \mu^{a|p} &= \mu^a + \Sigma^{a,p}(\Sigma^{p,p})^{-1}(\mathbf{c}^p - \mu^p), \\ \Sigma^{a|p} &= \Sigma^{a,a} - \Sigma^{a,p}(\Sigma^{p,p})^{-1}\Sigma^{p,a}, \end{aligned} \tag{C.5}$$

where $\mu^{a|p}$ can serve as the prediction of $\mathbf{c}^{\kappa,a} = [\mathbf{c}^{L,a\,T}, \mathbf{c}^{R,a\,T}]^T$. Essentially using the GMR, we are able to infer the contact locations of the agent given the contact locations of the partner according to $p(\mathbf{c}^{L,a}, \mathbf{c}^{R,a}|\mathbf{c}^{L,p}, \mathbf{c}^{R,p})$.

The last required step is to group the contact configurations into meaningful sets. The grouping was performed automatically, by counting the number of two-contacts, three-contacts and four-contacts sets, shown in Fig. C.4. Simply by counting the number of contacts and whether this number changes or not in consecutive measurement frames, we automatically segmented the dataset into these three groups. An additional benefit of the automatic segmentation is that the number of Gaussian components required for the fitting of the GMM can be also automatically calculated.

Using the methodology described above, we were able to predict the contact locations of an agent given the partner's contact locations. In Fig. C.4, we visualize a number of different contact scenarios where the contact configurations of the partner are denoted with the purple arrows and the predicted contact points of the agent are shown with the gold hexagons. As we mentioned in Chapter 4, this was an exploratory study towards *emergent coordination* that we decided not to pursue further. The main three hurdles that we faced were: (i) in the general case, the predictions generated from these type methods may not satisfy physical or task constraints, such as the contact locations being on the object surface, yet the work of (Vogt, Stepputtis, Jung, et al., 2018) could be used to address this concern; (ii) the generalisation capabilities—especially for dynamics tasks, like co-manipulation—with respect to the object, environment, partner, etc are typically limited (C.-M. Huang, Cakmak, and Mutlu, 2015; Peternel, Tsagarakis, and Ajoudani, 2017; Vogt, Stepputtis, Grehl, et al., 2017) and (iii) collecting collaborative manipulation data with two humans and an object is technically challenging (Mielke, Townsend, and Killpack, 2017).

D

SENSING HARDWARE

This appendix introduces the sensing hardware used to measure the state of the human partner and the manipulated object. Appendix D.1.1 describes a 3D vision motion capture based system, called VICON, which tracks the position of markers in space. Appendix D.1.2 describes a 3D motion capture system, called Xsens, which utilizes inertial sensors to obtain relative orientation, attitude and position information. Finally, Appendix D.2 provides details on a simple fusion method that can exploit the advantages of both the VICON tracking system and the Xsens motion capture suit. It is important to note here that this work was done in collaboration with Charayaphan Nakorn Boon Han.

D.1 HARDWARE

D.1.1 *VICON*

VICON is a passive optical motion capture system, which uses retroreflective markers that are tracked by infrared cameras. This technology has been initially used for gait analysis and now is widely used in a number of different industries, such as VFX studios, sports therapists, neuroscientists, computer vision and robotics. The system comprises a set of infrared cameras, a synchronization router box (named Lock Sync Box) and a set of retroreflective markers. The 3D position of the markers is obtained via triangulation, while using the proprietary VICON software markers can be grouped in subsets to describe the motion of rigid and articulated bodies. The primary advantage of the VICON optical motion capture system is its accuracy, while its main disadvantage is that it suffers from occlusions.

D.1.2 *Xsens*

Xsens is an inertial motion capture system, which uses inertial measurement units (IMUs) that contain a gyroscope, magnetometer, and accelerometer to measure rotational rates. These rotational rates are used to describe the motion of a skeleton via the Xsens proprietary software. The system comprises a set of IMUs, each IMU captures the full motion of a six degrees of freedom body and the respective data are transmit-

Figure D.1: Xsens and Vicon system used to capture the motion of a human.

ted wirelessly to a pc, where via the Xsens proprietary software these raw data are processed to generate the motion of a human skeleton. The primary advantage of the Xsens inertial motion capture system is its portability, while its main disadvantage is that it suffers from positional accuracy and positional drift.

D.2 HUMAN MOTION TRACKING VIA A FUSION OF VICON AND XSENS

The two motion capture systems described above have complementary strengths, and thus, next we describe a simple sensor fusion approach that can combine their individual strengths. The use of optical sensors enables us to have high accuracy measurements and in combination with the inertial sensors the overall system is robust to occlusions. Fig. D.1 depicts a setup where both the VICON and the Xsens motion capture systems are used. In particular, we use Xsens to capture the state of the human skeleton and the relative motion of the humans hands. In addition, we use VICON to capture the absolute state of the human hands and the state of the manipulated object. To realize the sensor fusion of the XSens and Vicon Motion Capture data we use a Kalman Filter, which is described next.

D.2.1 *Kalman Filter for hand position tracking*

The Kalman Filter (Welch, G. Bishop, et al., 1995) is a method that uses a series of noisy measurements over time to produce improved estimates of a variable of interest (system state). This is achieve via an iterative procedure that consist out of a prediction and an update step. The prediction step aims to predict system state, which in this case is the position of the human hands $\mathbf{c}_\mathbf{t}^\kappa \in \mathbb{R}^3$. To predict the state variable a linear motion model is used, while the update step aims to update $\mathbf{c}_\mathbf{t}^\kappa$ according to a linear observation model and based on a set of measurements obtained from the sensors.

The general idea of our implementation is the following. We build the transition model of $\mathbf{c}_\mathbf{t}^\kappa$ based on consecutive Xsens measurements and we use the VICON motion capture system to inform the measurement model. We manually configure the Kalman filter noise covariances to weight the Vicon observations, denoted with $\bar{\mathbf{c}}_t^\kappa$, as very accurate. In other words, as long as we receive an update from the VICON the update step is performed with very large certainty. In case there is an occlusion of the VICON markers, the update step is performed with the last VICON measurement and with very large uncertainty, which simply neglects the measurement. In parallel, the prediction step is realized based on the difference between two consecutive Xsens measurements, denoted with $\tilde{\mathbf{c}}_t^\kappa$, which enables us to obtain an estimate of the motion of the human hands even when they are occluded.

In more detail, the prediction step of the Kalman filter is written as $\hat{\mathbf{c}}_t^\kappa = \mathbf{A}\mathbf{c}_{t-1}^\kappa + \mathbf{B}\mathbf{u}_{t-1}$, where $\hat{\mathbf{c}}_t^\kappa$ is the estimate of human hand position. By setting $\mathbf{A}$ and $\mathbf{B}$ to be identity matrices and $\mathbf{u} = \tilde{\mathbf{c}}_t^\kappa - \tilde{\mathbf{c}}_{t-1}^\kappa$, we can re-write the prediction step as

$$\hat{\mathbf{c}}_t^\kappa = \mathbf{c}_{t-1}^\kappa + \tilde{\mathbf{c}}_t^\kappa - \tilde{\mathbf{c}}_{t-1}^\kappa.$$

Next, the observation model of the Kalman filter is $\mathbf{z} = \mathbf{H}\mathbf{c}_t^\kappa + \mathbf{v}$, where $\mathbf{z}$ denotes the measurement and $\mathbf{v}$ is zero mean Gaussian white noise. Given that we can obtain direct measurements on the position of the human hands, we set $\mathbf{H}$ to the identity matrix. Hence, the update step of the Kalman filter can be written as

$$\mathbf{c}_t^\kappa = \hat{\mathbf{c}}_t^\kappa + \mathbf{K}_t(\bar{\mathbf{c}}_t^\kappa - \hat{\mathbf{c}}_t^\kappa),$$

where $\mathbf{K}_t$ is the optimal Kalman gain. By selecting appropriately the corresponding covariance matrix of the observation model noise, the Kalman gain can be tuned to dismiss, in case there is an occlusion, or otherwise use the VICON measurements.

