Technical University of Munich
TUM School of Computation, Information and Technology

Predictable Motion Generation and Optimal Motion Control Design for Human Robot Collaboration

Khoi Hoang Dinh

Complete reprint of the dissertation approved by the TUM School of Computation, Information and Technology of the Technical University of Munich for the award of the

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

Chair: Prof. Dr.-Ing. Klaus Diepold

Examiners:

1. Priv.-Doz. Dr.-Ing. habil. Dirk Wollherr
2. Asst. Prof. Dr. Özgür S. Öğüz

The dissertation was submitted to the Technical University of Munich on 19.10.2023 and accepted by the TUM School of Computation, Information and Technology on 23.04.2024.

I dedicate my dissertation work to my family and many friends, whose unwavering support and encouragement have been instrumental in my academic journey. A special feeling of gratitude goes to my loving parents, Kien Hoang Dinh and Xuan Nguyen Thi, whose words of encouragement and unwavering belief in my abilities continue to resonate within me. To my beloved wife, Yen Nhi, and my son, Khoi Nguyen, your presence has been my source of strength and inspiration, and you have stood by my side through every challenge.

Preface

I would like to extend my heartfelt gratitude to everyone who played a significant role in the completion of this dissertation. The support, guidance, and valuable feedback from my esteemed colleagues, mentors, students, and friends have been invaluable in shaping the level of detail and scope of this work. While there are countless individuals who have inspired me over the years, I would like to express special thanks to a few who have made a profound impact on my research journey.

First and foremost, I am deeply thankful to my supervisor PD Dr.-Ing. habil. Dirk Wollherr for accepting me as a PhD candidate at the Chair of Automatic Control Engineering (LSR), TUM. His unwavering support and belief in my potential have been instrumental in conducting research on human-robot interaction. His guidance and continuous feedback throughout the series of experiments have been invaluable in shaping the direction of this study. Moreover, I am grateful for his assistance in navigating the administrative aspects, including the extension of my scholarship.

I am indebted to PD Dr.-Ing. habil. Marion Leibold for her pivotal role in developing the fundamental ideas of my work. Our numerous discussions and talks during my time at LSR have been enlightening and have enriched the content of this dissertation.

A heartfelt thank you goes to all my teammates in the Siemens project. The collaborative spirit, co-designing of experiments, co-authorship, and joint participation at conferences have provided a wealth of creative input that has contributed to the development of the core ideas and processes. I wish to express special appreciation to Volker Gabler, my roommate, and Gerold Huber for their invaluable support and constructive feedback, and for the cherished memories we have shared.

I am also grateful to the students who worked on their theses as part of my research project. Phillip Weiler, Bjoern Milke, and Mariam Elsayed, your dedication and motivation in exploring the topics I provided have been integral to the success of this dissertation.

Finally, I would like to extend my sincere thanks to Ms. Larissa Schmid, the secretary at LSR, for her unwavering assistance with all the administrative procedures that I encountered throughout this journey. Her support has been invaluable in ensuring a smooth and efficient process.

To all those mentioned and to many others who have contributed in various ways, I offer my profound gratitude. Your encouragement, feedback, and collaboration have been essential in making this dissertation a reality.

Vietnam, October 2023 Hoang Dinh, Khoi

Acknowledgments

The research leading to these results has received funding from the Siemens AG project.

Zusammenfassung

Diese Arbeit beschäftigt sich mit der Herausforderung, Roboter in die mensch-roboterische Zusammenarbeit in gemeinsamen Arbeitsbereichen zu integrieren und konzentriert sich auf zwei Hauptziele: die Generierung vorhersehbarer Roboterbewegungen während der Interaktion mit Menschen und die Gestaltung eines schnellen, optimalen Controllers, der in der Lage ist, gegebene Bewegungen auszuführen und sich dabei an Umgebungsänderungen und Aufgaben anzupassen. Die wachsende Nachfrage in der Industrie erfordert, dass Roboter als gleichwertige Teammitglieder neben Menschen arbeiten und sie in denselben Arbeitsbereichen unterstützen. Infolgedessen müssen die Bewegungen des Roboters vorhersehbar sein, um eine reibungslose Zusammenarbeit zu ermöglichen und gleichzeitig die Sicherheit der Menschen in allen Phasen der Planung und Ausführung zu gewährleisten. Frühere Forschungsergebnisse haben hauptsächlich die Vorhersagbarkeit von Bewegungen in isolierten Fällen ohne mensch-roboterische Interaktion untersucht, weshalb es entscheidend ist, einen Rahmen zu entwickeln, der die Vorhersagbarkeit von Bewegungen durch Interaktion ermöglicht. Die Ausführungsphase erfordert, dass der Roboter gegebene Trajektorien optimal ausführt und dabei Genauigkeits-, Kosten- und Einschränkungskriterien erfüllt, während er schnell auf Umgebungsänderungen reagiert, um die Sicherheit der Menschen zu gewährleisten. Ein Controller, der all diese Bedingungen vollständig erfüllt, ist jedoch in der aktuellen Literatur noch nicht ausreichend erforscht. Diese Arbeit zielt darauf ab, diese Forschungslücke zu schließen, indem sie verschiedene Ansätze vorstellt, die die Anwesenheit von Robotern in kollaborativen Umgebungen erleichtern, die Vorhersagbarkeit verbessern und effiziente und sichere mensch-roboterische Interaktionen in gemeinsamen Arbeitsbereichen gewährleisten.

Die Arbeit beginnt mit der Einführung eines Ansatzes in Kapitel 2, um die Vorhersage der menschlichen Bewegung in das Ausweichverhalten des Roboters zu integrieren und damit dessen Reaktionsfähigkeit in beengten Arbeitsbereichen zu verbessern. Die Auswirkungen dieses Ansatzes werden durch reale Experimente und Fallstudien mit Menschen bewertet (Kapitel 2.5). Aufbauend auf diesen Ergebnissen wird ein Framework vorgestellt, um vorhersehbare Roboterbewegungen durch Interaktion zu generieren und dabei das verbesserte Ausweichverhalten zu nutzen (Kapitel 3). Das vorgeschlagene Framework wird gründlich getestet und verifiziert, sowohl in virtuellen als auch in realen Szenarien, ergänzt durch verschiedene Fallstudien mit Menschen, die verschiedene Aspekte der mensch-roboterischen Interaktion untersuchen, um die Wirksamkeit zu validieren (Kapitel 3.5). Um die Anpassungsfähigkeit und Anwendbarkeit des Frameworks weiter zu verbessern, wird eine Methode zur Aufgabenverallgemeinerung vorgestellt, die den Transfer gelernter vorhersehbarer Bewegungspolicies von einer Aufgabe auf eine andere ermöglicht (Kapitel 3.4). Dies erhöht die Vielseitigkeit des Ansatzes und ermöglicht eine breitere Palette von Aufgaben und Anwendungen. Abschließend wird der TC-SAC Controller eingeführt, ein schneller und nahezu optimaler Ansatz, der entwickelt wurde, um gegebene Robotertrajektorien ab der Planungsphase auszuführen (Kapitel 4). Der TC-SAC Controller wird eingehend bewertet und mit dem ursprünglichen SAC sowie anderen indirekten optimalen Steuerungsmethoden verglichen, um Einblicke in seine Wirksamkeit und Leistungsfähigkeit zu liefern (Kapitel 4.5). Ein umfassender Stabilitätsbeweis für TC-SAC wird ebenfalls vorgelegt, um dessen Robustheit und Zuverlässigkeit zu gewährleisten (Kapitel 4.6). Diese kombinierten Bemühungen tragen dazu bei, die Herausforderungen in der mensch-roboterischen Zusammenarbeit und der Bewegungssteuerung zu bewältigen und eine effizientere, sicherere und intuitivere Interaktion zwischen Menschen und Robotern in gemeinsamen Arbeitsbereichen zu fördern.

Abstract

This thesis addresses the challenge of integrating robots into human-robot collaboration within shared workspaces, focusing on two main objectives: generating predictable robot motion during interactions with humans and designing a fast, optimal controller capable of executing given motions while adapting to environmental changes and tasks. The growing demand in industry necessitates robots to work alongside humans as equal team members, supporting them in the same workspace. Consequently, the motions of the robot must be predictable to facilitate seamless teamwork while ensuring human safety throughout both the planning and execution phases. Previous research has mainly explored predictable motion in isolated cases without human-robot interactions, making it crucial to develop a framework capable of generating predictable motion through interaction. The execution phase requires the robot to perform given trajectories optimally, meeting accuracy, cost, and constraint criteria, while responding quickly to environmental changes to maintain human safety. However, such a controller that fully satisfies all these conditions remains underexplored in the current literature. This thesis aims to fill this research gap by presenting diverse approaches that facilitate robot presence in collaborative settings, enhance predictability, and ensure efficient and safe human-robot interactions in shared workspaces.

The thesis begins by introducing an approach in Chapter 2 to integrate human motion prediction into the obstacle avoidance behavior of the robot, thereby enhancing its responsiveness in confined workspaces. The impact of this approach is assessed through real experiments and human case studies (Chapter 2.5). Building upon these findings, a framework is introduced to generate predictable robot motion through interaction, leveraging the improved obstacle avoidance behavior (Chapter 3). The proposed framework is rigorously tested and verified through a series of experiments in both virtual reality and real-world scenarios, complemented by various human case studies that explore different aspects of human-robot interactions to validate its effectiveness (Chapter 3.5). To further enhance the adaptability and applicability of the framework, a task generalization method is presented, enabling the transfer of learned predictable motion policies from one task to another (Chapter 3.4). This enhances the versatility of the approach, allowing it to accommodate a wider range of tasks and applications. Finally, the thesis introduces the TC-SAC controller, a fast and close-to-optimal method designed to execute given robot trajectories from the planning phase (Chapter 4). The TC-SAC controller is rigorously evaluated, being compared to the original SAC as well as other indirect optimal control methods, providing insights into its effectiveness and performance (Chapter 4.5). A comprehensive stability proof of TC-SAC is also presented to ensure its robustness and reliability (Chapter 4.6). These combined efforts contribute to addressing the challenges in human-robot collaboration and motion control, fostering a more efficient, safe, and intuitive interaction between humans and robots within shared workspaces.

Contents

List of Figures xi

List of Tables xiii

List of Algorithms xiii

1 Introduction 1
1.1 Motivation and Focus of the Thesis . . . 8

Part I Obstacle Avoidance and Predictable Motion Generation 17

Nomenclature of Part I 19

2 An Approach to Integrate Human Motion Prediction into Local Obstacle Avoidance in Close Human-Robot Collaboration 21
2.1 Introduction . . . 21
2.2 Related Work . . . 22
2.3 Design of the system architecture . . . 23
2.3.1 Way-point trajectory generation . . . 25
2.3.2 Compliance control . . . 28
2.3.3 Local collision avoidance . . . 28
2.3.4 Human motion prediction . . . 29
2.4 Implementation . . . 30
2.4.1 Damping factors of the compliance control . . . 31
2.4.2 Integration of human motion prediction into the compliance control . . . 32
2.5 Evaluation of the proposed framework in HRC scenario . . . 34
2.5.1 Safety Aspects . . . 34
2.5.2 Case Study . . . 34
2.5.3 Results . . . 35
2.6 Discussion and Conclusion . . . 36

3 Adaptation and Transfer of Robot Motion Policies for close Proximity Human-Robot Interaction 39
3.1 Introduction . . . 40
3.2 Related Work . . . 41
3.3 Legible Motion Framework in HRC in Close Proximity . . . 43
3.3.1 Dynamic Movement Primitives . . . 44
3.3.2 Policy Improvement through Black-box Optimization . . . 46
3.3.3 Safety Aspect in Close Proximity . . . 48
3.3.4 Cost Computation . . . 50
3.4 Task Generalization . . . 51
3.5 Results . . . 52
3.5.1 Experimental Setup in Virtual Reality . . . 53
3.5.2 Predictable Robot Motion for a specific Setup . . . 55
3.5.3 Task Generalization Evaluation . . . 59

3.5.4 Experimental Results on a Real Robot . . . 62
3.6 Discussion and conclusion . . . 65
3.6.1 Discussion . . . 65
3.6.2 Conclusion . . . 67

Part II Motion Control for Robots in Dynamic Environments 69

Nomenclature of Part II 71

4 Fast and Close to Optimal Receding Horizon Controller For Articulated Robots in Reaching Motions 73
4.1 Introduction . . . 74
4.2 Related Work . . . 76
4.3 Problem Formulation . . . 77
4.4 Target Constrained Sequential Action Control . . . 78
4.4.1 First-order gradient algorithm (FOGA) . . . 79
4.4.2 Sequential Action Control . . . 81
4.4.3 Extended Sequential Action Control with target constraints . . . 82
4.5 Trajectory Tracking Simulation and Comparison to Optimal Control . . . 84
4.5.1 Comparison of the proposed approach to other optimal-based control approaches . . . 84
4.5.2 Trajectory tracking in dynamic environment of a car-like system . . . 90
4.6 Stability Analysis . . . 96
4.6.1 Stability of FOGA . . . 96
4.6.2 Stability of TC-SAC . . . 99
4.7 Discussion . . . 99

5 Conclusion and Outlook 103
5.1 Summary and Discussion . . . 103
5.2 Outlook . . . 107

A Appendix 109

List of Figures

1.1 Applications of robots in industry . . . 2
1.2 An example of Human-Robot Collaboration in shared workspace . . . 6
1.3 HRC framework in joint collaboration scenarios . . . 7

2.1 Illustrative setup of the experiment . . . 24
2.2 Exponential characteristics of the velocity, acceleration and jerk . . . 26
2.3 Velocity, acceleration and jerk for one point-to-point motion . . . 27
2.4 Repulsive force profile . . . 29
2.5 Proposed system architecture . . . 31
2.6 Illustrative setup of the experiment with an example of a predicted motion . . . 32
2.7 The compliance response to an obstacle within the range of avoidance . . . 33
2.8 Boxplot of the questionnaire . . . 35
2.9 Comparison of total experiment time and number of iterations of the robot . . . 36

3.1 Pipeline of the human-guided policy improvement framework . . . 43
3.2 Experiment setup in virtual reality with different configurations . . . 53
3.3 Predictability evaluation from all subjects for each phase . . . 56
3.4 The mean and confidence interval of the total cost and human prediction time cost for all subjects . . . 56
3.5 Comparison of the total cost and human prediction time between adaptive robot and non-adaptive robot . . . 58
3.6 Different configurations of the robot goals . . . 59
3.7 Converged trajectories from different subjects and configurations . . . 60
3.8 Robot trajectories in the task generalization experiment . . . 61
3.9 Cost plots that show the difference between the control group that interacted with the untrained robot and the results for the interaction with the trained robot . . . 62
3.10 Real experiment setup on a KUKA LWR 4+ robot . . . 63
3.11 Results of the real experiment on a *KUKA LWR 4+* robot . . . 64

4.1 TC-SAC concept . . . 78
4.2 Setup of the two degrees of freedom robot used for the simulation . . . 84
4.3 States of 2DOF robotic arm when the designed position is upright . . . 85
4.4 States of 2DOF robotic arm in the case of arbitrary desired position . . . 86
4.5 Tracking performance of TC-SAC and FOGA . . . 87
4.6 States of 2DOF robotic arm in case of tracking an ellipse trajectory . . . 87
4.7 Control signal of the 2DOF robotic arm . . . 88
4.8 States of 2DOF robotic arm with obstacle avoidance . . . 89
4.9 Single Track Model . . . 90
4.10 Square path tracking with different horizons . . . 91
4.11 Static Obstacle Avoidance of TC-SAC in car-like system . . . 93
4.12 Other states of the single track model on a Lissajous curve trajectory . . . 94
4.13 Dynamic Obstacle Avoidance of TC-SAC . . . 95

List of Tables

4.1 Total cost and computation time. 85
4.2 Computation time of TC-SAC with different prediction horizons 92

List of Algorithms

1 TC-SAC 83

1 Introduction

Robots have been successfully employed in various industries for several decades. Within these environments, robots are primarily assigned to perform dangerous and repetitive tasks or operate in hazardous areas. Their role is to enhance productivity and carry out work that poses risks to humans. The success of robots in the current industrial landscape is evident in the substantial increase in their utilization across different sectors. This can be attributed to the numerous advantages they offer

- Robots excel in producing accurate and high-quality work. They demonstrate a remarkable level of precision and rarely make mistakes, surpassing human workers in terms of reliability. Moreover, robots exhibit the ability to generate a greater quantity of output within a shorter time frame. They operate at a consistent speed without breaks, days off, or holidays. Additionally, robots possess superior repeatability compared to humans when it comes to performing tasks.

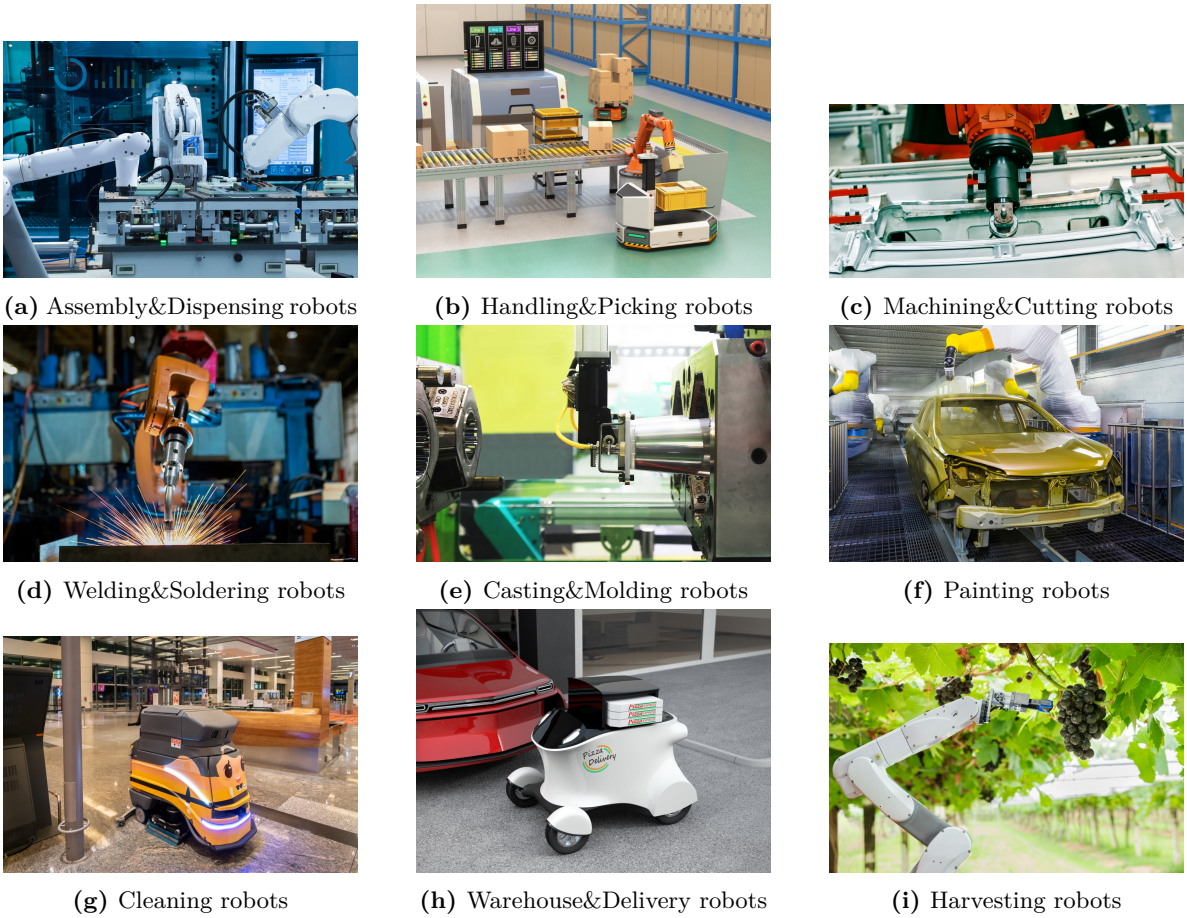
- Robots play a pivotal role in safeguarding workers by alleviating them from engaging in perilous tasks. They can operate effectively in hazardous conditions, such as environments with poor lighting, toxic chemicals, or confined spaces. Furthermore, robots are capable of lifting heavy loads without risking injury or fatigue. By assuming these risky responsibilities, robots significantly enhance worker safety by minimizing the occurrence of accidents caused by humans engaging in dangerous jobs.

- Robots contribute to time savings through their capacity to produce a larger volume of products. Additionally, their accuracy reduces the amount of wasted materials. In the long run, companies benefit financially from using robots due to quick return on investment (ROI), reduced worker injuries, and decreased material consumption. These cost-saving measures positively impact the overall profitability of businesses.

The increased utilization of robots in various industries has led to a significant surge in their numbers, owing to the multitude of benefits they offer. In manufacturing, for instance, robots are extensively employed to execute monotonous tasks such as material handling, processing operations, assembly, and inspection. Similarly, in the realm of agriculture, robots are harnessed to automate a diverse range of operations, including pruning, thinning, mowing, spraying, and weed removal. Moreover, within warehouse environments, robots play a crucial role in facilitating the pick-up, placement, and transportation of products between staging areas. Figure 1.1 provides a visual representation of several applications wherein robots find utility within the industrial domain.

However, as factories and plants experience rapid development, robots are confronted with increasingly sophisticated demands and requirements. Traditional robots, in particular, are approaching their limitations due to the disadvantages and drawbacks they encounter. One of the primary challenges lies in the high investment costs associated with establishing a robotic

assembly line, coupled with limited flexibility in its applications. These costs encompass installation, maintenance, additional components, programming, and more. Furthermore, each assembly line is typically designed for a specific set of tasks, necessitating a complete overhaul of the line if tasks need to be changed. Consequently, the cost of replacements and the time investment required for new tasks substantially inflate overall expenses.



(a) Assembly&Dispensing robots **(b)** Handling&Picking robots **(c)** Machining&Cutting robots

(d) Welding&Soldering robots **(e)** Casting&Molding robots **(f)** Painting robots

(g) Cleaning robots **(h)** Warehouse&Delivery robots **(i)** Harvesting robots

Figure 1.1: Applications of robots in industry[1]

Moreover, robots are confined to performing simple and repetitive tasks, rendering them unsuitable for more complex endeavors that demand high dexterity or on-the-fly adaptability. For instance, tasks such as sewing, which require intricate hand movements, or tasks involving object manipulation that necessitate real-time adjustments, are still beyond the capabilities of robots. In essence, robots are most effective in specific, narrowly defined tasks aimed at saving time and reducing labor, but their capabilities are limited in broader contexts. Additionally, robots have long been viewed as potential sources of danger on the factory floor. Given their substantial size and bulkiness, coupled with their ability to move at high speeds, robots pose inherent risks. Moreover, their limited sensory capabilities often render them incapable of detecting nearby humans, making them prone to hazardous collisions and accidents. Consequently, many manufacturers resort to implementing physical barriers or partitions to segregate robots from their human co-workers, ensuring safety within the workplace.

In light of the aforementioned challenges, it becomes increasingly evident that traditional

[1]Source: `https://www.howtorobot.com/expert-insight/industrial-robot-applications`

robots face formidable obstacles in meeting the evolving demands of modern factories and plants. The limitations related to cost, task complexity, and safety concerns highlight the urgent need for innovative approaches and advancements in robotic technology to effectively address these shortcomings. Consequently, there has been a recent shift in focus towards exploring the potential of robots to collaborate and work alongside human partners within a shared workspace. This transition is driven by the rapidly changing landscape of production automation, fueled by the growing demand for increased product variations and shorter product life cycles across multiple industries.

As the industry moves away from mass production towards individualized manufacturing tailored to specific customer requirements, a new and challenging research field emerges: the development of a new breed of robots capable of safely operating alongside humans as equal partners within a shared environment. This concept, often referred to as Human-Robot Collaboration (HRC) or Human-Robot Team (HRT), envisions a scenario where humans and robots seamlessly collaborate, combining their respective strengths to achieve optimal productivity and efficiency.

The concept of HRC stems from the recognition that both robots and humans possess different capabilities and limitations that can complement each other. Robots excel in executing fast, precise, repetitive, and heavy-duty tasks in manufacturing, but they lack the flexibility and adaptability exhibited by humans. Consequently, robots are often confined to specific tasks and require costly reconfiguration when transitioning to new ones. On the other hand, humans are highly flexible, capable of quickly adapting to different situations, and can collaborate with others to accomplish more complex tasks. However, humans have physical limitations and are prone to errors in repetitive tasks, which restricts the range of tasks they can perform. Therefore, the concept of HRC aims to integrate the repeatability and accuracy of robots with the flexibility and adaptability of humans. By combining the strengths of both entities, HRC endeavors to meet the new demands of the industry, resulting in enhanced productivity, improved task variety, and optimized utilization of human and robotic capabilities in industrial settings.

Several recent research studies have been conducted on various aspects of HRC. One early survey by Bauer et al. [BWB08] provides valuable insights into the concept of HRC and identifies the necessary components for its successful implementation. The survey comprehensively discusses classical robots, cognitive sciences, and psychology as essential components for HRC, highlighting their roles in facilitating effective collaboration. Moreover, the survey explores various methods, including intention estimation, action planning, joint action, and machine learning, which are considered crucial for the development of collaborative robots. The authors also address the significance of designing robot appearance to enhance human comfort, emphasizing the importance of creating a conducive and ergonomic environment for collaboration. Additionally, the paper acknowledges the vital aspect of safety in HRC and examines different techniques and approaches for ensuring safe physical interaction between humans and robots. It covers topics such as collision avoidance, force control, and compliant behavior, emphasizing the need for robust safety measures to prevent accidents and injuries throughout the HRC process.

In addition to Bauer et al., other literature presents different viewpoints and categorizations of human-robot collaboration (HRC). Hentout et al. [HAMA19] propose categorizing HRC into human-robot coexistence, human-robot cooperation, and human-robot collaboration, further dividing them into physical collaborations and contact-less collaborations. They provide distinct categories that capture the range of interactions between humans and robots. Zaatari et al. [EZMLU19] categorize HRC into independent, simultaneous, sequential, and

supportive levels, illustrating the evolving behaviors of the robot as it transitions from working independently alongside the human to actively supporting the human in the same task. Their categorization emphasizes the different modes of collaboration and the varying degrees of robot involvement. Adding to this discussion, Wang et al. [WLLW20] offer a more recent analysis that examines the relationship between humans and robots from five perspectives: workspace, direct contact, working task, simultaneous process, and sequential process. Through this analysis, they classify the levels of collaboration between humans and robots, spanning from mere coexistence to deeper collaboration levels, where joint activities and a shared working environment become essential to accomplish common tasks. Furthermore, these papers explore various characteristics required for successful HRC, such as flexibility and adaptability in configuration, the capability to support and assist human co-workers, intuitive interfaces for human-robot interaction or multimodal communication, and ensuring safety for human partners.

On one hand, Haddadin et al. [HC16] support a categorization based on the physical proximity between humans and robots. According to their interpretation, cooperative robot interactions occur in closer proximity compared to collaborative robot interactions. Therefore, human-robot cooperation is characterized by the closest possible distance between a robot and a human, while human-robot coexistence occurs when they are farthest apart. On the other hand, Kolbeinsson et al. [KLL19] emphasize that HRC is determined by how humans and robots share their workspace and tasks. They view HRC as more immersive than human-robot cooperation, highlighting the level of integration between humans and robots in their shared environment. Furthermore, Villani et al. [VPLS18] conducted a survey specifically focused on safety aspects and user interfaces in industrial robotic applications. They introduce various ISO safety standards that correspond to different operative modes of robots, reflecting the levels of collaboration between humans and robots. These modes range from monitoring, where either the human or the robot operates in the shared workspace at a time, to hand guiding, where humans physically teach robot positions without the need for an intermediate interface, and finally to the highest level known as power and force limiting. In the power and force limiting mode, the motor power and force of the robots are restricted to ensure safe coexistence with humans. The paper provides an overview of risk assessment methodologies, safety monitoring systems, and collision detection and avoidance techniques. It underscores the importance of advanced safety measures, including the use of force and torque sensors, to enhance the ability of robots to detect and respond to the presence of humans.

Regarding the safety aspect in HRC, the authors in [HASH07] delve into the topic of ensuring safe physical interaction between humans and robots. The authors emphasize the significance of accurately measuring and analyzing forces and contacts during collaborative tasks to ensure the safety of humans involved. They present novel measurement techniques and analysis methods that enable a comprehensive understanding of the physical interactions. By gaining insights into the forces exerted by humans and robots, the paper explores the development of control strategies that allow robots to actively respond and adapt to these forces, minimizing the risk of injuries. The authors also propose the concept of "soft robotics" as a means to design robots with compliant and safe physical properties. In another work by De Luca and Flacco [DLF12], the focus is on the integrated control of physical human-robot interaction (pHRI). The paper addresses multiple aspects of pHRI, including collision avoidance, detection, reaction, and collaboration, which are crucial for ensuring safe and efficient interactions between humans and robots. The authors propose control strategies and algorithms that enable robots to actively detect and avoid collisions with humans, react appropriately to unexpected interactions, and collaborate effectively in shared workspaces.

They emphasize the importance of integrating these control aspects to achieve seamless and safe human-robot collaboration.

In [BLOD17], the authors approach the topic of human-robot interaction from a different perspective, highlighting the intentional and informative nature of physical interaction during collaboration. They propose a framework that enables a human user to physically interact with a robot, demonstrating desired behaviors. The robot then learns from these interactions and infers the objectives the human intends to achieve. The paper addresses the challenge of accurately inferring human objectives and presents an algorithm that combines inverse reinforcement learning and reinforcement learning to learn from physical demonstrations. Building upon this work, the subsequent paper in [LBOD22] extends the framework by utilizing physical interaction as a form of communication to refine robot objectives. By incorporating corrections provided by the human, the robot can learn and adapt its objectives in an iterative manner.

Another recent survey in [HC22] discusses different key components necessary for a safe and successful implementation of HRC in the assembly and disassembly process. The authors focus on two main components: control strategies and human-robot communication techniques. Control strategies are divided into pre-collision and post-collision categories. Pre-collision control strategies aim to prevent harmful contact between the robot and its surrounding environment and can be referred to as obstacle avoidance [SNB19, CS18]. Post-collision control strategies aim to limit the contact force and energy exchange between the human and the robot to a safe level [YC16, LLH09, GS17, AAG$^+$19]. Human-robot communication represents the ways in which humans communicate with robots on the production floor. It is further divided into *verbal* communication, where voice and speech recognition play key roles, and *non-verbal* communication, which includes gesture recognition, human pose and skeleton tracking, gaze detection, intention recognition, and more. The paper also discusses task sequences related to different states of a product and how to distribute tasks between humans and robots to achieve optimal collaboration.

In addition to the aforementioned works, there are several other studies that approach HRC from different perspectives depending on the specific applications in which robots are integrated. Nevertheless, it can be concluded that the primary objective of HRC is to transform outdated and unresponsive robots into collaborative robots, commonly referred to as cobots. The aim is to enable efficient and intuitive human-robot collaboration within shared workspaces. To illustrate this goal further, consider a scenario in which humans and robots collaborate on joint assembly tasks, such as constructing a LEGO bridge as depicted in Figure 1.2. In such cases, various questions arise that require further investigation:

- *How to distribute tasks in human-robot teams?*
- *How to control and monitor the task execution?*
- *How to mutually understand and interpret movements and how to derive symbolic intentions?*
- *How to achieve reactive behavior of the robot, in particular to inconsistencies and errors?*

Under cooperation, humans and robots can coordinate their actions effectively, functioning as a cohesive team. To achieve this, robots must possess decision-making capabilities that enable them to choose cooperative actions from a set of ongoing tasks. However, several challenges arise in this context due to information uncertainties and rapidly increasing problem complexity in real-world scenarios. Information uncertainties stem from the lack of precise

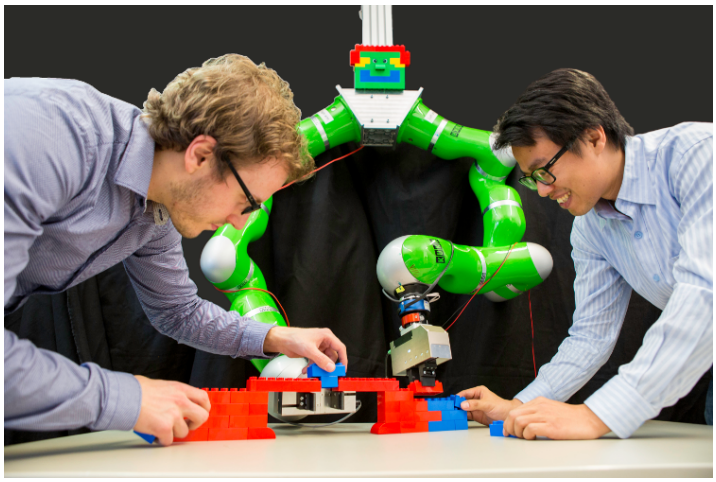


Figure 1.2: An example of Human-Robot Collaboration in shared workspace

knowledge about the current state of the environment, including the spatial position of task-relevant objects. This uncertainty arises from residual errors in sensor evaluation and the inability to fully observe the activities of human team members. Monitoring progress and handling errors during collaborative human-robot assembly are also crucial. Observing human actions and assigning them to individual elementary actions facilitates progress monitoring, and in the event of discrepancies or errors, action distribution among the agents can be recalculated, leading to the selection of a new action sequence.

Furthermore, the movements of robots need to be observed and planned to ensure safety and comfort for human partners during collaboration. Safety entails avoiding collisions between robots and humans while performing tasks. However, in shared workspaces, encounters between humans and robots are likely, requiring robots to dynamically adapt their motions while maintaining task efficiency. Additionally, robots need to move in a manner that instills comfort in human partners, encouraging them to approach and collaborate. This enhances the effectiveness of collaboration and increases success in joint tasks. Achieving this level of comfort requires the robot to understand and interpret human movements and actions in order to plan accordingly. Additionally, the robot must perform movements that allow for appropriate human interpretation.

Considering all the aforementioned challenges, we propose a comprehensive framework for joint collaboration between humans and robots, as illustrated in Figure 1.3. The framework consists of five major interdependent components categorized into two levels: the *cognitive reflection* level, which includes *perception & understanding* and *autonomous task allocation*, and the *execution* level, comprising *adaptive motion planning* and *versatile manipulation*.

Starting from the high-level concept of *cognitive reflection*, the *autonomous task allocation* component is responsible for the high-level interpretation of tasks and the autonomous decision making process in which tasks are efficiently allocated among the robot and human partners. This task allocation process faces challenges as it needs to adapt to changes in the environment and dynamically respond to human partners. The outcome of this component is the selection of actions by the robot that leads to the best collaborative performance.

The *perception & understanding* component focuses on the human side of HRC. To design robot co-workers that are accepted by humans, it is vital for the robot to comprehend human motion behavior in different settings. This component is responsible for modeling and learn-

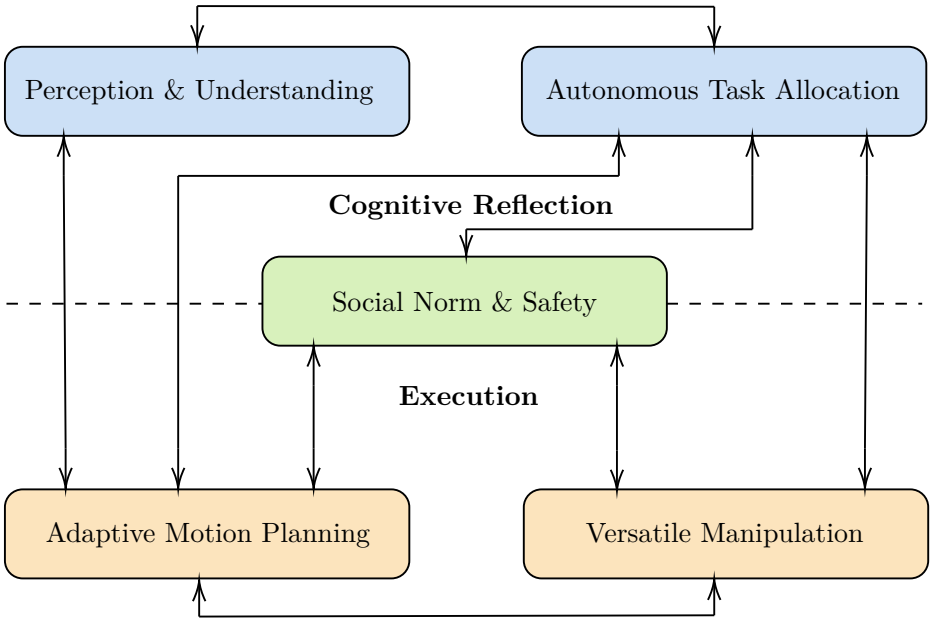


Figure 1.3: HRC framework in joint collaboration scenarios

ing human motion, their interaction with the robot and the surrounding environment, and integrating these factors to predict human motion behaviors in various HRC contexts. The outcome of this component facilitates a faster and more intuitive task allocation process and enables comfortable movements of the robot.

The *adaptive motion planning* component belongs to the *execution* level of the proposed HRC framework. Its objective is to develop a trajectory planner for the robot that not only ensures proper task execution but also provides comfort and acceptance to human collaborators. To achieve this, the planner considers the dynamics of the environment, task variations, and predictions of human motion to adapt the trajectories of the robot in real-time. This adaptation ensures safety for human partners during task execution and makes the movements of the robot predictable and understandable, facilitating a sense of collaboration and partnership between humans and the robot.

The *versatile manipulation* component aims to equip the robot with a diverse range of manipulation skills, enabling it to work effectively alongside human co-workers. In addition to standard tasks like picking and placing objects, the robot requires other cooperative skills such as guidance and handover, etc. These skills enhance the capabilities of the robot and broaden its range of tasks. A smooth and natural handover movement by the robot further enhances the comfort of human partners.