E

SUPPLEMENTARY MATERIAL

Five video recordings are included in the provided supplementary material along with two online videos.

1. **SV1**: This video demonstrates hybrid manipulation and co-manipulation motion plans in simulation, presented in Chapter 3 and Chapter 4, as well as real-world experiments, where a robot executes hybrid motion plans to manipulate objects jointly with a human partner, presented in Chapter 4. Format, MP4; size, 178 MB.

2. **SV2**: This video corresponds to the method and results presented in Chapter 3. It briefly introduces the main concept of the impact-aware manipulation and demonstrates impact-aware motion plans in real-world experiments, where the robot halts a 20 Kg object travelling at the speed of 0.66 m/s and 0.88 m/s. Also, the benefit of the impact-aware motion plans is demonstrated via a comparison against the standard KUKA compliance controller. Format, MP4; size, 10 MB.

3. **SV3**: This video corresponds to the techniques and results presented in Chapter 3 and Chapter 5. It demonstrates hybrid motion plans between a human and a robot collaboratively manipulating large objects both in simulation and in real-world experiments. Simulation results include (a) dynamic motion plans with a single agent bimanually handling an object, (b) comparisons between the two levels of the proposed bilevel optimization method, and (c) scenarios where the dyadic goals change in real-time during the execution of the plan. Hardware experiments show scenarios where a human-robot dyad manipulates (a) a box with the human being the follower and (b) a cylinder while the human's goal changes and the robot adapts its hybrid motion plans online. Format, MP4; size, 41 MB.

4. **SV4**: This video includes a presentation of the work presented in Section 3.4 and in Chapter 4. Format, MP4; size, 49 MB.

5. **SV5**: This video includes a presentation of the work presented in Chapter 3 and more specifically in Section 3.5. Format, MP4; size, 33 MB.

6. **OV1** and **OV2**: These videos correspond to the system and results presented in Appendix A. OV1 is available at `https://youtu.be/5jFU7oCP4vk`, demonstrates the various features of the system and displays the validation experiment performed on the arena shown in Fig. A.4. OV2 is a press coverage video from the actual dismantle and dispose challenge described in Appendix A and it is available at `https://youtu.be/iprK25e-uIs`.

BIBLIOGRAPHY

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

Agravante, D. J., A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar (2014). "Collaborative human-humanoid carrying using vision and haptic sensing." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 607–612.

Aine, S., S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev (2016). "Multi-heuristic A*." In: *The International Journal of Robotics Research* 35.1-3, pp. 224–243.

Ajoudani, A., A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib (2017). "Progress and prospects of the human–robot collaboration." In: *Autonomous Robots*, pp. 1–19.

Al Borno, M., M. de Lasa, and A. Hertzmann (2013). "Trajectory Optimization for Full-Body Movements with Complex Contacts." In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 19.8, pp. 1405–1414.

Alami, R., A. Albu-Schäffer, A. Bicchi, R. Bischoff, R. Chatila, A. De Luca, A. De Santis, G. Giralt, J. Guiochet, G. Hirzinger, et al. (2006). "Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges." In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1–16.

Alami, R., J.-P. Laumond, and T. Siméon (1997). *Two manipulation planning algorithms*.

Alami, R., T. Simeon, and J.-P. Laumond (1990). "A geometrical approach to planning manipulation tasks. The case of discrete placements and grasps." In: *In Proceedings International Symposium on Robotics Research*, pp. 113–119.

Albrecht, S. V. and P. Stone (2018). "Autonomous agents modelling other agents: A comprehensive survey and open problems." In: *Artificial Intelligence* 258, pp. 66–95.

Albu-Schaffer, A., C. Ott, U. Frese, and G. Hirzinger (2003). "Cartesian impedance control of redundant robots: Recent results with the DLR-light-weight-arms." In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. Vol. 3. IEEE, pp. 3704–3709.

Albu-Schäffer, A., C. Ott, and G. Hirzinger (2007). "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots." In: *The international journal of robotics research* 26.1, pp. 23–39.

Allport, F. H. (1924). *Social Psychology*. Mifflin Company.

Andersson, J. A. E., J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl (2018). "CasADi – A software framework for nonlinear optimization and optimal control." In: *Mathematical Programming Computation* 11.1, pp. 1–36.

Antsaklis, P. J., J. A. Stiver, and M. Lemmon (1993). "Hybrid system modeling and autonomous control systems." In: *Hybrid Systems*. Ed. by R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 366–392. ISBN: 978-3-540-48060-0.

Atmaca, S., N. Sebanz, W. Prinz, and G. Knoblich (2008). "Action co-representation: the joint SNARC effect." In: *Social neuroscience* 3.3-4, pp. 410–420.

Bajcsy, A., D. P. Losey, M. K. O'Malley, and A. D. Dragan (2017). "Learning robot objectives from physical human interaction." In: *Proceedings of Machine Learning Research* 78, pp. 217–226.

Banala, S. K., S. K. Agrawal, and J. P. Scholz (2007). "Active Leg Exoskeleton (ALEX) for gait rehabilitation of motor-impaired patients." In: *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*. IEEE, pp. 401–407.

Barto, A. G. and S. Mahadevan (2003). "Recent advances in hierarchical reinforcement learning." In: *Discrete event dynamic systems* 13.1-2, pp. 41–77.

Bauer, A., D. Wollherr, and M. Buss (2008). "Human–robot collaboration: a survey." In: *International Journal of Humanoid Robotics* 5.01, pp. 47–66.

Bauza, M. and A. Rodriguez (2018). "GP-SUM. gaussian processes filtering of non-gaussian beliefs." In: *In Proceedings of the 13th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*.

Beetz, M., G. Bartels, A. Albu-Schäffer, F. Bálint-Benczédi, R. Belder, D. Beßler, S. Haddadin, A. Maldonado, N. Mansfeld, T. Wiedemeyer, et al. (2015). "Robotic agents capable of natural and safe physical interaction with human co-workers." In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6528–6535.

Beker, O., C. Hollot, Y. Chait, and H. Han (2004). "Fundamental properties of reset control systems." In: *Automatica* 40.6, pp. 905–915.

Bekkering, H., A. Wohlschlager, and M. Gattis (2000). "Imitation of gestures in children is goal-directed." In: *The Quarterly Journal of Experimental Psychology: Section A* 53.1, pp. 153–164.

Bellman, R. (1966). "Dynamic programming." In: *Science* 153.3731, pp. 34–37.

Bennett, K. M. and U. Castiello (1994). *Insights into the reach to grasp movement*. Elsevier.

Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control*. Vol. 1. Athena scientific Belmont, MA.

Betts, J. T. (2010). *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2nd. SIAM.

Billard, A. and D. Kragic (2019). "Trends and challenges in robot manipulation." In: *Science* 364.6446, eaat8414.

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

Bonkenburg, T. (2016). *Robotics in Logistics. A DPDHL perspective on implications and use cases for the logistics industry*. Tech. rep. DHL Group.

Borrelli, F., A. Bemporad, and M. Morari (2017). *Predictive control for linear and hybrid systems*. Cambridge University Press.

Braun, D. J., F. Petit, F. Huber, S. Haddadin, P. Van Der Smagt, A. Albu-Schäffer, and S. Vijayakumar (2013). "Robots driven by compliant actuators: Optimal control under actuation constraints." In: *IEEE Transactions on Robotics* 29.5, pp. 1085–1101.

Braun, D., M. Howard, and S. Vijayakumar (2012). "Optimal variable stiffness control: formulation and application to explosive movement tasks." In: *Autonomous Robots* 33.3, pp. 237–253.