The *social norm & safety* component focuses on fault handling and addressing violations of social norms in joint assembly scenarios. Faults or errors occur when the robot deviates from the normal routine and behaves unexpectedly, leading to disruptions in the production process. This component explores strategies for handling faults, errors, and collisions that may occur during human-robot collaboration.

All five components are vital for enabling human-robot collaboration in industrial assembly and encompass a wide range of research topics currently under investigation by scientists and researchers. Within the proposed framework, this thesis primarily focuses on the *adaptive motion planning* component within the *execution* level of robots in HRC environments.

Specifically, the study aims to investigate how robot motions can be planned and controlled to execute tasks in dynamic environments following task distribution from the *cognitive reflection* level. The motivation, focus, and contributions of this thesis will be discussed in detail in the following sections.

1.1 Motivation and Focus of the Thesis

The areas of motion planning and control have long been subjects of research and exploration in both academia and industry within the field of robotics. These components are crucial for enabling robots to operate effectively in industrial settings and various other applications. Although there is some overlapping between them, they are usually categorized into two separated problems

1. *Motion planning*: refers to path planning and trajectory planning which involve the procedure of finding a collision-free path/trajectory for the robots. This process operates at the kinematic level, generating outputs such as position, velocity, acceleration, and jerk. A path represents a geometric representation, while a trajectory includes additional information on velocities, accelerations, and jerk along the path. When a path is provided, trajectory planning is often used to generate trajectories that enable robots to follow the given path while satisfying specific criteria, such as maximum velocities, accelerations, or minimum execution time.

2. *Motion control*: refers to the procedure of designing a controller such that the robots can execute the given path/trajectory properly without violating any requirements or constraints. This is done at the dynamic level, providing outputs in the form of force or torque that can be directly applied to the robot.

Both motion planning and motion control problems can be further classified into two types of algorithms: offline and online. Offline methods produce static results, and the calculations are performed prior to execution. In contrast, online methods allow for the recalculation of the path, trajectory, or controller in real-time, enabling the motion of the robot to be adjusted dynamically to react and adapt to changing environments.

In the context of this thesis, the focus lies on the *adaptive motion planning* component within the *execution* level of the proposed HRC framework. Consequently, both motion planning and motion control problems need to be explored, particularly in the online scenario. Therefore, we will first provide a brief overview of the current state-of-the-art in these two areas. Subsequently, we will discuss the motivation and specific focus of this thesis.

Path planning is a foundational problem within the field of motion planning, as it involves finding a geometric representation of a plan to move from a starting pose to a target pose. It is a computational process aimed at determining a collision-free path for a robot, allowing it to navigate from its initial configuration to a goal configuration while avoiding obstacles, static objects, as well as other robots or humans in the environment. The exploration of path planning began in the 1970s and has since evolved to address various challenges, ranging from simple 2-dimensional route planning for mobile robots to complex movements for articulated robots.

One of the early methods in this domain is the Dijkstra algorithm [Dij59] and its variants, which have found applications in diverse fields, including traffic routing systems like Google Maps [ZTC12]. The Dijkstra algorithm searches for the shortest path in an acyclic environment between two points. It selects the unvisited vertex with the lowest distance, calculates

the distance through it to each unvisited neighbor, and updates the neighbor's distance if it is smaller. The algorithm maintains newly discovered vertices in a priority-min queue, resulting in the determination of a single shortest path. Various modified versions of Dijkstra's algorithm have been developed to address different use cases and applications.

Despite its usefulness, the Dijkstra algorithm has certain limitations. It is memory-intensive [FJ09], as it needs to search through all possible outcomes to identify the shortest path, and it cannot handle negative edges. Consequently, alternative memory schemes [FAMJ15] and solutions to handle significant cost factors [GA20] have been proposed. Overall, the Dijkstra algorithm is best suited for static environments and global path planning, where most of the required data can be computed in advance.

Alternative approaches such as A* [HNR68] and its variants have been introduced in order to tackle the computational burden of Dijkstra's algorithm. A* follows a similar approach to Dijkstra's algorithm by constructing a lowest-cost path tree from the start to the end point. However, A* improves upon Dijkstra's algorithm by guiding its search towards the most promising direction, thereby significantly reducing computation time [FLS05]. A* is commonly employed in applications with available data sets or nodes, particularly in static environments. A variant of A* that incorporates moving obstacles into its calculations is the dynamic A* algorithm, also known as D* [HBA14]. D* generates optimal traverses in real-time by considering a graph with dynamically changing or updated arc costs. It takes advantage of the fact that most arc cost corrections occur near the vicinity of the robot, allowing for re-planning of only the portion of the path from these corrections to the current location. D* maintains a partial, optimal cost map focused on locations relevant to the robot, minimizing the need for extensive repairs. The conditions for determining when repairs can be halted are established by D*, whether it be due to finding a new optimal path or confirming that the old one remains optimal. Consequently, D* exhibits high computational and memory efficiency, and it is capable of operating in unbounded environments. However, one drawback of D* is that the simulation time increases as the complexity of the problem grows.

For robots with high degrees-of-freedom (DOF), such as articulated robots (typically more than six DOFs, like *KUKA* robots) and humanoid robots with over twenty DOFs, sampling-based algorithms have revolutionized the field of motion planning. Two well-known approaches in this regard are probabilistic roadmap (PRM) [KSLO96] and rapidly exploring random trees (RRT) [LaV98]. RRTs employ a dynamic expansion strategy wherein the tree grows incrementally from the starting point towards the goal point. In each iteration, a randomly generated vertex is selected and the tree expands towards its nearest vertex based on a specified distance metric. Notably, RRTs exhibit a bias towards unexplored regions of the configuration space, as they heavily expand in these uncharted territories. The vertices of an RRT follow a uniform distribution, ensuring a relatively simple algorithmic structure while maintaining connectivity, even with a minimal number of edges. Alternatively, PRM explores the configuration space using a distinct approach. It involves randomly sampling configurations within the space and testing them for collision with obstacles. These collision-free configurations are then connected to nearby configurations using a local planner. By adding the starting and goal configurations, a graph is constructed to represent feasible motions within the environment. The PRM planner consists of two distinct phases: the construction phase and the query phase. During the construction phase, the roadmap (graph) is incrementally built by adding configurations and establishing connections until the graph achieves the desired level of density. In the subsequent query phase, the start and goal configurations are connected to the graph, and a Dijkstra's shortest path query is employed to determine the optimal path between them.

These methods have been proven to be highly effective in overcoming the curse of dimensionality. The curse of dimensionality refers to the exponential increase in computation time as the number of DOFs of the robots increases, making traditional methods like Dijkstra's algorithm and A* impractical for higher-dimensional environments. In contrast, sampling-based algorithms rely on a collision checking module to evaluate the feasibility of sampled paths/trajectories and connect them to form a graph-like roadmap of viable trajectories. Although these methods do not fully represent the environment, they offer a relaxed guarantee known as probabilistic completeness, ensuring that a solution, if it exists, will be found as the number of samples approaches infinity. As a result, sampling-based methods have demonstrated great effectiveness in high-dimensional motion planning and have continued to attract the attention of researchers, leading to recent variants and improvements [KWP+11, KFT+08, WvdB13, BOvdS99, SLCS04].

The early works in motion planning, as discussed previously, primarily focused on finding paths or trajectories that adhere to environmental constraints. However, these methods faced inherent limitations that hindered their further extension. One major challenge was the difficulty in considering complex constraints imposed by robot dynamics, although some approaches were capable of handling simple kinodynamic constraints such as velocity or acceleration bounds [CRR91, LJJK01]. Another significant limitation of most sampling-based methods, including RRT and its variants, was their lack of consistency. Consistency refers to the ability to produce the same trajectories for two queries with identical start and goal states. Sampling-based planning, known for its lack of consistency, restricts its application in scenarios where trajectory predictability is crucial. Furthermore, evolving industrial requirements demanded more than just feasible solutions; the quality of the generated solution also became crucial. In many industrial applications, users sought solution paths that minimized execution time to maximize productivity. This gave rise to the challenging problem of optimizing the motion of the robot, which involves computing motion plans that minimize a given cost functional, such as path length or energy consumption. Even in basic cases, this problem has proven to be highly challenging to traditional approaches [CR87].

Trajectory optimal motion planning (TOMP) emerged as a subsequent solution to generate optimal paths or trajectories, aimed at enhancing overall performance. TOMP serves different purposes depending on the objectives of the problem. For instance, in scenarios where an initial trajectory is provided, TOMP is employed as a post-processing phase to optimize the given trajectory or removing dynamically infeasible segments. Conversely, in cases where no path or trajectory is provided, TOMP acts as an optimal planner, searching for collision-free trajectories while simultaneously optimizing them according to a given objective. The latter case presents additional challenges that require solving an extra problem. Extensive research has been conducted employing diverse optimization and optimal control techniques to effectively address the multitude of challenges that arise from the evolving demands in the fields of robotics and industry. Minimizing jerk, energy consumption, and execution time are commonly used as design objectives, while the joint configurations and motor torques of the robot serve as frequently employed design variables [YPW19]. If motor torques are used as design variables, the problem falls under the category of *optimal control*, which will be discussed in detail later.

In general, TOMP works by solving a constrained non-convex optimization problem within the trajectory space, treating the trajectory as a sequence of states. As mentioned earlier, the typical design objectives in TOMP involve minimizing jerk for smooth trajectory execution, as well as reducing energy consumption and execution time to enhance efficiency. One of the key advantages of TOMP over traditional motion planning methods lies in its ability

to incorporate constraints into the optimization process. This capability allows TOMP to tackle high-level complex problems that traditional approaches struggle to handle effectively. Inequality constraints commonly encountered in TOMP include obstacle avoidance, joint limits, and inverse singularity, while equality constraints encompass aspects such as end-effector positions and orientations, as well as the kinematics and dynamics of the controlled system.

Several TOMP approaches adopt a waypoint-based representation for trajectories, where each waypoint corresponds to a specific robot configuration. These methods leverage cost-gradient information to drive the minimization process. Notable examples of such techniques include CHOMP (Covariant Hamilton Optimization Motion Planning) [ZRD+13], STOMP (Stochastic Trajectory Optimization for Motion Planning) [KCT+11], and Trajopt (Trajectory Optimization) [SDH+14]. CHOMP begins by creating an initial path from the start to the end position, which is then optimized using gradient descent to obtain a smooth and collision-free trajectory. While CHOMP efficiently handles high-dimensional cases, it requires the cost function to be differentiable, which limits its applicability to a narrower range of scenarios. STOMP takes a different approach to address the limitations of CHOMP. It explores the space around an initial trajectory by generating noisy trajectories and combines them to produce an updated trajectory with a lower cost. The cost function in STOMP encompasses both obstacle and smoothness costs, without the requirement of gradient information. This characteristic allows STOMP to accommodate more general cost functions, including cases where derivatives may not be available, such as those involving motor torque constraints. Trajopt formulates trajectory planning as an optimization problem and employs sequential convex optimization techniques to solve it. A notable strength of Trajopt lies in its efficient formulation of the no-collision constraint, which accounts for continuous time safety. This formulation enables Trajopt to reliably solve motion planning problems, even in scenarios involving thin and complex obstacles. These methods, including their variants, have garnered significant attention, finding application and extensions in numerous scenarios and domains [BBSF14, LL22, DMDB16, MHB15]. They excel at planning smooth and optimal trajectories in high-dimensional spaces, such as six degrees of freedom or more, while still considering various constraints.

However, it is important to note that these methods do not explicitly consider the physical body or dynamical model of the controlled system. The output of CHOMP, STOMP, Trajopt, and similar techniques consists solely of a sequence of waypoints, providing information such as position, velocity, acceleration, jerk, etc. Consequently, when applying these approaches to real robots, an additional controller is required to guide the robot in following this sequence of waypoints effectively.

In the domain of optimal control, extensive and closely related work exists, which primarily focuses on systems with complex dynamical properties. In these approaches, the dynamical system is treated as equality constraints within the optimal control problem, and the outputs are forces or torques directly applied to the robot. Such methods fall under the category of *motion control*. It is worth noting that solving an optimal control problem generally demands more computational effort compared to the optimization procedure of TOMP, particularly when dealing with high DOF robots. This increased computational complexity arises due to the quadratic scaling of the dynamic model with the number of DOFs of the robot. Consequently, a key challenge in this area revolves around efficiently solving the optimal control problem to reduce computation time for the algorithms.

The two most common methods employed to tackle the optimal control problem are known as indirect and direct methods [VSB92]. Indirect methods, pioneered by Pontryagin, have played a significant role in the applications of space manufacturing, largely due to their high

precision. These methods rely on the first-order necessary conditions summarized in Pontryagin's minimum principle and strive to identify control and state trajectories that satisfy these conditions. Direct methods, introduced later, apply numerical analysis to discretize the cost function and constraints directly, thereby transforming the optimal control problem into a nonlinear programming problem. Both indirect and direct methods have been successfully applied and integrated into various robotic systems and applications.

For instance, in [ADRVM13], indirect methods were utilized to minimize the execution time between two configurations, and the approach was tested on a *PUMA 560* robot. Gregory et al. [GOS12] also employed an indirect method to solve an optimal control problem, resulting in an optimal energy-consumption trajectory for a two-revolute planar manipulator. In [PDE+14], the authors aimed to improve the energy efficiency of a robot using direct methods through the ACADO toolkit [HFD11] for trajectory optimization. The experiments were conducted on a real *IRB1600* industrial ABB robot. Through the optimal control approach, they achieved up to a 5% energy improvement compared to most trajectories generated by the ABB software. In [KNGD02], a different direct method called SNOPT was employed to calculate trajectories for four-wheeled omnidirectional vehicles. Optimal control has also been utilized in various other applications, such as finding smooth trajectories for each foot of a service robot [MPMK17], minimizing energy and jerk in controlling a 6-DOF chain manipulator [GZ07], optimizing gaits for a 1.5m-tall and 39kg-weight bipedal robot [CEA16], and minimizing fuel consumption in space robots [CMW+17].

Recently, learning-based methods have been integrated into the optimization procedures for solving optimal control problems. These methods include particle swarm optimization using population-based stochastic approaches [K.17, CSZT16, MCTK17, DBP16, AS17], artificial bee colonization inspired by the foraging behavior of bee swarms [GSK15], ant colony optimization built upon the foraging performance of ant colonies [BEL17], as well as hybrid methods combining genetic algorithms with traditional direct methods [EMA17, SSD14, SD15], among others. However, a major drawback of learning-based methods is the lack of stability proof due to their probabilistic nature. Despite numerous modifications and improvements, optimal control methods, in general, still require a significant amount of computational effort, limiting their applicability to offline planning or applications that do not necessitate on-the-fly trajectory recalculations.

Model Predictive Control (MPC) [DBDW07] has emerged as a highly sought-after approach in dealing with optimal control problems in online scenarios. Stemming from the optimal control background, MPC treats trajectory planning as an optimal control problem. However, it differs from traditional methods by focusing on finding a close-to-optimal solution within a short horizon. The optimal control inputs are applied to the system, and the process is iteratively repeated by updating the state estimate, predicting future states, optimizing control inputs, and applying them to the system.

MPC exhibits two key advantages over optimal control methods. Firstly, it offers fast computation times due to the shortened horizon and relaxed optimality requirements. Secondly, it enables online trajectory recalculations, allowing for the consideration of new constraints or changes in the environment at each iteration. While the optimality of the solution may be compromised to some extent, the significant benefits provided by MPC outweigh this drawback. Surveys have summarized key works and developments in this area [Lee11, May14]. In the domain of motion planning, MPC has garnered significant attention, leading to advancements in online motion planning methods [NdCF+16, FJS+17].

For instance, in [LVS20], the authors propose a distributed MPC approach for multi-robot motion planning, decentralizing trajectory generation among multiple robots to optimize their

trajectories while considering inter-robot interactions. Experimental validation with a swarm of up to 20 drones demonstrates the successful generation of collision-free trajectories aligned with global mission objectives. In [FFLS18], a Perception-Aware MPC method is introduced for quadrotor UAVs, which integrates perception-based information into the control framework to enhance performance in dynamic environments. This approach outperforms traditional MPC methods by incorporating perceptual cues, resulting in improved trajectory tracking accuracy, obstacle avoidance capabilities, and overall flight performance. In [NSG+18], a comprehensive approach is presented for controlling the motion of quadrupedal robots using whole-body nonlinear MPC techniques with contact constraints. This method enables agile and dynamic movements, as demonstrated through simulations and real-world experiments on challenging surfaces. Similarly, in [KK22], a method called model hierarchy predictive control (MHPC) is developed to enhance balance and disturbance recovery in humanoid robots by leveraging arm motions. The MHPC formulation combines a full-body kino-dynamic model and a single rigid body model, allowing for the generation of ground wrenches while maintaining stability within the support polygon. Furthermore, a more recent work in [MSV+23] introduces the BiConMP framework, a nonlinear MPC approach for online planning of whole-body motions in legged robots. By efficiently utilizing the structure of the robot dynamics, BiConMP generates real-time whole-body trajectories. Its performance is evaluated on a physical quadruped robot, highlighting its capability to generate cyclic gaits on different terrains, withstand unforeseen disturbances, and seamlessly transition between gaits during online operation.

Another branch of MPC is Learning-based MPC [HWMZ20], which has emerged as a promising approach for enhancing control performance by integrating learning techniques into the MPC framework. This advancement encompasses three main directions. Firstly, learning the system dynamics focuses on adapting the system model during operation or between instances, enabling the MPC controller to continually improve its understanding of the system [SMTA18, TFFS19, TFG19]. This adaptive modeling approach enhances the accuracy and predictive capabilities of the MPC controller. Secondly, learning the controller design involves optimizing the MPC formulation, including the cost function, constraints, and terminal components, to ensure favorable closed-loop control behavior [FKDZ19, RB20]. By leveraging learning algorithms, the controller can adapt and optimize its parameters to achieve desired control objectives effectively. Lastly, MPC for safe learning utilizes MPC techniques to provide safety guarantees for learning-based controllers by addressing constraint satisfaction [KBTK18, WHCZ21]. This ensures that the learning-based controller operates within predefined safety bounds, mitigating risks and improving system reliability. The integration of learning techniques into the MPC framework expands its capabilities and allows for more adaptive, optimized, and safe control in various applications.

Up to this point, humans have not been extensively involved in the industrial pipeline. They mainly serve as supervisors who control, operate, and maintain robotic systems to perform assigned tasks. However, with the increasing demands in robotics and the emergence of challenges where robots have to collaborate with humans in shared workspaces, robot motion planning in these scenarios has reached a new level of complexity. In addition to fulfilling given tasks with considerations for smoothness, low energy consumption, and environmental constraints, the presence of humans in the same workspace as robots introduces new challenges for trajectory motion planning.

- It is essential to plan and adapt robot motions in coordination with human motions to ensure the safety of human partners working in the same environment as the robot.

Simply avoiding humans passively when they come close is no longer sufficient due to the confined workspace shared by both the robot and human partners. Instead, the robot needs to proactively understand and predict human motions, integrating this information into the planning process. By incorporating human motion prediction, the robot can take preemptive action, such as avoiding human partners even before they reach the safety margin around the robot. This proactive approach enhances the safety aspect and improves the overall comfort level for human partners when collaborating with the robot.

- Robot motions must be *readable* to human partners. In this context, *readable* means that by observing a portion of robot motion, the human partner is able to understand the intentions of the robot, and the motion or behavior of the robot should meet the expectations of the human partner. Terms such as predictable motion or legible motion are used to describe this type of robot motion. Achieving *readable* motions increases the comfort of human partners and fosters trust, enabling more efficient collaboration between humans and robots.

- The robot needs to be able to adapt to the highly dynamic environment and adjust its tasks or goals while in motion. This requires the controller of the robot to have fast computation capabilities and the ability to recalculate trajectories on-the-fly, ensuring responsiveness and adaptability in real-time scenarios.

While motion planning has been a topic of research for several decades, there is a limited number of studies that specifically address the emerging challenges in HRC. In the domain of human motion prediction, considerable research has been conducted, as highlighted in the survey by Aggarwal et al. [AC99] and Rudenko et al. [RPH+20]. In [LFS17a], the authors provide a comprehensive overview of how human motion prediction is harnessed to facilitate safe interaction between humans and robots across various robotic systems. They categorize the techniques into two main approaches: goal intent and motion characteristic-based methods. Goal intent techniques focus on inferring human intentions and predicting trajectories, employing methods such as the early prediction approach proposed by Ryoo [Ryo11] and the mixed observability Markov decision process developed by Nikolaidis et al. [NRGS15]. On the other hand, motion characteristic-based approaches concentrate on observing natural human motion patterns, exemplified by Takano et al.'s predictive model [TIN11] that utilizes motion capture data, as well as Xiao et al.'s approach [XWF15] that incorporates support vector machine classifiers. However, it is important to note that these methods primarily explore human motion in isolation, rather than in the specific context of human-robot collaboration within the same workspace.

In addition to human motion prediction, the ability of human agents to anticipate the actions and movements of robots is also crucial for enhancing collaboration in HRC, and this aspect has only recently gained attention. One approach to making robot behavior more predictable to human teammates is through explicit communication of intent, where the robot conveys its planned actions and motions using visual and auditory cues [SCM15, LHS+13, VWE13]. However, a more intuitive and natural way for humans to understand the intentions of the robot is by observing its motion, akin to how humans collaborate with each other. This implicit communication fosters cooperation between humans and robots, creating a sense of partnership. Building on this concept, the authors in [TDJ11] used animation principles to enhance robot intent readability. Participants interpreted actions in simulated robot scenarios, with forethought, such as height adjustments and directed gaze, increasing confidence.

Forethought also led to participants perceiving the robot as more approachable and appealing. Furthermore, when the robot exhibited reactions to the success or failure of the task it was performing, participants regarded it as more intelligent. Dragan et al. [DLS13] investigated how robot motion can convey intent through mathematical definitions of predictability and legibility. They conducted experiments where participants predicted the goals of robots and humans using predictable and legible motions. In a subsequent study [Dra15], they examined the impact of legible and predictable motion on human-robot collaboration. Participants had to infer robot intent for efficient task completion, and the results showed that legible motion, despite having longer trajectories, facilitated more efficient interaction in terms of coordination time compared to functional motion alone. Other related works exist, such as [KTS+10, DWK+05, LK16, BBB13], exploring predictable motion in different scenarios. However, these studies typically involve the robot performing tasks alone or separate human-robot tasks without interaction. Limited recent works have started to apply these concepts in real-life HRC scenarios, although industrial use-cases have not been extensively considered yet, as seen in works such as [BHFK+19, FKBF+20, FPM+23].

Therefore, there is a pressing need for in-depth research on the integration of human motion prediction within the collaborative framework of HRC to enhance the behaviors of the robot. The ability to accurately anticipate human motions and intentions is crucial for the robot to effectively collaborate with its human partners. Current research in human motion prediction has primarily focused on analyzing patterns in human behavior and utilizing machine learning techniques, such as deep learning models, to forecast future motions based on past observations. However, the challenge lies in incorporating these predictive capabilities into the context of HRC, where the robot needs to react responsively to its human partner as well as changes of the environment.

Moreover, the generation of *readable* robot motions represents another critical area that necessitates further investigation and validation within the framework of effective collaboration. Readability refers to the ability of the robot's motions to convey its intentions and actions in a clear and understandable manner to human partners. Achieving readability in robot motions is vital for establishing a shared understanding and facilitating efficient coordination between humans and robots. However, it is imperative to explore and validate the effectiveness of readable robot motions in real HRC scenarios and actual collaborative tasks, rather than solely relying on observations of human observers from outside the working area.

While significant progress has been made in laboratory settings, there is a scarcity of real-life applications and scenarios that explore the dynamics of human and robot motions and their interaction during collaborative tasks. Understanding the challenges and dynamics of human-robot collaboration in real-world environments is essential for designing robust and adaptable systems. Additionally, the complexity and highly dynamic nature of HRC environments pose challenges for motion control. Although MPC has shown promise in generating optimal robot motions, its computational demands and response times may hinder its applicability in fast-paced collaborative scenarios. Therefore, there is a need for motion control methods that not only ensure accurate execution of trajectories but also exhibit fast computation to adapt to the changing demands of the environment and human partners.

As the demand for collaborative robots or *cobots* continues to rise across various industries, addressing these research challenges becomes paramount in establishing *cobots* as equal team members alongside humans. To tackle these issues comprehensively, this thesis focuses on the development of a cohesive framework. This framework encompasses a motion generation approach that incorporates human motions and intentions, enabling faster response times and improved synchronization between human and robot actions. Additionally, online adaptive

motion planning methods will be explored to generate robot motions that are both readable and responsive to the expectations and preferences of human partners. Furthermore, advanced motion control methods will be developed to ensure the precise execution of trajectories, taking into account environmental conditions and safety constraints while maintaining fast computation capabilities to adapt to the dynamic nature of the surrounding environment. The contributions of this thesis can be summarized as follows

- In Chapter 2, an approach is proposed to enhance the performance of robots in close collaboration with humans by integrating human motion prediction into local obstacle avoidance. This integration allows the robots to anticipate and respond to human motions in real-time while avoiding obstacles, ensuring safer and more efficient human-robot interaction.
- Building upon the integration of human motion prediction, Chapter 3 presents a novel approach for generating readable robot motions and adapting motion policies based on task requirements and human preferences. Machine learning techniques and human demonstrations are leveraged to enable the robot to learn and generalize motion policies, promoting flexibility and adaptability in human-robot interaction.
- Finally, in Chapter 4, a novel receding horizon controller scheme is developed to execute given trajectories precisely at the force/torque level. This controller scheme addresses the challenges of limited computation time and on-the-fly trajectory adaptation, enhancing the capabilities of robots in terms of speed and accuracy during reaching motions. This contribution is particularly valuable in close human-robot collaboration scenarios, ensuring smooth and safe cooperation.

Part I

Obstacle Avoidance and Predictable Motion Generation

Nomenclature of Part I

Scalars are written in plain lower case, vectors in bold-face lower case. Matrices are bold-face upper case, while plain upper case symbols refer to coordinate frames, mathematical spaces and sets. Subscript annotations are reserved for index notation of multi-dimensional variables, whereas superscript annotations are part of the variable specification.

Also note that we omit explicit listing of function parameters whenever it is clear from the context, to not unnecessarily clutter the notation. A list of the most frequently used variables in this work is given.

$\boldsymbol{\epsilon_t}$	exploration noise
$\boldsymbol{\mathcal{T}}$	subset
$\boldsymbol{D}$	damping matrix
$\boldsymbol{F}$	force
$\boldsymbol{g}$	task
$\boldsymbol{J}$	jacobian
$\boldsymbol{K}$	stiffness matrix
$\boldsymbol{M}$	mass matrix
$\|\cdot\|$	2-norm of a vector
ω_i	modifiable weight of DMP
$\pi_{\boldsymbol{\theta}}$	policy with parameters $\boldsymbol{\theta}$
$\psi_i(x)$	Gaussian basis functions
τ	time constant
$a_{\max}$	maximum acceleration
$f(t)$	function
$H(x)$	Heaviside function
$j_{\max}$	maximum jerk
P	Probabilities
u	one-dimensional time-scaling factor
$v_{\max}$	maximum velocity

2

An Approach to Integrate Human Motion Prediction into Local Obstacle Avoidance in Close Human-Robot Collaboration

This chapter was previously published in [HDOH+15].

Within the field of HRC, ensuring safety is of utmost importance during collaborative tasks. In confined workspaces where humans and robots work together, the risk of collisions and potential injuries is high. These unintended collisions can lead to a loss of trust in the robot by the human partners. Therefore, it is crucial to prevent collisions in a Human-Robot-Team (HRT) setting. Additionally, the comfort of human partners must be taken into consideration. Balancing the behaviors of the robot between task performance and avoiding human interference is a challenge. If the robot avoids humans too early, task efficiency is compromised as it spends more time evading them. Conversely, if the robot reacts late and humans get too close, discomfort may arise, resulting in unwillingness to cooperate. Addressing this challenge, this chapter proposes an approach to enhance robot avoidance behavior by integrating human motion prediction into the obstacle avoidance strategy. The collision avoidance algorithm incorporates compliance control to enable fast reactions to dynamic obstacles, such as humans, without requiring extensive computational resources. The human motion prediction algorithm is based on the minimum-jerk model. Experimental analysis conducted on a case-study involving collecting *LEGO*-bricks on a table with various subjects investigates the impact of integrating human motion prediction on the reaction time of the robot and the human's perception of the robot as a co-worker. The study concludes by analyzing subjective human feedback questionnaires to assess the comfort and acceptance of the robot colleague by the human collaborator.

2.1 Introduction

To enable the collaboration between human and robot, one of the goals is to overcome the still very commonly separated workspaces of robots and humans in industrial assembly lines and form a new joint collaboration space instead. In order to achieve that, one of the key aspect is to guarantee safety for the human collaborators. And the most fundamental action is that the robot should avoid the human at all time to prevent any collisions. Here we mainly focuses on obstacle avoidance behaviors of the robot and how to generate comprehensive and efficient motions for the robot in a shared and confined workspace such that the tasks are carried effectively and the human feels comfortable when working with the robots.

In order to avoid the human, the robot should be able to sense the environment autonomously, and respond according to its human coworker. One of the fundamental sensory information for the robot can be acquired by observing human motions during the task execu-

tion. By knowing where the human is, collision can be avoided. This can be easily achieved by sufficient hardware setups i.e. cameras, tracking system, etc. And with the acquired sensory information, the basic obstacle avoidance strategy [Kha90] is to setup a bounded threshold around the robot such that whenever the human goes over it, the robot will perform movements to avoid the human. This strategy is simple in calculation to be applied in real-time and hence it becomes a very common approach being used for obstacle avoidance. However, in the context of HRC, this original obstacle avoidance strategy becomes less efficient in the sense of task completion due to the increasing of time spend for the robot to avoid the human. The reason for this is because of the high probability of interference between the human and robot in their shared workspace. As the consequence, it leads to the high chance of collisions that might happen during the collaboration, which means that the robot has to spend more time in pending until the human moves out of the predefined safety area to continue his tasks. In particular, there are cases where the actual current task of the human does not interfere with the task of the robot, but on the way performing the task, the human crosses over the boundary which also triggers the avoidance action. As the result, the time needed for the robot to complete his own tasks increases and the efficiency of the collaboration between the human and robot decreases as well. One might come up with an idea of reducing the safety area around the robot to reduce the spending time. However this solution only works to a certain extend as if the distance between the human and robot is too small, the human will naturally feel uncomfortable even if there is no collision happen.

Building upon the aforementioned considerations, we propose an approach that integrates human motion prediction into local obstacle avoidance for HRC in a shared workspace. By leveraging human motion models, the robot can proactively interpret and respond to the anticipated actions of its human coworkers. This integration of human motion prediction with obstacle avoidance allows the robot to react faster, enhancing the overall efficiency and safety of the collaboration. To facilitate collision avoidance, we employ compliance control as it enables the robot to avoid dynamic obstacles, such as its human collaborator, without requiring complete trajectory replanning. The human motion models are based on a minimum-jerk model, which provides valuable insights into the kinematic and dynamic properties of human movement.

To evaluate the effectiveness of our proposed approach, a typical manipulation task is considered for this study. A specific scenario in which the workspace of the robot and the human overlap distinctly and collisions are provoked is explored: Sorting *LEGO*-bricks together in confined space (Fig. 2.1). Within this experiment, the effect of the motion prediction algorithm is analyzed by investigating the behavior of the robot within a HRT. We also examine the acceptance and comfort levels of the overall system by analyzing the feedback questionnaires provided by the 16 participants involved in the experiment.

In Section 2.2, we discuss related research projects and highlight the results achieved in the field. Section 2.3 provides an explanation of the collision avoidance and intention prediction algorithms employed in our approach. The overall system setup is outlined in Section 2.4, and finally, the experiment itself is described and evaluated in Section 2.5.

2.2 Related Work

Safety within a HRT has been studied extensively in [BPC08] and[HHK+12]. They proved the importance of safety for the human team members at any time, independent of the resulting use-case. For trajectory generation, numerical optimal control methods as a direct solution

to generate collision free trajectories are proposed in [vSS94]. As these methods compute the entire trajectory from start- to end-point they are computationally expensive and result in slow convergences. In [PNS+15], this method is used to generate motion trajectories offline while only deforming them online. In [PCN13], the computational effort is reduced by enabling the kinodynamic planning in the configuration space. Thus, reducing the search dimension distinctly.

In [ZRD+13], a functional gradient technique is applied to iteratively improve the convergence as well as the trade-off between obstacle avoidance and trajectory smoothness. It uses Hamilton Monte Carlo method to overcome the problem of infeasible solutions due to convergences in high-cost local minima, which numerical optimal control methods tend to converge in. Even though this technique reduces the computation time and the resulting trajectory distinctively, the computational costs of this algorithm are still respectively high.

Besides these optimization methods, online motion adaption algorithms with high computational efficiency have been introduced. In [LRS14] a generic collision avoidance approach is presented. It proposes the idea of altering the velocity of each joint adaptively to the current distance of the human hand and the robot end-effector independent of the used robot hardware and sensors. Furthermore, it highlights the impact of communication delays within the proposed system to the real-time capability of the overall system.

For obstacle avoidance, the potential field theory has been introduced by Khatib [Kha90] and outlined more in detail in [Qui95] and [BK02]. Within this approach, objects and targets are represented as virtual repulsive force fields to generate a collision free motion trajectory without the need of heavy computations in terms of online optimization. In [FKDLK12] and [DLF12] a collision avoidance algorithm based on this method is proposed. It uses virtual spring and damper elements as it was previously proposed in [HUP+10] to model the virtual force of dynamic objects.

While the proposed collision avoidance algorithm in [FKDLK12] and [DLF12] is based on altering the desired velocity, in order to avoid collisions during execution, our approach is based on changing the the overall desired position. As it will be shown in Sec. 2.3.3, the robot is not exceeding a maximum modification of the actual desired position, while providing safety at the same time. This solution assures the ability to guarantee a collision free motion of the robot over all time without the expensive pay-off of the high computational cost of numerical online optimization.

Regarding the human motion prediction, there are several mathematical models for human hand motion such as minimum-jerk [FH85], minimum-torque change [KMUS90] and minimum-variance [HW98]. Those models have been shown to match the experimental data on various conditions, with some variances on velocity and acceleration profile depending on the optimization criterion in use. The HRC scenario presented in this paper usually consists of point-to-point movements, and requires a fast motion trajectory computation for avoiding contact with the human partners proactively. Hence, we use a minimum-jerk model for our trajectory estimation calculations since it provides a good compromise between accuracy and mathematical simplicity.

2.3 Design of the system architecture

In order to create a realistic and challenging environment for human-robot collaboration, we carefully construct an HRC scenario as depicted in Figure 2.1. The scenario involves placing the human and robot on opposite sides of a table, with a shared center area as their workspace.

The human's task is to pick up the blue bricks, while the task of the robot is to pick up the red bricks. By positioning the bricks in opposite directions, the human and robot are required to cross the common area, leading to a high probability of interference and collisions. To ensure the human's continued participation until the robot completes its task, the number of blue bricks is set to 15, while only 3 red bricks are provided. This discrepancy accounts for the human's typically faster speed and necessitates the robot to temporarily pause its task to avoid collisions with the human.

Given the HRC scenario and the specific requirements, the collaborated robot must satisfy the following requirements

- The robot has to be able to perform point-to-point motions repetitive for the pick and place task.
- The movements of the robot should be smooth and its speed should be limited to ensure the human's comfort during the collaboration.
- The robot must prevent collision with its human partner at all times.

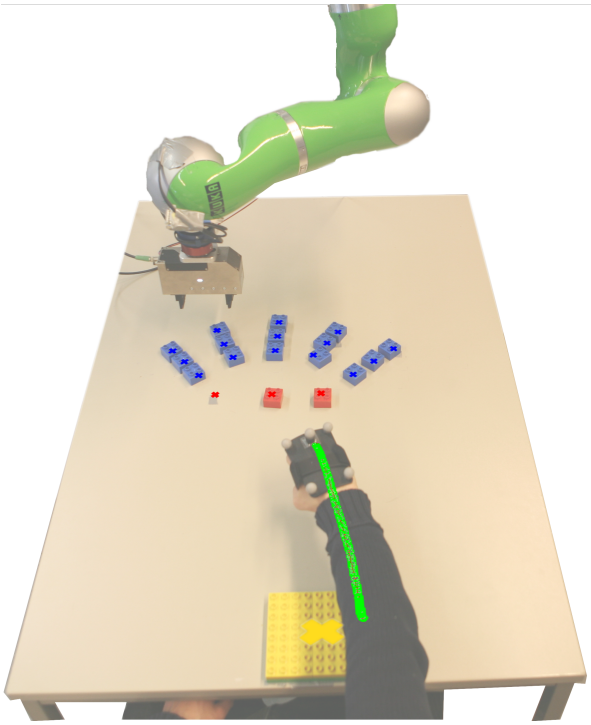


Figure 2.1: Illustrative setup of the experiment

To meet these requirements, we have designed a comprehensive control system architecture for the robot. The architecture consists of multiple components: a way-point trajectory generation module responsible for generating smooth trajectories, a compliance control module for tracking the generated trajectories, and a potential field approach for collision avoidance. Additionally, the avoidance strategy is further supported by a model-based human motion estimation method. In the following sections, each component of the control architecture will be outlined in detail, providing a clear understanding of the functionality and effectiveness of the proposed system.

2.3.1 Way-point trajectory generation

Point-to-point motions play a crucial role in various mechatronic applications such as machine tools, CNCs, and robots, where high speed and smooth motion are essential. The objective is to determine a trajectory profile that consists of position, velocity, acceleration, jerk, and other parameters at each timestamp. This profile enables the robot to accurately navigate through all the specified way-points in the correct order. However, it is equally important to ensure that the trajectory profile satisfies certain conditions:

(i) While operating at high speeds, the velocity, acceleration, and jerk must adhere to the limitations of the robot. This constraint helps reduce wear and tear, prolonging the life of the machine and its tools.