Bruijn, E. R. de, S. F. Miedl, and H. Bekkering (2011). "How a co-actor's task affects monitoring of own errors: evidence from a social event-related potential study." In: *Experimental brain research* 211.3-4, p. 397.

Bryson, A. E. (2018). *Applied Optimal Control: Optimization, Estimation And Control*. Routledge.

Buck, J. (1988). "Synchronous rhythmic flashing of fireflies. II." In: *The Quarterly review of biology* 63.3, pp. 265–289.
Burdet, E., R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato (2001). "The central nervous system stabilizes unstable dynamics by learning optimal impedance." In: *Nature* 414.6862, pp. 446–449.
Busch, B., G. Maeda, Y. Mollard, M. Demangeat, and M. Lopes (2017). "Postural optimization for an ergonomic human-robot interaction." In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2778–2785.
Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*. John Wiley & Sons.
Calinon, S., F. Guenter, and A. Billard (2007). "On learning, representing, and generalizing a task in a humanoid robot." In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.2, pp. 286–298.
Campbell, J. and H. B. Amor (2017). "Bayesian interaction primitives: A slam approach to human-robot interaction." In: *Conference on Robot Learning*, pp. 379–387.
Cannon-Bowers, J. A. and E. Salas (1998). "Team Performance and Training in Complex Environments: Recent Findings From Applied Research." In: *Current Directions in Psychological Science* 7.3, pp. 83–87.
Carpentier, J., S. Tonneau, M. Naveau, O. Stasse, and N. Mansard (2016). "A versatile and efficient pattern generator for generalized legged locomotion." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3555–3561.
Carruthers, P. and P. K. Smith (1996). *Theories of theories of mind*. Cambridge University Press.
Chavan-Dafle, N., R. Holladay, and A. Rodriguez (2018). "In-hand manipulation via motion cones." In: *Robotics: Science and Systems (RSS)*.
Chen, L., L. F. Figueredo, and M. R. Dogar (2018). "Planning for muscular and peripersonal-space comfort during human-robot forceful collaboration." In: *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 1–8.
Chen, L., L. F. Figueredo, and M. R. Dogar (2020). "Manipulation planning under changing external forces." In: *Autonomous Robots* 44.7, pp. 1249–1269.
Choset, H. M., S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, S. Thrun, and R. C. Arkin (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.
Clark, H. H. (1996). *Using language*. Cambridge university press.
Colgate, E., A. Bicchi, M. A. Peshkin, and J. E. Colgate (2008). "Safety for physical human-robot interaction." In: *Springer handbook of robotics*. Springer, pp. 1335–1348.
Colgate, J. E., M. Peshkin, and S. H. Klostermeyer (2003). "Intelligent assist devices in industrial applications: a review." In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*. Vol. 3. IEEE, pp. 2516–2521.
Colgate, J. E. and M. A. Peshkin (Feb. 23, 1996). "Cobots." US 5952796A.
Converse, S., J. Cannon-Bowers, and E. Salas (1993). "Shared mental models in expert team decision making." In: *Individual and group decision making: Current issues* 221, pp. 221–46.
Dafle, N. C., A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge (2014). "Extrinsic dexterity: In-

hand manipulation with external forces." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1578–1585.
Dai, H., A. Valenzuela, and R. Tedrake (2014). "Whole-body motion planning with centroidal dynamics and full kinematics." In: *International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 295–302.
Dayan, P. and N. D. Daw (2008). "Decision theory, reinforcement learning, and the brain." In: *Cognitive, Affective, & Behavioral Neuroscience* 8.4, pp. 429–453.
De Santis, A., B. Siciliano, A. De Luca, and A. Bicchi (2008). "An atlas of physical human–robot interaction." In: *Mechanism and Machine Theory* 43.3, pp. 253–270.
Deits, R. and R. Tedrake (2014). "Footstep planning on uneven terrain with mixed-integer convex optimization." In: *International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 279–286.
der Spaa, L. v., M. Gienger, T. Bates, and J. Kober (2020). "Predicting and Optimizing Ergonomics in Physical Human-Robot Cooperation Tasks." In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1799–1805.
Diehl, M., H. Bock, H. Diedam, and P.-B. Wieber (2006). "Fast direct multiple shooting algorithms for optimal robot control." In: *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*. Ed. by M. Diehl and K. Mombaur. Lecture Notes in Control and Information Sciences. Springer, pp. 65–93.
Dragan, A. D. (2017). "Robot planning with mathematical models of human state and action." In: *arXiv preprint arXiv:1705.04226*.
Dragan, A. D., K. C. Lee, and S. S. Srinivasa (2013). "Legibility and predictability of robot motion." In: *International Conference on Human-Robot Interaction (HRI)*. ACM/IEEE, pp. 301–308.
Duschau-Wicke, A., J. von Zitzewitz, A. Caprez, L. Lunenburger, and R. Riener (2010). "Path control: a method for patient-cooperative robot-aided gait rehabilitation." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18.1, pp. 38–48.
Dyer, P. and S. McReynolds (1968). "On optimal control problems with discontinuities." In: *Journal of Mathematical Analysis and Applications* 23.3, pp. 585–603.
Dyer, P. and S. R. McReynolds (1970). *The computation and theory of optimal control*. New York, Academic.
Ebert, D. M. and D. D. Henrich (2002). "Safe human-robot-cooperation: Image-based collision detection for industrial robots." In: *IEEE/RSJ international conference on intelligent robots and systems*. Vol. 2. IEEE, pp. 1826–1831.
Edward, J., W. Wannasuphoprasit, and M. Peshkin (1999). "Cobots: Robots For Collaboration With Human Operators." In:
Erlhagen, W. and D. Jancke (2004). "The role of action plans and other cognitive factors in motion extrapolation: A modelling study." In: *Visual Cognition* 11.2-3, pp. 315–340.
Ermolov, I. (2020). "Industrial Robotics Review." In: *Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms*. Ed. by A. G. Kravets. Cham: Springer International Publishing, pp. 195–204.
Escande, A., N. Mansard, and P.-B. Wieber (2014). "Hierarchical quadratic programming: Fast online humanoid-robot motion generation." In: *The International Journal of Robotics Research* 33.7, pp. 1006–1028.
Evrard, P., E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar (2009). "Teaching physical collaborative tasks: Object-lifting case study with a humanoid." In: *2009 9th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 399–404.

Evrard, P. and A. Kheddar (2009). "Homotopy switching model for dyad haptic interaction in physical collaborative tasks." In: *World Haptics 2009-Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, pp. 45–50.

Featherstone, R. (2014). *Rigid Body Dynamics Algorithms*. Springer.

Flanagan, J. R., M. C. Bowman, and R. S. Johansson (2006). "Control strategies in object manipulation tasks." In: *Current opinion in neurobiology* 16.6, pp. 650–659.

Flash, T. and N. Hogan (1985). "The coordination of arm movements: an experimentally confirmed mathematical model." In: *Journal of neuroscience* 5.7, pp. 1688–1703.

Fogassi, L., P. F. Ferrari, B. Gesierich, S. Rozzi, F. Chersi, and G. Rizzolatti (2005). "Parietal lobe: from action organization to intention understanding." In: *Science* 308.5722, pp. 662–667.

Franklin, D. W., G. Liaw, T. E. Milner, R. Osu, E. Burdet, and M. Kawato (2007). "Endpoint stiffness of the arm is directionally tuned to instability in the environment." In: *Journal of Neuroscience* 27.29, pp. 7705–7716.

Gams, A., B. Nemec, A. J. Ijspeert, and A. Ude (2014). "Coupling movement primitives: Interaction with the environment and bimanual tasks." In: *IEEE Transactions on Robotics* 30.4, pp. 816–830.

Ganesh, G., A. Albu-Schäffer, M. Haruno, M. Kawato, and E. Burdet (2010). "Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks." In: *2010 IEEE International Conference on Robotics and Automation*. IEEE, pp. 2705–2711.