(ii) To minimize tracking errors and prevent jerky motions that could lead to a lack of trust from human collaborators, the trajectory profile must exhibit continuous derivatives, particularly in acceleration and jerk. This ensures smooth and seamless movements throughout the path of the robot.

(iii) The trajectory computation process needs to be simple and efficient to facilitate real-time execution. This requirement allows for on-the-fly trajectory generation, enabling the robot to quickly adapt to dynamic changes in its environment.

There are several methods that have been proposed to address all of the requirements of this point-to-point motion generation task. For example, in [HWR08] a trapezoidal velocity profile, also known as Linear Segments with Parabolic Blends (LSPB) was proposed to minimize motion time while respecting the maximum actuator accelerations. However, this method is not suitable for our application as it results in abrupt changes in acceleration, leading to discomfort and roughness in velocity. Another approach is the S-curve velocity profile based on third-order polynomial trajectory planning [HMH09]. By generating continuous acceleration profiles that respect jerk limitations, this method achieves smoother velocity profiles with higher accuracy. A notable contribution in this field is the Reflexxes libraries developed by T. Kroeger [KW10], which provide algorithms for deterministic and instantaneous computation of motion trajectories.

In our work, we have implemented an exponential function-based trajectory generation method [RIS13]. This approach offers several advantages over the aforementioned methods. It provides a smoother approximation of the trapezoidal velocity profile, ensuring continuity up to the third derivative (jerk). As a result, the intermediate points of the trajectory can be tracked more accurately compared to other methods. Additionally, the exponential function takes into account the limitations of the velocity, acceleration, and jerk, guaranteeing safety and prolonging the lifespan of the actuators. Moreover, this method offers reduced computational effort due to its simpler formulation, and the calculation of the exponential function can be performed offline, requiring only one-time computation.

According to [RIS13] a third order exponential function

$$f(t) = v_{\max}\left(1 - \exp\left(-u^3\right)\right), \tag{2.1}$$

with u as a one-dimensional time-scaling factor

$$u = \alpha t, \tag{2.2}$$

is used as the velocity profile. The first and second derivative of (2.1) are then the representations of the acceleration and jerk profiles respectively

$$\dot{f}(t) = 3\alpha v_{\max} u^2 e^{-u^3} \tag{2.3}$$

$$\ddot{f}(t) = \alpha^2 v_{\max}(6u - 9u^4)e^{-u^3} \tag{2.4}$$

The exponential function exhibits a crucial characteristic that makes it suitable for generating smooth velocity, acceleration, and jerk profiles. Its smooth shape allows it to maintain the same exponential form through differentiation, resulting in continuous and smooth profiles, as depicted in Fig. 2.2. The steepness of the exponential function is determined by the tuning factor α. A higher value of α leads to greater acceleration generated by the function. Consequently, the selection of this tuning factor is essential and should be appropriately chosen to meet the specific requirements of the application, as further discussed later in this work. These profiles guarantee the smoothness of the robot motion which is the reason why it is chosen as the trajectory generation method for our point-to-point motion.

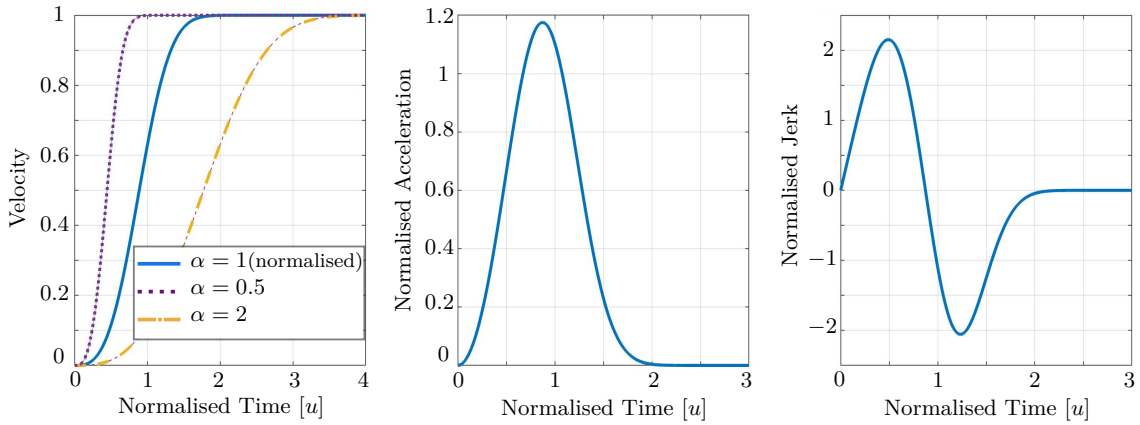


Figure 2.2: Exponential characteristics of the velocity $f(u)$ on the left, acceleration $\dot{f}(u)$ in the middle and jerk $\ddot{f}(u)$ on the right

The next problem is how to bound the maximum values of the velocity, acceleration and jerk within the given thresholds. The maximum velocity can be set with $v_{\max}$ in Eq. (2.1). For the maximum velocity and jerk, they are defined by analyzing the maximum points of the functions in Eq. (2.3) and (2.4). They occur at

$$t_{a_{\max}} = \frac{1}{\alpha}\sqrt[3]{\frac{2}{3}} \approx \frac{0.8736}{\alpha} \tag{2.5}$$

$$t_{j_{\max}} = \frac{1}{\alpha}\sqrt[3]{\frac{3-\sqrt{7}}{3}} \approx \frac{0.4906}{\alpha} \tag{2.6}$$

where $t_{a_{\max}}$ and $t_{j_{\max}}$ are the times at which the maximum acceleration and jerk happens respectively. Substituting these values into Eq. (2.3), (2.4) we found

$$|a_{\max}| = |0.3918 v_{\max}\alpha| \tag{2.7}$$

$$|j_{\max}| = \left|0.7652 v_{\max}\alpha^2\right| \tag{2.8}$$

where $a_{\max}$ and $j_{\max}$ represent the maximum values of acceleration and jerk respectively. Thus, given the actuator velocity, acceleration and jerk limits, we can find the maximum value of α that allows the fastest possible motion within the limits, as follow

$$\alpha_{\max} = \min \left\{ \frac{|a_{\max}|}{\lambda |v_{\max}|}; \sqrt{\frac{|j_{\max}|}{\mu |v_{\max}|}} \right\}, \tag{2.9}$$

with $\lambda = 1.1754$ and $\mu = 2.1524$, regarding the maximum velocity $v_{\max}$, acceleration $a_{\max}$ and jerk $j_{\max}$.

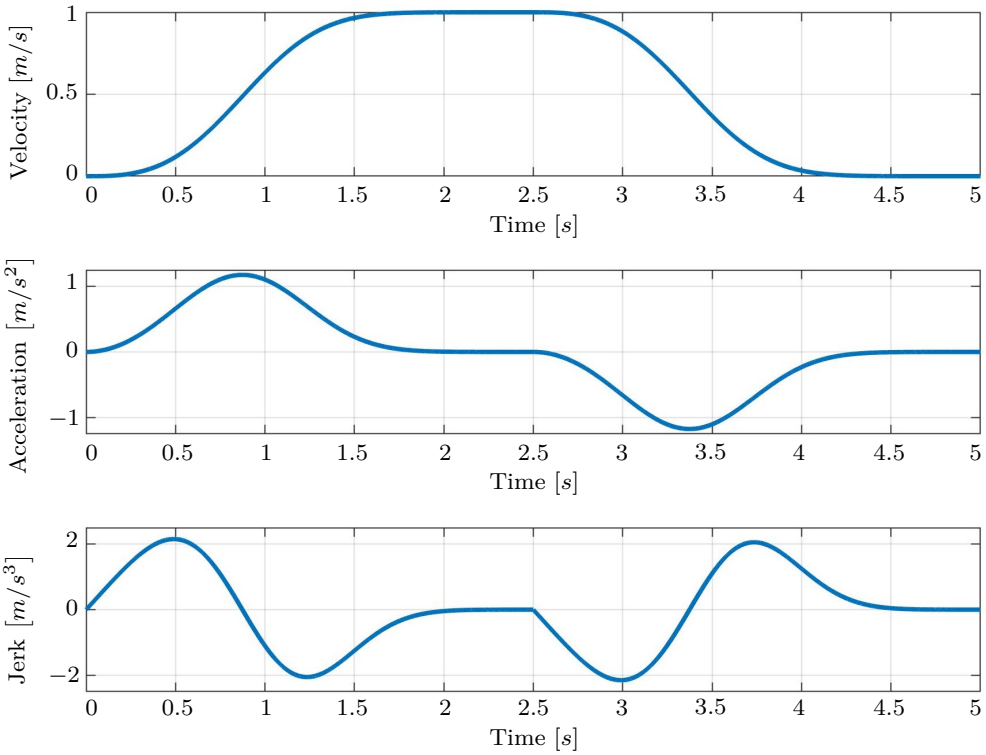


Figure 2.3: Velocity, acceleration and jerk for one point-to-point motion

Note that the expression presented in Eq. (2.1) and illustrated in Fig. 2.2a only shows the increment part of the velocity profile. To complete the trapezoidal velocity profile between two points with a given time interval T_d, two of the velocity profile defined in Eq. (2.1) are used in sequence

$$v(t) = f(t) - f(t - T_d) \cdot H(t - T_d). \tag{2.10}$$

with $H(x)$ is a Heaviside function, defined as

$$H(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}, \tag{2.11}$$

The complete velocity, acceleration and jerk profiles for one point-to-point motion are illustrated in Fig. 2.3. Finally, in order to generate a trajectory via several points, the velocity profile of the next step overlaps with the profile of the current step to keep the velocity always non-zero until the motion is finished.

2.3.2 Compliance control

The subsequent component of the system architecture focuses on designing a controller that can effectively track the trajectories generated in Section 2.3.1 while also being responsive to the changing environment surrounding the robot. Specifically, our objective is for the robot to avoid collisions with its human partners whenever they come into close proximity. For simplicity and ease of verification, we narrow our focus to collisions between the human arm and the end-effector of the robot, even though collisions could potentially occur at different parts of the robot (e.g., elbow). This selective consideration aligns with our aim of enhancing the reaction behavior of the robot. The scenario depicted in Fig. 2.1 is also tailored towards these types of collisions, where the robot and human partner are positioned on opposite sides of the table, and the human only utilizes their hand to perform tasks. Thus, by default, the potential collision occurs between the end-effector of the robot and the human's hand.

To follow the generated trajectories as well as to react to the dynamic of the environment, we implemented a compliance controller based on the concept introduced by N. Hogan [Hog85]. This controller enables the robot to emulate the behavior of a mass-spring-damper system in response to external forces. Since our focus is primarily on the collision occurring around the end-effector of the robot, the compliance controller is calculated in Cartesian space as follows

$$\mathbf{F}_{\text{ext}} = \mathbf{M}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d) + \mathbf{D}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}(\mathbf{x} - \mathbf{x}_d). \tag{2.12}$$

The environmental reaction force $\mathbf{F}_{\text{ext}} \in \mathbb{R}^3$ represents both attraction towards the desired trajectory as well as repulsion from obstacles within the environment (its composition is discussed in Sec. 2.4.2). As a result, the robot can react to uncertainties and changes in the environment without computing a new trajectory from current position $\mathbf{x} \in \mathbb{R}^6$ to the desired end-pose $\mathbf{x}_d \in \mathbb{R}^6$. In (2.12) $\mathbf{M}, \mathbf{D}, \mathbf{K} \in \mathbb{R}^{3\times3}$ are positive definite matrices that represent the mass, damping and stiffness. As a consequence, the introduced error dynamics guarantee stability and robustness. In the proposed setup $\mathbf{M}$ is chosen as the identity matrix, $\mathbf{K}$ and $\mathbf{D}$ are diagonal matrices chosen to adapt the desired reaction to virtual forces. Increasing the damping results in a slower reaction but smoother movement of the robot. This is outlined in more detail in Sec. 2.4.1.

2.3.3 Local collision avoidance

The idea of the presented obstacle avoidance is based on potential fields w.r.t. the distance between the end-effector of the robot and obstacles. Whenever an obstacle is inside a threshold region of the end-effector, a virtual force vector is generated and added to $\mathbf{F}_{\text{ext}}$ in (2.12) of the compliance control to push the end-effector away. A simple but effective function based on the same idea of repulsive vectors in [FKDLK12] is used to generate a smooth reaction force

$$\mathbf{F}_{\text{react}} = \frac{\mathbf{F}_{\max}}{1 + \exp\left(\left(\|\mathbf{d}(E,O)\|\,(2/\rho) - 1\right)\alpha\right)}, \tag{2.13}$$

where $\mathbf{F}_{\max}$ is the maximum force applied, $\|\mathbf{d}(E,O)\|$ is the distance between obstacle and end-effector, ρ is the threshold distance that defines the collision region around the end-effector and α is a shape factor. The force reaches its maximum if the distance equals zero, and zero if the obstacle is outside the region respectively. The steepness of the force profile within the threshold region regarding the distance can be adjusted by the shape factor α. An example of the virtual force profile is shown in Fig. 2.4. In our experiment, $\mathbf{F}_{\max}$ is chosen to guarantee the end-effector of the robot does not collide with the human arm while working in the same

workspace. As mentioned in Sec. 2.2, the approach presented alters the desired trajectory on position level rather than velocity level (see [FKDLK12]). This yields in a saturation of the position alteration, proportional to $\mathbf{F}_{\max}$. This effect is preferred in this use-case, as it results in a more human like behavior. As a consequence, the robot stays closer to its planned trajectory, rather than letting it being chased around by the human.

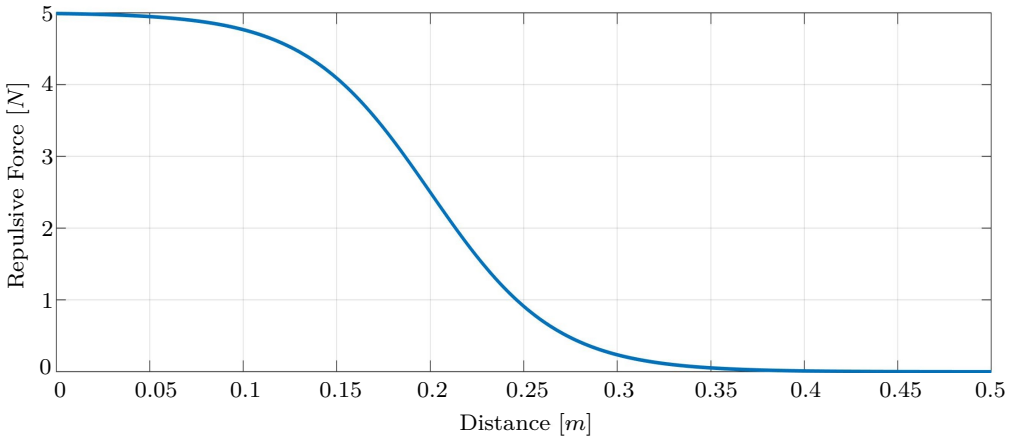


Figure 2.4: Repulsive force profile from (2.13), with $\mathbf{F}_{\max} = 5N$, $\rho = 0.4m$, and $\alpha = 6$

2.3.4 Human motion prediction

In order to enhance the collision avoidance behavior of the robot, we aim to integrate the prediction of the human motion to provide additional information for the robot to react faster. Given a part of a human motion trajectory as an input, we need to predict how this motion is expanded in the future and extract this information for the robot. In general, predicting human motion is a complex task, intricately dependent on various tasks and necessitating a comprehensive analysis of the human body's movements [AC99]. Moreover, in our specific context, real-time prediction is imperative, demanding computational efficiency that matches the speed of human motion. Nevertheless, by leveraging specific known conditions within our scenario, we can simplify the problem and reduce computational overhead. Since the assigned task for the human involves seated object manipulation, the majority of the human body remains constrained, leaving the wrist as the primary agent of motion. Hence, our focus narrows to capturing the point-to-point motion of the human wrist as a singular entity, disregarding the influence of the upper body and the elbow. Given these assumptions, the minimum-jerk model emerges as the optimal candidate for representing human wrist motion in our scenario. Consequently, we proceed to introduce the minimum-jerk model and outline our method for estimating human motion using this model.

Flash and Hogan [FH85] hypothesized that humans follow a smooth trajectory for point-to-point movements by minimizing the jerk. For a trajectory $\mathbf{x}(t)$, that starts at time t_s and ends at time t_f, the minimum-jerk criterion G can be expressed by

$$G(\mathbf{x}) = \frac{1}{2} \int_{t_s}^{t_f} \left(\frac{d^3\mathbf{x}}{dt^3} \right)^T \left(\frac{d^3\mathbf{x}}{dt^3} \right) dt. \tag{2.14}$$

The optimization criterion can be solved for given start time t_s, final time t_f, and boundary

conditions for $\mathbf{x}_s = \mathbf{x}(t_s)$ and $\mathbf{x}_f = \mathbf{x}(t_f)$. The solution to this optimization problem is a fifth-order polynomial [FH85], and for point-to-point movements, where start and final velocities and accelerations are equal 0, the solution takes the following form:

$$\mathbf{x} = \mathbf{x}_s + (\mathbf{x}_f - \mathbf{x}_s)\left(10\frac{\tau^3}{T^3} - 15\frac{\tau^4}{T^4} + 6\frac{\tau^5}{T^5}\right), \tag{2.15}$$

where $\tau = t - t_f$ and $T = t_f - t_s$. Thus, the trajectory, i.e. the spatial as well as the temporal course of the movement, can be calculated exactly with given coefficients.

However, as mentioned earlier, the motion estimation demands the solution of the inverse problem: Given a part of a human motion trajectory, find the coefficients of the model which would result in the trajectory that best matches the observed one. In essence, the human motion estimation process can be described as a curve-fitting problem.

The physical trajectory point $\mathbf{x}_i$ is sampled by the camera tracking system at discrete time instants t_i. Hence, the observed trajectory can be represented as

$$\mathbf{x}_i = \mathbf{f}(t_i), \qquad t_s \leq t_i < t_f. \tag{2.16}$$

As stated earlier, the function for the model which is used to approximate the data, is a fifth-order polynomial with coefficient vector $\boldsymbol{q}$

$$\hat{\mathbf{x}}_i(\mathbf{q}) = \mathbf{h}(t_i, \mathbf{q}). \tag{2.17}$$

A sequence of n time stamps t_i and corresponding positions $\mathbf{x}_i$, with $i = 1, \ldots n$, which represents the observed movement, is the input to the estimation procedure. Finally, the motion prediction algorithm optimizes the cost function $G(\mathbf{q})$ in order to find the parameters $\boldsymbol{q}$ of the model function in (2.17):

$$G(\mathbf{q}) = \frac{1}{2}\sum \|\hat{\mathbf{x}}_i(\mathbf{q}) - \mathbf{x}_i\|^2 = \frac{1}{2}\sum \|\mathbf{h}(t_i, \mathbf{q}) - \mathbf{x}_i\|^2 \tag{2.18}$$

Solving this optimization procedure would yield in unreliable results without any constraints. However, due to the nature of the task, several constraints can be imposed. As the human needs to pick up and then place an object, final velocity and acceleration should be zero, i.e. $\dot{\mathbf{x}}(t_f) = 0$, $\ddot{\mathbf{x}}(t_f) = 0$ respectively. Moreover, we can approximately identify the object that the human is going to pick up (see Sec. 2.4.2). The position of that object, i.e. $\mathbf{x}(t_f) = \mathbf{x}_f$, can also be used as a constraint for the optimization. Then, the cost function is minimized only to find the final time t_f that the motion would end at. As a result, the full trajectory of the human can be estimated with just a given part of the trajectory. Additionally, our proposed method is fast enough to be executed online. It means that every time the human starts to move, our approach starts to publish the estimated trajectory and adjust it on-the-fly based on the change of the human motion. An illustrated example of the proposed approach is shown in Fig. 2.6 where the green line is the trajectory of the human up to current time and the magenta line is the predicted motion generated by the algorithm.

2.4 Implementation

In this section, we provide a comprehensive overview of the implementation of the system architecture proposed in Section 2.3. All the components of the control scheme are summarized in Fig. 2.5. To begin, we delve into the selection process for the parameters of the compliance

control, as discussed in Section 2.4.1. This section outlines the rationale behind our parameter choices, which are crucial for achieving effective collision avoidance. Moving forward, in Section 2.4.2, we delve into the utilization and integration of human motion prediction within the compliance control framework, emphasizing how this integration enhances the avoidance behavior of the robot.

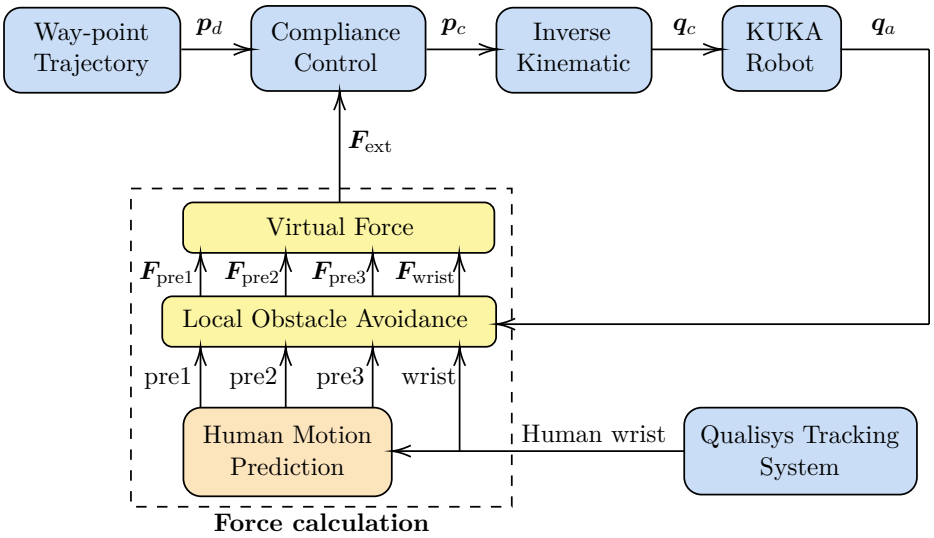


Figure 2.5: Proposed system architecture. The point-to-point motion $\boldsymbol{p}_d$ is generated by the way-point trajectory and served as one of the inputs of the compliance control. The human wrist is tracked using the Qualisys tracking system and the data is stored for prediction. For every human motion, three prediction points pre1, pre2 and pre3 in 0.1s, 0.2s and 0.3s ahead are used in combination with the current human wrist position to form the corresponding repulsive forces $\boldsymbol{F}_{\text{pre1}}$, $\boldsymbol{F}_{\text{pre2}}$, $\boldsymbol{F}_{\text{pre3}}$, and $\boldsymbol{F}_{\text{wrist}}$ for obstacle avoidance. The total virtual force $\boldsymbol{F}_{\text{ext}}$ is then normalized from these repulsive forces and added into the compliance control to generate the actual motion $\boldsymbol{p}_c$ for the robot.

2.4.1 Damping factors of the compliance control

For the compliance control, the stiffness factor for every coordinate is set to $K_{i,i} = 100\,\text{N/m}$. The damping factor is chosen dependent on the position of the human wrist. When the human wrist is outside of the predefined safety region the damping factor is set to the critical damping value

$$D_{i,i} = \sqrt{4 \cdot K_{i,i}}. \tag{2.19}$$

This results in the maximum possible reaction speed without oscillation. Upon the human entering the safety region, defined by a radius of $\rho = 15$ cm, the damping value is amplified by a factor of 10. This amplification serves the purpose of generating smoother robot reactions. Consequently, the heightened damping value reduces the sensitivity to virtual forces that represent human motions, preventing abrupt and jerky responses during collision avoidance, thereby enhancing safety. Nonetheless, this amplification results in a trade-off with reaction speed, as the response of the robot becomes relatively slower. This limitation prompted the integration of the motion prediction method outlined in Section 2.3.4 into the collision avoidance strategy. By incorporating the prediction into the calculation of virtual forces, the robot can anticipate potential collisions at an earlier stage. As a result, this approach mitigates the drawback of reduced reaction speed caused by increased damping, effectively enhancing the overall performance of the robot in collision avoidance scenarios.

2.4.2 Integration of human motion prediction into the compliance control

The human motion prediction strategy, as discussed in Section 2.3.4, utilizes tracking data from the motion tracking system. To ensure avoidance of the human's hand, a tracking marker patch is attached to each participant's wrist for accurate hand position tracking (refer to Fig. 2.6).

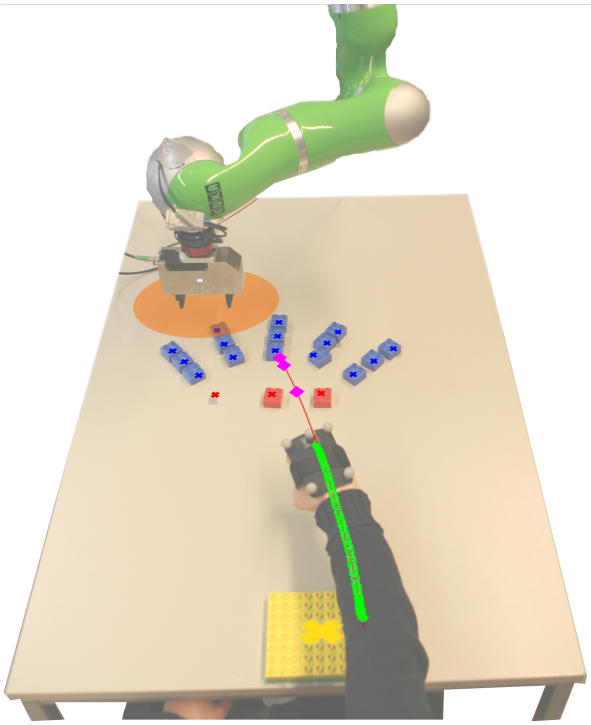


Figure 2.6: Illustrative setup of the experiment with an example of a predicted motion. Robot bricks (red), human bricks (blue), human home base (yellow), tracked motion (green line), predicted motion (red line), prediction points $\{0.1, 0.2, 0.3\}$ s ahead (magenta)

As explained in Section 2.3.4, the human motion prediction involves solving the optimization problem described in Eq. (2.18). An essential piece of information required for solving this problem is the starting and ending points of the human motion. Considering that the human's task involves picking up objects and placing them at a designated location, the prediction algorithm sets the starting point at the base location and resets it whenever the tracked velocity falls below a specified threshold. The end position of the target is determined by identifying the closest *LEGO* brick to the current velocity vector. By analyzing the collected tracked points, the minimum jerk trajectory is obtained, as detailed in Sec.2.3.4. Three future points from this predicted trajectory are utilized for obstacle avoidance. To ensure meaningful integration of the avoidance strategy without prediction, both the wrist point and its three predictions are normalized in terms of their weight for potential field calculations within the compliance control discussed in Sec.2.3.3. The setup and utilization of the prediction are shown in Fig. 2.6.

To calculate the total virtual force considering the prediction, the current position of the human wrist and three arbitrary prediction points at intervals of 0.1s, 0.2s, and 0.3s ahead are employed. To avoid collisions with the table, only the virtual forces in the X- and Y-dimensions are calculated in (2.13), while the force in the Z-direction is disregarded. The

three prediction points, along with the current wrist position, are weighted based on their prediction time in the future and subsequently normalized. This weighting accounts for the increasing uncertainty associated with human motion prediction over time. As a result, the total virtual force then is summed up as

$$\mathbf{F}_{\text{virtual}} = \frac{k_1\mathbf{F}_{\text{wrist}} + k_2\mathbf{F}_{\text{pre1}} + k_3\mathbf{F}_{\text{pre2}} + k_4\mathbf{F}_{\text{pre3}}}{\sum_{i=1}^{4} k_i}, \tag{2.20}$$

with $k_1 > k_2 > k_3 > k_4$. Finally, this virtual force is incorporated as an input to the external force within the compliance control, as mentioned in (2.12).

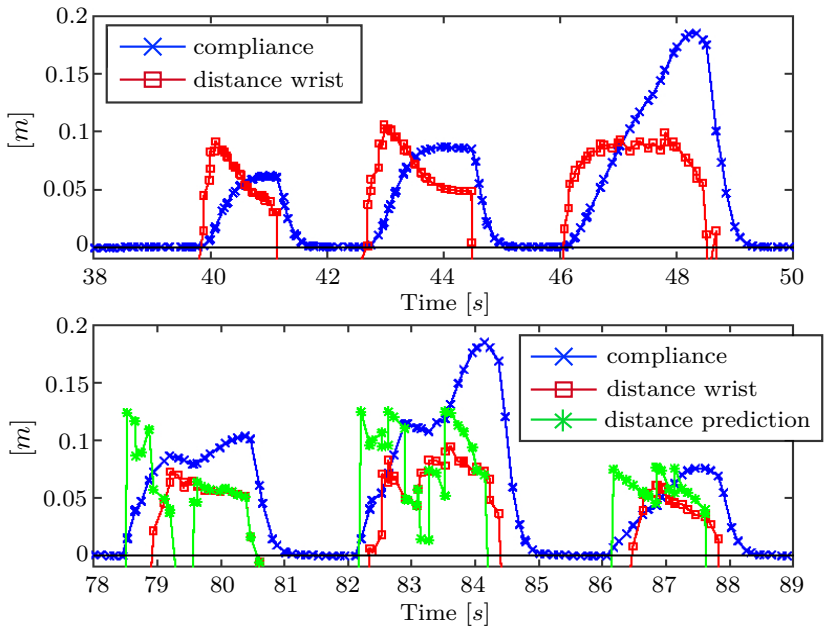


Figure 2.7: The compliance response to an obstacle within the range of avoidance

The improvement in terms of responsiveness can be observed in Fig. 2.7, where the upper plot illustrates the response without prediction and the lower plot demonstrates the response with prediction. To enable an intuitive visual comparison, both plots utilize Euclidean distances in the XY-plane. In these plots, the blue line represents the alteration of the desired position in response to an obstacle within the avoidance range. The red line indicates the distance between the end-effector of the robot and the wrist, while the green line represents the distance to the third prediction point (0.3s ahead). It is important to note that the plot depicts the extent to which the tracking/prediction enters the avoidance area of the robot, rather than directly displaying the Euclidean distance. This allows for a more accurate visual verification of the reaction of the robot.

Upon analysis, it becomes evident that by incorporating the prediction, the robot initiates its reaction approximately 0.25s to 0.5s before the obstacle actually enters the avoidance range. Consequently, the performance of the system is enhanced, particularly when facing communication delays between the motion tracking system and the robot.

2.5 Evaluation of the proposed framework in HRC scenario

In the experimental setup, a collaborative scenario is created using a 7-DOF *KUKA-LBR4+* arm with a *WSG 50* gripper from Schunk interacting with a human participant. The human participant exclusively utilizes one arm, which is tracked using a Qualisys motion capture system comprising of four cameras. The strategic arrangement of these cameras ensures comprehensive coverage of the joint workspace, accounting for a wide range of possible robot-human positions. To induce collisions during the experiment, the task of the robot involves iterative reordering of two red *LEGO* bricks positioned at three predetermined locations within the middle of the workspace. The role of the human participant is to pick up 15 blue bricks situated behind the workspace of the robot and systematically collect them on his home base located in front of him. This task results in point-to-point movements of the human, thereby justifying the utilization of the minimum-jerk model. The experimental setup is visually depicted in Fig. 2.6.

2.5.1 Safety Aspects

The experimental setup prioritizes safety by incorporating measures to minimize occlusion and prevent accidents. The camera redundancy of the tracking system greatly reduces the occurrence of occlusions, and in the rare event of an occlusion, the last tracked position is utilized. To ensure the safety of the human participant, the setup is designed to keep the human body out of reach of the robot, except for the participant's arm, which is tracked using markers on the wrist and elbow. However, since the elbow never enters the avoidance region of the robot, its position is not taken into account for obstacle avoidance purposes. Reactive movements of the robot towards the human are also disabled. This prevents any potential accidents or collisions between the robot and the human participant. Additionally, in addition to the *KUKA* internal impedance control, safety measures such as a maximum velocity threshold for the robot arm and collision detection using the joint torque sensors of the robot are implemented. If any emergency situations arise, these safety measures trigger the activation of the emergency brakes.

2.5.2 Case Study

Each participant in the experiment was asked to perform two separate runs: one utilizing the plain obstacle avoidance based solely on wrist position, and the other incorporating the proposed prediction strategy. The order of the runs was randomized for each person and conducted on different days, effectively eliminating any potential learning effects or biases. Further, to eliminate any perceptual issues related to the robot and the prediction strategy, the experiment was designed with predefined positions for the *LEGO* bricks and predetermined tasks conducted in a specific order. This ensured that the position of the bricks and the tasks were known beforehand, allowing participants to focus solely on evaluating the behavior of the robot and performance.

Both runs involved global path planning based on the positions of the bricks and boxes, utilizing the exponential trajectory generation method mentioned in Section 2.3.1. Additionally, the local obstacle avoidance technique described in Section 2.3.3 was implemented in both runs. The key difference between the runs was the enabling or disabling of the motion prediction from Section 2.3.4.

Following each experiment, participants were asked to complete a brief questionnaire. The

questionnaire aimed to gather their impressions regarding various aspects of the behavior of the robot, including safety, comfort in working with the robot, responsiveness, smoothness of movements, and whether they perceived the robot as a co-worker or more as an obstacle. The questionnaire utilized a 4-level Likert scale without a neutral position, avoiding the issue of acquiescence bias. The response options included *strongly agree*, *agree*, *disagree*, *strongly disagree*. This approach ensured that participants could provide their opinions without being influenced by a middle or neutral position.

2.5.3 Results

A total of 16 individuals participated in the experiment, carrying out both runs with and without the prediction strategy. The participants' responses to the questionnaire after each experiment are visualized in a boxplot presented in Fig. 2.8. The median of each group is indicated by the mark, while the box represents 50% of all responses, reflecting the variance. The whiskers indicate the most extreme answers.

Regarding the feeling of safety in close proximity to the robot, the majority of participants state either agreement or strong agreement for both runs. However, when the human motion prediction is employed, the median of all responses improves to *strongly agree*, indicating a clear enhancement in perceived safety. An improvement was observed in terms of comfort, in which the prediction strategy leads to a reduction in variance, with responses predominantly falling in the agreement category. Without prediction, the boxplot displays a wider spread between *agree* and *disagree*.

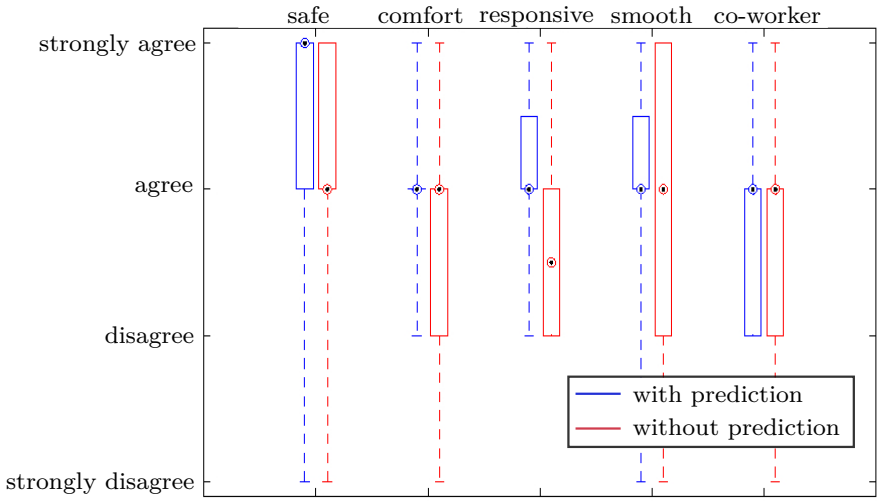


Figure 2.8: Boxplot of the questionnaire

The responsiveness of the robot was also enhanced in the case with the prediction strategy. As the robot reacted promptly when the human's motion intersected the avoidance range, most of the participants agree that the robot was responsive. The assessment of smoothness did not yield a clear distinction as it primarily resulted from the high damping factor mentioned in Section 2.5.2. However, it is worth noting that the focus of the experiment was on collision avoidance behavior, and the perception of smoothness might not have been a primary concern. Further, no higher acceptance rate of the robot being seen as a co-worker could be

achieved. This outcome is not surprising considering that the scenario did not involve genuine collaboration. The two tasks were chosen individually to provoke collisions and study obstacle avoidance behavior.

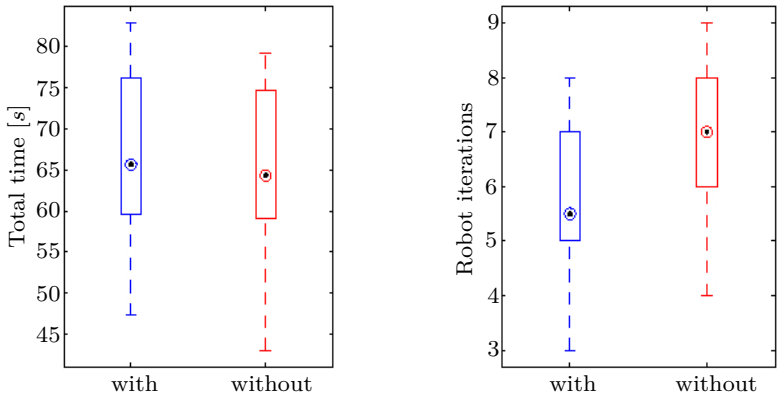


Figure 2.9: Comparison of total experiment time and number of iterations of the robot

In addition to the participants' perceptions, Fig. 2.9 illustrates the overall duration of the experiments and the number of iterations performed by the robot. No definitive conclusion can be drawn regarding the duration, as the variance in the total time of the subjects does not provide a significant trend. However, the decreased number of block reordering iterations by the robot indicates that, due to its increased responsiveness, the robot switches to avoidance mode earlier and more frequently.