Ganesh, G., A. Takagi, R. Osu, T. Yoshioka, M. Kawato, and E. Burdet (2014). "Two is better than one: Physical interactions improve motor performance in humans." In: *Scientific reports* 4, p. 3824.

Garrido-Jurado, S., R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez (2014). "Automatic generation and detection of highly reliable fiducial markers under occlusion." In: *Pattern Recognition* 47.6, pp. 2280–2292.

Ghadirzadeh, A., J. Bütepage, A. Maki, D. Kragic, and M. Björkman (2016). "A sensorimotor reinforcement learning framework for physical Human-Robot Interaction." In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2682–2688.

Gienger, M., D. Ruiken, T. Bates, M. Regaieg, M. Meißner, J. Kober, P. Seiwald, and A.-C. Hildebrandt (2018). "Human-Robot Cooperative Object Manipulation with Contact Changes." In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Gienger, M., M. Toussaint, and C. Goerick (2010). "Whole-body motion planning–building blocks for intelligent systems." In: *Motion Planning for Humanoid Robots*. Springer, pp. 67–98.

Gill, P. E., W. Murray, and M. A. Saunders (2005). "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization." In: *SIAM Review* 47.1, pp. 99–131.

Glassmire, J., M. O'Malley, W. Bluethmann, and R. Ambrose (2004). "Cooperative manipulation between humans and teleoperated agents." In: *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS'04. Proceedings.* IEEE, pp. 114–120.

Goebel, R., R. G. Sanfelice, and A. R. Teel (2009). "Hybrid dynamical systems." In: *IEEE control systems magazine* 29.2, pp. 28–93.

Goertz, R. C., J. H. Grimson, and F. A. Kohut (1961). *Manipulator for slave robot*. US Patent 2,978,118.
Goh, T. B., Z. Li, B. M. Chen, T. H. Lee, and T. Huang (2001). "Design and implementation of a hard disk drive servo system using robust and perfect tracking approach." In: *IEEE Transactions on Control Systems Technology* 9.2, pp. 221–233.
Goodrich, M. A. and A. C. Schultz (2008). *Human-robot interaction: a survey*. Now Publishers Inc.
Goyal, S., A. Ruina, and J. Papadopoulos (1991). "Planar sliding with dry friction part 1. limit surface and moment function." In: *WEAR* 143.2, pp. 307–330.
Gribovskaya, E., A. Kheddar, and A. Billard (2011). "Motion learning and adaptive impedance for robot control during physical interaction with humans." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4326–4332.
Grigore, E. C., A. Roncone, O. Mangin, and B. Scassellati (2018). "Preference-based assistance prediction for human-robot collaboration tasks." In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4441–4448.
Gyugyi, L. (1979). "Reactive power generation and control by thyristor circuits." In: *IEEE Transactions on Industry applications* 5, pp. 521–532.
Haddadin, S., A. Albu-Schaffer, A. De Luca, and G. Hirzinger (2008). "Collision detection and reaction: A contribution to safe physical human-robot interaction." In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3356–3363.
Haddadin, S., A. Albu-Schäffer, and G. Hirzinger (2009). "Requirements for safe robots: Measurements, analysis and new insights." In: *The International Journal of Robotics Research* 28.11-12, pp. 1507–1527.
Haddadin, S., S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer (2012). "On making robots understand safety: Embedding injury knowledge into control." In: *The International Journal of Robotics Research* 31.13, pp. 1578–1602.
Haddadin, S., S. Parusel, R. Belder, and A. Albu-Schäffer (2013). "It is (almost) all about human safety: a novel paradigm for robot design, control, and planning." In: *International Conference on Computer Safety, Reliability, and Security*. Springer, pp. 202–215.
Hansel, D. and H. Sompolinsky (1992). "Synchronization and computation in a chaotic neural network." In: *Physical Review Letters* 68.5, p. 718.
Hargraves, C. and S. Paris (1987). "Direct Trajectory Optimization Using Nonlinear Programming and Collocation." In: *Journal of Guidance, Control, and Dynamics* 10.4, pp. 338–342.
Hart, P. E., N. J. Nilsson, and B. Raphael (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." In: *IEEE Transactions on Systems Science and Cybernetics (TSSC)* 4.2, pp. 100–107.
Hauser, K. (2010). "Task planning with continuous actions and nondeterministic motion planning queries." In: *Proc. of AAAI Workshop on Bridging the Gap between Task and Motion Planning*.
Hillier, F. S. and G. J. Lieberman (2001). *Introduction to Operations Research*. Seventh. New York, NY, USA: McGraw-Hill.
Hirzinger, G., K. Landzettel, D. Reintsema, C. Preusche, A. Albu-Schäffer, B. Rebele, and M. Turk (2005). "Rokviss-robotics component verification on ISS." In: *Proc. 8th Int. Symp. Artif. Intell. Robot. Autom. Space (iSAIRAS)(Munich 2005) p. Session2B*.

Hoare, J. R. and L. E. Parker (2010). "Using on-line Conditional Random Fields to determine human intent for peer-to-peer human robot teaming." In: *International Conference on Intelligent Robots and Systems (IROS)*. Vol. IEEE, pp. 4914–4921.

Hogan, F. R., M. Bauza, and A. Rodriguez (2018). "A Data-Efficient Approach to Precise and Controlled Pushing." In: *Conference on Robot Learning (CoRL)*, pp. 336–345.

Hogan, F. R., E. R. Grau, and A. Rodriguez (2018). "Reactive planar manipulation with convex hybrid MPC." In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 247–253.

Hogan, F. R. and A. Rodriguez (2020). "Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics." In: *Algorithmic Foundations of Robotics XII*. Springer, pp. 800–815.

Hogan, N. (1985). "Impedance control: An approach to manipulation." In: *Journal of dynamic systems, measurement, and control* 107, p. 17.

Hogan, N. and S. P. Buerger (2018). "Impedance and interaction control." In: *Robotics and automation handbook*. CRC press, pp. 375–398.

Huang, C.-M., M. Cakmak, and B. Mutlu (2015). "Adaptive Coordination Strategies for Human-Robot Handovers." In: *Proceedings of Robotics: Science and Systems (RSS)*.

Hung, J. Y., W. Gao, and J. C. Hung (1993). "Variable structure control: A survey." In: *IEEE transactions on industrial electronics* 40.1, pp. 2–22.

Ikemoto, S., H. B. Amor, T. Minato, H. Ishiguro, and B. Jung (2009). "Physical interaction learning: Behavior adaptation in cooperative human-robot tasks involving physical contact." In: *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, pp. 504–509.

Ikuta, K., H. Ishii, and M. Nokata (2003). "Safety evaluation method of design and control for human-care robots." In: *The International Journal of Robotics Research* 22.5, pp. 281–297.

Al-Jarrah, O. M. and Y. F. Zheng (1997). "Arm-manipulator coordination for load sharing using reflexive motion control." In: *Proceedings of International Conference on Robotics and Automation*. Vol. 3. IEEE, pp. 2326–2331.

Jarrassé, N., V. Sanguineti, and E. Burdet (2013). "Slaves no longer: review on role assignment for human–robot joint motor action." In: *Adaptive Behavior* 22.1, pp. 70–82.

Jezernik, S., G. Colombo, and M. Morari (2004). "Automatic gait-pattern adaptation algorithms for rehabilitation with a 4-DOF robotic orthosis." In: *IEEE Transactions on Robotics and Automation* 20.3, pp. 574–582.

Johansson, R. S. and K. J. Cole (1992). "Sensory-motor coordination during grasping and manipulative actions." In: *Current opinion in neurobiology* 2.6, pp. 815–823.

Kaelbling, L. P. and T. Lozano-Pérez (2011). "Hierarchical task and motion planning in the now." In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 1470–1477.

Kaess, M., A. Ranganathan, and F. Dellaert (2008). "iSAM: Incremental smoothing and mapping." In: *IEEE Transactions on Robotics* 24.6, pp. 1365–1378.

Kao, P.-C. and D. P. Ferris (2005). "The effect of movement frequency on interlimb coupling during recumbent stepping." In: *Motor control* 9.2, pp. 144–163.

Kavraki, L. E., P. Svestka, J.-C. Latombe, and M. H. Overmars (1996). "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." In: *IEEE transactions on Robotics and Automation* 12.4, pp. 566–580.

Kelly, M. (2017). "An introduction to trajectory optimization: How to do your own direct collocation." In: *SIAM Review* 59.4, pp. 849–904.
King, J. E., J. A. Haustein, S. S. Srinivasa, and T. Asfour (2015). "Nonprehensile whole arm rearrangement planning on physics manifolds." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2508–2515.
Kirk, D. E. (2004). *Optimal control theory: an introduction*. Courier Corporation.
Knoblich, G., S. Butterfill, and N. Sebanz (2011). "Psychological research on joint action: theory and data." In: *Psychology of learning and motivation*. Vol. 54. Elsevier, pp. 59–101.
Kohler, E., C. Keysers, M. A. Umilta, L. Fogassi, V. Gallese, and G. Rizzolatti (2002). "Hearing sounds, understanding actions: action representation in mirror neurons." In: *Science* 297.5582, pp. 846–848.
Konidaris, G. (2019). "On the necessity of abstraction." In: *Current opinion in behavioral sciences* 29, pp. 1–7.
Koppula, H. S. and A. Saxena (2016). "Anticipating human activities using object affordances for reactive robotic response." In: *IEEE transactions on pattern analysis and machine intelligence* 38.1, pp. 14–29.
Kosuge, K. and N. Kazamura (1997). "Control of a robot handling an object in cooperation with a human." In: *Proceedings 6th IEEE International Workshop on Robot and Human Communication. RO-MAN'97 SENDAI*. IEEE, pp. 142–147.
Krüger, J., T. K. Lien, and A. Verl (2009). "Cooperation of human and machines in assembly lines." In: *CIRP annals* 58.2, pp. 628–646.
Kuffner, J. J. and S. M. LaValle (2000). "RRT-connect: An efficient approach to single-query path planning." In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 2. IEEE, pp. 995–1001.
Kulić, D. and E. Croft (2007). "Pre-collision safety strategies for human-robot interaction." In: *Autonomous Robots* 22.2, pp. 149–164.
Kulić, D. and E. A. Croft (2005). "Safe planning for human-robot interaction." In: *Journal of Robotic Systems* 22.7, pp. 383–396.
Lackner, J. R. and P. Dizio (1994). "Rapid adaptation to Coriolis force perturbations of arm trajectory." In: *Journal of neurophysiology* 72.1, pp. 299–313.
Lamy, X., F. Colledani, F. Geffard, Y. Measson, and G. Morel (2009). "Achieving efficient and stable comanipulation through adaptation to changes in human arm impedance." In: *2009 IEEE International Conference on Robotics and Automation*. IEEE, pp. 265–271.
Lanini, J., H. Razavi, J. Urain, and A. Ijspeert (2018). "Human Intention Detection as a Multiclass Classification Problem: Application in Physical Human–Robot Interaction While Walking." In: *IEEE Robotics and Automation Letters* 3.4, pp. 4171–4178.
Latombe, J.-C. (2012). *Robot motion planning*. Vol. 124. Springer Science & Business Media.
LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press.
Lavalle, S. M. (1998). *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. Tech. rep.
LaValle, S. M. and J. J. Kuffner Jr (2001). "Randomized kinodynamic planning." In: *The international journal of robotics research* 20.5, pp. 378–400.

Lawitzky, M., A. Mörtl, and S. Hirche (2010). "Load sharing in human-robot cooperative manipulation." In: *International Symposium on Robot and Human Interactive Communication RO-MAN*. IEEE, pp. 185–191.
Lee, E., J. Park, K. A. Loparo, C. B. Schrader, and P. H. Chang (2003). "Bang-bang impact control using hybrid impedance/time-delay control." In: *IEEE/ASME transactions on mechatronics* 8.2, pp. 272–277.
LePine, J. A. (2005). "Adaptation of teams in response to unforeseen change: effects of goal difficulty and team composition in terms of cognitive ability and goal orientation." In: *Journal of Applied Psychology* 90.6, p. 1153.
Li, W. and E. Todorov (2004). "Iterative linear quadratic regulator design for nonlinear biological movement systems." In: *ICINCO (1)*, pp. 222–229.
Li, Y., K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu (2015). "Continuous role adaptation for human–robot shared control." In: *IEEE Transactions on Robotics* 31.3, pp. 672–681.
Lii, N. Y., Z. Chen, B. Pleintinger, C. H. Borst, G. Hirzinger, and A. Schiele (2010). "Toward understanding the effects of visual-and force-feedback on robotic hand grasping performance for space teleoperation." In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3745–3752.
Likar, N., B. Nemec, L. Žlajpah, S. Ando, and A. Ude (2015). "Adaptation of bimanual assembly tasks using iterative learning framework." In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 771–776.
Likhachev, M., G. J. Gordon, and S. Thrun (2003). "ARA*: Anytime A* with Provable Bounds on Sub-Optimality." In: *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, pp. 767–774.
Lin, H.-C., J. Smith, K. K. Babarahmati, N. Dehio, and M. Mistry (2018). "A Projected Inverse Dynamics Approach for Multi-arm Cartesian Impedance Control." In: *IEEE International Conference on Robotics and Automation*.
Liu, S., K. Mohta, N. Atanasov, and V. Kumar (2018). "Search-based motion planning for aggressive flight in SE (3)." In: *IEEE Robotics and Automation Letters (RA-L)* 3.3, pp. 2439–2446.
Losey, D. P., M. Li, J. Bohg, and D. Sadigh (2020). "Learning from my partner's actions: Roles in decentralized robot teams." In: *Conference on Robot Learning*. PMLR, pp. 752–765.
Lozano-Pérez, T. and L. P. Kaelbling (2014). "A constraint-based method for solving sequential manipulation planning problems." In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3684–3691.
Lynch, K. M., H. Maekawa, and K. Tanie (1992). "Manipulation and active sensing by pushing using tactile feedback." In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 416–421.
Madan, C. E., A. Kucukyilmaz, T. M. Sezgin, and C. Basdogan (2015). "Recognition of Haptic Interaction Patterns in Dyadic Joint Object Manipulation." In: *IEEE Transactions on Haptics* 8.1, pp. 54–66.
Maeda, G., M. Ewerton, G. Neumann, R. Lioutikov, and J. Peters (2017). "Phase estimation for fast action recognition and trajectory generation in human–robot collaboration." In: *The International Journal of Robotics Research* 36.13-14, pp. 1579–1594.
Maeda, Y., T. Hara, and T. Arai (2001). "Human-robot cooperative manipulation with motion estimation." In: *International Conference on Intelligent Robots and Systems, (IROS)*. IEEE, pp. 2240–2245.