2.6 Discussion and Conclusion

In this chapter, we have introduced a simple yet efficient approach for enhancing the safety and performance of robots operating in close proximity to humans. By incorporating human motion prediction into the local obstacle avoidance algorithm, we have improved the capability of the robot to anticipate and react to human movements, ultimately reducing the risk of collisions and enhancing overall collaboration. The case study we presented, although specific to one scenario, involved a considerable number of participants and demonstrated the potential effectiveness of our approach in human-robot collaboration.

One of the major advantages of our approach lies in its ability to enhance safety in close proximity human-robot collaboration. By integrating human motion prediction, our approach enables the robot to proactively avoid potential collisions, thereby reducing the likelihood of accidental impacts and injuries. Moreover, our approach improves the responsiveness of the robot. By predicting the future positions of the human, the robot can adjust its trajectory and avoid potential conflicts before they occur. This results in improved responsiveness and smoother interactions, enhancing the overall user experience and increasing the efficiency of collaborative tasks, as evidenced by our case study where human participants reported feeling more comfortable when working with the robot.

The improved safety and responsiveness achieved through our approach also lead to better task coordination. By aligning its movements with the intended actions of the human, the robot can effectively coordinate tasks and adapt its trajectory accordingly. This facilitates

seamless collaboration and enhances the synchronization between the robot and the human participant, leading to more effective and efficient task execution. Additionally, our approach offers the advantage of mitigating communication delays. By utilizing human motion prediction, the robot can compensate for potential communication delays between the motion tracking system and itself. Predicting the future positions of the human allows the robot to start reacting in advance, bridging the gap caused by delays. This mitigates the impact of communication latency, ensuring timely and appropriate responses from the robot and improving the overall performance of the system.

Despite implementing known assumptions and restrictions on human movement to facilitate human motion prediction and improve accuracy, inherent uncertainty persists in predicting human behavior. Factors such as sudden changes in movement patterns, unpredictability, and external influences can introduce errors and inaccuracies in the predictions. This uncertainty can lead to false positives or false negatives in the obstacle avoidance decisions of the robot, potentially affecting the overall performance and reliability of the system. The dependency on accurate tracking is also one of the problems that affect not only our experiment but most of the HRC scenarios. Any inaccuracies or limitations in the tracking system can directly impact the quality of the predictions and subsequently affect the performance of the obstacle avoidance algorithm. The reliability and precision of the tracking system become critical factors in the success of integrating human motion prediction into local obstacle avoidance.

Another limitation of the experiment presented in this chapter is the lack of diversity in the scenarios tested. The experiment presents currently focus on a specific and rather simple pick and place scenario and there are restrictions on the movements of the human as well as the robot. The trajectories of the robot in the experiment is predefined as point-to-point motion. While such setups allow for precise evaluation, they may not adequately represent the dynamic and complex nature of real-world human-robot collaboration. Incorporating a broader range of scenarios, including crowded or cluttered environments, varying speeds, and unpredictable human behaviors, would enhance the external validity of the results.

With the successful integration of human motion prediction into obstacle avoidance, an important next step in advancing the field of human-robot collaboration (HRC) is to explore how to generate robot motions that foster seamless interaction and establish a sense of comfort and trust between humans and robots. By leveraging the insights gained from predicting human behavior, we can delve deeper into developing sophisticated algorithms and techniques to dynamically adapt the motion of the robot in real-time, considering both responsiveness and naturalness. While this chapter has primarily focused on enhancing the capability of the robot to anticipate and react to human movements, the generation of robot motions has been omitted, leaving room for further investigation.

To ensure a harmonious collaboration, it is crucial to consider how the movements of the robot can be synchronized with the human's actions. This involves designing robot motion policies that not only align with human intentions but also adapt to changing circumstances, maintaining a level of fluidity and naturalness in their execution. Additionally, incorporating human feedback into the adaptation process is essential. By allowing humans to provide feedback on the behavior of the robot, the system can iteratively refine its motion policies, tailoring them to the specific preferences and requirements of the human partner. Various machine learning techniques, such as reinforcement learning algorithms or other adaptive strategies, can be employed to utilize this feedback and continuously improve the behavior of the robot over time, leading to a more personalized and optimized human-robot interaction experience.

However, integrating human feedback in real-time presents a challenge. The robot needs

to consider human feedback on-the-fly and promptly adapt its motions to maintain a smooth and natural collaboration, similar to human-to-human interactions. This requires developing efficient algorithms and mechanisms that allow for real-time processing of human feedback and seamless integration into the motion generation process. By addressing all aspects and challenges that we have just mentioned, the next chapter will present a comprehensive solution and approach that enables the robot to generate and adapt its motions based on the feedback of its human partner in real-time, fostering a highly interactive and mutually beneficial collaboration in close proximity HRC scenarios.

3

Adaptation and Transfer of Robot Motion Policies for close Proximity Human-Robot Interaction

This chapter was previously published in [HDOEW19]

In the context of human-robot collaboration in close proximity, ensuring safety and comfort is crucial for efficient task execution. Building upon the findings of Chapter 2, which investigated the integration of human motion prediction into robot actions, particularly focusing on obstacle avoidance behaviors, we recognize the importance of enhancing safety and comfort in HRC scenarios. The results obtained in Chapter 2 underscored the potential of improving the understanding of the robot in human behavior, enabling it to plan and react in a more natural and intuitive manner. From the human perspective, when the behaviors of the robot align with their expectations, trust in the robot increases, fostering a greater sense of comfort and willingness to collaborate. This mutual trust and cooperation are essential for achieving efficient joint tasks.

To establish effective collaboration, the behaviors of the robot must be comprehensive and easily interpretable by humans, allowing them to reliably predict its intentions. For instance, in a scenario involving the selection of different objects, the motion of the robot should clearly indicate which object it intends to pick, enabling humans to anticipate its actions by observing specific parts of its movements. However, determining how the robot should execute its motions to convey its objectives, especially in the presence of multiple objects, poses a challenging question. Furthermore, each individual may have their own unique way of predicting robot motions, implying that the predictability of robot behavior may vary among different people.

Continuing from the previous work, this chapter explores various aspects of robot motions in HRC and addresses the generation of such motions for the robot. Our focus lies in determining how the robot should execute its motions when interacting with humans in close proximity to ensure their safety and comfort. For safety, the robot must possess reliable and efficient capabilities to avoid dynamic obstacles, including human arms. In terms of comfort, it is essential for the the trajectories and avoidance behavior of the robot to be predictable to humans. To address these requirements, we present a framework that utilizes policy improvement methods to generate predictable motions with dynamic obstacle avoidance for the robot in human-robot interaction scenarios.

In our framework, trajectories are generated using Dynamic Motion Primitives, augmented with a potential field term that penalizes trajectories that may lead to collisions with obstacles. Additionally, we employ a data-driven approach to predict human movements, facilitating proactive avoidance strategies. To ensure comfort and predictability for human co-workers, we define a cost function that considers various factors such as human response time and

joint jerk. During human-robot interaction, this cost function is minimized using policy improvement through black-box optimization, resulting in robot trajectories that adapt to human preferences while avoiding obstacles.

To evaluate the effectiveness of our approach, user studies are conducted to assess the trust and comfort experienced by human co-workers when collaborating with the robot. Furthermore, these studies are extended to various scenarios and different users to analyze the transferability of tasks, enhancing the learning performance of the robot when switching to new tasks or adapting to different co-workers. By comprehensively investigating safety, comfort, and task transferability, this chapter contributes to the development of more advanced and adaptive human-robot interaction systems.

3.1 Introduction

The new generation of robots are no longer industrial machines behind fences but rather being integrated more in our daily lives as well as in collaborative manufacturing scenarios. They are expected to assist elderly people in daily tasks, to support customers in markets, to work as a partner with humans in factories, etc. For all of these tasks, the robots are required to interact with the human. Especially in collaborative scenarios, where robots work with humans as co-partners in joint tasks, they need to interact more efficiently since it will increase the overall performance. Looking at the case when two humans perform a joint task as an example, the humans can anticipate each others' movements and perform a complementary action without the need of verbal communication. This facilitates teamwork and increases the efficiency of joint tasks [EMCB07]. Similarly, robots are expected to move in a natural way, similar to human-human interaction. To achieve such an interaction between humans and robots, the first requirement is the motion of the robot must be readable to the human [KTS+10], which means the human partner is able to understand its intentions and the motion/behavior of the robot has to meet the expectations of the human partner. In the work of [LK16], this is defined as legible robot behavior. Another requirement is that the robot has to be aware of its surroundings to provide a safe environment, while still being efficient in performing its task. Legibility and safety are therefore the two important criteria that increase the efficiency of joint collaboration between human and robot.

In order for humans to feel comfortable working with robots, especially in close proximity, they have to understand the behavior of the robot and be able to infer their actions or in other words, the behavior of the robot must be legible to the human partner. Identifying the factors that contribute to these natural movements is not trivial. According to a study conducted by [DWK+05], participants want robots assisting at home to be predictable, controllable and have human-like communication. Another study [KSS+07] that investigated the subjective effects of direction of approach and distance of robots when handing an object over to humans, came to the conclusion that the frontal approach is subjectively preferred most by the participants since it is the most predictable. In addition, [BBB13] discovered that understanding and predicting the behavior of the robot increases the well-being of humans.

As such, the next question arises is how such legible robot motion can be generated. [DLS13] tried to find one mathematical metric for legibility. However, this is insufficient as robot motion gets perceived differently by individual humans and depends on several factors including the configuration of tasks, robot positions and human positions. It is therefore necessary to have a framework in which the robot is able to learn legible motions by interacting directly with the human. In this way, all possible influencing factors will indirectly be included.

In addition to legibility, ensuring the safety of humans in close proximity scenarios and facilitating joint collaboration require the robot to be aware of the human's position and potentially predict their motion ([OSHDW17]). This information allows the robot to modify its trajectories in real-time, effectively avoiding collisions with humans and ensuring a safe interaction. By combining this safe behavior with legibility, the robot can enhance human comfort and foster natural collaboration.

It is also important to address that the main drawback of many learning approaches is the training time.The iterative nature of the learning process and the need for repeated training for each new task and human partner are time-consuming. In practical scenarios, having a flexible algorithm that can adapt to parameter changes, such as variations in robot position, task configuration, and human perspective, without the need for retraining would be highly beneficial. This flexibility enables the algorithm to extend to different scenarios and tasks, saving valuable training time.

In this work, we present a framework that tackles the interdependency between legibility, safety, and efficiency to achieve natural human-robot interaction. Our approach employs a reinforcement learning methodology to generate legible robot motion that is transferable to different tasks and enables safe collaboration in close proximity. The framework involves a joint scenario where both the human and robot are engaged in reaching similar objects. Over time, the robot adapts its motions based on the reactions and predictions of the human partner. Through training, the robot becomes more efficient and predictable in performing its tasks, thereby increasing human comfort and enhancing the effectiveness of collaboration. Furthermore, our framework is designed to be generalizable to similar tasks using learned policies, reducing the need for extensive training time.

3.2 Related Work

The investigation of safety and legibility in robot motion in close proximity has traditionally been conducted separately. Some methods have focused on generating real-time obstacle-avoiding trajectories, while others have developed optimization-based algorithms for legible robot motions.

Legible (or predictable) robot motion was first introduced in [DWK+05]. The result from their survey confirms the necessity of predictable behaviors in future robot companions. However, the paper does not focus on how to generate predictable behaviors for the robot. In the works from the Robotics and Artificial Intelligence Group at LAAS/CNRS [ACM+05, SMAS07, SCAR08, SMUB+10, SA12], they developed a human aware motion and manipulation framework which is able to generate safe, comfortable and socially acceptable motions. The framework is verified on a mobile robot manipulator in simulated environment and in a hand-over scenario on real setup. The safety criterion introduced in their works, however, is based on the distance between the robot and the human, i.e. the robot should keep its distance from the human when performing tasks. While their framework achieves safe and legible motion, it does not support interaction between humans and robots in joint tasks conducted in close proximity. The absence of collaboration between human and robot is evident in their results, where only the robot performs the tasks.

The work from [DLS13] focuses explicitly on generating predictable and legible robot motion. In their work, the authors differentiate between legibility and predictability and provide a mathematical model to produce and evaluate such motions. They assume that humans expect robots to be efficient in their movements and compare all possible goals in the scene

to determine the most probable one. This probability is formulated mathematically and is being maximized for the targeted goal. This approach has some limitations. The algorithm was tested only with two goals for the robot, which the human had to predict when pausing a video which showed the robot moving to one of the two. This setup was very simple as the probability of selecting a goal (randomly) is already 50%. Another limitation is that the subjective evaluation of robot efficiency differs from one individual to another and the algorithm does not allow to adjust the movements of the robot to individual preferences of each participant.

In the work of [SGBL15], the team generates robot motions that learn from the observation of a human participant and iteratively reduce the human's reaction time. Here, Dynamic Motion Primitives (DMPs) are used for motion planning. Policy Improvement through Black Box Optimization (PIBBO) [SS12] is applied to improve the legibility of the robot to the human iteratively. This is done by only optimizing human guessing time about the action of the robot and the correctness of the prediction without defining formal criteria about legibility. This approach provides flexibility in choosing the relevant parameters to be optimized to obtain legible motion.

Recently, [BGLS17] showed that transferring the learned policy to other individuals leads to better prediction in the beginning and can thus lead to shorter adaptation times for new subjects. However, in this work, no close interaction scenarios were considered as no necessary collision avoidance methods were integrated and only the policy transfer to other subjects was investigated, not the policy transfer to new tasks.

Safety for humans during interaction with the robot, in general, involves several aspects and criteria [RBL+17]. There are also different categories of methods to ensure safety for the human partner [LFS17b] i.e. safety through control, motion planning, consideration of psychological factors, etc. Within this work, we limit the safety aspect to the obstacle avoidance behavior of the robot and therefore only mention about methods that are able to provide this functionality to the robot. In this aspect, potential field [Kha90] is a very popular and widely used approach due to its simplicity and real-time capability. [FKDLK12] and [HDOH+15] utilize the potential field idea in their works to provide obstacle avoidance behavior on the end-effector of an articulated robot. In the work of [PHPS08], the authors introduce the dynamic potential field to adapt robot trajectories while avoiding obstacles in mid-motion. This dynamic potential field is used with the inverse kinematics with null-space constraints to further ensure collision avoidance between the human and robot's links. However, the aim of these approaches was not to enable the robot to interact with humans, but rather to perform desired movements in the presence of obstacles.

In a recent study by [OSHDW17], a stochastic motion planning algorithm is introduced that predicts human motions and adjusts the robot's trajectories on-line to avoid the predicted region. For the prediction of the human movement, Probabilistic Movement Primitives (ProMPs) were used, which were first introduced by [PDPN13]. This method learns the distribution of the motion during training and allows prediction of human motion in the online phase. This allows close interaction between humans and robots, but does not examine predictable or legible motion.

Inspired by the work of [SGBL15] and considering the requirements of joint human-robot collaboration in close proximity, in this Chapter, we extend the learning approach in [SGBL15] with the potential field method. Our contribution is therefore a learning framework incorporating real-time obstacle avoidance to allow humans and robots working together in close proximity and therefore both legibility and safety aspects are tackled within our framework. The human partner no longer remains outside the workspace of the robot as a passive observer

but actively cooperates with the robot in joint tasks within the same workspace. Additionally, we develop a task generalization method that facilitates the generation of policies for new tasks based on previously learned tasks. This approach enables the robot to adapt more quickly to new tasks, reducing training time. We evaluate our approach using an articulated *KUKA* robot in both virtual reality (VR) and real robot experiments, and complement our evaluation with a human study.

In the following sections, we introduce our legible motion framework in Section 3.3 and present our task generalization method in Section 3.4. We then evaluate the improvement of our framework and task generalization approach through experiments in Section 3.5. Finally, we provide further discussion and conclude our work in Section 3.6.

3.3 Legible Motion Framework in HRC in Close Proximity

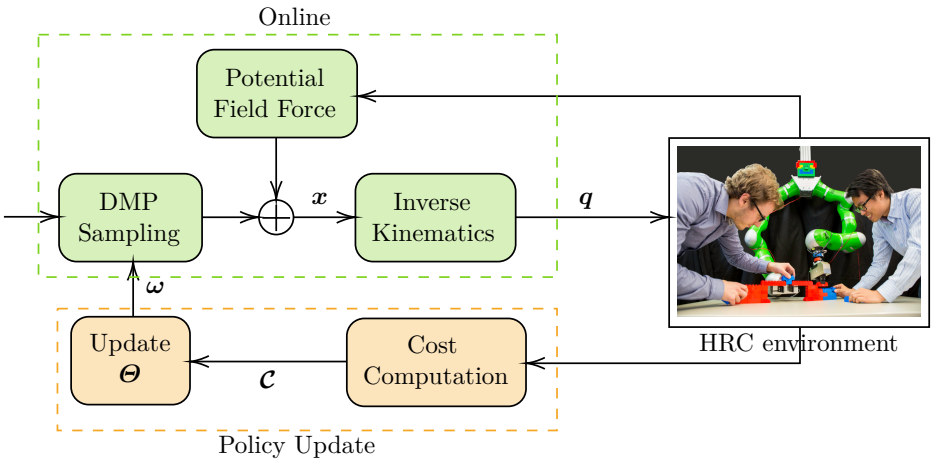


Figure 3.1: Overview of the human-guided policy improvement framework. In each iteration, DMPs generate Cartesian trajectories $\boldsymbol{x}$, which are converted to joint angles $\boldsymbol{q}$ using inverse kinematics. During interaction, potential field force is added to modify the trajectories online to provide safety when the human gets close to the robot. After execution, a cost function $\boldsymbol{C}$ is evaluated to update the policy $\boldsymbol{\Theta}$ of the DMPs using PIBBO for the next iteration.

A general overview of our framework is depicted in Figure 3.1, illustrating the key steps involved. The primary objective of our framework is to facilitate the generation of legible robot motion through direct interaction between the human and the robot. Both entities collaborate in a joint scenario, such as the task of reaching similar objects. The robot will adapt its motions over time corresponding to the reaction/prediction of the human partner. Through training, the robot acquires the ability to perform tasks more efficiently and predictably. This not only enhances human comfort but also improves the overall effectiveness of the collaboration. In summary, the framework can be described in three essential steps as outlined below

1. Firstly, Dynamic Movement Primitives (DMPs) are used to generate smooth trajectories with modifiable parameters. These trajectories are generated in Cartesian space and converted into the joint space of the robot using inverse kinematics.

2. DMP trajectories are then executed by the robot in the online phase where the robot collaborates with the human in a joint task. During execution, a potential field force is added to modify the DMP trajectories to ensure safety of the human.

3. A cost function which evaluates how the human partner perceives each trajectory is computed. These costs are then used to update the policy, which comprises the parameters of the DMPs in our framework. In the next iteration, new trajectories are sampled based on the updated policy and the procedure repeats until it converges to an optimal predictable trajectory or the maximum number of iterations is reached.

DMP trajectories are the trials/samples that the robot performs to understand how his human partner perceives a legible motion. By changing the parameters of the DMPs, the robot is able to exploit the working area and approach the goal from different angles. The human reacts to the robot by moving to his corresponding task. Each trajectory performed by the robot is then evaluated based on the human reaction formulated in a predefined cost function. This cost function reflects the perception of the human on how legible this trajectory is. Base on the evaluation of the cost function of each trajectory, the DMP parameters will be modified in favor of the ones that are more predictable to the human (smaller costs). This is done by the policy update method called Policy Improvement through Black Box Optimization (PIBBO). After the DMP parameters (policies) are updated, the robot rolls out new samples from these parameters for the next iteration. The procedure is then repeated until the trajectories converge or the maximum number of iterations is reached. Note that all of these computations are done at the beginning of each iteration.

To prevent collision between human and robot during execution (online phase), the DMP trajectories are modified using a potential field force. This potential field force is proportional to the relative distance between the human and robot and returns an error vector that is added into the current DMP trajectory. As a result, the robot will move away when the human comes close, and recovers his task when the area is free. Additionally, in order to increase safety in close proximity, human motion is predicted using Probabilistic Movement Primitives (ProMPs) [PDPN13] and serves as supplementary information added into the potential field force. ProMPs is a recent approach that is able to generate/represent movement from a given trajectory distribution. After training with a set of human motion observations, we used ProMPs in the online phase to predict the movement of the human hand and incorporate this information into the potential field. This helps the robot react faster and can avoid the human more actively.

In Section 3.3.1, we provide a concise introduction to DMPs and outline how they generate smooth trajectories. The policy update method, PIBBO, is subsequently introduced and explained in Section 3.3.2. We then proceed to clarify how safety for the human partner is ensured through the potential field force in collaboration with ProMPs in Section 3.3.3. Finally, we delve into the detailed explanation of the cost function, which evaluates the performance of each trajectory with a specific emphasis on collaboration effectiveness, in Section 3.3.4.

3.3.1 Dynamic Movement Primitives

Dynamic Movement Primitives (DMPs) were initially introduced by Stefan Schaal's group [Sch06] and later updated by Auke Ijspeert [INH+13]. The motivation behind DMPs stemmed from the need to represent complex motor actions that could be easily adjusted without manual parameter tuning or concerns about instability.

One key advantage of DMPs is their ability to modify trajectories online without the need for complete recalculation. This feature proves particularly beneficial in scenarios where the environment undergoes frequent changes, such as in HRC scenarios. In these cases, robot trajectories often require constant online modifications based on the interaction between the

robot and the human. Therefore, the ability to generate modulated trajectories online becomes a critical component for enabling the robot to effectively participate in HRC scenarios. It is worth noting that, aside from DMPs, classical spline methods [MMM00] can also generate robot motions by imitating pre-defined or observed trajectories. However, due to the time-based nature of splines, they are not easily modulated online by external variables, making DMPs more suitable for scenarios with dynamic changes.

In DMPs, complex movements are considered as sequences or parallel sets of primitive actions, and DMPs serve as a mathematical formalization of these primitives. The fundamental concept behind DMPs is the combination of two systems: the first system defines a dynamical system with well-specified and stable behavior, while the second system introduces an additional forcing term that enables the dynamical system to follow desired trajectories on demand.

The dynamical system is a closed loop spring-damper system

$$\tau\ddot{y} = \alpha(\beta(y_g - y) - \dot{y}) \tag{3.1}$$

that converges to the defined attractor state y_g where τ is the time constant, α and β are positive constants. By setting β to $\alpha/4$ we get a critically damped system. The variables y, $\dot{y}$ and $\ddot{y}$ are the position, velocity and acceleration, respectively. The benefit of having a second-order dynamical system is that it can be applied to motor control problems which are usually described by second order differential equations and require knowledge of position, velocity and acceleration for control. A controller would convert them into motor commands, which account for non-linearities in the dynamics of the system.

The forcing term, which forms the second part of the DMPs, deforms the trajectory to match a desired shape. Thus, the spring-damper system is modulated to

$$\tau\ddot{y} = \alpha(\beta(y_g - y) - \dot{y}) + f(x), \tag{3.2}$$

where $f(x)$ is the forcing term consisting of a weighted sum of Gaussian basis functions multiplied by a canonical dynamical system, denoted as x. The canonical system x is obtained by

$$\dot{x} = -\alpha_x x, \tag{3.3}$$

where α_x is a constant. The canonical system state x in (3.3) starts at some arbitrary value and goes to 0 as time goes to infinity. This ensures convergence to the goal while keeping the forcing term not directly dependent on time. The forcing function $f(x)$ hence has the form

$$f(x) = \frac{\sum_{i=1}^{N} \psi_i(x)\omega_i}{\sum_{i=1}^{N} \psi_i(x)} x, \tag{3.4}$$

where

$$\psi_i(x) = \exp\left(-\frac{1}{2\sigma_i^2}(x - c_i)^2\right) \tag{3.5}$$

defines the Gaussian basis functions with means c_i and variances σ_i. In (3.4), N is the number of basis functions and ω_i are modifiable weights, which are adjusted to match the desired trajectory. They are optimized by the policy improvement method explained in Section 3.3.2.

Since the mass spring-damper system leads to high initial accelerations, which is not desirable for robots, we use a goal system, which moves the attractor state of the system from

the initial state y_0 to the goal state y_g during the movement. This delayed goal attractor y_{gd} itself is given as an exponential dynamical system that starts at y_0 and converges to y_g.

$$\dot{y}_{gd} = -\alpha_g(y_g - y_{gd}) \tag{3.6}$$

Thus the equation for the DMPs resolves to

$$\tau\ddot{y} = \alpha(\beta(y_{gd} - y) - \dot{y}) + f(x) \tag{3.7}$$

The DMPs has several advantages, which make it suitable for our framework:

- It is guaranteed to converge to the goal, since the canonical system is 0 at the end of every movement.
- The weights ω_i can be adapted to generate any desired trajectory. In our case this is especially relevant, since we want to learn the optimal trajectory and adjust the weights online with each interaction.
- As there is no time-dependency, the duration of the movement can simply be altered by adjusting τ.

3.3.2 Policy Improvement through Black-box Optimization

In the proposed framework depicted in Figure 3.1, the DMP trajectories generated in Section 3.3.1 play a crucial role in the online phase, where they are executed by the robot during joint tasks with the human partner. The parameters of the DMPs are continuously updated based on the collaboration and reactions of the human. Specifically, the weight factors of the Gaussian basis functions, which serve as the deformation term in the trajectory (as explained in Section 3.3.1), are the parameters subject to modification. Typically, this update process involves solving an optimization problem using methods like gradient descent or policy improvement. However, given that the update procedure relies on trial and error with respect to a utility function, policy improvement methods prove to be more suitable for addressing this specific problem.

Policy improvement methods seek to optimize the parameters of a policy w.r.t. a utility function and in general, have two basic steps:

1. Exploration by perturbation: The exploration noise $\boldsymbol{\epsilon_t}$ can be either added to the actions, i.e. the output of the policy $(\pi_{\boldsymbol{\theta}}(x)+\boldsymbol{\epsilon_t})$, or directly to the input parameters of the policy $(\pi_{\boldsymbol{\theta}+\boldsymbol{\epsilon}_t}(x))$.
2. Policy update: Here, the parameters of the policy are updated in order to minimize a predefined cost metric C. Usually, gradient descent is applied to iteratively converge to a local minimum. Another method is the reward-weighted averaging, which is used in our application.

Reward-weighted averaging does not require differentiability of the cost function, which makes it more stable than gradient descent if the cost function is not continuous.

Specifically for this work, we choose Policy Improvement through Black-box Optimization (PIBBO) as our policy improvement method [SS12]. PIBBO treats the whole control trajectory as a black-box, i.e. no assumptions are made about the search space or the cost function.

An important property of PIBBO is that the search is done in the space of policy parameters, thus it is a parameter perturbing approach. The output u_t of the policy is computed as:

$$u_k = \pi_{\boldsymbol{\theta}+\boldsymbol{\epsilon}_k}(x), \quad \text{with} \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \Sigma) \tag{3.8}$$

In our case the policy $\pi_{\boldsymbol{\theta}}$, is the DMP and $\boldsymbol{\theta}$ are the corresponding weights for the Gaussians.

The parameter update is done using reward-weighted averaging. First, the cost C_k for each trajectory roll-out is computed. Then we assign higher probabilities P_k to trajectories with a lower cost and vice versa.

$$P_k = \frac{e^{-1/\lambda C_k}}{\sum_{k=1}^{K} e^{-1/\lambda C_k}} \tag{3.9}$$

k is the number of roll-outs and λ is the decay factor that varies between 0 and 1. Small value in the decay factor results in the decreased covariance matrix in the exploration step which then increase the convergence speed of the algorithm.

The parameter update is then given as

$$\delta\boldsymbol{\theta} = \sum_{k=1}^{K} P_k \boldsymbol{\epsilon}_k \tag{3.10}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \delta\boldsymbol{\theta}. \tag{3.11}$$

After taking the weighted average of all roll-outs, the new DMP with updated parameters $\boldsymbol{\theta}$ follows the trend of trajectories with high probabilities (i.e. low costs). This process of perturbing and updating is repeated until the desired cost value is achieved or the maximum number of updates is reached. Here is a brief summary for the steps of policy improvement with black box optimization:

1. Exploration
   a) Get $\boldsymbol{\theta}_{\text{init}}$
   b) Get k samples from Gaussian distribution, $\boldsymbol{\theta}_k \sim N(\boldsymbol{\theta}_{\text{init}}, \Sigma)$
   c) Execute the trajectories generated by DMP parameters $\boldsymbol{\theta}_k$ to get C_k for each execution.
2. Parameter Update
   a) Convert the costs C_k into weights P_k using (3.9)
   b) Update the parameters using reward weighted averaging in (3.10), (3.11) to get $\boldsymbol{\theta}_{\text{new}}$
   c) Decrease the covariance matrix, $\Sigma \Leftarrow \lambda\Sigma$ with a decay factor $0 < \lambda \leq 1$
   d) Return to the exploration step with $\boldsymbol{\theta}_{\text{new}}$ instead of $\boldsymbol{\theta}_{\text{init}}$

In our proposed framework, we assign a dedicated DMP to each goal, and these DMPs are constructed with a fixed number of radial basis functions. During the training phase, the robot executes various trajectories by perturbing the parameters of the DMPs. These trajectories are then evaluated based on the cost values obtained from the interaction with the human partner. Trajectories that yield lower costs are associated with higher rewards, influencing the exploration behavior of the PIBBO algorithm in subsequent updates. This iterative process continues until the trajectories converge or a predefined number of iterations is reached.

The result of this training process is the identification of optimal trajectories for each goal with respect to the evaluated cost function. These trajectories exhibit desirable characteristics in terms of legibility and collaboration effectiveness. However, before delving into the detailed explanation of the cost function, it is essential to address the safety aspect concerning the human partner during close proximity interactions. By considering safety measures, we ensure the well-being of the human partner throughout the collaboration process.

3.3.3 Safety Aspect in Close Proximity

As the human works together with the robot in close proximity, safety of the human needs to be considered. It is crucial to implement measures that allow the robot to avoid physical collisions with the human. In our previous work, we introduced an obstacle avoidance method that serves this purpose. In this study, we build upon the ideas presented in Chapter 2 but with some modifications. In order to prevent collisions, we adopt the concept of creating an artificial repulsive force that pushes the robot away from potential obstacles, including the human partner. However, instead of relying solely on the minimum-jerk model to improve the reactivity of the robot, we employ a different approach known as Probabilistic Movement Primitives (ProMPs). By leveraging ProMPs, we can predict the motion of the human and incorporate its anticipated effects into the repulsive force. In Section 3.3.3.1, we will introduce our method for generating the repulsive force, which facilitates obstacle avoidance. Subsequently, in Section 3.3.3.2, we will provide an overview of ProMPs and explain how we extract predictions of human motion using this approach.

3.3.3.1 Repulsive Force with Artificial Potential Field

The robot trajectory is generated by the DMP at the beginning of every update. We want to modify this trajectory to avoid the human partner while still generating smooth motions and following the original DMP trajectory when the human is out of reach. As the DMP trajectory is already smooth based on its formulation (see 3.3.1), the artificial repulsive force also has to generate a smooth transition on the robot. This is important for the human partner to feel comfortable when working with the robot. This kind of behavior can be achieved by using the concept of a virtual mass-spring-damper system regarding to external forces as presented is Chapter 2

$$\boldsymbol{F}_{\text{ext}} = \boldsymbol{M}\ddot{\boldsymbol{e}} + \boldsymbol{D}\dot{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{e}, \tag{3.12}$$

where $\boldsymbol{F}_{\text{ext}} \in \mathbb{R}^3$ represents an external virtual force, which is excited whenever the human enters the safety area around the end-effector of the robot. This virtual mass-spring-damper system results in a smooth transition in the vector $\boldsymbol{e} \in \mathbb{R}^3$ regardless $\boldsymbol{F}_{\text{ext}}$. This vector indicates the modification length and direction to be added to the DMP. $\boldsymbol{M}, \boldsymbol{D}, \boldsymbol{K} \in \mathbb{R}^{3\times3}$ are positive definite matrices that represent the mass, damping and stiffness of the virtual system. In our proposed setup, $\boldsymbol{M}$ is chosen as the identity matrix, $\boldsymbol{K}$ and $\boldsymbol{D}$ are diagonal matrices chosen to adapt the desired reaction to virtual forces. Increasing the damping results in a slower reaction but smoother movement of the robot. The external virtual force $\boldsymbol{F}_{\text{ext}}$ is computed based on potential fields w.r.t the distance between the end-effector of the robot and obstacles.

For the external forces $\boldsymbol{F}_{\text{ext}}$, we use the same idea as in Chapter 2 to generate a smooth profile

$$\boldsymbol{F}_{\text{ext}} = \frac{\boldsymbol{F}_{\max}}{1 + \exp\left(\left(\|\boldsymbol{d}(E,O)\|\,(2/\rho) - 1\right)\gamma\right)}, \tag{3.13}$$

where $\mathbf{F}_{\max}$ is the maximum force applied, $\|\boldsymbol{d}(E,O)\|$ is the distance between obstacle O and end-effector E, ρ is the threshold distance that defines the collision region around the end-effector and γ is a shape factor. The force reaches its maximum if the distance equals zero, and zero if the obstacle is outside the region, respectively. The steepness of the force profile within the threshold region regarding the distance can be adjusted by the shape factor γ. With $\boldsymbol{F}_{\text{ext}}$, the error vector $\boldsymbol{e}$ is obtained from (3.12) which return in the deviation needs to be added into the DMP to avoid the obstacle. Note that the approach presented alters the desired trajectory on position level rather than velocity level (see [FKDLK12]). This yields in a saturation of the position alteration, proportional to $\mathbf{F}_{\max}$. This effect is preferred in this use-case, as it results in a more human like behavior. As a consequence, the robot stays closer to its planned trajectory, rather than letting it being chased around by the human.

3.3.3.2 Human Motion Prediction with ProMP

Although the robot is able to avoid the human with the repulsive force generated from the potential field, its reaction time is an important factor that needs to be considered. In a confined workspace where the human usually interferes with the robot, the robot might not have enough time to react and fail to avoid the human partner. Increasing the safety region around the robot can improve the reaction time but results in a smaller workspace. Thus, in our framework, we estimate and predict the human motion and add this additional information into the repulsive force to increase the responsiveness of the robot.

In general, human motion estimation requires a specialized prediction method due to the inter- and intra-personal movement variations [Tod04]. To imitate such behavior online, we use Probabilistic movement primitives (ProMPs) and learn a distribution of a motion behavior by training with multiple trajectories performed for a specific task [PDPN13]. ProMPs represent a discrete trajectory $X = \{x_n\}$, $n = 0 \dots N$ defined by states x_n over time N with the formulation

$$\boldsymbol{y}_n = [x_n, \dot{x}_n]^\top = \boldsymbol{\Phi}_n^\top \boldsymbol{\omega} + \boldsymbol{\epsilon}_y, \tag{3.14}$$

where $\boldsymbol{\omega} \in \mathbb{R}^{k\times 2}$ is the weighting matrix over the $k \times 2$ dimensional time-dependent basis matrix $\boldsymbol{\Phi}_n = [\phi_n, \dot{\phi}_n]$ with k being the number of basis functions and $\boldsymbol{\epsilon}_y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_y)$ is zero-mean independent Gaussian noise, while $\boldsymbol{\Phi}_n^\top \boldsymbol{\omega}$ gives the mean of the trajectory. Introducing a Gaussian distribution to also represent variance $p(\boldsymbol{\omega}; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\omega}|\boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega)$ over the weighting vector $\boldsymbol{\omega}$ results in the following distribution for the trajectory:

$$\begin{aligned} p(\boldsymbol{y}_n; \boldsymbol{\theta}) &= \int \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{\Phi}_n^\top \boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_y)\mathcal{N}(\boldsymbol{\omega}|\boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega) d\boldsymbol{\omega} \\ &= \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{\Phi}_n^\top \boldsymbol{\mu}_\omega, \boldsymbol{\Phi}_n^\top \boldsymbol{\Sigma}_\omega \boldsymbol{\Phi}_n + \boldsymbol{\Sigma}_y). \end{aligned} \tag{3.15}$$

Using a set of motion observations, the parameters $\boldsymbol{\mu}_\omega$, $\boldsymbol{\Sigma}_\omega$ can be computed by maximum likelihood estimation [LG10].

By using this formulation, an online human motion prediction, where a trajectory along with the variance for each discretized time point is generated. This predicted trajectory can be used in different ways within our framework. An intuitive way is to select some predictions at different time points along the trajectory. These predictions represent the points in space where the human *might* occlude in the future and thus are treated as *incoming* obstacles that the robot has to avoid. This triggers the reaction of the robot even if the human is not currently within the safety region, which in turn increases the responsiveness of the robot. In case the human does not move toward the robot, these *incoming* obstacles do not create any disturbance, thus do not alter the robot desired position.

3.3.4 Cost Computation

In this section, we will explain how the cost function in our framework (Fig. 3.4) is defined. There are different aspects that we want to evaluate through the cost function:

- First is the legibility of the robot trajectories. There are different methods to measure this aspect. In the works of [DSAC11] and [LLK11], they show the participants robot motions and afterwards ask them to rate how legible the motions were perceived. In a quantitative level, [DLS13], [BGLS17] show the participants robot motions through videos/experiments and ask them to indicate immediately or press a button when they feel certain about the intention of the robot. Time and correctness of the prediction are used as the indicators for legibility in their works. Using the same approach as in [BGLS17], we also use the human prediction time and accuracy to form the cost of legibility.
- Second is the smoothness of the trajectories. This helps the human partners feel comfortable when working with the robot and be more confident approaching their goals. Smoothness also contributes in the legibility aspect since a jerky motion does not meet the expectation of the human. In our framework, we use the third derivative of the trajectories to form the cost of smoothness.