Mainprice, J. and D. Berenson (2013). "Human-robot collaborative manipulation planning using early prediction of human motion." In: *International Conference on Intelligent Robots and Systems* (*IROS*). IEEE, pp. 299–306.
Mainprice, J., R. Hayne, and D. Berenson (2015). "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning." In: *2015 IEEE International Conference on Robotics and Automation* (*ICRA*). IEEE, pp. 885–892.
Marcucci, T., R. Deits, M. Gabiccini, A. Bicchi, and R. Tedrake (2017). "Approximate hybrid model predictive control for multi-contact push recovery in complex environments." In: *IEEE-RAS International Conference on Humanoid Robots* (*Humanoids*). IEEE, pp. 31–38.
Marsh, K. L., M. J. Richardson, and R. C. Schmidt (2009). "Social connection through joint action and interpersonal coordination." In: *Topics in Cognitive Science* 1.2, pp. 320–339.
Mason, M. T. (1986). "Mechanics and planning of manipulator pushing operations." In: *The International Journal of Robotics Research* (*IJRR*) 5.3, pp. 53–71.
Mason, M. T. (1999). "Progress in nonprehensile manipulation." In: *The International Journal of Robotics Research* 18.11, pp. 1129–1141.
Mason, M. T. (2018). "Toward Robotic Manipulation." In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1, pp. 1–28.
Mastalli, C., R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard (2020). "Crocoddyl: An efficient and versatile framework for multi-contact optimal control." In: *2020 IEEE International Conference on Robotics and Automation* (*ICRA*). IEEE, pp. 2536–2542.
Mathieu, J. E., T. S. Heffner, G. F. Goodwin, E. Salas, and J. A. Cannon-Bowers (2000). "The influence of shared mental models on team process and performance." In: *Journal of applied psychology* 85.2, p. 273.
Mayer, R. E. (1992). *Thinking, problem solving, cognition*. WH Freeman/Times Books/Henry Holt & Co.
Merkt, W., V. Ivan, and S. Vijayakumar (2019). "Continuous-time collision avoidance for trajectory optimization in dynamic environments." In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*). IEEE, pp. 7248–7255.
Meyer, M., R. P. van der Wel, and S. Hunnius (2013). "Higher-order action planning for individual and joint object manipulations." In: *Experimental brain research* 225.4, pp. 579–588.
Meyer, M., R. P. van der Wel, and S. Hunnius (2016). "Planning my actions to accommodate yours: joint action development during early childhood." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 371.1693, p. 20150371.
Mielke, E., E. Townsend, and M. Killpack (2017). "Analysis of Rigid Extended Object Co-Manipulation by Human Dyads: Lateral Movement Characterization." In: *Proceedings of Robotics: Science and Systems*. Cambridge, Massachusetts.
Mitrovic, D., S. Klanke, R. Osu, M. Kawato, and S. Vijayakumar (2010). "A computational model of limb impedance control based on principles of internal model uncertainty." In: *PloS one* 5.10, e13601.
Mordatch, I., E. Todorov, and Z. Popović (2012). "Discovery of complex behaviors through contact-invariant optimization." In: *ACM Transactions on Graphics* (*TOG*) 31.4, pp. 43–51.

Mörtl, A., M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche (2012). "The role of roles: Physical cooperation between humans and robots." In: *The International Journal of Robotics Research* 31.13, pp. 1656–1674.

Munzer, T., M. Toussaint, and M. Lopes (2018). "Efficient behavior learning in human–robot collaboration." In: *Autonomous Robots* 42.5, pp. 1103–1115.

Murphy, R. R., K. L. Dreger, S. Newsome, J. Rodocker, E. Steimle, T. Kimura, K. Makabe, F. Matsuno, S. Tadokoro, and K. Kon (2011). "Use of remotely operated marine vehicles at Minamisanriku and Rikuzentakata Japan for disaster recovery." In: *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, pp. 19–25.

Nagurka, M. and S. Huang (2004). "A mass-spring-damper model of a bouncing ball." In: *Proceedings of the 2004 American control conference*. Vol. 1. IEEE, pp. 499–504.

Nakanishi, J., A. Radulescu, D. J. Braun, and S. Vijayakumar (2016). "Spatio-temporal stiffness optimization with switching dynamics." In: *Autonomous Robots* 41.2, pp. 273–291.

Nakanishi, J., A. Radulescu, and S. Vijayakumar (2013). "Spatio-temporal optimization of multi-phase movements: Dealing with contacts and switching dynamics." In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 5100–5107.

Narciso, D. A. C., N. P. Faísca, and V. Dua (2011). "Multiparametric Nonlinear Programming." In: *Multi-Parametric Programming*. John Wiley & Sons, Ltd. Chap. 2, pp. 25–51. ISBN: 9783527631216.

Nau, D. S., V. Kumar, and L. Kanal (1984). "General branch and bound, and its relation to A* and AO*." In: *Artificial Intelligence* 23.1, pp. 29–58.

Nemec, B., N. Likar, A. Gams, and A. Ude (2018). "Human robot cooperation with compliance adaptation along the motion trajectory." In: *Autonomous robots* 42.5, pp. 1023–1035.

Neunert, M., F. Farshidian, A. W. Winkler, and J. Buchli (2017). "Trajectory optimization through contacts and automatic gait discovery for quadrupeds." In: *IEEE Robotics and Automation Letters* 2.3, pp. 1502–1509.

Neunert, M., M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli (2018). "Whole-body nonlinear model predictive control through contacts for quadrupeds." In: *IEEE Robotics and Automation Letters* 3.3, pp. 1458–1465.

Ng, A. Y., S. J. Russell, et al. (2000). "Algorithms for inverse reinforcement learning." In: *Icml*. Vol. 1, p. 2.

Nikolaidis, S., A. Kuznetsov, D. Hsu, and S. Srinivasa (2016). "Formalizing human-robot mutual adaptation: A bounded memory model." In: *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. IEEE Press, pp. 75–82.

Nikolaidis, S., S. Nath, A. D. Procaccia, and S. Srinivasa (2017). "Game-theoretic modeling of human adaptation in human-robot collaboration." In: *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, pp. 323–331.

Nocedal, J. and S. Wright (2006). *Numerical optimization*. Springer Science & Business Media.

Nokata, M., K. Ikuta, and H. Ishii (2002). "Safety-optimizing method of human-care robot design and control." In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. Vol. 2. IEEE, pp. 1991–1996.

Noohi, E., M. Žefran, and J. L. Patton (2016). "A Model for Human–Human Collaborative Object Manipulation and Its Application to Human–Robot Interaction." In: *IEEE Transactions on Robotics* 32.4, pp. 880–896.

Nottensteiner, K., T. Bodenmueller, M. Kassecker, M. A. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas (2016). "A complete automated chain for flexible assembly using recognition, planning and sensor-based execution." In: *Proceedings of ISR 2016: 47st International Symposium on Robotics*. VDE, pp. 1–8.

Nurkanovic, A., S. Albrecht, and M. Diehl (2020). "Limits of MPCC Formulations in Direct Optimal Control with Nonsmooth Differential Equations." In: *European Control Conference (ECC)*. IEEE.

Obhi, S. S. and N. Sebanz (2011). "Moving together: toward understanding the mechanisms of joint action." In: *Experimental Brain Research* 211.3, p. 329.

Önol, A. Ö., P. Long, and T. Padır (2019). "Contact-Implicit Trajectory Optimization Based on a Variable Smooth Contact Model and Successive Convexification." In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2447–2453.

Onol, A. O., P. Long, and T. Padlr (2018). "A comparative analysis of contact models in trajectory optimization for manipulation." In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1–9.

Otani, K., K. Bouyarmane, and S. Ivaldi (2018). "Generating Assistive Humanoid Motions for Co-Manipulation Tasks with a Multi-Robot Quadratic Program Controller." In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3107–3113.

Ott, C., R. Mukherjee, and Y. Nakamura (2010). "Unified impedance and admittance control." In: *2010 IEEE International Conference on Robotics and Automation*. IEEE, pp. 554–561.

Pacherie, E. (2011). "14 The Phenomenology of Joint Action: Self-Agency versus Joint Agency." In: *Joint attention: New developments in psychology, philosophy of mind, and social neuroscience*, p. 343.

Pajarinen, J., V. Kyrki, M. Koval, S. Srinivasa, J. Peters, and G. Neumann (2017). "Hybrid Control Trajectory Optimization under Uncertainty." In: *International Conference on Intelligent Robots and Systems (IROS)*, pp. 5694–5701.

Patel, A., S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler (2019). "Contact-implicit trajectory optimization using orthogonal collocation." In: *IEEE Robotics and Automation Letters* 4.2, pp. 2242–2249.