From the two aspects that we want to evaluate, several components are identified and also mixed up depending on the experimental setup. Here, we list all the costs used in this work:

- End-effector jerk V_{ej}: the sum of the third derivative of the end-effector position of the robot at each sample along trajectory.
- Angular jerk V_{θ}: the sum of the third derivative of the angular positions of the controlled joints of the robot at each sample along trajectory.
- Human prediction time V_{pred}: the time taken by the human to make a prediction about the robot's target. It starts when the robot starts moving and ends when the human reaches one of the targets.
- Accuracy V_{task}: whether the human prediction was correct, translating to 0 cost ($V_{\text{task}} = 0$), or if the prediction was wrong which results to a cost of 1 ($V_{\text{task}} = 1$).
- Human duration V_{dur}: the duration of the human movement between when the human starts moving and reaches the goal. It is a measurement of human's confidence in the presence of the robot.
- The weighted distance between the robot trajectories, V_{δ}, which measures how distinct the trajectory to the targeted goal is in comparison to the trajectories to the other goals. This cost is calculated using the following equation:

$$V_{\delta} = \left(\sum_{g=1}^{G} \sum_{t=0}^{T} \frac{1}{t} \ \| \boldsymbol{p}_t, \boldsymbol{q}_t \|_2 \right)^{-1} \tag{3.16}$$

  where G is the number of the goals excluding the targeted goal, g is the other goal whose trajectory is compared to the targeted goal trajectory, t is the time step at which we calculate the distance, T is the total time of the trajectory, $\boldsymbol{p}_t$ is the point at t in the trajectory to the targeted goal, $\boldsymbol{q}_t$ is the position at t in the trajectory to the goal g and $\| \boldsymbol{p}_t, \boldsymbol{q}_t \|_2$ is the Euclidean distance between $\boldsymbol{p}_t$ and $\boldsymbol{q}_t$.

In summary, the cost function has the form

$$V = \lambda_{\text{ej}} V_{\text{ej}} + \lambda_{\theta} V_{\theta} + \lambda_{\text{pred}} V_{\text{pred}} + \lambda_{\text{task}} V_{\text{task}} + \lambda_{\text{dur}} V_{\text{dur}} + \lambda_{\delta} V_{\delta} \tag{3.17}$$

where each cost component is weighted differently. In general, $\lambda_{\text{pred}}, \lambda_{\text{task}} > \lambda_{\text{ej}}, \lambda_{\theta}, \lambda_{\text{dur}}, \lambda_{\delta}$ as we want to have a high reward for trajectories that are more predictable to the human partner.

3.4 Task Generalization

While our framework produces policies that exhibit predictability, it is important to acknowledge that the learning process demands a significant amount of data and time to achieve convergence. Additionally, the trained policies are inherently tied to specific setups. Consequently, when the environment undergoes changes, such as alterations in the start and goal positions of the robot, or variations in the relative position of the human with respect to the robot, the robot must adapt to these new configurations.

To address this challenge, we can leverage the existing knowledge acquired from a fixed number of policies trained on specific settings. By doing so, we can exploit this knowledge to facilitate the adaptation to similar task variations, even when limited data is available. In essence, the previously learned policies already capture certain preferences of human perception. Therefore, they can be utilized to enhance the convergence rate of learning in cases where the robot has not been explicitly trained. We propose an approach to incorporate this generalization capability into the policy improvement framework within the context of HRC settings. This approach allows the robot to adapt more efficiently to new or modified scenarios, capitalizing on the insights gained from previous training.

Suppose that the set of tasks for the robot is defined as

$$\boldsymbol{\Phi} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_M \mid M \in \mathbb{N}\}, \tag{3.18}$$

where M is the number of available tasks. Within the scope of this work, a task is defined as a reaching motion, where the starting position is the same for all of the tasks and $\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_M$ are M different goal positions. Learning via PIBBO is done by selecting a subset $\boldsymbol{\mathcal{T}_i}$ out of $\boldsymbol{\Phi}$ and training trajectories for each goal in $\boldsymbol{\mathcal{T}}_i$, where

$$\boldsymbol{\mathcal{T}}_i = \{\boldsymbol{g}_{i1}, \boldsymbol{g}_{i2}, \cdots, \boldsymbol{g}_{iS}\} \subset \boldsymbol{\Phi}, S \in \mathbb{N}, \boldsymbol{g}_{ij} \neq \boldsymbol{g}_{ik}, \forall j \neq k \tag{3.19}$$

with a predefined $S < M$. The result of PIBBO is S policies that generate predictable trajectories for each $\boldsymbol{g}_{ij}$ over $\boldsymbol{\mathcal{T}_i}$. Each policy is parameterized by $\boldsymbol{\Theta}^{ij}_{\boldsymbol{\mathcal{T}}_i}$, e.g. in our case given as the weighted basis functions of the DMP. Note that the policy of $\boldsymbol{g}_{ij}$ depends on the remaining goals in $\boldsymbol{\mathcal{T}}_i$, which means a similar task will have different policies if it belongs to a different subset. We then denote the generated policy for a goal $\boldsymbol{g}_{ij}$ from $\boldsymbol{\mathcal{T}_i}$ as

$$\boldsymbol{\pi}_{\boldsymbol{\mathcal{T}_i}}(\boldsymbol{g}_{ij} \mid \boldsymbol{g}_{i\setminus j}) = \boldsymbol{\pi}_{\boldsymbol{\mathcal{T}_i}}(\boldsymbol{g}_{ij}) = \boldsymbol{\pi}(\boldsymbol{\Theta}^{ij}_{\boldsymbol{\mathcal{T}}_i}) \tag{3.20}$$

where $\boldsymbol{g}_{i\setminus j}$ is an abbreviation of all tasks in $\boldsymbol{\mathcal{T}_i}$ except j. This can be interpreted as the policy that generates the most predictable motion for goal $\boldsymbol{g}_{ij}$ given the remaining tasks in $\boldsymbol{\mathcal{T}}_i$.

Given a training set $\mathbb{T} = \{\boldsymbol{\mathcal{T}}_1, \boldsymbol{\mathcal{T}}_2, \cdots, \boldsymbol{\mathcal{T}}_k\}$ consisting of k batches of S elements from $\boldsymbol{\Phi}$ each, a new $\widetilde{\boldsymbol{\mathcal{T}}} \notin \mathbb{T}$ is drawn from $\boldsymbol{\Phi}$. The objective here is to find a new policy for a goal

$\boldsymbol{g}_m \subset \widetilde{\boldsymbol{\mathcal{T}}}$ such that the DMPs initialized using this policy improve the convergence rate of the learning procedure of $\boldsymbol{g}_m$ afterwards. This requires finding a mapping

$$\boldsymbol{\pi}_{\widetilde{\boldsymbol{\mathcal{T}}}}(\boldsymbol{g}_m) = \boldsymbol{h}\left(\boldsymbol{\pi}_{\boldsymbol{\mathcal{T}_1}}(\boldsymbol{g}_{11}), \ldots, \boldsymbol{\pi}_{\boldsymbol{\mathcal{T}_1}}(\boldsymbol{g}_{1S}), \ldots, \boldsymbol{\pi}_{\boldsymbol{\mathcal{T}_k}}(\boldsymbol{g}_{kS})\right) \tag{3.21}$$

with $\boldsymbol{h}(\cdot)$ is a function of all policies obtained from the training set $\mathbb{T}$. In fact, solving (3.21) is equivalent to finding the parameterized vector $\boldsymbol{\Theta}^m_{\widetilde{\boldsymbol{\mathcal{T}}}}$ in Eq. (3.20) for goal $\boldsymbol{g}_m$ in the new subset $\widetilde{\boldsymbol{\mathcal{T}}}$.

We claim that a predictable trajectory for each goal in $\boldsymbol{\mathcal{T}_i}$ depends on a set of features $\boldsymbol{\chi}$. These features characterize the interrelation between $\boldsymbol{g}_{ij}$ and $\boldsymbol{g}_{i\setminus j}$ in the subset $\boldsymbol{\mathcal{T}}_i$. They can be relative distances, angles, etc, depending on how the set of tasks $\boldsymbol{\Phi}$ is defined. These features vary for each goal in each subset. Given a predefined set of p features for goal $\boldsymbol{g}_{ij}$ in $\boldsymbol{\mathcal{T}}_i$, we denote the resulting feature vector for each goal as $\boldsymbol{\chi}_{\boldsymbol{\mathcal{T}_i}}(\boldsymbol{g}_{ij}) \in \mathbb{R}^p$. We now want to establish a relation between $\boldsymbol{\chi}_{\boldsymbol{\mathcal{T}_i}}(\boldsymbol{g}_{ij})$ and vector $\boldsymbol{\Theta}^{ij}_{\boldsymbol{\mathcal{T}}_i}$, which is the policy of $\boldsymbol{g}_{ij}$ in $\boldsymbol{\mathcal{T}}_i$. Furthermore, the weighted basis functions of the DMP in $\boldsymbol{\Theta}^{ij}_{\boldsymbol{\mathcal{T}}_i}$ are independent from each other, hence can be evaluated individually. Therefore, we propose an approximation to initialize each individual weight $\Theta \in \boldsymbol{\Theta}^{ij}_{\boldsymbol{\mathcal{T}}_i}$ as follow

$$\Theta = \beta_0 + \sum_{k=1}^{p} \beta_k \chi_k(\boldsymbol{g}_{ij}), \tag{3.22}$$

where $\chi_k \in \boldsymbol{\chi}_{\boldsymbol{\mathcal{T}}_i}$ represents a single feature in the set of p features for goal $\boldsymbol{g}_{ij}$. Given the trained policies from $\mathbb{T}$ and a predefined set of features $\boldsymbol{\chi}$, $\boldsymbol{\beta} = \{\beta_0, \boldsymbol{\beta}_k\}$ in (3.22) is obtained by solving the linear regression problem. Assuming $\boldsymbol{\Theta}^{ij}_{\boldsymbol{\mathcal{T}}_i}$ has N basis functions, then N linear regression problems of (3.22) are solved individually to obtain N sets of $\boldsymbol{\beta}$, denoted as $\boldsymbol{\beta}_i$ where i denotes the according basis function index.

From there, given the new subset $\widetilde{\boldsymbol{\mathcal{T}}}$, the generalized policy for a goal $\boldsymbol{g}_m$ in $\widetilde{\boldsymbol{\mathcal{T}}}$ is initialized by the approximate value as

$$\boldsymbol{\Theta}^m_{\widetilde{\boldsymbol{\mathcal{T}}}} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_N \end{bmatrix} \boldsymbol{\chi}_{\widetilde{\boldsymbol{\mathcal{T}}}}(\boldsymbol{g}_m), \tag{3.23}$$

where $\boldsymbol{\chi}_{\widetilde{\boldsymbol{\mathcal{T}}}}(\boldsymbol{g}_m)$ is the features of $\boldsymbol{g}_m$ in $\widetilde{\boldsymbol{\mathcal{T}}}$. We then use this policy as an initialization for the DMP when learning predictable motion for the new subset $\widetilde{\boldsymbol{\mathcal{T}}}$. Details about our implementation and results are outlined in Section 3.5.3.

3.5 Results

This section encompasses a series of experiments designed to assess the effectiveness of our framework and task generalization method. We begin by outlining the experimental setup in virtual reality (VR) in Section 3.5.1, providing crucial context for the subsequent evaluations. To evaluate the legibility of the generated motions, we present the corresponding results in Section 3.5.2. Following that, in Section 3.5.3, we analyze the outcomes of our task generalization approach, which aims to extend the capabilities of the framework to new tasks based on prior training. Furthermore, we address the safety aspect of our approach by conducting an experiment involving the physical implementation of a *KUKA* robot. The results of this real-world experiment are detailed in Section 3.5.4. Through these experiments, we aim to provide comprehensive insights into the performance and versatility of our framework, along with its potential applications in practical scenarios.

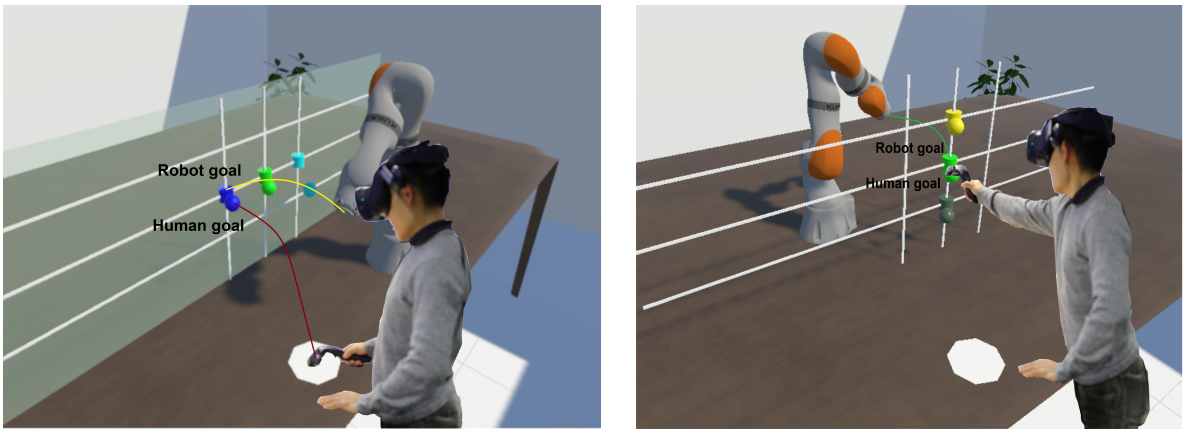


Figure 3.2: Experiment setup with different configurations: **a)** Human and robot are on the same side, **b)** Robot is on the opposite side of the human

3.5.1 Experimental Setup in Virtual Reality

Our main experiments are conducted within a VR environment, as depicted in Fig. 3.2. VR offers several advantages that greatly facilitate our work. Firstly, it provides a high degree of flexibility, allowing us to easily modify the environment or switch between different robot configurations. This flexibility is crucial for exploring various scenarios and assessing the adaptability of our framework. Secondly, VR provides a first-person perspective, closely resembling how humans perceive their surroundings. This immersive experience aligns well with the objectives of our work, making VR an ideal platform for our experiments.

During the experiment, the participant wears a *VIVE Pro* headset and stands in front of a table, upon which the robot is mounted within the VR environment. To enhance the realism and interaction, we incorporate a physical table that precisely matches its virtual counterpart. The positioning of the robot varies depending on the specific experiment. We consider two configurations: (i) the robot is mounted on the same side as the participant, and (ii) the robot is mounted on the opposite side of the participant, relative to the collaborative task area. The first configuration emphasizes the side-by-side perspective of the human observer, while the second configuration simulates a direct viewpoint of the motions of the robot as seen by the human. By comparing these two perspectives, we aim to investigate whether the observer's position influences the predictable motion of the robot. To facilitate collaboration between the human and the robot, we design tasks in which both parties are required to reach designated goals. The goals of the robot are visualized as cylinders, while the human's goals are represented as spheres. Each goal of the robot corresponds to a goal of the human, with matching colors. The close proximity of these goals allows us to evaluate the effectiveness of obstacle avoidance behavior, as illustrated in Fig. 3.2.

For each experiment, there are three different goals for the robot and three corresponding goals for the human. The robot starts first by moving to one randomly chosen goal and the participant has to predict which one the robot is aiming at and moves the VR controller to the corresponding goal with same color when they feel confident about the target of the robot. After that, both the participant and robot move back to their starting positions and the procedure repeats. The participant is informed that this is a collaborative task, therefore they are expected to find a balance between making a correct prediction or being fast and reacting early. For example, making many wrong predictions results in failing the tasks,

whereas having long prediction time increases the total amount of time for both to finish their tasks. Both cases reduce the efficiency of the collaboration.

Each experiment consists of a habituation phase and an evaluation phase. The purpose of the habituation phase is to get the participants acquainted to the VR environment and the used equipment as well as familiarized to the robot motions and their own task. This habituation phase reduces the learning effect during the main evaluation phase. During the evaluation phase, the participants are asked to answer a questionnaire. The answers are scaled onto 5 different levels: *strongly disagree*, *disagree*, *neutral*, *agree*, *strongly agree*. There are 11 questions in total, that are classified into 5 categories:

- How does the participant feel about the smoothness of the trajectories?
- Does the participant feel safe when working with the robot?
- Are the robot trajectories predictable?
- How natural and comfortable the participant feel about the robot trajectories?
- How does the participant like and want to work with the robot again?

For all experiments, if not mentioned specifically, we use configurations and parameters described as follow:

- For DMP, we use three goal systems for the three Cartesian goal positions of the end-effector. These goal systems are first initialized with straight lines. The DMP has 5 equally spaced Gaussian radial basis functions and there are 5 samples per update for each goal. In the sampling phase, we add perturbations with the covariance size as 200 to the DMP parameters and run the policy for each sample. With each iteration, we let the variance factor for the perturbations decay as it helps reducing the search space for the parameters over time.
- For obstacle avoidance, we use a motion capture system to detect the position of the human (and the velocity), which are then used to compute the repulsive force. The maximum force $\boldsymbol{F}_{\max}$ is set to $300N$ and the obstacle threshold is $20cm$ around the end-effector of the robot.
- The weights of cost components used in the experiment are: $\lambda_{\text{ej}} = 1$, $\lambda_{\theta} = 2$, $\lambda_{\text{pred}} = 8$, $\lambda_{\text{task}} = 10$, $\lambda_{\text{dur}} = 1$, $\lambda_{\delta} = 3$. The weights of the human prediction time and accuracy costs are relatively higher than the others.

Over time, the policy of the robot is updated to adapt to the preferences of the human and produces more predictable movements to the human partner. The results of this adaptation are presented in the following section.

In order to convert the Cartesian trajectory produced by the DMP into joint positions, we use traditional inverse differential kinematics:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^{+}\dot{\boldsymbol{x}} \tag{3.24}$$

with $\boldsymbol{J}^{+}$ being the pseudo inverse of the Jacobian $\boldsymbol{J}$ of the end-effector ([Pen55]), $\boldsymbol{\theta} \in \mathbb{R}^7$ is joint configuration and $\boldsymbol{x} \in \mathbb{R}^3$ is Cartesian position. The pseudo inverse gives the least square approximation to the real inverse. In our case only the pseudo inverse is applicable, as we map three Cartesian values to seven joint positions, which makes the Jacobian not quadratic

and thus not regular. We constrain the covariance size of the DMPs to avoid generating trajectories out of the range of the robot. In addition, the joint configuration corresponding to the starting position is fixed for all trajectories. In this way, the elbow position of the robot resulting from joint redundancy does not change significantly during the experiment. Hence, the adaptation effect is mainly visible on the end-effector movement. The motion of the end-effector is formed based on the DMPs trajectories and the potential field force applied to it and it is the major factor for the human partner to differentiate between different robot motions.

3.5.2 Predictable Robot Motion for a specific Setup

Given a specific setup, which in our case comprises the goals of the human and the robot in addition to the robot mounting position (either in the same side or opposite side of the human), the predictable trajectories are obtained through the learning framework. We conduct experiments with different participants on different configurations to evaluate overall performance. To quantify the performance of our framework, we look at the following criteria:

- The total cost V and human prediction time cost V_{pred} (Section 3.3.4) for each update. V_{pred} is used to quantify the legibility of the robot motion while V shows the overall efficiency of the learning framework.
- The opinion of the subject about how legible robot motions are after each phase.
- The converged trajectory for each goal after learning w.r.t each subject.

The first two criteria will be discussed in Section 3.5.2.1 and 3.5.2.2 while the last one will be analyzed in Section 3.5.2.3.

3.5.2.1 Evaluation of the learning framework

Fifteen participants took part in this study. As mentioned, each experiment consists of a habituation phase and an evaluation phase. In the habituation phase, 30 trials are executed using invariable DMP trajectories. After its completion, the evaluation phase starts, which consists of 10 updates with 5 trials per update for each of the three goals, resulting in 150 trials in total. This number is comparable to [SGBL15, BGLS17]. To evaluate the participants' perception during the experiment, this phase is divided in three blocks with two breaks after the 4^{th} and 7^{th} update respectively, in which the participants are asked to fill a short questionnaire (see results in Fig. 3.3).

The prediction time and accuracy from all participants is collected using a motion capturing system after each trial to update the cost function and evaluate the framework over time. The human prediction time is calculated by measuring the time between the start of the robot's motion until the participant reaches their goal. Since each human being has a different inherent reaction speed, we normalize the measurement of the human prediction time of each participant by their responses on the first update, which is computed as the average of 15 values of the human prediction time.

Both total cost V and human prediction time cost V_{pred} for all subjects are presented in Fig. 3.4. For each update, there are 225 data points (15 trials per update for 15 participants), each data point represents the measurement of one single movement of the participants. The red line depicts the mean while the blue area illustrates the 95% confidence interval. As shown,

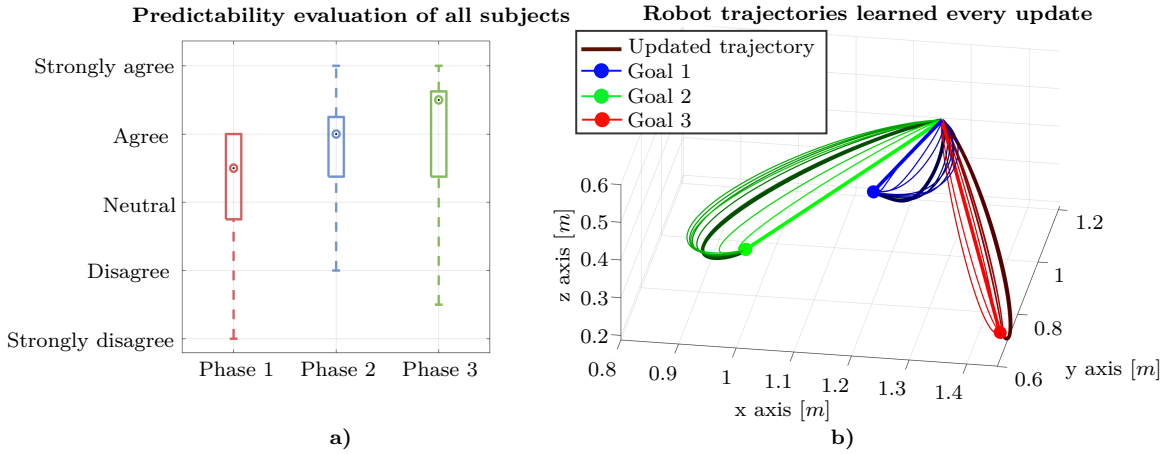


Figure 3.3: a) Predictability evaluation from all subjects for each phase. The evaluation of participants for the first 4 updates, then the next 3 updates and the last 3 ones are shown in phase 1, phase 2 and phase 3 respectively. The mark shows the median of each group. The box contains 50% of the middle half of all given answers thus representing the interquartile range of the data. The whiskers mark the most extreme answers. **b)** Robot trajectories learned for each goal every update. The first updates (straight lines) and last updates are marked thicker than the others.

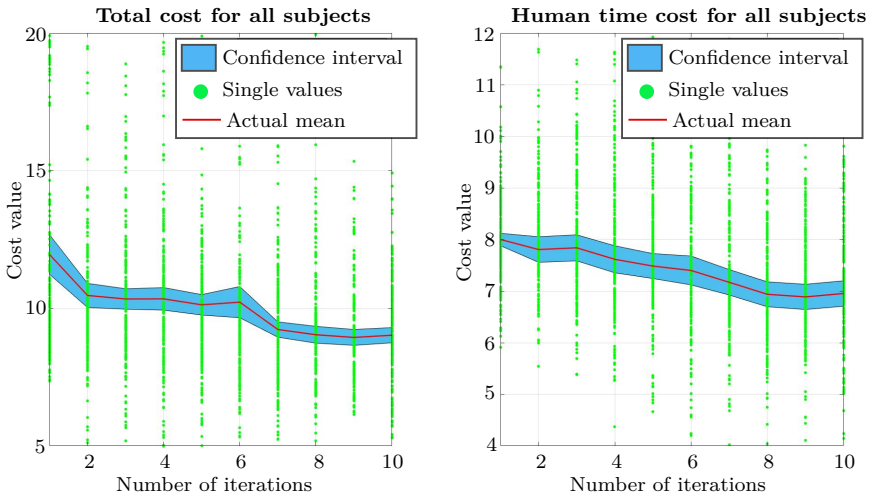


Figure 3.4: The mean and confidence interval of the total cost and human prediction time cost for all subjects.

both cost values decrease over time. The human prediction time cost V_{pred} drops around 13%, while the total cost V drop is around 24%. Comparing the data between the first and the last update, a pair-sampled t-test indicates that they are both significantly different from each other ($t = 7.142, p < .001$ for V and $t = 7.437, p < .001$ for V_{pred}). The decrease of human prediction time indicates that the subjects are able to predict and react faster to the robot motions while the reduction of total cost also implies that the subjects predict more accurately over time (accuracy cost has the highest weight).

The subjective legibility of robot trajectories is measured by the questionnaire during the breaks and after the last update. Here, we asked the participants' opinion on two statements:

the robot's intention was clear and *it was easy to predict which goal the robot is targeting.* We get the average of the two answers as the measurement of legibility aspect from the human perspective. The trajectories become more predictable as the median increases over time (Fig. 3.3**a**). An interesting result that can be observed here is that the interquartile range is reduced from phase 1 to phase 2, however it slightly increases from phase 2 to phase 3. This means the improvement from phase 2 to phase 3 is not very clear as the mean increase but the data spread is also larger. One reason for this is due to the trajectories of the robot start to get close to the converged one after a few updates and the updated trajectories of phase 2 and phase 3 are quite close together. An example of this behavior is shown in Fig. 3.3**b**, where the trajectories start as a straight line toward the goals and after a few updates, get close to the converged trajectories depicted as the bold and dark curves for each goal.

Overall, it can be concluded that, given a specific setup, human prediction time and subjective legibility can be improved through our framework and therefore can boost the efficiency of the collaboration between human and robot. However, the question arises here whether the learning effect of the participants plays a significant role in the improvement of the results, since the experiment is designed as a repetitive task. This will be discussed further in the next section.

3.5.2.2 Comparison with non-adaptive robot

In this section, we compare our method with a non-adaptive baseline. Even though we reduce the learning effect from the participants through the habituation phase, there is still probability that the human adapts to the motions of the robot over time. Therefore, the goal of this section is to investigate if the prediction of the human is improved due to the legible motions of the robot or because of human adaption. We design two experiments with the same environment setup, i.e. the tasks and the positioning of human and robot are the same. We use the counterbalanced ABBA design and define the following two groups:

- Group I: Subjects within this control group first interact with the non-adaptive robot, then with the adaptive robot subsequently.
- Group II: Subjects within this control group first interact with the adaptive robot, then with the non-adaptive robot subsequently.

In the case of non-adaptive robot, we also use our framework, but the policies (the parameters of the DMP) will not be updated. Therefore, the non-adaptive robot will always follow a straight line from the start toward the goal in every motion. As there is no adaption from the robot, the results from the non-adaptive robot solely reflect the learning capability of the human over time. This configuration also guarantees that the trajectory of the robot is smooth based on the DMP formulation (Section 3.3.1) and the avoidance behavior is identical to the adaptive robot. The only difference between the two robots is the method to generate their motions which can be evaluated by comparing the results from the two experiments.

The experiments are then conducted on 14 new subjects, divided into 2 groups of 7 participants each. The procedure for each experiment is identical to the experiment described in Section 3.5.2.1.

The total cost and human prediction time for both cases, adaptive and non-adaptive robot, are shown in Fig. 3.5. The error bar represents the mean value and standard deviation for each update. For the adaptive robot, there is a clear tendency for decreasing in both total cost and human time cost over the course of iterative updates. On average, the total cost deceases

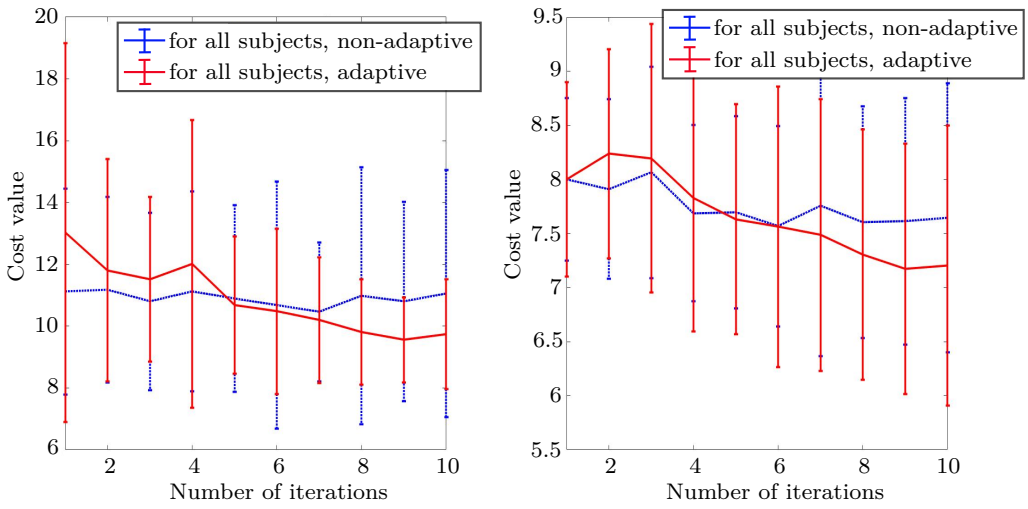


Figure 3.5: Comparison of the total cost and human prediction time between adaptive robot and non-adaptive robot

around 22% and human time cost decreases around 10%. In the case of non-adaptive robot, these values are 3.8% and 4.5%, respectively. It can also be seen that for the first few updates, the subjects collaborate better with the non-adaptive robot as both of the costs are lower. This is due to the fact that the adaptive robot uses a trial and error method to understand how the human perceives legibility by exploiting different motions. Motions that are harder to predict result in a higher cost, as shown in the slightly increasing in the human time cost on the second and third updates of the adaptive robot. But overtime, its motions become more predictable and easier for the subjects to predict compared to the non-adaptive robot, as indicated by the better performance in both cost values from the sixth update and after.

We also perform pair-sampled t-test to evaluate how significantly different is the performance between the adaptive and non-adaptive robot. On the first update, the performance between both robots is not significantly different ($t = -2.464, p > .001$ for the total cost V and $t = -0.266, p > .001$ for the human prediction time cost V_{pred}). In contrast, on the last update, the t-test results in $t = 4.139, p < .001$ for V and $t = 3.185, p < .001$ for V_{pred}, which indicates that the difference is significant. Overall, our conclusion drawn from this section is that the improvement in the human prediction time and the overall performance is mainly from the legible behavior of the robot. The learning effect from the human partner, while also reducing the human time and cost, does not have a significant contribution within our framework.

3.5.2.3 Predictable Trajectory Evaluation

To analyze the converged trajectories from the policy improvement framework, we first pick three different configurations: 3 goals in a horizontal line, 3 goals in a vertical line and 3 goals in a diagonal line. These configurations are illustrated in Fig. 3.6. Combined with two different mounting positions of the robot (same or opposite to the human), we have 6 cases in total. The experiments are conducted with several participants for each case. In Fig. 3.7 we representatively show 3 converged robot trajectories for each goal configuration.

For the horizontal configuration, **a** and **b** are with the robot on the same side and **c** is with

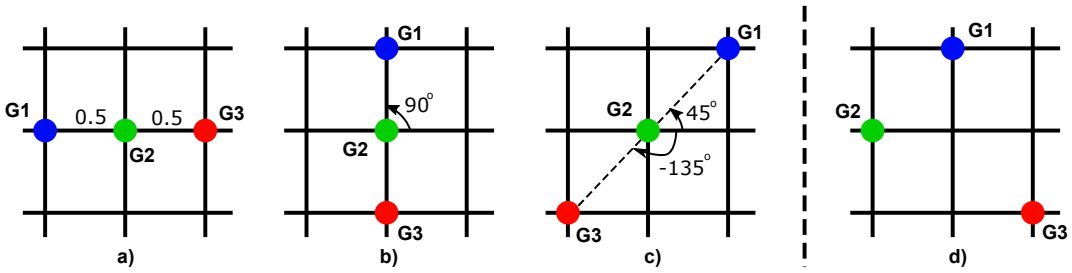


Figure 3.6: Different configurations of the robot goals: **a**, **b**, **c** are used for training, **d** is used for testing the task generalization approach.

the robot on the opposite side. The robot tends to bend more on the left or right for the blue or red goal, respectively, while for the green goal, the robot tries to keep the trajectory in the middle with a small variance, i.e. the green line diverges slightly to the left side in **a**. Another variance is the length of the trajectories, e,g. the red line is the shortest in **a** and longest in **b**. All trajectories tend to go downward for all three results.

The vertical configuration is one of the most interesting case as the trajectories converge quite differently. For example, the green line curves to the left in **d** and **e** but keeps in the middle-left in **f**. The red line is the only one bending to the left in all three results. However, we observe the same pattern for all three results. For each case, one trajectory bends to the left side, one to the right side and one stays in the middle. This creates a divergence between the three trajectories and makes them easier to predict. The difference in trajectory shape toward each goal comes from the random sampling of DMPs during the rollout phase. For example, if there are more rollouts for the green goal to the left side and being predicted correctly by the human, these rollouts will be rewarded more and push the next update to the left. Another reason is the personal preference of each participant, i.e. for the blue goal, it is easier for one participant to predict if it bends to the left side, but for another the right side is favorable. Hence, these trajectories are rewarded differently.

For the diagonal configuration, we observe similar behaviors as in the horizontal one. The green line stays in the middle while the two others diverge to the corresponding directions. Also in this configuration, the distance between two goals is larger than previous cases, therefore it is easier for the human to predict in this configuration. The blue line is one example as it tends to go straight toward the goal in **h**.

For all configurations, we observed slightly different trajectories w.r.t the mounting position of the robot. It seems the perspective affects the shape, but it's not always significant. This is probably because from the human point of view, the shape of trajectories does not change a lot, therefore it does not affect the predictability too much.

In summary, there are differences between trajectories w.r.t different subjects and configurations i.e. length, bending angle, etc. However, we also observed several similarities and patterns in the robot trajectories that make them become more predictable to the human. This motivates us to learn these patterns such that they can be generalized to other cases.

3.5.3 Task Generalization Evaluation

As learning a policy for each task and each configuration requires considerable amount of time, it is preferable to take advantage of the knowledge of the prior polices as it already encodes some preference of human perception. In this section, we evaluate our task generalization presented in Section 3.4. To generalize the policy for task $\boldsymbol{g}_m$ in a new set $\widetilde{\mathcal{T}}$, we have to

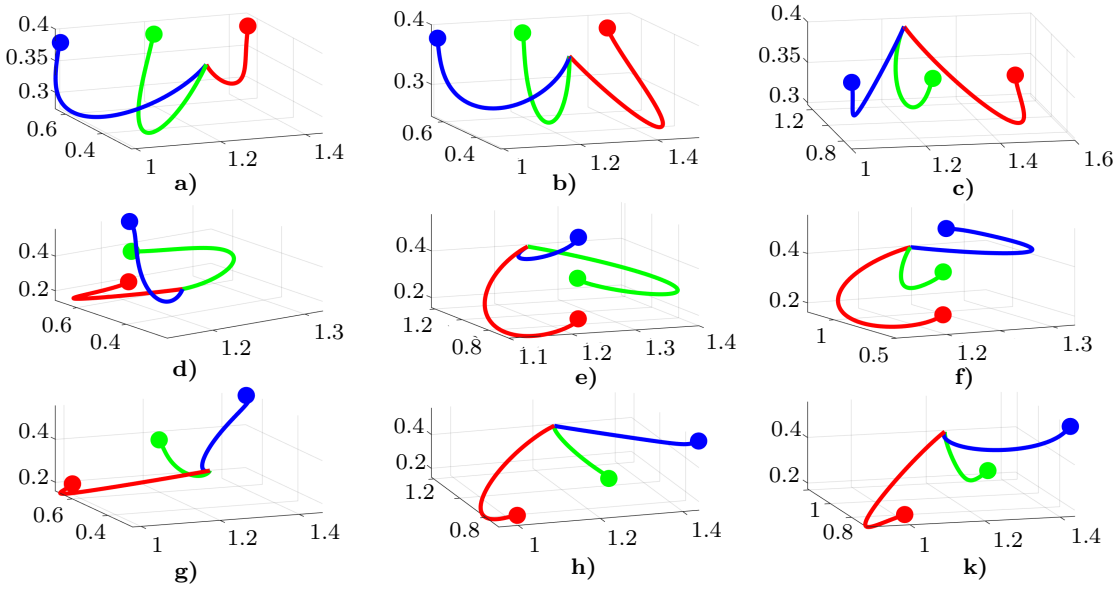


Figure 3.7: Converged trajectories from different subjects and configurations, *top*: horizontal, *middle*: vertical, *bottom*: diagonal configurations.

find a set of features $\boldsymbol{\chi}_{\widetilde{\mathcal{T}}}(\boldsymbol{g}_m)$ (see Section 3.4). From our observation and from the results in Section 3.5.2.3, we identified some critical features that a predictable trajectory depends on:

- The relative distances from the target goal $\boldsymbol{g}_m$ to other goals in $\widetilde{\mathcal{T}}$.
- The angles between the target goal $\boldsymbol{g}_m$ to other goals in $\widetilde{\mathcal{T}}$ w.r.t the horizontal line.
- The relative angle between the human and the robot.

Without loss of generality, we illustrate our idea for the case $\widetilde{\mathcal{T}}$ consisting of 3 goals as depicted in Fig. 3.6. The workspace of the robot is divided into a 3×3 lattice where robot goals can be located in 9 different positions. For the sake of simplicity, the height of the workspace is normalized as 1. Fig. 3.6 depicts some possible configurations and how $\boldsymbol{\chi}_{\widetilde{\mathcal{T}}}(\boldsymbol{g}_m)$ is calculated. For example, for **G1** in **a**, the relative distances to **G2** and **G3** are 0.5 and 1 respectively, the angles to **G2** and **G3** are both 0^o. For **G2** in **b**, the angles are 90^o and -90^o while for the same **G2** in **c**, these values are 45^o and -135^o. The relative angles between the human and robot is set 0^o if the robot in mounted on the same side with the human and 180^o if the robot is mounted on the opposite side of the human. Within the scope of this work, we only investigate these two mounted positions of the robot, but it can be extended to other cases, e.g. the robot is positioned on one side of the table such that the perspectives of the human and robot are orthogonal.