Pecchinenda, A. (1996). "The affective significance of skin conductance activity during a difficult problem-solving task." In: *Cognition & Emotion* 10.5, pp. 481–504.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Perzanowski, D., A. C. Schultz, and W. Adams (1998). "Integrating natural language and gesture in a robotics domain." In: *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell*. IEEE, pp. 247–252.

Peshkin, M. A., J. E. Colgate, W. Wannasuphoprasit, C. A. Moore, R. B. Gillespie, and P. Akella (2001). "Cobot architecture." In: *IEEE Transactions on Robotics and Automation* 17.4, pp. 377–390.

Peternel, L., N. Tsagarakis, and A. Ajoudani (2017). "A Human-Robot Co-Manipulation Approach Based on Human Sensorimotor Information." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.7, pp. 811–822.

Plaku, E. and G. D. Hager (2010). "Sampling-based motion and symbolic action planning with geometric and differential constraints." In: *2010 IEEE International Conference on Robotics and Automation*. IEEE, pp. 5002–5008.

Posa, M., C. Cantu, and R. Tedrake (2014). "A direct method for trajectory optimization of rigid bodies through contact." In: *The International Journal of Robotics Research (IJRR)* 33.1, pp. 69–81.

Prinz, W. (1997). "Perception and action planning." In: *European journal of cognitive psychology* 9.2, pp. 129–154.

Qi, C. R., H. Su, K. Mo, and L. J. Guibas (2017). "Pointnet: Deep learning on point sets for 3d classification and segmentation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660.

Ratliff, N., M. Zucker, J. A. Bagnell, and S. Srinivasa (2009). "CHOMP: Gradient optimization techniques for efficient motion planning." In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 489–494.

Rawlings, J. B., D. Q. Mayne, and M. Diehl (2017). *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing.

Reed, K. B. and M. A. Peshkin (2008). "Physical collaboration of human-human and human-robot teams." In: *IEEE Transactions on Haptics* 1.2, pp. 108–120.

Riener, R. (2013). "Rehabilitation Robotics." In: *Foundations and Trends® in Robotics* 3.1–2, pp. 1–137.

Rijnen, M., E. de Mooij, S. Traversaro, F. Nori, N. van de Wouw, A. Saccon, and H. Nijmeijer (2017). "Control of humanoid robot motions with impacts: Numerical experiments with reference spreading control." In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4102–4107.

Rijnen, M., A. Saccon, and H. Nijmeijer (2015). "On optimal trajectory tracking for mechanical systems with unilateral constraints." In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 2561–2566.

Roa, M. A. and R. Suárez (2014). "Grasp quality measures: review and performance." In: *Autonomous robots* 38.1, pp. 65–88.

Roveda, L., N. Iannacci, F. Vicentini, N. Pedrocchi, F. Braghin, and L. M. Tosatti (2015). "Optimal impedance force-tracking control design with impact formulation for interaction tasks." In: *IEEE Robotics and Automation Letters* 1.1, pp. 130–136.

Roy, J. and L. L. Whitcomb (2002). "Adaptive force control of position/velocity controlled robots: theory and experiment." In: *IEEE Transactions on Robotics and Automation* 18.2, pp. 121–137.

Rozo Castañeda, L., S. Calinon, D. Caldwell, P. Jimenez Schlegl, and C. Torras (2013). "Learning collaborative impedance-based robot behaviors." In: *the Twenty-Seventh Conference on Artificial Intelligence (AAAI)*. AAAI, pp. 1422–1428.

Ruina, A. L. and R. Pratap (2002). *Introduction to statics and dynamics*. Pre-print for Oxford University Press.

Sadigh, D., S. Sastry, S. A. Seshia, and A. D. Dragan (2016). "Planning for autonomous cars that leverage effects on human actions." In: *Robotics: Science and Systems*. Vol. 2. Ann Arbor, MI, USA.

Sagardia, M., T. Hulin, K. Hertkorn, P. Kremer, and S. Schätzle (2016). "A platform for bimanual virtual assembly training with haptic feedback in large multi-object en-

vironments." In: *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pp. 153–162.
Salas, E., J. A. Cannon-Bowers, and E. L. Blickensderfer (1993). "Team Performance and Training Research: Emerging Principles." In: *Journal of the Washington Academy of Sciences* 83.2, pp. 81–106.
Schmidt, R., M. Bienvenu, P. Fitzpatrick, and P. Amazeen (1998). "A comparison of intra-and interpersonal interlimb coordination: Coordination breakdowns and coupling strength." In: *Journal of Experimental Psychology: Human Perception and Performance* 24.3, p. 884.
Schulman, J., Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel (2014). "Motion planning with sequential convex optimization and convex collision checking." In: *The International Journal of Robotics Research* 33.9, pp. 1251–1270.
Schultz, G. and K. Mombaur (2009). "Modeling and optimal control of human-like running." In: *IEEE/ASME Transactions on mechatronics* 15.5, pp. 783–792.
Sebanz, N., H. Bekkering, and G. Knoblich (2006). "Joint action: bodies and minds moving together." In: *Trends in cognitive sciences* 10.2, pp. 70–76.
Sebanz, N. and G. Knoblich (2009). "Prediction in joint action: What, when, and where." In: *Topics in Cognitive Science* 1.2, pp. 353–367.
Sebanz, N., G. Knoblich, and W. Prinz (2003). "Representing others' actions: just like one's own?" In: *Cognition* 88.3, B11–B21.
Sebanz, N., G. Knoblich, and W. Prinz (2005). "How two share a task: corepresenting stimulus-response mappings." In: *Journal of Experimental Psychology: Human Perception and Performance* 31.6, p. 1234.
Sendhoff, B. and H. Wersing (2020). "Cooperative Intelligence-A Humane Perspective." In: *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*. IEEE, pp. 1–6.
Shah, J. and C. Breazeal (2010). "An Empirical Analysis of Team Coordination Behaviors and Action Planning With Application to Human–Robot Teaming." In: *Human Factors* 52.2, pp. 234–245.
Sheridan, T. B. (1997). "Eight ultimate challenges of human-robot communication." In: *in the 6th International Symposium on Robot and Human Interactive Communication RO-MAN*. IEEE, pp. 9–14.
Siciliano, B. and O. Khatib (2016). *Springer handbook of robotics*. Springer.
Siméon, T., J.-P. Laumond, J. Cortés, and A. Sahbani (2004). "Manipulation planning with probabilistic roadmaps." In: *The International Journal of Robotics Research (IJRR)* 23.7-8, pp. 729–746.
Sleiman, J.-P., J. Carius, R. Grandia, M. Wermelinger, and M. Hutter (2019). "Contact-Implicit Trajectory Optimization for Dynamic Object Manipulation." In: *2018 International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1–8.
Slotine J J, S. B. (1991). "A general framework for managing multiple tasks in highly redundant robotic systems." In: *proceeding of 5th International Conference on Advanced Robotics*. Vol. 2, pp. 1211–1216.
Srivastava, S., E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel (2014). "Combined task and motion planning through an extensible planner-independent interface layer." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 639–646.