To verify our task generalization approach, three configurations **a**, **b** and **c** in Fig. 3.6 combined with two different robot positions are used for the training phase (6 different cases in total). The training phase consists of 18 subjects, equally distributed for all cases. For each experiment, we obtain the policy w.r.t each subject for each case. The weights of the converged trajectories are extracted to construct a regression model. Then, we use a new setup depicted in Fig. 3.6**d** with the robot mounted on the same side with the human as a testing sample. Using the corresponding features for the new setup as the input, we initialize the DMP with the output of the regression function in Eq. (3.23).

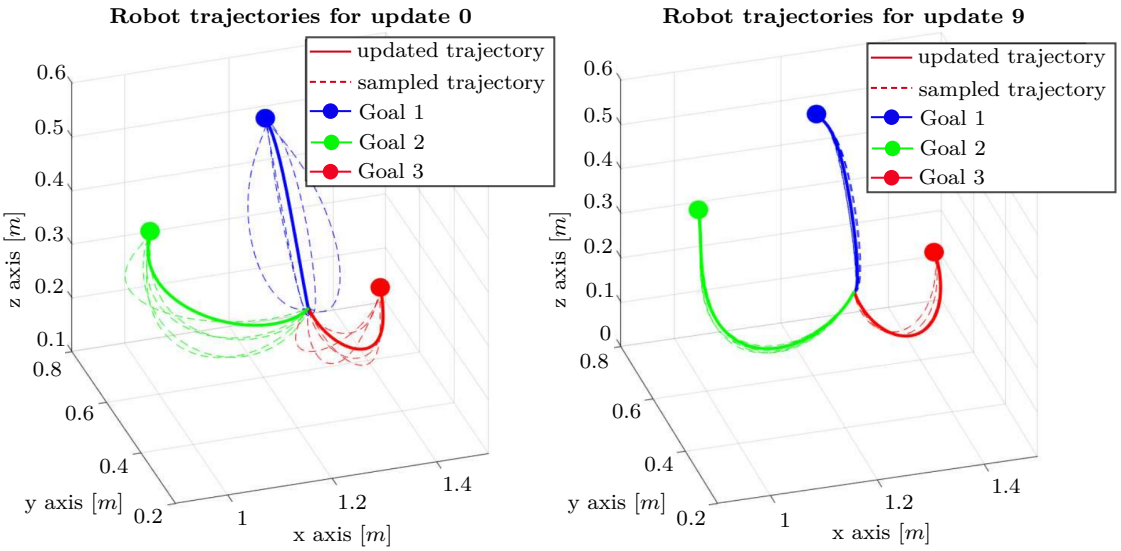


Figure 3.8: Robot trajectories in the task generalization experiment. The trajectories were initialized by the weights generated from the regression model.

The robot trajectories in the first and final update are depicted in Fig. 3.8. The trajectories are initialized as curves toward the three goals in the first update instead of straight lines in the non-trained case. For **G1**, the curve bends upward while for **G2** and **G3**, the curves deviate downward, more to the left and right from the human point of view respectively. These behaviors match the expectation that we observed in Section 3.5.2.3. During the updates, the robot continues exploring new motions around the initial ones. The covariance size of the DMP perturbation is set to half of the value of the non-trained case so that the rollout trajectories are sampled in a smaller area. The converged trajectories for each goal are shown in the final update in Fig. 3.8. Compared to the first update, the shape of the trajectories does not change a lot, which indicates that the learning algorithm stays close to the minimum from the beginning.

Next, we analyze the outcome of the total cost and the human prediction time. Our goal here is to compare the performance of the learning method to the non-trained case. Therefore, we establish two groups with 6 new participants each:

- Group A: Subjects within this control group interact with the untrained robot on a specific experimental setup different from the ones used for training the data.
- Group B: Subjects interact with the robot, whose trajectories are initialized by the regression model. The experimental setup is identical to the one of Group A.

The experiment procedure is the same as described in Section 3.5.2.1. The human prediction time cost of each subject is also normalized for cross comparison. The means and standard deviations of the total cost and human prediction time cost from both groups are plotted together for comparison (Fig. 3.9). A clear improvement of the trained robot can be observed directly from the result as both the total cost and the human prediction time cost are lower than the untrained robot. In addition, the cost values of the trained robot start decreasing from the start while in the case of untrained robot, they start increasing at first then decrease due to high exploration in the beginning. As the experiment is designed exactly

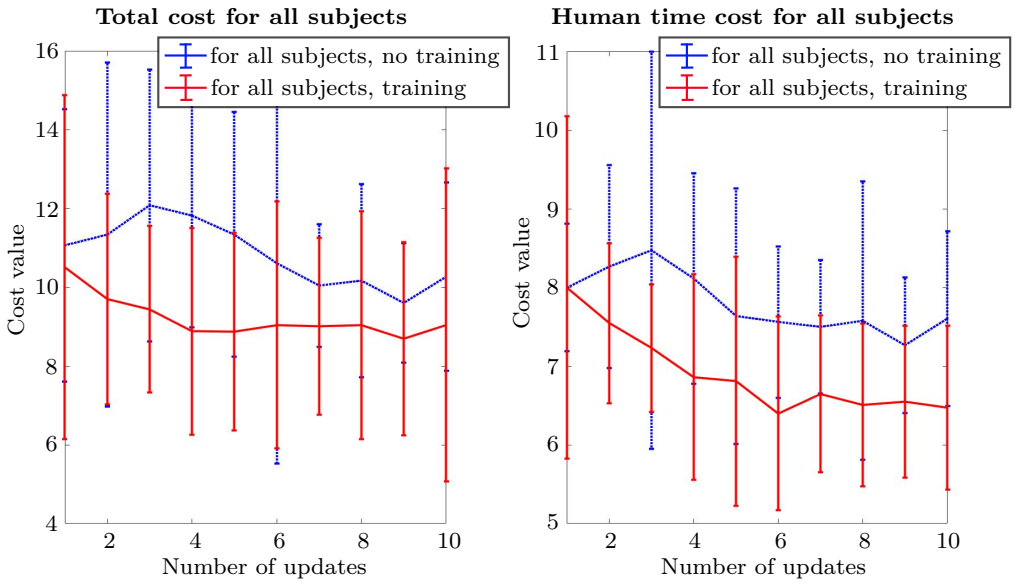


Figure 3.9: Cost plots that show the difference between the control group that interacted with the untrained robot and the results for the interaction with the trained robot

the same between both groups, the improvement of the trained robot comes from the initialized trajectories derived from our task generation approach. Instead of exploring the whole area, the trained robot only needs to search around the given trajectories, which inherit the properties of legibility from training data. As a result, the human predicts easier and faster over time, i.e. the human time cost drops substantially 20% in the case of the trained robot compare to 10% of the untrained robot after 10 updates.

As a conclusion, the task generalization approach that we proposed increases the efficiency of the learning framework. Starting from an initial trajectory generated from the approach, the robot trajectory converges quickly to the predictable one, which is also close to the initial trajectory. This helps to reduce the number of updates and the number of sampled trajectories per update, which in turn reduces the amount of time needed for training.

3.5.4 Experimental Results on a Real Robot

As demonstrated in the previous sections, our approach proves effective in learning predictable robot motions through interaction in a VR environment. To further advance our framework, we take a significant stride by implementing it on a real robot. While conducting experiments in VR allows us to evaluate our hypotheses across various setups and configurations without being limited by system constraints or safety concerns, it becomes challenging to assess the safety aspect from the human perspective due to the absence of real collision possibilities. Hence, in this section, we provide additional evaluation of the safety aspect.

In order to evaluate safety, we design an experiment as depicted in Fig. 3.10, with the robot positioned opposite the human. In this experiment, we utilize the *KUKA LWR 4+* robot, which possesses 7 DOFs. We employ the same inverse kinematics methodology introduced in Section 3.5.1 to convert Cartesian positions into joint configurations for the robot. The trajectories generated by our framework are then transmitted to the robot using ROS (Robot Operating System) at a frequency of $100Hz$. The *KUKA* robot internally employs joint

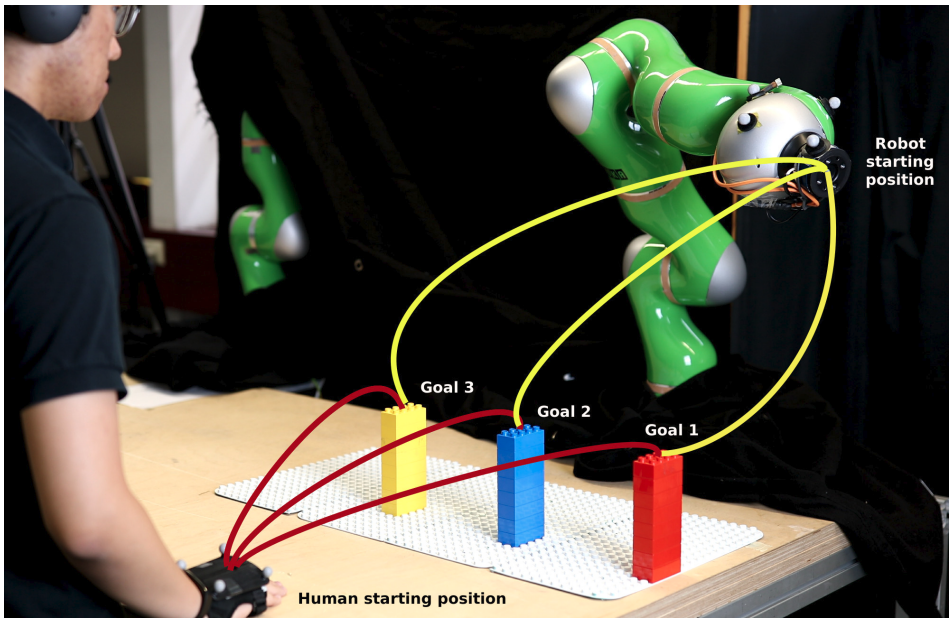


Figure 3.10: Real experiment setup on a KUKA LWR 4+ robot

position control to execute the provided trajectories.

Slightly deviating from the VR setup, in this real robot experiment, we use identical goals for both the human and the robot. The goals are represented by three *LEGO* blocks, distinguished by their colors (red, blue, and yellow). Consequently, the human is required to enter the workspace of the robot to reach these goals, thereby introducing a potential collision risk with every movement. To ensure safety during the experiment, both the human hand and the end-effector of the robot are equipped with passive retroreflective markers, which are tracked by a Qualisys tracking system [Qua19]. This tracking information is utilized by the robot to avoid collision with the human, thereby guaranteeing a safe working environment.

The experiment procedure follows the same structure as described in the previous sections, comprising a habituation phase and three main blocks in the evaluation phase. The first block consists of 4 updates, while the second and third blocks each consist of 3 updates. Following each block, a short break is provided for the subjects to complete a questionnaire. The questionnaire includes the same legibility-related questions as presented in Section 3.5.2.1, aiming to assess the perceived legibility of the robot motions. In addition to the existing questions, new inquiries are included to evaluate the safety aspect and the participants' comfort.

Regarding safety, the participants are asked to provide their opinions on three statements: *The robot is responsive to my movement*, *The robot does not hit me while moving*, and *I feel safe working with the robot.* The first two statements specifically focus on the avoidance behavior of the robot, as it is a crucial feature for ensuring the safety of the human partner. The third statement directly gauges the participants' overall perception of safety when collaborating with the robot. The average of the responses to these three statements is used as the measurement for evaluating the safety aspect. Similarly, for assessing comfort, two statements are included: *The motion of the robot is natural to me* and *I feel comfortable working with the robot.* These statements aim to evaluate whether our framework also provides a comfortable working experience for the human partner.

The entire experiment duration is approximately 30 minutes. Throughout the experiment,

participants are instructed to wear headphones playing concentration music to minimize potential distractions from the surrounding environment.

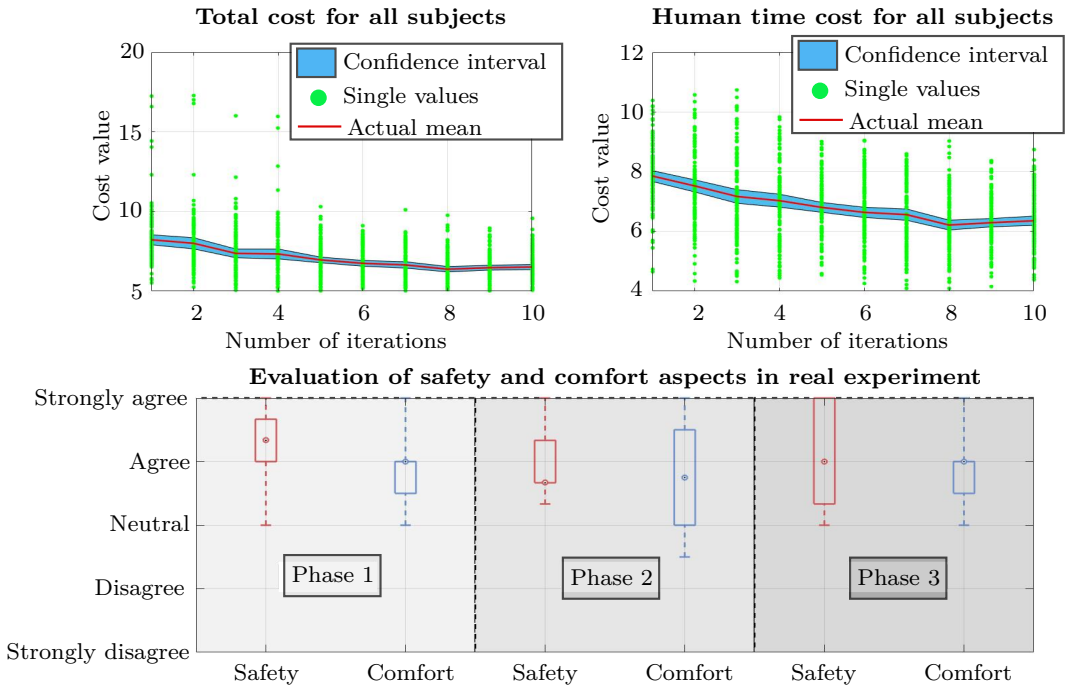


Figure 3.11: Results of the real experiment on a *KUKA LWR 4+* robot. The top side of the figure shows the total cost and human prediction time cost while the bottom side shows the human evaluation in the safety and comfort aspects when working with the real robot

We gathered data from a group of 10 new participants who had no prior knowledge or experience with the VR experiments. Consequently, the results solely reflect the performance of the real robot, offering insights into its behavior. We observe similar patterns in the total cost V and the human prediction time cost V_{pred} as in the VR experiments. Both cost values exhibit a gradual decrease over time (Fig. 3.11). Specifically, the total cost V decreases by approximately 20%, while the human prediction time cost V_{pred} sees a reduction of around 19%. These improvements in cost values indicate that the trajectories of the KUKA robot became increasingly predictable over time.

The lower section of Fig. 3.11 presents the box-plots illustrating the evaluation of safety and comfort aspects. Concerning safety, the data distribution shows almost no negative responses from the participants. The responses range from *neutral* to *strongly agree* in all three phases of the experiment. The boxes, representing 50% of the answers, mainly spread around the *agree* level in phase 1 and phase 2. In phase 3, there is a larger variation, with the box extending from above *neutral* to *strongly agree*. Overall, the data demonstrates positive feedback, indicating that the participants were confident that the robot would not collide with them during its movements, fostering a sense of safety when working with the robot. During the experiment, we even observed some participants expressing interest in the behavior of the robot by repeatedly interacting with it after completing their tasks, deliberately moving their hand toward the robot to observe its reactions.

Regarding comfort, the participants provide predominantly positive feedback, with most answers surpassing the *neutral* level. Only in phase 2, one of the whiskers extends below

the *neutral* level. However, this is an isolated case, representing a single participant out of the 10 subjects, highlighting individual differences in personal preferences. Overall, we can conclude that our framework is capable of providing a safe and comfortable environment for human-robot interaction during the learning process.

3.6 Discussion and conclusion

3.6.1 Discussion

Our learning framework represents a significant advancement in integrating learning and interaction, thereby transitioning from a passive "learning from observation" approach to an active "learning through interaction" paradigm. The findings presented in Sections 3.5.2.1 and 3.5.2.2 provide compelling evidence that our framework enables the generation of legible robot motions during human-robot interaction. Notably, the improvement observed in comparison to the non-adaptive baseline highlights the inherent adaptability of the robot motion over time, with the own adaptive capabilities of the robot playing a pivotal role, while the learning effect from human diminishes throughout the learning process. Furthermore, our investigation into task generalization has yielded promising preliminary results. By initially learning the policies of three sampled tasks and subsequently utilizing our approach to generate policies for new tasks, we have demonstrated the potential for successful policy transfer and task generalization. The results discussed in Section 3.5.3 validate our hypothesis that legibility can indeed be effectively transferred to similar tasks, reinforcing the generalizability of our framework through the application of our task generalization method. We also verified our framework in a real experiment setup, affirming its capacity to create a safe and secure environment for human-robot interaction. While the presented results substantiate the effectiveness of our framework, it is important to acknowledge that there are additional aspects warranting detailed consideration and further exploration.

In our study, we not only evaluate and verify various hypotheses, as detailed in Section 3.5, but we also recognize the importance of exploring additional case studies through further experimentation. One intriguing avenue for investigation lies in understanding how the predictable trajectories learned from our framework are influenced by the relative perspective of the human and robot. This particular case study stems from the recognition that the human partner is unlikely to remain stationary during interactions, often moving around the robot. Consequently, the trajectories of the robot are perceived differently from the human's vantage point. While our work has examined two distinct mounting positions for the robot and yielded preliminary results, it is imperative to explore additional positions to fully substantiate this proposition. Another compelling case study centers on examining the variations in perception among participants with different backgrounds. For instance, participants with a robotics background may exhibit distinct behaviors and reactions when collaborating with the robot, as compared to those without prior experience in robotics. By comparing the outcomes of the learning framework across these participant groups, we can gain deeper insights into the interplay between individual backgrounds and the effectiveness of the framework.

Task generalization is a concept that allows us to estimate the policy for a new task by leveraging the existing policies of previously trained tasks and considering the relationship between human perception, specifically in terms of predicting robot trajectories, and task specifications. By starting from a trajectory that is more predictable to the human for a new task, our approach enhances the convergence rate of the learning framework. We have demonstrated the effectiveness of this idea in a 3×3 lattice environment, implementing three tasks

per configuration, and showcasing the positive outcomes of our approach. The advantages of our method are noteworthy. Firstly, it does not necessitate precise task positions but instead relies on the relative positions between tasks, estimating the basis functions of the DMP. This characteristic enhances the flexibility and adaptability of our approach. Secondly, our method can be easily extended to an $n \times n$ case, accommodating a larger number of tasks per configuration without requiring extensive modifications. This scalability ensures its applicability to complex scenarios with numerous tasks. However, it is essential to acknowledge the limitations associated with the current formulation of our approach. Since the features that capture task differences are defined at the outset and remain unchanged during the learning phase, the range of new tasks that can be estimated using our approach is somewhat restricted. This constraint arises from the requirement that these new tasks must be described using the same features. For instance, in our work, all robot tasks involve reaching a goal on a vertical plane. Consequently, tasks with fundamentally different features or characteristics may not be effectively addressed using our current task generalization method.

The promising outcome of the task generalization method poses several intriguing questions that warrant further investigation. Firstly, we need to explore how to identify the relevant features and determine the influence of each feature on the trajectories of the robot. Although we have primarily relied on observations from a limited number of participants to identify critical features, a more extensive dataset is necessary to adequately validate and establish the significance of these features. Secondly, it is important to determine the optimal number of training cases required during the training phase of the task generalization approach. Finding the right balance is crucial, as too many training cases can significantly increase training time and diminish the efficiency of the approach. Conversely, too few training cases may lack the necessary variation to capture the interrelation between tasks effectively, potentially impacting the outcome of the generalization method.

Finally, the experiment conducted on the *KUKA LWR 4+* robot is our first step to bring our framework to real-world applications. The avoidance behavior of the robot is reliable such that the human feels safe and confident to cooperate with the robot. The implementation of avoidance behavior in the robot plays a crucial role in ensuring the safety and building trust between the human and the robot during their collaborative interactions. The reliability of the avoidance behavior is instrumental in fostering a sense of security and confidence in the human partner. It allows for a smooth and consistent behavior from the human, both in terms of prediction and hand movement coordination. It is worth emphasizing the significance of this avoidance behavior and its contribution to the overall success of the learning process. A seamless interaction between the human and the robot is essential for accurate measurement of human prediction time. For instance, in the event of a collision during the experiment, the human may feel uncomfortable and hesitant to proceed with subsequent movements. Such interruptions can potentially introduce inconsistencies and inaccuracies in the measurement of human prediction time, underscoring the importance of reliable avoidance behavior.

However, our setup with the *KUKA LWR 4+* robot does have certain limitations, primarily related to the working area of the robot. Due to joint limits, especially the elbow joint, as well as the chosen mounting position and configuration to avoid singularities, the workspace of the robot is relatively small. As a result, we were constrained to setting up only three goals with a distance of approximately $20cm$ between them. This limited workspace presents challenges in extending the framework to different tasks and evaluating the effectiveness of the task generalization method in a real-world setup. In order to overcome this limitation, alternative solutions can be explored. One possible approach is to change the mounting position of the robot, such as mounting it on the ceiling, which would provide a larger range of motion for

the elbow joint of the robot. Alternatively, employing a different robot with a larger working space could also address this constraint and enable the evaluation of the task generalization method in a more diverse set of tasks and scenarios.

3.6.2 Conclusion

In this work, we have developed a comprehensive framework that integrates human motion prediction with our learning framework to generate predictable robot motions that can adapt to human preferences and avoid collisions. We have successfully verified the effectiveness of our framework through extensive experiments in both VR and real-world scenarios. The results obtained from these experiments demonstrate the ability of robots to adapt their behavior based on human preferences while ensuring a safe and comfortable working environment. The humans involved in the experiments showed increased confidence and improved prediction capabilities, leading to faster and more accurate responses. Furthermore, our task generalization approach has proven to be successful, as evidenced by the improved performance of the robot when executing a new task using the learned policy. This confirms our hypothesis that the policies learned through our framework are transferable to similar tasks. This adaptability and transferability highlight the versatility and efficacy of our framework for generating predictable robot motion in both offline and online cases.

Building upon these advancements in trajectory generation and adaptation, the next crucial challenge is the control of the robot to effectively execute the generated motions and swiftly adapt to changes in the environment or task requirements. While our framework successfully generates trajectory profiles comprising position, velocity, and jerk information, the execution of these trajectories by the robot remains an open question that needs to be addressed. To execute the generated trajectories, a force/torque profile must be generated to serve as the control input for the robot. This aspect falls under the domain of motion control, which was previously discussed in Chapter 1.1. Motion control plays a critical role in achieving optimal robot performance, particularly for high degree-of-freedom articulated or humanoid robots that are commonly utilized in close proximity HRC scenarios. The motion control system must take into account various physical properties of the robot, including its mass, inertia, and dynamic characteristics. Additionally, it needs to consider optimization requirements such as energy consumption, smoothness of motion, and adherence to user-defined constraints. While the existing internal controllers of the robot may enable basic functionality, they often lack the necessary adaptability, flexibility, and optimality required for complex HRC tasks. Manual configuration of controller parameters for each experiment or scenario is time-consuming and inefficient. Furthermore, the use of high DOF robots with complex kinematic structures and physical models introduces additional challenges. These robots often require computationally intensive algorithms to generate optimal control solutions. Fast and efficient motion control algorithms are crucial to ensure optimal robot performance, enable swift adaptation to changing task requirements and environmental conditions, and maintain smooth and predictable trajectories that prioritize the safety and comfort of human partners.

In the upcoming chapter, we will delve deeper into the topic of motion control and explore novel techniques and strategies to address these challenges. By considering the intricate dynamics of the robots and the dynamic nature of the environment, our goal is to design fast and efficient control methods that can generate and execute precise trajectories on the robot while quickly adapting to changes in real-time, ultimately facilitating more productive and harmonious interactions between humans and robots in close proximity collaboration scenarios.

Part II

Motion Control for Robots in Dynamic Environments

Nomenclature of Part II

Scalars are written in plain lower case, vectors in bold-face lower case. Matrices are bold-face upper case, while plain upper case symbols refer to coordinate frames, mathematical spaces and sets. Subscript annotations are reserved for index notation of multi-dimensional variables, whereas superscript annotations are part of the variable specification.

Also note that we omit explicit listing of function parameters whenever it is clear from the context, to not unnecessarily clutter the notation. A list of the most frequently used variables in this work is given.

$\boldsymbol{\nu}$	Lagrange multiplier vector
$\boldsymbol{\Phi}$	set of terminal constraints
$\boldsymbol{\rho}$	co-state, Lagrange multiplier vector
$\boldsymbol{f}$	dynamic system
$\boldsymbol{R}$	matrix of influence functions
$\boldsymbol{S}, \boldsymbol{Q}$	positive definite matrix
$\boldsymbol{u}$	control input
$\boldsymbol{W}$	weighting matrix
$\boldsymbol{x}$	state
$\hat{V}$	optimal cost
λ	duration
$\mathbb{H}$	Hamiltonian
$\|\cdot\|$	2-norm of a vector
τ	sampling time
J, V	cost functional
l, L	performance cost
m_f	terminal cost
T_P	prediction horizon

4

Fast and Close to Optimal Receding Horizon Controller For Articulated Robots in Reaching Motions

This chapter was previously published in [HDWLW17, HDLW22]

One of the fundamental aspects of the discussed HRC scheme in Chapter 1 is the development of a controller for the robot that can effectively execute commands generated by high-level task allocation or trajectories generated by motion planning, while taking into account various factors and challenges specific to HRC environments. It is important to note that the controller operates at the force/torque level, thereby considering the dynamic behavior of the robot. In the context of HRC, especially with the presence of humans as collaborative partners, the controller must exhibit adaptability and real-time responsiveness to changing situations. This adaptability is crucial throughout the HRC framework and has been addressed to some extent in the motion planning frameworks presented in Chapter 2 and Chapter 3. For instance, the obstacle avoidance framework (discussed in Chapter 2) incorporates a simple waypoint trajectory generation method [RIS13], facilitating easy trajectory replanning, while a potential field approach enables rapid response to obstacles that may hinder the actions of the robot. Similarly, the motion generation framework in Chapter 3 leverages DMPs to generate versatile trajectories by adjusting the DMP goal. However, up until this point, our previous works have primarily focused on the kinematic level of the robot. The outputs of the motion planning frameworks presented in previous chapters are trajectories, represented as sets of waypoints with timestamps. Each waypoint contains kinematic information such as position, velocity, acceleration, jerk, etc., that the robot is expected to follow.

In essence, the motion planners in these works assume that the robot will execute the trajectories flawlessly and precisely adhere to the provided instructions. However, due to the inherent dynamic behavior of the robot, it is generally impossible for the robot to accurately track the given trajectories without a dedicated controller. Thus, the connection between the planning and execution components is currently lacking. The aim of this chapter is to address this issue by proposing a control approach that enables the robot to track the given trajectories while considering the diverse requirements of HRC environments. Given the challenges of HRC, the designed robot controller needs to fulfill the following properties

- The controller should enable the robot to precisely follow the given trajectories, while ensuring the overall stability of the system. Achieving precise tracking and stability is essential in order to fulfill the intended task objectives.

- Given the dynamic nature of HRC environments, the controller needs to be capable of recalculating and adapting in real time. The ability to dynamically adjust the control inputs and adapt to changes in goals or environmental conditions is crucial for maintaining

optimal performance and safety.

- In addition to trajectory tracking, it is desirable for the controller to incorporate obstacle avoidance capabilities. Although high-level planning techniques, as discussed in Chapter 3, may also address this issue, having a dedicated controller that accounts for obstacle avoidance adds an extra layer of safety and ensures smooth and secure collaboration between humans and robots within the same workspace.

- Beside tracking, different aspects need to be considered such as optimizing energy consumption, human comfort, etc. Hence, the designed controller requires the capability of coupling these terms.

- Finally, it is crucial for the controller to achieve fast computation times to meet the demands of real-life applications.

Designing a controller that satisfies all the aforementioned requirements is not a trivial task. One intuitive approach is to formulate an optimal control problem (OCP) that incorporates the desired properties through an objective function. By solving the OCP, an optimally calculated controller can be obtained. However, the computational effort required for most optimal control approaches is a significant challenge, limiting their usage to offline cases and static environments. Extensive research has been conducted to reduce the computation time of the OCP, with model predictive control (MPC) emerging as an effective approach that can solve the OCP in a shorter time frame. Despite its advantages, the computation time of MPC, particularly for complex systems, remains a limitation, restricting its application in online scenarios and HRC settings.

To address the computational challenges while considering all the essential properties, this work presents a novel approach for fast and optimal trajectory planning of nonlinear systems in dynamic environments. Given the start and end goals with an objective function, the problem is to find an optimal trajectory from start to end that minimizes the objective while taking into account the changes of the environment. The main challenge here is that the optimal control sequence needs to be computed in a limited amount of time and needs to be adapted on-the-fly. The proposed control method consists of two stages: an initial phase that utilizes a first-order gradient algorithm to compute an initial guess of the control sequence satisfying constraints but not yet optimal, and a sequential action control phase that optimizes the control sequence for the portion to be applied in the next iteration. This approach reduces the computational effort while maintaining optimality w.r.t. the objective function, making it suitable for online computation and dynamic environment scenarios. Moreover, the proposed controller is proven to be asymptotically stable under mild conditions. Simulated results demonstrate the efficacy of the controller in solving various tracking problems for different systems in the presence of dynamic obstacles. A comparison with related indirect optimal control approaches and sequential action control is conducted in terms of cost and computation time to evaluate the superiority of the proposed method.

4.1 Introduction

In recent years, trajectory planning and control in robotics and automation has garnered significant attention due to emerging demands in various applications. In addition to reaching a goal, it is crucial for controlled robots to respond to highly dynamic environments, such as

avoiding vehicles and pedestrians in autonomous driving scenarios or ensuring safety by avoiding collisions with human co-workers in close proximity collaborations. This necessitates the ability of the robot to adapt rapidly to changing situations. Furthermore, in addition to safety considerations, there is a growing need to optimize energy consumption and enhance human comfort. Optimal control methods offer a suitable approach as they enable the definition of these requirements as cost functions or constraints. However, the extensive computational effort required by these methods limits their use to offline trajectory computation, rendering them unable to effectively handle dynamic environments [Rao10].

Among the few optimal-based methods capable of addressing dynamic scenarios, MPC [RMD17] stands out. By utilizing the concept of a receding horizon, MPC computes a sub-optimal solution over a finite horizon, implementing only the first portion of the trajectory and waiting for the next measurement before proceeding. This approach offers two key advantages. First, MPC significantly reduces the computation time compared to purely optimal control approaches, as it solves the optimal control problem for a short horizon rather than the entire trajectory. Second, MPC can effectively respond to changes in the environment by incorporating them as additional costs and constraints at the beginning of each iteration. However, the trade-off is the optimality of the solution, as MPC provides a sub-optimal solution. Nevertheless, MPC still struggles with nonlinear systems due to their complexity, resulting in increased computation time. As a result, MPC is primarily applicable to simple or slow systems. For instance, in the case of ground vehicles, point-mass or kinematic models are commonly used, which overlook many of the physical behaviors of actual cars.

Sequential Action Control (SAC) offers a promising alternative to traditional optimal control methods for trajectory planning. Introduced in [AM16], SAC employs a receding horizon approach but focuses on finding the optimal control signal for a single interval rather than the entire horizon. By utilizing theoretical foundations from switching systems [EWA06], the author derives an analytical solution that allows for faster execution compared to numerical approaches. The potential of SAC for online applications is demonstrated through various examples presented in the work. However, SAC has limitations when it comes to incorporating constraints from the optimal control problem. The neglect of these constraints in favor of a more efficient analytical solution restricts the applicability of SAC to trajectory planning tasks that require the fulfillment of additional constraints, such as target or final constraints.

Building upon the effectiveness of SAC, the contribution of this work is a development of an optimal-based control algorithm called Target Constraint Sequential Action Control (TC-SAC) for the trajectory planning tasks. TC-SAC combines two steps to handle target constraints effectively while maintaining computational efficiency. In the first step, a first-order gradient approach is employed to address the target constraints and generate an initial guess for the trajectory planning problem. Then, SAC is utilized to further refine the initial guess w.r.t. the cost function. A notable contribution of this work is the extension of the original SAC algorithm to derive a new analytical solution that ensures target constraints are not violated. By integrating the first-order gradient approach with SAC, TC-SAC achieves a balance between constraint fulfillment and computational speed. Comparative evaluations between TC-SAC, SAC, and indirect optimal control methods are conducted to highlight the advantages of the proposed method. Furthermore, we demonstrate the versatility of TC-SAC by applying it to different systems and applications with minimal modifications required.

The remaining sections of this work are organized as follows: Section 4.2 provides an overview of related works in the field, while Section 4.3 formulates the general problem that serves as the foundation for this research. In Section 4.4, we present the fundamental concept of TC-SAC along with a comprehensive mathematical formulation of the algorithm. Subse-

quently, Section 4.5 presents the simulation results obtained using TC-SAC, highlighting the trajectory tracking performance and conducting comparative analyses. The stability proof of TC-SAC is discussed in Section 4.6, providing a rigorous assessment of the stability properties of the algorithm. Finally, Section 4.7 concludes our work with a thorough discussion and an outlook on future directions for research and development in this field.

4.2 Related Work

Optimal control approaches play a crucial role in trajectory planning and control, enabling robots and autonomous systems to generate trajectories that optimize certain performance criteria. By formulating the problem as an optimization task, optimal control methods aim to find the control inputs that minimize a given cost function while satisfying system dynamics and constraints. These approaches provide a systematic framework to handle complex system behaviors, and address various objectives such as energy efficiency, smoothness, and obstacle avoidance. With advancements in computational techniques and algorithm design, optimal control has become a prominent approach in achieving precise and efficient trajectory planning and control for a wide range of applications in robotics, automation, and autonomous systems.

There are two main categories of optimal control methods: indirect methods [VSB92] and direct methods [DBDW07]. Indirect methods focus on deriving a boundary value problem (BVP) in ordinary differential equations (ODE) by considering the necessary conditions of optimality for the infinite OCP. On the other hand, direct methods transform the infinite OCP into a finite nonlinear programming problem, which can then be solved numerically. However, due to the computational complexity associated with these methods, their online applicability is limited. To address this challenge, modifications have been proposed to perform offline pre-computations and reduce online computation time. For instance, some studies [LNTC$^+$11] compute a finite number of global optimal solutions offline and generalize them using techniques such as Support Vector Machines or Gaussian process regression, which are then used as training data for online adaptation. Additionally, machine learning and motion primitives [WTLL15, ALB16] have been employed to learn precomputed optimal trajectories. However, these approaches have limitations in dynamic environments where an infinite number of scenarios can arise, making it difficult to cover all possibilities with a finite number of precomputed solutions. Dynamic Motion Primitives (DMPs) [WHHAS13] have emerged as an alternative, allowing online adaptation of DMP parameters to respond to dynamic environments. However, in this method, the optimality is lost due to the deformation process and therefore the trajectory is not optimal.

Recently, MPC [All05] has garnered considerable attention and found applications in various domains [QB00, VZQD16, OUT16, MNK$^+$13, LSZL14]. The popularity of MPC can be attributed to the development of efficient numerical toolboxes and the availability of powerful CPUs that enable rapid solution of optimal control problems. MPC operates as a feedback optimal control framework by solving an optimal control problem over a finite receding horizon. The control input for the first interval is applied until new state measurements are obtained, at which point the horizon is shifted forward and the process is repeated. The key advantages of MPC include its computational efficiency compared to traditional optimal control approaches [VSB92] and its ability to account for dynamic environments. Consequently, MPC has gained significant attraction in the past decade, with notable advancements in algorithms and software implementations that substantially reduce computation time. Some approaches, such as those presented in [Oht04] and [FE16], employ limited iterations to approximate the solution,

leading to reduced computation time at the expense of suboptimal solutions. Additionally, the ACADO toolkit [HFD11] has emerged as a robust numerical solver for nonlinear MPC, capable of handling a wide range of applications and problems, as demonstrated in various research studies [KAAS15, ZHFD18]. Despite these developments, the computation time of MPC remains significant, particularly for nonlinear and complex models like articulated robots or car-like systems. As a result, most existing works focus on simple or slow systems in static environments to achieve computational feasibility.

SAC, introduced in [AM16], is a model-based algorithm that computes a sub-optimal control signal for nonlinear systems. Unlike NMPC, SAC derives a closed-form expression for a single interval of the control signal, which is applied for a short duration. This analytical solution, obtained without the need for numerical solvers, makes SAC faster than NMPC and other optimal control approaches, positioning it as a promising candidate for online applications. However, the analytical solution of SAC is only feasible when there are no constraints in the control problem. SAC employs the mode insert gradient method [EWA06], which measures the first-order sensitivity of the cost function with respect to the control signal applied for a short duration. An auxiliary cost function is cleverly formulated using this sensitivity, enabling the derivation of the analytical solution. Nevertheless, this approach inherently neglects the consideration of constraints, rendering SAC unsuitable for tasks that require hard constraints such as tracking tasks.

Motivated by SAC, this work proposes the Target Constrained Sequential Action Control (TC-SAC) method, which not only computes an optimal solution quickly but also handles target constraints in trajectory planning tasks. TC-SAC presents a promising solution for real-time optimal control in dynamic environments. Furthermore, we extend the original SAC algorithm in this work to address scenarios where target constraints may be violated, broadening its applicability to a wider range of applications. We perform comprehensive comparisons to demonstrate the effectiveness of the proposed TC-SAC method. Additionally, we showcase the capability of TC-SAC to handle dynamic environments, specifically addressing the challenge of avoiding moving obstacles. The stability analysis of the proposed method is also discussed in Section 4.6.

4.3 Problem Formulation

In general, trajectory tracking task can be formulated as an OCP in the receding horizon where the terminal constraints is used to for trajectory generation task. For the OCP, a dynamic system with n states and m control inputs is considered and described by a set of ordinary differential equations

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \;\; \mathbf{x}(t_0) = \mathbf{x}_0 \tag{4.1}$$

with $\mathbf{f} : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^n$ being nonlinear in state $\mathbf{x} \in \mathbb{R}^n$ and control input $\mathbf{u} \in \mathbb{R}^m$. The initial state of the system is denoted by $\mathbf{x}_0$ and the target configuration $\mathbf{x}_\mathrm{d}(t)$ denotes the final goal. The cost functional

$$J_1 = m_\mathrm{f}(\mathbf{x}(t_\mathrm{f})) + \int_{t_0}^{t_\mathrm{f}} l_1(t, \mathbf{x}(t), \mathbf{u}(t))dt \tag{4.2}$$

with the performance cost $l_1(\cdot)$ and the terminal cost $m_\mathrm{f}(\cdot)$ being used to measure the performance in the horizon $[t_0, t_\mathrm{f}]$. The optimal control problem over a receding prediction horizon

T_P is given by

$$\min_{\mathbf{u}} \; J_1 = m_\mathrm{f}(\mathbf{x}(t_\mathrm{f})) + \int_{t_0}^{t_\mathrm{f}} l_1 dt \tag{4.3a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \; \mathbf{x}(t_0) = \mathbf{x}_0, \; \mathbf{x} \in \mathbb{R}^n \tag{4.3b}$$

$$\boldsymbol{\Phi} = \begin{pmatrix} x_1(t_\mathrm{f}) - x_{\mathrm{d},1}(t_\mathrm{f}) \\ \vdots \\ x_q(t_\mathrm{f}) - x_{\mathrm{d},q}(t_\mathrm{f}) \end{pmatrix} = 0, \; q \leq n, \tag{4.3c}$$

where $t_0 = t_\mathrm{cur}$ denotes the current time, $t_\mathrm{f} = t_\mathrm{cur} + T_\mathrm{P}$ is the final time at the end of the prediction horizon and $(x_{\mathrm{d},1}(t_\mathrm{f}), ..., x_{\mathrm{d},q}(t_\mathrm{f}))$, defines q constrained states at the end of the prediction horizon t_f. $\boldsymbol{\Phi}$ is then the set of terminal constraints at t_f.

In this paper, the dynamic system in (4.1) is assumed to be given in control-affine form, i.e.

$$\dot{\mathbf{x}} = \mathbf{f}\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) = \mathbf{g}\left(t, \mathbf{x}(t)\right) + \mathbf{h}\left(t, \mathbf{x}(t)\right) \mathbf{u}(t) \tag{4.4}$$

with $\mathbf{f}$ being non-linear with respect to state $\mathbf{x}$ and linear in control input $\mathbf{u}$.

4.4 Target Constrained Sequential Action Control

This section outlines the proposed approach called Target Constrained Sequential Action Control (TC-SAC) [HDWLW17] for trajectory planning tasks. The idea of TC-SAC is utilizing the advantage of SAC in term of fast computation time and extending the original method with an additional controller to tackle constraints. The overall structure of TC-SAC is given in Fig. 4.1.

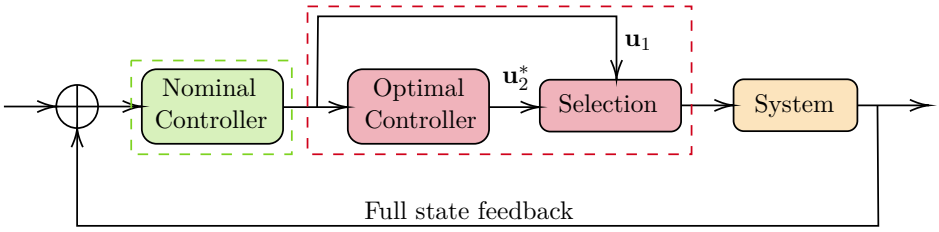


Figure 4.1: Overview of the controller scheme. First, a nominal control input is computed with the first order gradient algorithm (green box). After this, the first portion of the control input is updated with SAC (red box)

The proposed method consists of two parts:

1. A nominal controller based on first-order gradient algorithm (FOGA) [BHS79]
2. An optimal controller based on Sequential Action Control (SAC)

In TC-SAC, an initial guess is first computed by an indirect optimal control method, which is FOGA in our case in order to consider terminal constraints. Due to time limitation, FOGA is only run for one iteration to obtain the nominal control $\mathbf{u}_1$. Obviously, with one iteration, $\mathbf{u}_1$ is not yet close to the optimal solution. The first interval of $\mathbf{u}_1$ is then improved by SAC since it will be applied on the system in the next iteration. As mentioned, SAC utilizes the concept of mode insert gradient [EWA06] and a proper selection of an additional auxiliary cost function to derive an analytical solution for the optimal control $\mathbf{u}_2^*$, which improves the

performance over $\mathbf{u}_1$. Note that $\mathbf{u}_2^*$ differs from $\mathbf{u}_1$ only by the first portion while the rest remains the same. Then in the Selection step (see Fig. 4.1), $\mathbf{u}_2^*$ is compared to $\mathbf{u}_1$ in term of performance and terminal constraint costs. If the comparison shows the improvement of $\mathbf{u}_2^*$, then it is applied to the system for the next iteration.

Looking in depth into the difference between TC-SAC and SAC, our method has a better choice of the nominal controller $\mathbf{u}_1$. To be precise, the original SAC only computes one interval of the control signal $\mathbf{u}_2^*$ per iteration while assuming $\mathbf{u}_1 \equiv 0$. In the case of TC-SAC instead, $\mathbf{u}_1$ is updated every iteration. Therefore, TC-SAC always improve the optimality of the control signal over time while SAC does not. The update of $\mathbf{u}_1$ also plays a crucial role for TC-SAC to incorporate target constraints for trajectory tracking tasks and is part of the stability proof that will be discussed later in Section 4.6. Clearly this update requires more computational effort than SAC, but it is negligible compare to the benefits that it provides. Next, both parts of our approach will be outlined and explained in detail.

4.4.1 First-order gradient algorithm (FOGA)

The first part of the controller serves the purpose of incorporating constraints i.e. target constraints into the OCP which are crucial for trajectory generation tasks. Since the theoretical background of SAC uses the co-state equation from the Pontryagin's principle[Pon87] as part of the calculation (see Section 4.4.2 for more detail), FOGA is selected as the solver here to utilize this equation to reduce the amount of steps needed for implementation. FOGA then solves problem (4.3) for one iteration to find an initial guess. Since this is an optimal control problem with equality constraints at the final state, the idea from [BHS79] is used. First, the dynamic constraint is adjoined to the performance equation l_1 by introducing time-varying Lagrange multiplier vector $\boldsymbol{\rho}$, whose elements are called the co-states of the system. This constructs the Hamiltonian $\mathbb{H} \in \mathbb{R}^1$ defined for all $t \in [t_0, t_\mathrm{f}]$:

$$\mathbb{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\rho}, t) = \boldsymbol{\rho}^\mathsf{T} \mathbf{f}(\mathbf{x}, \mathbf{u}) + l_1(\mathbf{x}, \mathbf{u}) \tag{4.5}$$

Follow the Pontryagin's maximum principle[Pon87], the co-state equation

$$\dot{\boldsymbol{\rho}} = -\left(\frac{\partial \mathbb{H}}{\partial \mathbf{x}}\right)^\mathsf{T} = -\left(\frac{\partial l_1}{\partial \mathbf{x}}\right)^\mathsf{T} - \left(\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}}\right)^\mathsf{T} \boldsymbol{\rho} \tag{4.6}$$

must be satisfied. Solving (4.6) requires a terminal condition, which is usually chosen as $\boldsymbol{\rho}(t_\mathrm{f}) = \left(\frac{\partial m_\mathrm{f}}{\partial \mathbf{x}}(t_\mathrm{f})\right)^\mathsf{T}$ if the state $\mathbf{x}$ is not fixed at t_f. In our case, this terminal condition is slightly modified to consider the terminal constraints in (4.3c):

$$\boldsymbol{\rho}_i(t_\mathrm{f}) = \begin{cases} 0, & i = 1, \cdots, q, \\ \left(\frac{\partial m_\mathrm{f}}{\partial \mathbf{x}_i}\right)^\mathsf{T}_{t=t_\mathrm{f}}, & i = q+1, \cdots, n, \end{cases} \tag{4.7}$$

with n and q as defined above. Next, the matrix of influence functions $\mathbf{R} \in \mathbb{R}^{n \times q}$ is introduced

$$\dot{\mathbf{R}} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^\mathsf{T} \mathbf{R}, \tag{4.8}$$

where

$$\mathbf{R}_{ij}(t_\mathrm{f}) = \begin{cases} 1, & i = j, \quad i = 1, \cdots, n, \\ 0, & i \neq j, \quad j = 1, \cdots, q. \end{cases} \tag{4.9}$$

By defining the matrix $\mathbf{R}$, it is able to predict how changes in the control input, $\delta\mathbf{u}(t)$, affect the cost function J_1 and the q terminal constraints in $\boldsymbol{\Phi}$ by the following, check [BHS79] for more details

$$\delta J_1 = \int_{t_0}^{t_\mathrm{f}} \left(\boldsymbol{\rho}^\mathsf{T} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \frac{\partial l_1}{\partial \mathbf{u}} \right) \delta\mathbf{u}(t) dt \tag{4.10}$$

$$\delta\boldsymbol{\Phi} \triangleq \begin{bmatrix} \delta x_1 \\ \vdots \\ \delta x_q \end{bmatrix}_{t=t_\mathrm{f}} = \int_{t_0}^{t_\mathrm{f}} \mathbf{R}^\mathsf{T} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta\mathbf{u}(t) dt \tag{4.11}$$

Now we want to minimize (4.10) s.t. constraints (4.11). However, both have linearized relations w.r.t $\delta\mathbf{u}$ so there is no minimum for δJ_1. A simple way to create a minimum is to add a quadratic integral penalty function in $\delta\mathbf{u}$ to (4.10)

$$\delta J_\mathrm{ex} = \delta J_1 + \frac{1}{2} \int_{t_0}^{t_\mathrm{f}} (\delta\mathbf{u})^\mathsf{T} \mathbf{W} \delta\mathbf{u} \; dt, \tag{4.12}$$

where $\mathbf{W}(t) \in \mathbb{R}^{m \times m}$ is an arbitrary positive-definite weighting matrix. The problem then becomes a minimization problem of δJ_ex subject to (4.11). Adjoining (4.11) to (4.12) with another constant Lagrange multiplier $\boldsymbol{\nu}$, one obtains

$$\delta \bar{J} = \delta J_\mathrm{ex} + \boldsymbol{\nu}^\mathsf{T} \left[\int_{t_0}^{t_\mathrm{f}} \mathbf{R}^\mathsf{T} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta\mathbf{u}(t) dt - \delta\boldsymbol{\Phi} \right]. \tag{4.13}$$

If we neglect the change in coefficients, the first derivative of (4.13) is given by

$$\delta(\delta \bar{J}) = \int_{t_0}^{t_\mathrm{f}} \left[\frac{\partial l_1}{\partial \mathbf{u}} + (\boldsymbol{\rho} + \mathbf{R}\boldsymbol{\nu})^\mathsf{T} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + (\delta\mathbf{u})^\mathsf{T} \mathbf{W} \right] \delta(\delta\mathbf{u}) dt \tag{4.14}$$

Set (4.14) to be zero, one can find a solution of

$$\delta\mathbf{u} = -\mathbf{W}^{-1} \left[\frac{\partial l_1}{\partial \mathbf{u}} + (\boldsymbol{\rho} + \mathbf{R}\boldsymbol{\nu})^\mathsf{T} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right]^\mathsf{T} \tag{4.15}$$

that minimizes (4.13). Substitute this into (4.11), we find that

$$\delta\boldsymbol{\Phi} = -\mathbf{I}_{\boldsymbol{\Phi}\mathbf{J}} - \mathbf{I}_{\boldsymbol{\Phi}\boldsymbol{\Phi}}\boldsymbol{\nu} \tag{4.16}$$

where $\mathbf{I}_{\boldsymbol{\Phi}\boldsymbol{\Phi}} \in \mathbb{R}^{n \times n}$ and $\mathbf{I}_{\boldsymbol{\Phi}\mathbf{J}} \in \mathbb{R}^{1 \times n}$ are computed by

$$\mathbf{I}_{\boldsymbol{\Phi}\boldsymbol{\Phi}} = \int_{t_0}^{t_\mathrm{f}} \mathbf{R}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \mathbf{W}^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\mathsf{T} \mathbf{R} dt \tag{4.17}$$

and

$$\mathbf{I}_{\mathbf{J}\boldsymbol{\Phi}} = \mathbf{I}_{\boldsymbol{\Phi}\mathbf{J}}^\mathsf{T} = \int_{t_0}^{t_\mathrm{f}} \left(\boldsymbol{\rho} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \frac{\partial l_1}{\partial \mathbf{u}} \right) \mathbf{W}^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\mathsf{T} \mathbf{R} dt \tag{4.18}$$

Assuming that $\mathbf{I}_{\boldsymbol{\Phi}\boldsymbol{\Phi}}$ is non-singular, we can solve (4.16) for the value of $\boldsymbol{\nu}$

$$\boldsymbol{\nu} = -\mathbf{I}_{\boldsymbol{\Phi}\boldsymbol{\Phi}}^{-1} (\delta\boldsymbol{\Phi} + \mathbf{I}_{\boldsymbol{\Phi}\mathbf{J}}) \tag{4.19}$$

with $\delta\boldsymbol{\Phi} = -\epsilon\boldsymbol{\Phi}[\mathbf{x}(t_\mathrm{f})]$ and a constant $\epsilon \in (0, 1]$. Finally the new control $\mathbf{u}_1$ is updated to

$$\mathbf{u}_1 = \mathbf{u}_{1,\mathrm{old}} + \delta\mathbf{u} \tag{4.20}$$

with the old control input $\mathbf{u}_{1,\text{old}}$ and the update $\delta\mathbf{u}$ from (4.15).

In summary, the procedure of FOGA follows these steps:

1. Initialize a set of control input $\mathbf{u}(t)$.
2. Forward integrate (4.4) with the initial conditions $\mathbf{x}(t_0)$ and the initial guess of the control input from step 1.
3. Determine the co-state vector $\boldsymbol{\rho}$ and the matrix of influence functions $\mathbf{R}$ by backward integration through (4.6) and (4.8) with the terminal conditions (4.7) and (4.9).
4. Calculate $\mathbf{I}_{\boldsymbol{\Phi\Phi}}$, $\mathbf{I}_{\boldsymbol{\Phi J}}$ through the integrals (4.17), (4.18) simultaneously with step 3.
5. Determine $\boldsymbol{\nu}$ from (4.19) and compute and estimation of $\delta\mathbf{u}$ via (4.15).
6. Update the control input $\mathbf{u}_1$ using (4.20)

Remark: In the standard OCP, step (2)-(6) are repeated until the optimal solution is found. However, since we pursue fast computation and real-time capability, these steps are only performed once. Obviously, $\mathbf{u}_1$ is not yet close to the optimal solution. SAC is then used to further improve the performance without trading much of the computation time.

4.4.2 Sequential Action Control

The motivation of using Sequential Action Control (SAC) is that, instead of computing the control input for the whole prediction horizon, it is more crucial to consider the next interval since it will be applied to the system first. On the other hand, in dynamic environments, the predicted controller is affected by moving obstacles or might be completely changed if the final goals changes on-the-fly. SAC therefore aims to improve the control input for only the next interval, but still uses the same prediction horizon $[t_0, t_\text{f}]$ for the performance cost evaluation.

Assuming that the nominal controller $\mathbf{u}_1$ is obtained using FOGA described in 4.4.1, we want to find a control $\mathbf{u}_2^*$, denoted as the optimal control, that further improves the cost function (4.2) w.r.t the dynamic system (4.4). SAC then computes a triplet consisting of the control value $\mathbf{u}_2^*$, its application time τ_m and the application duration λ. This triplet is called an action and the control signal can be written as

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_1, & t \notin \left[\tau_m - \frac{\lambda}{2}, \tau_m - \frac{\lambda}{2}\right] \\ \mathbf{u}_2^*, & t \in \left[\tau_m - \frac{\lambda}{2}, \tau_m - \frac{\lambda}{2}\right] \end{cases} \tag{4.21}$$

with the nominal controller $\mathbf{u}_1$ and the optimal controller $\mathbf{u}_2^*$. This can be interpreted as a switching controller where SAC switches between two modes. These two modes are given by

$$\mathbf{f}_1(t) \triangleq \mathbf{f}(\mathbf{x}(t), \mathbf{u}_1(t)) \tag{4.22}$$

for the nominal controller $\mathbf{u}_1$ and

$$\mathbf{f}_2(t, \tau_m) \triangleq \mathbf{f}(\mathbf{x}(t), \mathbf{u}_2^*(\tau_m)) \tag{4.23}$$

for the optimal controller $\mathbf{u}_2^*$. In this paper, the application time τ_m is deterministic with $\tau_m = t_\text{cur} + \tau_\text{sample}/2$ where t_cur denotes the current time and τ_sample denotes the sampling time. Now remember that our aim is to improve the cost function (4.2) with the new controller

$\mathbf{u}_2^*$ applied within the duration λ. We then rely on the mode insert gradient[EWA06], which evaluates the first-order sensitivity of the cost (4.2)

$$\frac{dJ_1}{d\lambda^+}(\tau_m) = \boldsymbol{\rho}(\tau_m)^\mathsf{T}(\mathbf{f}_2(\cdot,\tau_m) - \mathbf{f}_1(\cdot,\tau_m)) \tag{4.24}$$

This equation measures how the cost is influenced from varying the length λ of the application of the optimal control $\mathbf{u}_2^*$. The co-state $\boldsymbol{\rho}$ in (4.24) is computed based on (4.6) and (4.7). To reduce the cost J_1, (4.24) should be driven to a desired negative value $\alpha_d \in \mathbb{R}^-$. This can be done by simply introducing an auxiliary cost function

$$\begin{aligned} l_2(s) &= l_2(\mathbf{x}(s), \mathbf{u}_1(s), \mathbf{u}_2(s), \boldsymbol{\rho}(s)) \\ &= \frac{1}{2}\left[\frac{\mathrm{d}J_1}{\mathrm{d}\lambda^+} - \alpha_d\right]^2 + \frac{1}{2}\|\mathbf{u}_2(s)\|_\mathbf{S}^2 \end{aligned} \tag{4.25}$$

with $\mathbf{S} > 0$ and $\|\mathbf{u}_2(s)\|_\mathbf{S}^2 = \mathbf{u}_2(s)^T\mathbf{S}\mathbf{u}_2(s)$. Solving the minimization problem of (4.25) results in a control that achieves the desired sensitivity α_d. For models in control-affine form (4.4), the solution for this minimization is given analytically by

$$\mathbf{u}_2^* = (\boldsymbol{\Lambda} + \mathbf{S}^\mathsf{T})^{-1}[\boldsymbol{\Lambda}\mathbf{u}_1 + \mathbf{h}(\mathbf{x})^\mathsf{T}\boldsymbol{\rho}\alpha_d] \tag{4.26}$$

with $\boldsymbol{\Lambda} = \mathbf{h}(\mathbf{x})^\mathsf{T}\boldsymbol{\rho}\boldsymbol{\rho}^\mathsf{T}\mathbf{h}(\mathbf{x})$.

4.4.3 Extended Sequential Action Control with target constraints

One problem arises from the theoretical background of SAC presented in Section 4.4.2 is that it cannot handle constraints and hence using SAC solely in the next step can lead to the violation of the constraint (4.3c). To prevent this happen, we extend the the original SAC method by using the knowledge from FOGA. Re-define the auxiliary cost function in (4.25) into

$$\begin{aligned} l_2^{\text{ext}}(s) &= l_2(\mathbf{x}(s), \mathbf{u}_1(s), \mathbf{u}_2(s), \boldsymbol{\rho}(s)) \\ &= \frac{1}{2}\left[\frac{\mathrm{d}J_1}{\mathrm{d}\lambda^+} - \alpha_d\right]^2 + \frac{1}{2}\left[\frac{dl_\mathrm{c}}{d\lambda^+} - \alpha_c\right]^2 + \frac{1}{2}\|\mathbf{u}_2(s)\|_\mathbf{S}^2 \end{aligned} \tag{4.27}$$

where $l_c = \frac{1}{2}\boldsymbol{\Phi}^\mathsf{T}\mathbf{Q}_c\boldsymbol{\Phi}$, $\mathbf{Q}_c \in \mathbb{R}^{q\times q}$ is a positive definite matrix and $\boldsymbol{\Phi}$ is the vector of target constraints defined in (4.3c). l_c can be interpreted as the additional constraint cost and therefore $\frac{dl_c}{d\lambda^+}$ measures how this constraint cost is influenced from varying the length λ of the optimal control $\mathbf{u}_2^*$. Hence solving the new minimization problem of (4.27) also drives this sensitivity to a negative value $\alpha_c \in \mathbb{R}^-$. This leads to the reduction of l_c which then helps SAC prevent the constraints from being violated.

The only problem now is the evaluation of $\frac{dl_c}{d\lambda^+}$. We have

$$\frac{dl_c}{d\lambda^+}(\tau_m) = \frac{dl_c}{d\mathbf{x}(t_\mathrm{f})}\frac{d\mathbf{x}(t_\mathrm{f})}{d\lambda^+}(\tau_m) = \boldsymbol{\Phi}^\mathsf{T}\mathbf{Q}_c\frac{d\mathbf{x}(t_\mathrm{f})}{d\lambda^+}(\tau_m) \tag{4.28}$$

From (4.11), we have

$$\begin{aligned} \delta\boldsymbol{\Phi} = \delta\mathbf{x}(t_\mathrm{f}) &= \int_{t_0}^{t_\mathrm{f}} \mathbf{R}^\mathsf{T}\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\delta\mathbf{u}(t)dt \\ &= \int_{\tau_m}^{\tau_m+\delta\lambda} \mathbf{R}^\mathsf{T}\frac{\partial\mathbf{f}}{\partial\mathbf{u}}(\mathbf{u}_2 - \mathbf{u}_1)dt \end{aligned} \tag{4.29}$$

with $[\tau_m, \tau_m + \delta\lambda]$ is the infinitesimal duration where $\mathbf{u}_2$ is applied. As $\delta\lambda \to 0$, (4.29) can be written as

$$d\mathbf{x}(t_\mathrm{f}) = \mathbf{R}^\mathsf{T} \frac{d\mathbf{f}}{d\mathbf{u}}(\tau_m)(\mathbf{u}_2 - \mathbf{u}_1)d\lambda \tag{4.30}$$

or

$$\frac{d\mathbf{x}(t_\mathrm{f})}{d\lambda^+}(\tau_m) = \mathbf{R}^\mathsf{T}\mathbf{h}(\tau_m)(\mathbf{u}_2 - \mathbf{u}_1) \tag{4.31}$$

Substitute this into (4.28) and solve the minimization problem of (4.27) the same way as in Section 4.4.2, we obtain a new analytical solution of $\mathbf{u}_2^*$

$$\mathbf{u}_2^* = (\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2 + \mathbf{S})^{-1}(\boldsymbol{\Lambda}_1\mathbf{u}_1 + \boldsymbol{\Lambda}_2\mathbf{u}_1 + \mathbf{h}^\mathsf{T}\boldsymbol{\rho}\alpha_d + \mathbf{h}^\mathsf{T}\mathbf{R}_c^\mathsf{T}\alpha_c) \tag{4.32}$$

with $\boldsymbol{\Lambda}_1 = \mathbf{h}^\mathsf{T}\boldsymbol{\rho}\boldsymbol{\rho}^\mathsf{T}\mathbf{h}$, $\boldsymbol{\Lambda}_2 = \mathbf{h}^\mathsf{T}\mathbf{R}_c^\mathsf{T}\mathbf{R}_c\mathbf{h}$, $\mathbf{R}_c = \boldsymbol{\Phi}^\mathsf{T}\mathbf{Q}_c\mathbf{R}^\mathsf{T}$.

Algorithm 1: TC-SAC

Initialize $\mathbf{x}_0$, $\mathbf{x}_\mathrm{d}$, current time t_curr, prediction horizon T_P, sampling time τ_sample, end time T_end , initial guess for nominal control $\mathbf{u}_1$.

while $t_\mathrm{curr} < T_\mathrm{end}$ **do**
  if $t_\mathrm{curr} \geq \tau_m$ **then**
    $\tau_m = t_\mathrm{curr} + \tau_\mathrm{sample}/2$
    Simulate $(\mathbf{x}, \boldsymbol{\rho}, \mathbf{R})$ using $\mathbf{u}_1$ for $t \in [t_0, t_\mathrm{f}]$
    Compute $\boldsymbol{I}_{\psi\psi}$, $\boldsymbol{I}_{\boldsymbol{J}\psi}$ as in (4.17) and (4.18), $\boldsymbol{\nu}$ by (4.19), change in control $\delta\mathbf{u}$ by (4.15)
    Update $\mathbf{u}_1 \leftarrow \mathbf{u}_1 + \delta\mathbf{u}$
    Saturate $\mathbf{u}_1$ to $[u_\mathrm{min}, u_\mathrm{max}]$
    Simulate $(\mathbf{x}, \boldsymbol{\rho})$ using new $\mathbf{u}_1$ for $t \in [t_0, t_f]$
    Compute new costs $J_\mathrm{1,init}, l_\mathrm{c,init}$
    Specify α_d
    Compute $\mathbf{u}_2^*$ from (4.26) and saturate it to $[u_\mathrm{min}, u_\mathrm{max}]$
    Initialize $k = 0$, $J_\mathrm{1,min} \leftarrow J_\mathrm{1,init}$, $J_\mathrm{1,SAC} \leftarrow \infty$, $l_\mathrm{c,min} \leftarrow l_\mathrm{c,init}$, $l_\mathrm{c,SAC} \leftarrow \infty$
    while $k < k_\mathrm{max}$ **do**
      $\lambda = \tau_\mathrm{sample}$
      $(\tau_0, \tau_f) = (\tau_m - \frac{\lambda}{2}, \tau_m + \frac{\lambda}{2})$
      Re-simulate $(\mathbf{x}, \boldsymbol{\rho})$ applying control (4.21)
      Compute new cost $J_\mathrm{1,SAC}, l_\mathrm{c,SAC}$
      $k = k + 1$
      if $J_\mathrm{1,SAC} < J_\mathrm{1,init}$ **and** $l_\mathrm{c,SAC} < l_\mathrm{c,init}$ **then**
        $J_\mathrm{1,min} \leftarrow J_\mathrm{1,SAC}, l_\mathrm{c,min} \leftarrow l_\mathrm{c,SAC}$
      end if
    end while
    if $J_\mathrm{1,SAC} < J_\mathrm{1,init}$ **and** $l_\mathrm{c,SAC} < l_\mathrm{c,init}$ **then**
      $\mathbf{u}_1 \leftarrow \mathbf{u}_2^*$ for $t \in (\tau_m - \frac{\lambda}{2}, \tau_m + \frac{\lambda}{2})$
    end if
  else
    Apply control $\mathbf{u}_1$
  end if
  $(t_0, t_f) = (t_0 + \tau_\mathrm{sample}, t_0 + T_\mathrm{pre} + \tau_\mathrm{sample})$
end while

In this work, the application time λ is set to be equal to τ_{sample} such that $\mathbf{u}_2^*$ is applied for one interval. Additionally, $\mathbf{u}_2^*$ is just applied if it results smaller in both performance cost J_1 and terminal constraint cost l_c over the prediction horizon T_{P} compared to the nominal control $\mathbf{u}_1$. This is considered as the Selection procedure in the control scheme as illustrated in Fig. 4.1.

In order to obtain a feasible solution, a boundary of control signals is given by box-constraints, i.e. $U = [u_{\min}, u_{\max}]^m$, by simply saturating the control output computed above. Overall, Algorithm 1 outlines the general structure of the proposed method.

4.5 Trajectory Tracking Simulation and Comparison to Optimal Control

In this section, we provide several examples to demonstrate the effectiveness of the proposed approach. Specifically, we apply our method to the reaching task and trajectory tracking task for a 2DOF robotic arm, comparing the results with the original SAC. From a theoretical standpoint, an intriguing question arises regarding the impact of the extended SAC presented in Section 4.4.3 on system performance. To address this, we introduce two additional controllers for comparison purposes. Firstly, we compare the performance of the extended SAC with target constraints to the original SAC. Secondly, we evaluate the performance of FOGA with multi-steps in comparison to TC-SAC. Additionally, we extend our analysis to include a car-like system, highlighting the versatility of TC-SAC in handling diverse and complex systems with minimal modifications. By conducting these extensive evaluations, we aim to provide comprehensive evidence of the efficacy and applicability of our proposed approach in a variety of practical scenarios.

4.5.1 Comparison of the proposed approach to other optimal-based control approaches

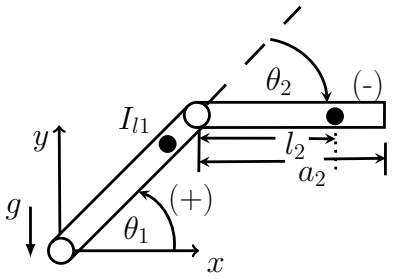


Figure 4.2: Setup of the two degrees of freedom robot used for the simulation.

In this Section, we consider a basic reaching motion task and evaluate the performance of the proposed approach on a 2DOF robotic arm in vertical plane as shown in Fig. 4.2. The full dynamics of the robot are considered, including friction and gravitation. The task of the robot is to reach a pre-defined position. The performance cost is chosen to penalize the error between the current and desired states

$$l_1(t) = (\mathbf{x}(t) - \mathbf{x}_d)^T \mathbf{Q} (\mathbf{x}(t) - \mathbf{x}_d), \tag{4.33}$$

with $\mathbf{Q} \succeq 0$. The target constraint is also set to be the desired position. The prediction horizon is chosen $T_{\text{p}} = 0.3$s and the sampling rate is set to 1kHz. The initial position of the

robot is $\mathbf{x}_0 = (-\frac{\pi}{2}, 0, 0, 0)$ and the control $\mathbf{u}$ is bounded by $\mathbf{u} \in [-50, 50]$Nm. The sensitivity α_d was chosen proportional to the cost J_1, $\alpha_\mathrm{d} = \omega_{\alpha_d} J_1$ with $\omega_{\alpha_d} = -10$. Similarly, the sensitivity α_c is set to be $\alpha_c = \omega_{\alpha_c} l_c$ with $\omega_{\alpha_c} = -5$.

4.5.1.1 Upright as Desired Position

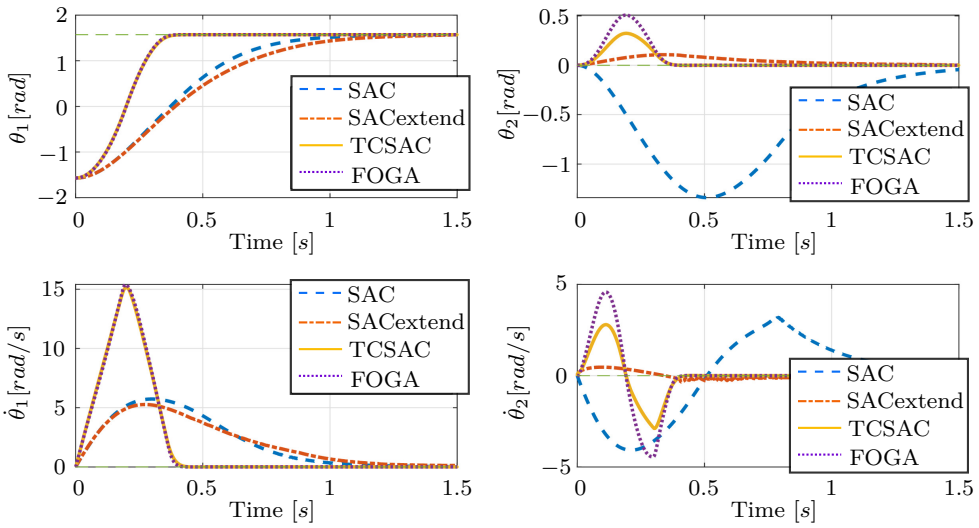


Figure 4.3: States of 2DOF robotic arm when the designed position is upright

First, the results are evaluated in the case of $\mathbf{x}_\mathrm{d}$ being the upright position, $\mathbf{x}_\mathrm{d} = (\frac{\pi}{2}, 0, 0, 0)$. The weighting matrix for the performance cost (4.33) is $\mathbf{Q} = \mathrm{diag}([100,\ 50,\ 0.0001,\ 0.0001])$. Having the first two values of $\mathbf{Q}$ much higher than the latter means that the position errors are penalized more heavily than the velocity errors, which is necessary for tracking tasks. The weighting matrix for l_c in (4.27) is $\mathbf{Q}_c = \mathrm{diag}([10,\ 1000,\ 10,\ 10])$. For FOGA, we run in two iterations to have a fair comparison to TC-SAC, in term of cost improvement and computation time. The simulation is run for 1.5s and the result is shown in Fig. 4.3. All methods succeed in controlling and stabilizing the robot. However, TC-SAC and FOGA needs only 0.5s to converge to $\mathbf{x}_d$, while SAC and the extended SAC take more than 1s to be converged. The overall cost is shown in Table 4.1. Further more, each method is run for 100 times to obtain the average computation time. Note that the code is run on Matlab environment so the computation time is only used to evaluate the speed of these methods relatively.

Table 4.1: Total cost and computation time.

	SAC	Extended SAC	TC-SAC	FOGA
Total cost	316.74	291.02	152.21	150.60
Computation time	18.18s	19.60s	89.82s	124.63s

When comparing SAC to the extended SAC, it is evident that the incorporation of the new sensitivity term improves the constraint satisfaction in the states of the robot. This is particularly noticeable in the case of θ_2, where the extended SAC minimizes the deviation between

the current and desired states to a greater extent than SAC. This observation highlights the enhanced performance of the extended SAC in terms of tightening the target constraints.

In the comparison between TC-SAC and FOGA, both approaches exhibit similar performance outcomes. However, it is worth noting that TC-SAC demonstrates a computational advantage, with a computation time that is 34s shorter than that of FOGA with 2 iterations. This discrepancy in computation time can be attributed to the fact that each iteration of SAC is considerably faster than each iteration of FOGA, as SAC benefits from a direct analytical solution. Consequently, TC-SAC emerges as the preferable choice, as it achieves comparable performance in a shorter time frame.

4.5.1.2 Arbitrary position

In this section, we want to analyze the performance of the methods in the case of any arbitrary desired position. In this case, the controller has to compensate the gravitational force, which is zero if the robot is at the upright position (equilibrium point). This highlights the role of the target constraint in TC-SAC and FOGA in term of convergence. The desired position is set to be $\mathbf{x}_\mathrm{d} = (\frac{3\pi}{4}, \frac{-3\pi}{4}, 0, 0)$ and other parameters are set to be the same as the case of the upright position. The simulation is run for 2s and the result is shown in Fig. 4.4. It can be seen that TC-SAC and FOGA quickly converges to the desired position in approximately 0.5s while both SAC and the extended SAC methods can not converge to the desired position in θ_1 and θ_2. These offsets appear in any desired position that is not the equilibrium point. Therefore, TC-SAC and FOGA is preferable for reaching/tracking tasks even though the computation time is larger.

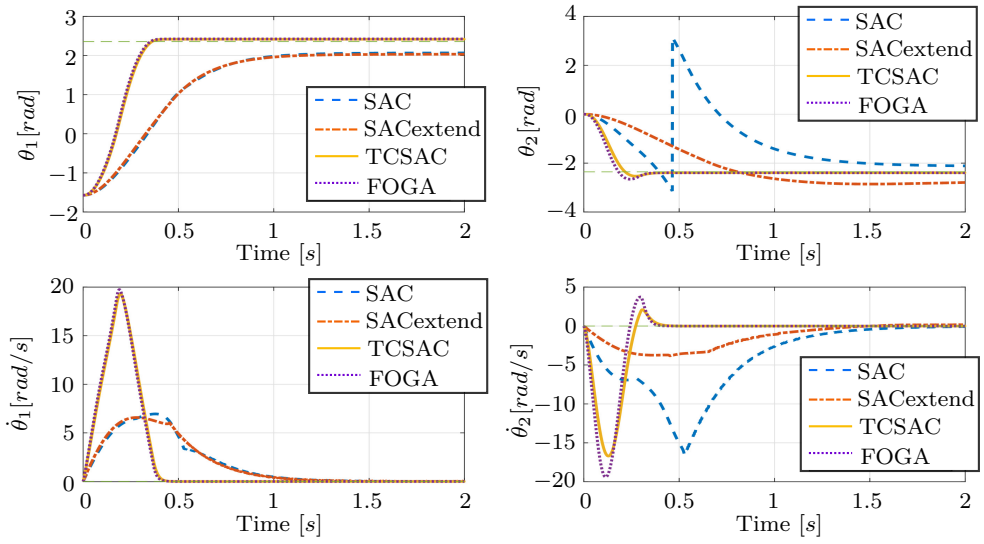


Figure 4.4: States of 2DOF robotic arm in the case of arbitrary desired position

4.5.1.3 Tracking an ellipse trajectory

Taking one step further, we evaluate the methods in trajectory tracking task. Since SAC and the extended SAC are incapable of reaching arbitrary position, they are excluded from this task. The trajectory is set to be an ellipse with the radii along x-axis and y-axis are $0.3m$ and

$0.18m$ respectively. For the sake of simplicity, the robot follows the ellipse with a constant velocity and the whole ellipse takes 3s to finish. The starting position is set to be upright. In this simulation, the weighting matrices $\mathbf{Q}, \mathbf{Q}_c$ are set to be $\mathbf{Q} = \text{diag}([100, 50, 10, 10])$ and $\mathbf{Q}_c = \text{diag}([10, 100, 0.1, 0.1])$. Other parameters remain the same. Fig. 4.5 shows the tracking performance between TC-SAC and FOGA in XY graph while Fig. 4.6 shows it in each state of the robot. Fig. 4.5 also illustrates the configuration of the robot at different positions along the ellipse.