Stückler, J. and S. Behnke (2011). "Following human guidance to cooperatively carry a large object." In: *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 218–223.
Sung, G. T. and I. S. Gill (2001). "Robotic laparoscopic surgery: a comparison of the da Vinci and Zeus systems." In: *Urology* 58.6, pp. 893–898.
Sutton, R. S. and A. G. Barto (2018). *Reinforcement learning: An introduction*. MIT press.
Takagi, A., G. Ganesh, T. Yoshioka, M. Kawato, and E. Burdet (2017). "Physically interacting individuals estimate the partner's goal to enhance their movements." In: *Nature Human Behaviour* 1, p. 0054.
Takagi, A., F. Usai, G. Ganesh, V. Sanguineti, and E. Burdet (2018). "Haptic communication between humans is tuned by the hard or soft mechanics of interaction." In: *PLOS Computational Biology* 14.3, pp. 1–17.
Tassa, Y., N. Mansard, and E. Todorov (2014). "Control-limited differential dynamic programming." In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1168–1175.
Tee, K. P., E. Burdet, C.-M. Chew, and T. E. Milner (2004). "A model of force and impedance in human arm movements." In: *Biological cybernetics* 90.5, pp. 368–375.
Thobbi, A., Y. Gu, and W. Sheng (2011). "Using human motion estimation for human-robot cooperative manipulation." In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2873–2878.
Todorov, E. (2006). "Optimal control theory." In: *Bayesian brain: probabilistic approaches to neural coding*, pp. 269–298.
Tonneau, S., A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard (2018). "An Efficient Acyclic Contact Planner for Multiped Robots." In: *IEEE Transactions on Robotics (T-RO)* 34.3, pp. 586–601.
Toussaint, M. (2015). "Logic-geometric programming: An optimization-based approach to combined task and motion planning." In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
Toussaint, M. (2017). "A tutorial on Newton methods for constrained trajectory optimization and relations to SLAM, Gaussian Process smoothing, optimal control, and probabilistic inference." In: *Geometric and numerical foundations of movements*. Springer, pp. 361–392.
Toussaint, M., K. Allen, K. Smith, and J. B. Tenenbaum (2018). "Differentiable physics and stable modes for tool-use and manipulation planning." In: *Robotics: Science and Systems (RSS)*.
Toussaint, M., L. Jung-Su, and D. Danny (2020). "Describing Physics For Physical Reasoning:Force-based Sequential Manipulation Planning." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.
Toussaint, M. and M. Lopes (2017). "Multi-bound tree search for logic-geometric programming in cooperative manipulation domains." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4044–4051.
Turski, M. R., J. Norby, and A. M. Johnson (2020). "Contact-Implicit vs. Hybrid Trajectory Optimization: Performance Comparison." In: *Dynamic Walking*.
Umilta, M. A., E. Kohler, V. Gallese, L. Fogassi, L. Fadiga, C. Keysers, and G. Rizzolatti (2001). "I know what you are doing: A neurophysiological study." In: *Neuron* 31.1, pp. 155–165.
Unhelkar, V. V. and J. A. Shah (2019). "Learning models of sequential decision-making with partial specification of agent behavior." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 2522–2530.

Utkin, V. I. (2013). *Sliding modes in control and optimization*. Springer Science & Business Media.
Valenzuela, A. K. (2016). "Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain." PhD thesis. Massachusetts Institute of Technology.
Vallery, H., A. Duschau-Wicke, and R. Riener (2009a). "Generalized elasticities improve patient-cooperative control of rehabilitation robots." In: *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*. IEEE, pp. 535–541.
Vallery, H., A. Duschau-Wicke, and R. Riener (2009b). "Optimized passive dynamics improve transparency of haptic devices." In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, pp. 301–306.
Verfaillie, K. and A. Daems (2002). "Representing and anticipating human actions in vision." In: *Visual Cognition* 9.1-2, pp. 217–232.
Vijayakumar, S., A. D'souza, and S. Schaal (2005). "Incremental online learning in high dimensions." In: *Neural computation* 17.12, pp. 2602–2634.
Vogt, D., S. Stepputtis, S. Grehl, B. Jung, and H. B. Amor (2017). "A system for learning continuous human-robot interactions from human-human demonstrations." In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, pp. 2882–2889.
Vogt, D., S. Stepputtis, B. Jung, and H. B. Amor (2018). "One-shot learning of human–robot handovers with triadic interaction meshes." In: *Autonomous Robots* 42.5, pp. 1053–1065.
Wächter, A. and L. T. Biegler (2006). "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." In: *Mathematical Programming* 106.1, pp. 25–57.
Wang, S. and K. Hauser (2018). "Unified Multi-Contact Fall Mitigation Planning for Humanoids via Contact Transition Tree Optimization." In: *International Conference on Humanoid Robots (Humanoids)*, pp. 1–9.
Wang, Y. and V. Chiew (2010). "On the cognitive process of human problem solving." In: *Cognitive systems research* 11.1, pp. 81–92.
Welch, G., G. Bishop, et al. (1995). *An introduction to the Kalman filter*.
Wenke, D., S. Atmaca, A. Holländer, R. Liepelt, P. Baess, and W. Prinz (2011). "What is shared in joint action? Issues of co-representation, response conflict, and agent identification." In: *Review of Philosophy and Psychology* 2.2, pp. 147–172.
Williams, C. K. and C. E. Rasmussen (2006). *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA.
Wimpenny, J. H., A. A. Weir, L. Clayton, C. Rutz, and A. Kacelnik (2009). "Cognitive processes associated with sequential tool use in New Caledonian crows." In: *PLoS One* 4.8, e6471.
Winkler, A. W., C. D. Bellicoso, M. Hutter, and J. Buchli (2018). "Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization." In: *IEEE Robotics and Automation Letters (RA-L)* 3.3, pp. 1560–1567.
Winkler, A. W., C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini (2015). "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain." In: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5148–5154.
Wolbrecht, E. T., V. Chan, D. J. Reinkensmeyer, and J. E. Bobrow (2008). "Optimizing compliant, model-based robotic assistance to promote neurorehabilitation." In:

IEEE Transactions on Neural Systems and Rehabilitation Engineering 16.3, pp. 286–297.

Wolpert, D. M., K. Doya, and M. Kawato (2003). "A unifying computational framework for motor control and social interaction." In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 358.1431, pp. 593–602.

Wu, Z., S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao (2015). "3d shapenets: A deep representation for volumetric shapes." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920.

Yanco, H. A. and J. Drury (2004). "Classifying human-robot interaction: an updated taxonomy." In: *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*. Vol. 3. IEEE, pp. 2841–2846.

Yanco, H. A. and J. L. Drury (2002). "A taxonomy for human-robot interaction." In: *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*. sn, pp. 111–119.

Yang, C., G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet (2011). "Human-like adaptation of force and impedance in stable and unstable interactions." In: *IEEE transactions on robotics* 27.5, pp. 918–930.

Yang, C., P. Liang, A. Ajoudani, Z. Li, and A. Bicchi (2016). "Development of a robotic teaching interface for human to human skill transfer." In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 710–716.

Yang, Y., V. Ivan, W. Merkt, and S. Vijayakumar (2016). "Scaling sampling-based motion planning to humanoid robots." In: *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, pp. 1448–1454.

Yang, Y., W. Merkt, V. Ivan, Z. Li, and S. Vijayakumar (2017). "HDRM: A resolution complete dynamic roadmap for real-time motion planning in complex scenes." In: *IEEE Robotics and Automation Letters* 3.1, pp. 551–558.

Zhou, J., Y. Hou, and M. T. Mason (2019). "Pushing revisited: Differential flatness, trajectory planning, and stabilization." In: *The International Journal of Robotics Research (IJRR)* 38.12-13, pp. 1477–1489.

Zhou, J., R. Paolini, A. M. Johnson, J. A. Bagnell, and M. T. Mason (2017). "A probabilistic planning framework for planar grasping under uncertainty." In: *IEEE Robotics and Automation Letters (RA-L)* 2.4, pp. 2111–2118.

Zhu, S., S. Zwiebel, and G. Bernhardt (1999). "A theoretical formula for calculating damping in the impact of two bodies in a multibody system." In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 213.3, pp. 211–216.

Zinn, M., B. Roth, O. Khatib, and J. K. Salisbury (2004). "A new actuation approach for human friendly robot design." In: *The international journal of robotics research* 23.4-5, pp. 379–398.

Zucker, M., N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa (2013). "Chomp: Covariant hamiltonian optimization for motion planning." In: *The International Journal of Robotics Research* 32.9-10, pp. 1164–1193.