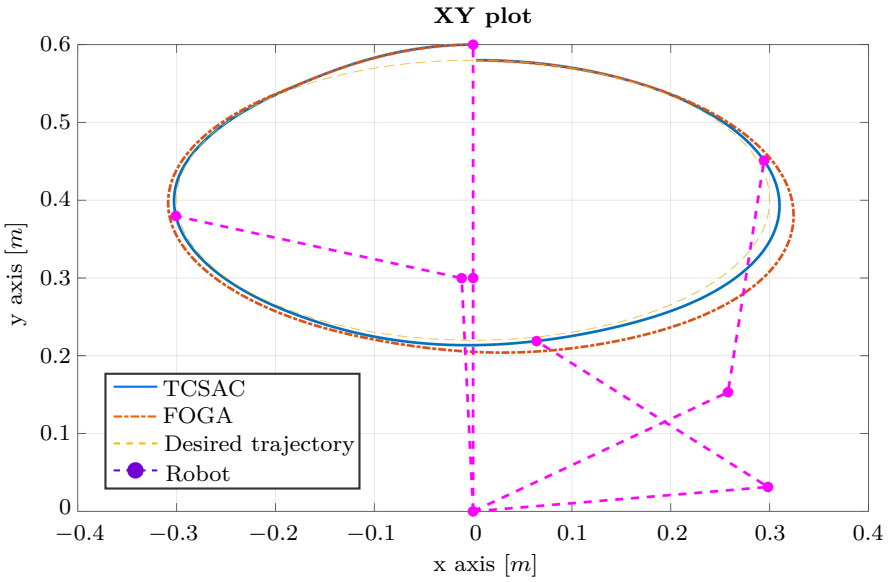


Figure 4.5: Tracking performance of TC-SAC and FOGA

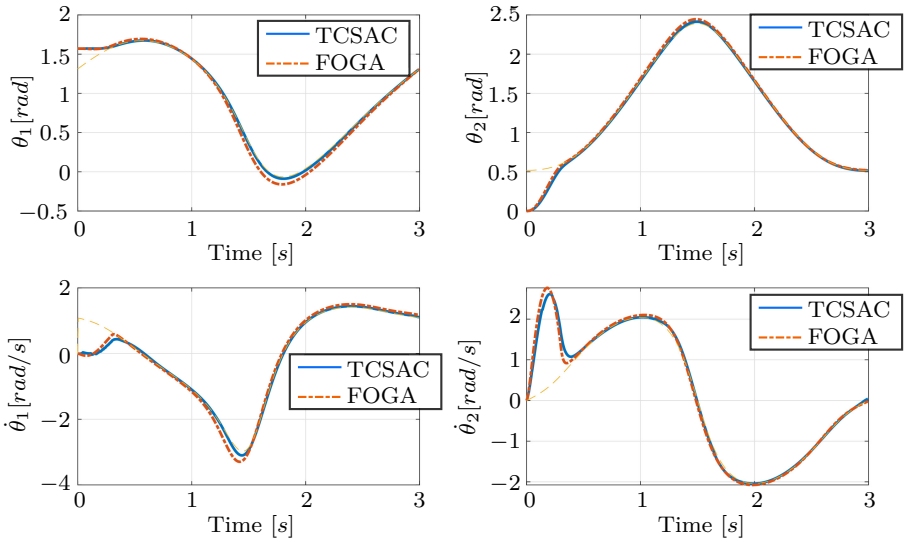


Figure 4.6: States of 2DOF robotic arm in case of tracking an ellipse trajectory

Both TC-SAC and FOGA demonstrate successful tracking of the given trajectory. However,

TC-SAC outperforms FOGA in this task, particularly on the right side of the ellipse. From a mathematical perspective, TC-SAC can be viewed as a combination of one iteration of FOGA followed by an update from SAC. Therefore, it can be inferred that the SAC step in TC-SAC improves the cost function further compared to the second iteration step of FOGA, all while being executed in a significantly shorter timeframe. This emphasizes the advantage of TC-SAC in scenarios where the controller computation must be fast due to time restrictions, such as in dynamic environments. TC-SAC effectively enhances the overall cost performance without incurring excessive computational overhead. Fig. 4.7 also shows the control signal of TC-SAC on the first and second joint of the robot. The final graph in Fig. 4.7 indicates the time steps at which SAC is activated (denoted as 1), signifying instances when the control signal computed by SAC contributes to cost reduction. By observing the frequency at which SAC takes over FOGA ($\mathbf{u}_2^*$ is applied instead of $\mathbf{u}_1$), we can justify the effectiveness of the additional SAC step in TC-SAC. It can be seen that $\mathbf{u}_2^*$ is used most of the time which means SAC often improves the solution of FOGA ($\mathbf{u}_1$). This proves the effectiveness of this additional SAC step, benefiting from its fast computation. In conclusion, TC-SAC is the preferred choice for applications that require rapid computation of an optimal controller. Its ability to swiftly generate superior solutions makes it well-suited for time-sensitive scenarios, effectively striking a balance between computational speed and optimality.

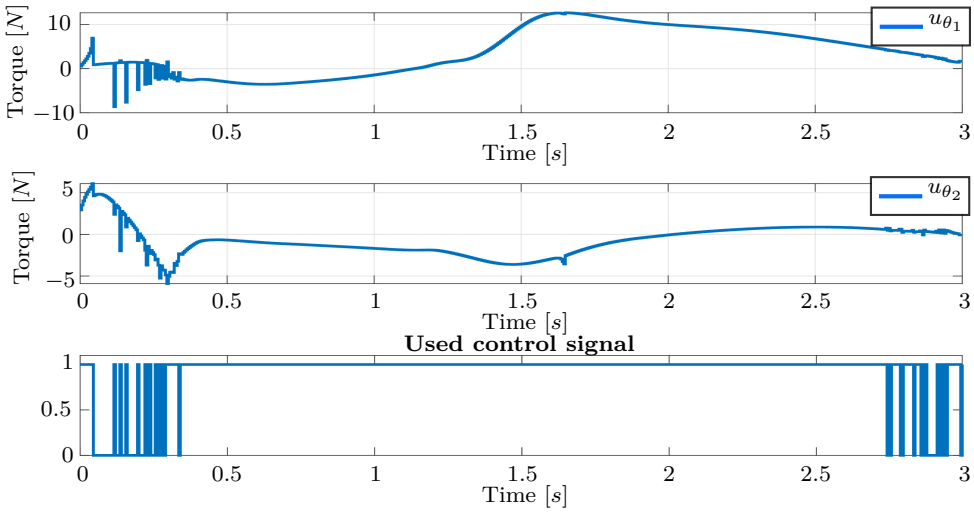


Figure 4.7: Control signal of the 2DOF robotic arm

4.5.1.4 Validation of Obstacle Avoidance

In this section, we focus on validating the obstacle avoidance capability of the 2DOF robot discussed earlier. Here, we examine the capability of the robot to avoid obstacles in a static scenario. Specifically, the robot is tasked with tracking an ellipse trajectory from the bottom to an upright position while encountering a rounded object placed in its path. This setup allows us to assess the effectiveness of the obstacle avoidance behavior and evaluate its performance.

The cost function has the form

$$l_1(t) = l_{1,\text{track}}(t) + l_{1,\text{limit}}(t) + l_{1,\text{avoid}}(t) \tag{4.34}$$

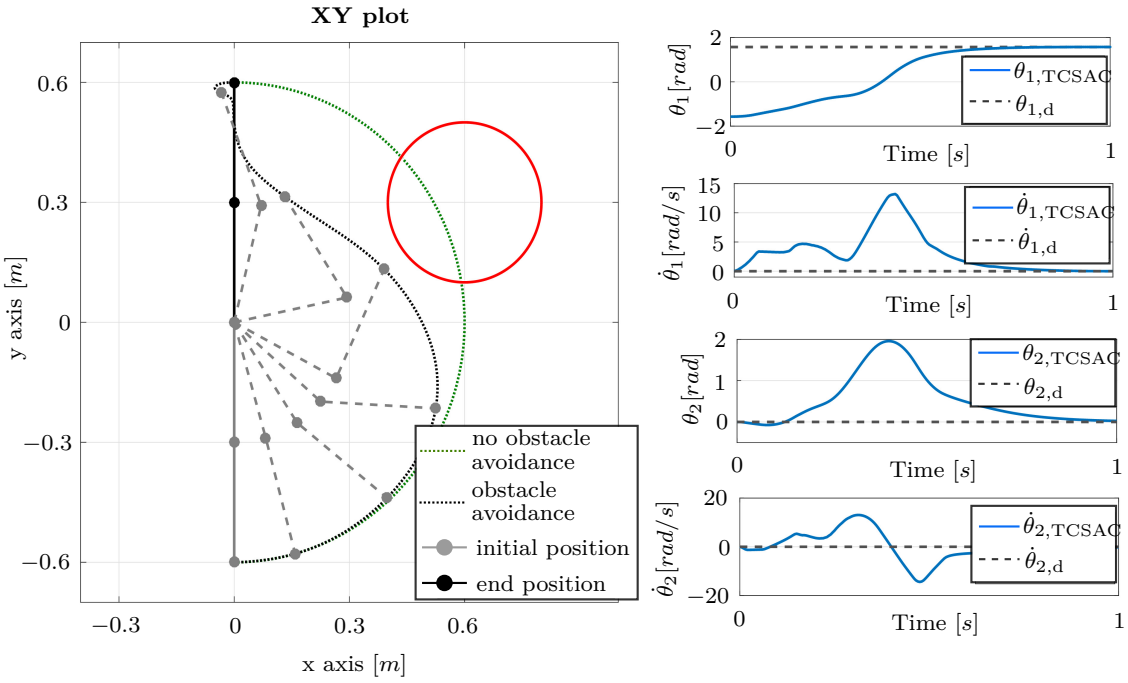


Figure 4.8: States of 2DOF robotic arm when considering obstacle avoidance cost is applied on the right. On the left is the movement of the end-effector with or without obstacle avoidance cost.

in which each component has its own role. Since the main task of the robot is moving toward a desired configuration, a tracking cost in a quadratic form

$$l_{1,\mathrm{track}}(t) = (\mathbf{x}(t) - \mathbf{x}_d)^T \mathbf{Q} (\mathbf{x}(t) - \mathbf{x}_d), \tag{4.35}$$

with $\mathbf{Q} \succeq 0$, is chosen. Thus, an error between current state x and desired one x_d is penalized. In addition, to prevent the robot violating its physical joint limits, an additional cost function $l_{1,\mathrm{limit}}$ is defined as follows:

1. This cost increases whenever the system is close to its joint limit constraints $x_{i,\max}$ and $x_{i,\min}$ within a given tolerance $x_{i,\mathrm{tol}}$.

2. The cost is zero if the system is not close to its joint limit constraints.

The joint limit cost function then e.g. have the form

$$l_{1,\mathrm{limit},i} = \begin{cases} C[x_i - (x_{i,\mathrm{tol,\ min}})]^2, & x_i \leq x_{i,\mathrm{tol,\ min}} \\ 0, & \text{otherwise} \\ C[x_i - (x_{i,\mathrm{tol,\ max}})]^2, & x_i \geq x_{i,\mathrm{tol,\ max}} \end{cases}, \tag{4.36}$$

with x_i being the i-th component of state $\mathbf{x}$, $x_{i,\mathrm{tol,\ min}} = x_{i,\min} + x_{i,\mathrm{tol}}$, $x_{i,\mathrm{tol,\ max}} = x_{i,\max} - x_{i,\mathrm{tol}}$ and a shape parameter $C > 0$. Thus, $l_{1,\mathrm{limit}} = \sum l_{1,\mathrm{limit},i}$ is smooth and satisfies all the demanded properties.

The third component $l_{1,\mathrm{avoid}}$ represents the obstacle avoidance cost and it is considered similar to the case of a joint limit cost as the requirements concerning the cost function are

the same. Furthermore, considering only spherical obstacles suffices as any object can be approximated by spheres. Therefore, $l_{1,\text{avoid}}$ is defined as

$$l_{1,\text{avoid}} = \begin{cases} \tilde{C}[\text{dist}(o,l) - d]^2, & \text{dist}(o,l) < d \\ 0, & \text{dist}(o,l) > d \end{cases}, \tag{4.37}$$

with parameters $\tilde{C}$ and d to be chosen. Here, $\text{dist}(o,\cdot)$ denotes the distance between the center o of an object and the robot. This distance can be measured by discretizing the setup of the robot and evaluating the distance w.r.t these discrete points.

Specifically, the chosen parameters for tracking cost (4.35), the cost for joint limit constraints (4.36) and the cost for obstacle avoidance (4.37) are: $\mathbf{Q} = \text{diag}([100, 50, 1, 1])$, $C = 1 \times 10^9$, $\tilde{C} = 1 \times 10^8$. Fig. 4.8 presents simulation results and the movement of the robotic arm. The movement of the robot shows that it successfully prevents collisions with the obstacle. This validation strengthens our confidence in the obstacle avoidance capability of the controller. Moving forward, the next section will demonstrate the versatility and applicability of TC-SAC in various other applications, highlighting its potential in diverse scenarios.

4.5.2 Trajectory tracking in dynamic environment of a car-like system

In recent years, autonomous driving has emerged as a prominent field of interest for both industry and researchers. In the realm of autonomous cars, it is crucial for the controlled vehicle to effectively respond to highly dynamic environments. Given the unpredictability of other participants in traffic, the car must rapidly adapt to changing situations. This section focuses on the implementation of TC-SAC for trajectory tracking tasks, with a particular emphasis on obstacle avoidance in both static and dynamic scenarios. To represent vehicles, we employ the widely used single-track model, also referred to as the bicycle model [RNH10, RA13, PCY+16, LLLW17, VZQD16]. The single-track model, depicted in Fig. 4.9, serves as an effective representation in capturing the essential dynamics of vehicles.

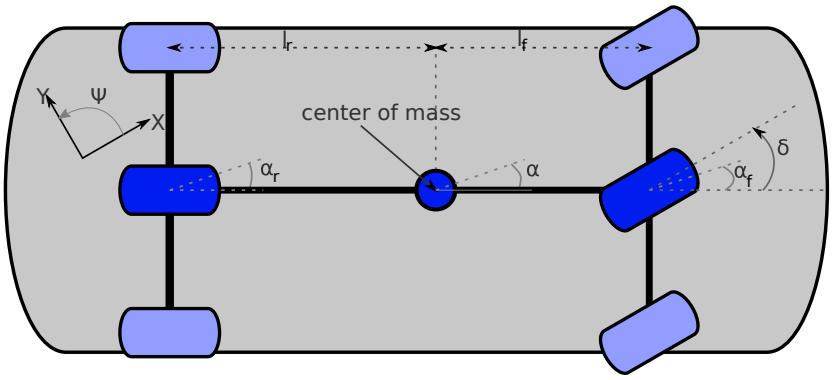


Figure 4.9: Schematic Diagram of the Single Track Model used in this work

The single track model is used in this work is nonlinear with 7 states and 2 control input

$$\mathbf{x} = \begin{bmatrix} X, Y, \Psi_c, v, \alpha, \dot{\Psi}_c, \delta \end{bmatrix} \tag{4.38}$$

$$\mathbf{u} = \begin{bmatrix} \dot{\delta}_{\text{input}} \\ M \end{bmatrix} \tag{4.39}$$

with

$$\dot{\delta} = \dot{\delta}_{\text{input}} \tag{4.40}$$

with the x and y-coordinates X and Y, the orientation Ψ_c, the velocity v, the side slip angle α, the change in orientation $\dot{\Psi}_c$, the steering angle δ and the applied torque M. The last state in x is the additional state added so that the single track model is in control-affine form. Further details about this car model are in Appendix A. We first present the trajectory tracking task for the car-like system with different prediction horizons in Section 4.5.2.1 to quantify the effectiveness of the prediction lengths side by side with the computational effort. Then the static obstacle avoidance and dynamic obstacle avoidance scenarios will be presented in Section 4.5.2.2 and 4.5.2.3 respectively.

4.5.2.1 Trajectory Tracking Task with Different Prediction Horizons

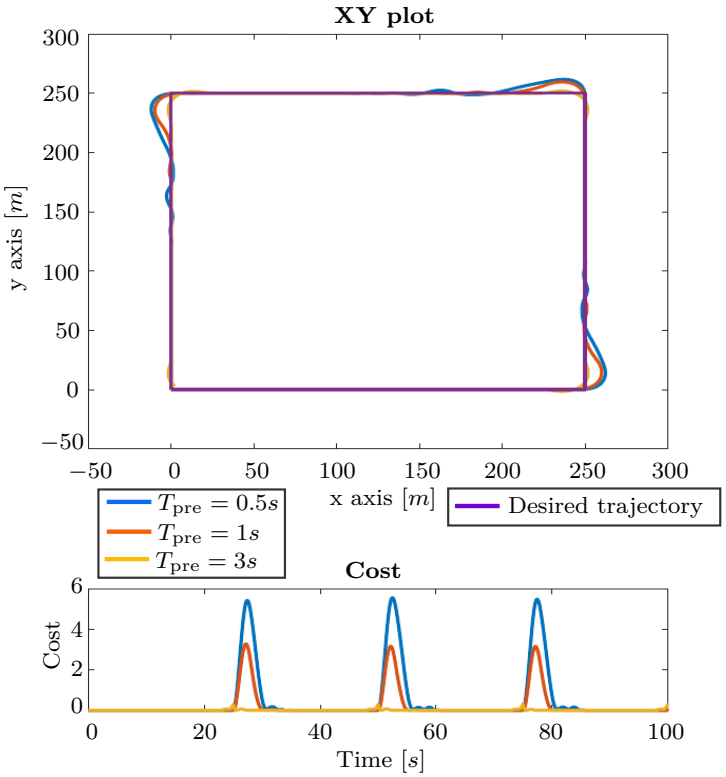


Figure 4.10: Comparison of different prediction horizons ($0.5s$,$1.0s$ and $3.0s$) for TC-SAC for tracking a square reference trajectory

In this Section, we assess the effectiveness of TC-SAC in a fundamental trajectory tracking task. Furthermore, we investigate the impact of varying receding horizons on the performance of TC-SAC. Specifically, we compare the results obtained with three different prediction horizons: 0.5s, 1.0s, and 3.0s. By examining the influence of prediction horizon length on the performance of TC-SAC, we aim to gain insights into the trade-off between optimality and computational effort. Typically, longer prediction horizons yield superior results in terms of optimality but require increased computational resources. Therefore, our objective is to find a balance that minimizes computational burden while maintaining satisfactory performance.

The sample time T_{sample} is set to 20ms in all scenarios if not mentioned differently. The same tracking cost $l_{1,\text{track}}$ is used

$$l_{1,\text{track}}(t) = (\mathbf{x}(t) - \mathbf{x}_d)^T \mathbf{Q} (\mathbf{x}(t) - \mathbf{x}_d), \tag{4.41}$$

with the weighting matrix $\mathbf{Q} \geq 0$ with $q_{1,1} = q_{2,2} = 1$ and $q_{4,4} = 0.1$ and all other elements equal to 0, thus only the distance to the $x-$ and $y-$ coordinates and the velocity at each time step influence the tracking cost while the other states are not considered. Furthermore, a final cost m is

$$m = (\mathbf{x}_{\text{t}_f}(t) - \mathbf{x}_d)^T \mathbf{P} (\mathbf{x}_{\text{t}_f}(t) - \mathbf{x}_d), \tag{4.42}$$

with the weighting matrix $\mathbf{P} \geq 0$ with $p_{5,5} = 10$ and all other elements equal 0 just the side slip angle will be considered. This is due to the characteristic of the car in which a large side slip angle would result in a state of the car which is much more difficult to control compared to having a small side slip angle. Thus (4.42) guarantees that more opportunities to follow the reference trajectory are left for the car at the end of the prediction horizon. As previously mentioned, the prediction horizon is set to be 0.5s, 1.0s and 3.0s. Setting the prediction horizon as short as possible is generally desirable to minimize computation time. However, it is important to maintain a minimum prediction horizon to ensure stable trajectory tracking and allow sufficient time for the car to react to potential obstacles.

A square shaped path with the length of each side equal 250m is chosen for this tracking task. This reference trajectory has the straight parts of the square which is easy to control the car and also has the corner parts where the dynamics of the car are violated by the reference trajectory which returns to be a more difficult case for the car to be optimally controlled.

The results of TC-SAC for the three prediction horizon cases are presented in Figure 4.10. It is observed that the cost in the straight segments is close to zero for all prediction horizons, indicating effective tracking. However, the cost increases in the corner segments as the prediction horizon decreases. This behavior is expected, as a longer prediction horizon allows the car more time to react and adjust its trajectory.

To evaluate the computation time of TC-SAC for each prediction horizon, simulations are conducted for a duration of 100s on an *i7-7700k* computer. The corresponding computation times are summarized in Table 4.2. As anticipated, the computation time increases with the prediction horizon. Notably, for a prediction horizon of 0.5s, the computation time is faster than the simulation time, demonstrating the real-time capability of TC-SAC. Even for a prediction horizon of 1.0s, the computation time remains close to real-time, while a prediction horizon of 3.0s exceeds real-time requirements. Thus, a trade-off is necessary to balance tracking accuracy and computation time. In this case, a prediction horizon of 1.0s is deemed the optimal choice. This prediction horizon is used for further evaluations of TC-SAC on a car-like system in this work, unless otherwise stated.

Table 4.2: Computation time of TC-SAC with different prediction horizons.

Prediction horizon	Computation time
0.5s	79s
1.0s	159s
3.0s	563s

4.5.2.2 Static Obstacle Avoidance

In this part, the performance of TC-SAC in an environment with static obstacle blocking the reference trajectories and in further scenarios which might occur in real world situations such as overtaking and frontal crash avoidance is evaluated.

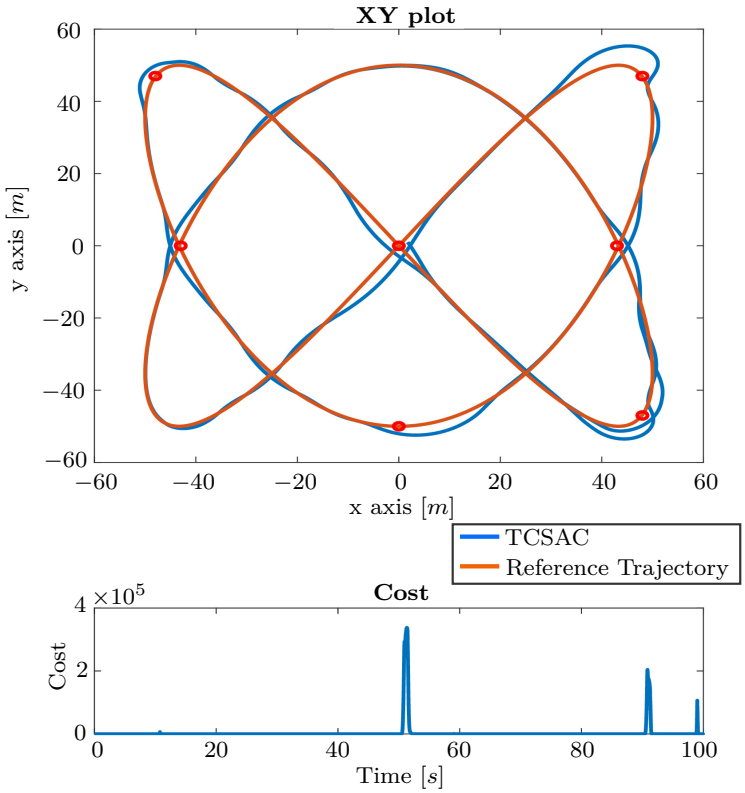


Figure 4.11: Performance of TC-SAC on a Lissajous curve as reference trajectory with avoidance of static obstacles

A Lissajous curve with a ratio of the frequencies of $\frac{2}{3}$ and a phase shift of $\frac{\pi}{2}$ is used for this tracking task.

$$\begin{aligned} x_{\text{ref}}(t) &= 50\ \sin(0.05\pi t) \\ y_{\text{ref}}(t) &= 50\ \cos(0.075\pi t) \\ v_{\text{ref}}(t) &= \sqrt{\dot{x}_{\text{ref}}^2(t) + \dot{y}_{\text{ref}}^2(t)} \end{aligned} \tag{4.43}$$

This reference trajectory is chosen since it results in different curvature and velocity at every point which makes it challenging for tracking. The velocity varies between about $3.65\frac{\text{m}}{\text{s}}$ and $14.16\frac{\text{m}}{\text{s}}$, which does not exceed the limit of the dynamic of the car.

In order to avoid the obstacles, an additional term is added into the cost function, which is defined as

$$l_{1,\text{avoid}} = \begin{cases} C_{\text{obstacle}} \cdot (\text{dist}(\cdot) - r)^2, & \text{dist}(\cdot) < r \\ 0, & \text{dist}(\cdot) > r \end{cases} \tag{4.44}$$

with the distance between the car from the center to an obstacle $\text{dist}(\cdot) = \text{dist}(x, y, x_c, y_c) =$

$\sqrt{(x-x_c)^2+(y-y_c)^2}$, the weighting factor C_{obstacle} and the obstacle radius r. The overall performance cost l_1 is than given by $l_1 = l_{1,\text{track}} + l_{1,\text{avoid}}$. Since obstacle avoidance is most crucial for the safety of passengers and other traffic participants, C_{obstacle} is set to 10^6 and therefore it is considered with a much higher weighting factor than the tracking task. It is assumed that the obstacle are 2m in diameter, which is close to the width of real cars. However, since avoidance cannot completely be guaranteed with soft constraints, a safety margin of 1m around the obstacle is added which leads to an obstacle radius $r = 2$m to increase safety. The position of the obstacles is given in Fig. 4.11 and marked by the circles.

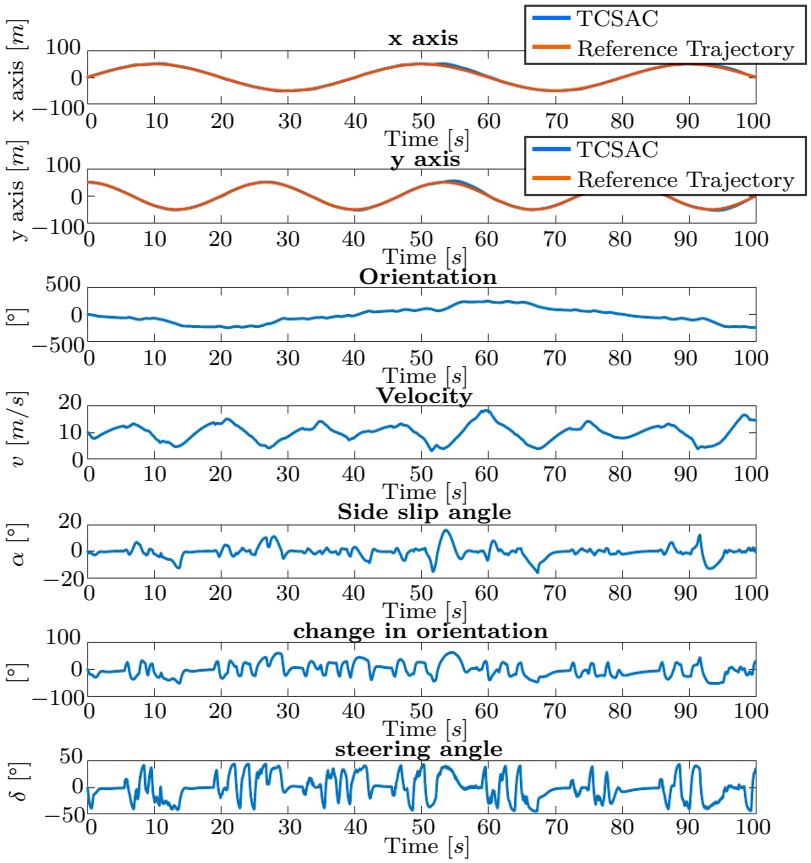


Figure 4.12: Other states of the single track model on a Lissajous curve as reference trajectory with avoidance of static obstacles

The results clearly demonstrate that the car successfully avoids all obstacles while accurately tracking the reference trajectory. Analyzing the cost measurements, we observe two peaks at around 51s and 91s, corresponding to the obstacles located in the top right and bottom right sections of the curve. Although there are no collisions, these peaks indicate that the obstacles affected the safety margin around the car. This highlights the significance of selecting an appropriate safety margin to ensure the safety of the car during trajectory tracking. Additionally, Figure 4.12 presents the other states of the single-track model, providing a comprehensive view of the behavior of the car during the obstacle avoidance task.

4.5.2.3 Dynamic Obstacle Avoidance

In the dynamic obstacle avoidance scenario, we once again utilize the Lissajous curve as the reference trajectory. To tackle the more challenging task of dynamic obstacle avoidance, the prediction horizon is increased to 2s. The scenario includes a total of 13 obstacles: one static obstacle positioned at the center, six obstacles moving on a circular trajectory with a radius of 40m, and six obstacles moving on a circular trajectory with a radius of 50m. Both circles share the same center as the Lissajous curve. The obstacles are evenly distributed along these two circular paths and complete a full lap in 20s. It is assumed that the car has knowledge of the future paths of the obstacles, allowing TC-SAC to precisely account for their movements. To ensure safety, a safety margin of 4m is maintained around the obstacles.

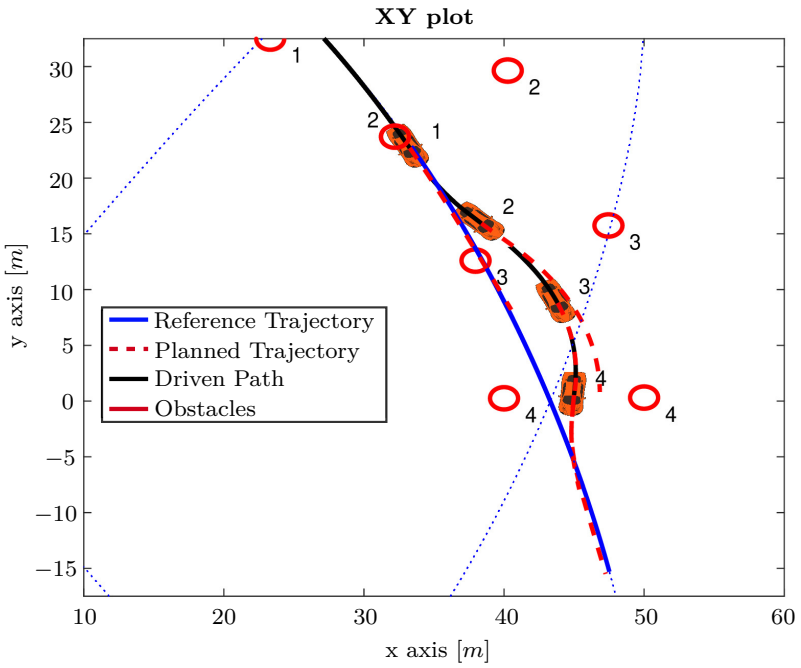


Figure 4.13: Dynamic obstacle avoidance. The position at different time steps is labeled by the number

Figure 4.13 provides a snapshot of the dynamic behavior between 7s and 10s, with each frame representing a 1s interval. It is evident that the car successfully navigates around the obstacles, ultimately returning to the reference trajectory. To further support the validity of the simulation results, a video demonstrating the scenario has been uploaded at `https://github.com/khoilsr/tcsac_trajectory_generation`. It is important to note that while this scenario may not perfectly reflect real-world situations, it encompasses a variety of obstacle avoidance scenarios, including obstacles approaching from different directions and consecutive obstacle avoidance situations. This serves as a proof of concept, showcasing the effectiveness of TC-SAC in efficiently solving optimal control problems in diverse applications.

4.6 Stability Analysis

It is crucial for any controller to have its stability verified and proven. Thus, in this Section, the stability of our control method TC-SAC is outlined and discussed. We consider the class of systems to be controlled by the following general nonlinear set of ODEs

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0 \tag{4.45}$$

with state vector $\mathbf{x}(t) \in \mathbb{R}^n$, input vector $\mathbf{u}(t) \in \mathbb{R}^m$. We also assume that

1. $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is twice continuously differentiable and $\mathbf{f}(0, 0) = 0$. Thus, $0 \in \mathbb{R}^n$ is an equilibrium of the system,
2. system (4.45) has a unique solution for any initial condition $\mathbf{x}_0 \in \mathbb{R}^n$ and any piecewise continuous $\mathbf{u}(\cdot) \in \mathbf{R}^m$.

For the sake of simplicity, we consider the stability of the system around the origin $\mathbf{x} = 0$. The stability of an arbitrary position is achieved similarly by shifting the state and control signals such that this arbitrary position becomes the origin. In addition, to keep the notation the same as in literature, the objective cost function is denoted as V, which is equivalent to the notation J_1 presented in previous Sections. Recall the OCP with target constraints

$$\min_{\mathbf{u}} \; V(\mathbf{x}, t; \mathbf{u}) = \int_0^T L(\mathbf{x}, \mathbf{u}, t)dt \tag{4.46a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \; \mathbf{x}(0) = \mathbf{x}_0, \tag{4.46b}$$

$$\mathbf{x}(T) = 0 \tag{4.46c}$$

where L has a quadratic form:

$$L(\mathbf{x}, \mathbf{u}, t) = \frac{1}{2}\left[\mathbf{x}^\mathsf{T}(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\mathsf{T}(t)\mathbf{R}\mathbf{u}(t)\right] \tag{4.47}$$

Here let $\mathbf{x}(\cdot; \mathbf{x}_0, 0)$ denote the corresponding trajectory of (4.45) with initial condition $\mathbf{x}(0) = \mathbf{x}_0$ and $\hat{\mathbf{u}}(\cdot; \mathbf{x}_0, 0)$ denote the optimal control sequence that minimizes the objective function $V(\mathbf{x}_0, 0; \mathbf{u})$. From the methodology point of view, TC-SAC is a combination of FOGA in the first step and the extended SAC in the consequence. Hence, an intuitive way to analyze the stability of TC-SAC is to establish the stability conditions for FOGA first, then the stability of TC-SAC can be concluded after that. Further more, TC-SAC uses the same concept of receding horizon as MPC, therefore, it is straightforward to derive the stability conditions of TC-SAC from the stability literature of MPC. In the following, we first look at the stability conditions of FOGA in the absence of the extended SAC in Section 4.6.1. Then we will discuss about the stability of TC-SAC in Section 4.6.2.

4.6.1 Stability of FOGA

There have been several works that investigate and deploy sufficient conditions for the closed-loop MPC system to be stable. An overview about most of the works in this area can be found in [MRRS00]. Specifically for our problem in (4.46), we are seeking for the stability conditions of a receding horizon control with terminal constraint for nonlinear continuous system. The constraint imposed at terminal time T provides a relatively simple procedure to establish the stability of the closed-loop system. In fact, Chen and Shaw [CS82] derived sufficient conditions for the closed-loop receding horizon control of (4.46) to be asymptotically stable. Such these conditions are described by the following assumptions:

Assumption 1. *There exists an optimal control function* $\hat{\mathbf{u}}(\cdot;\mathbf{x}_0,0)$ *which gives the minimal cost* $\hat{V}(\mathbf{x}_0,0;\mathbf{u})$ *and satisfies the terminal constraint* (4.46c).

Assumption 2. *The optimal cost* $\hat{V}(\mathbf{x}_0,0;\mathbf{u})$ *satisfies the following conditions for any* $T>0$

- $\hat{V}(\mathbf{x}_0,0;\mathbf{u})=0$ *and* $\hat{V}(\mathbf{x},t;\mathbf{u})>0$ *for* $\mathbf{x}\neq 0$
- $\hat{V}(\mathbf{x},t;\mathbf{u})\to\infty$ *when* $\|\mathbf{x}\|\to\infty$
- $\frac{\partial\hat{V}(\mathbf{x},t;\mathbf{u})}{\partial\mathbf{x}}$ *exists for any* $\mathbf{x}$

Theorem 1. *Suppose that Assumption 1 and 2 are satisfied, then for any fixed* $T>0$*, the closed-loop system is asymptotically stable at the origin.*

Proof. See the Appendix A. □

Assumption 2 can be fulfilled easily by having the cost function in quadratic form i.e.

$$V(\mathbf{x},t;\mathbf{u})=\frac{1}{2}\int_t^{t+T}\left(\|\mathbf{x}(\tau)\|^2_{\mathbf{Q}}+\|\mathbf{u}(\tau)\|^2_{\mathbf{R}}\right)d\tau \tag{4.48}$$

Assumption 1 is however more difficult to be satisfied. It requires the optimal solution of (4.46) in which the terminal constraint (4.46c) also has to be satisfied for each receding horizon. We will first prove that, there exists an optimal solution that satisfies the terminal constraint when using FOGA as the solver.

Recall the changes in the cost function J_1 and the terminal constraints $\boldsymbol{\Phi}$

$$\delta J_1=\int_{t_0}^{t_\mathrm{f}}\left(\boldsymbol{\rho}^\mathsf{T}\frac{\partial\mathbf{f}}{\partial\mathbf{u}}+\frac{\partial l_1}{\partial\mathbf{u}}\right)\delta\mathbf{u}(t)dt \tag{4.49}$$

$$\delta\boldsymbol{\Phi}\triangleq\begin{bmatrix}\delta x_1\\ \vdots\\ \delta x_q\end{bmatrix}_{t=t_\mathrm{f}}=\int_{t_0}^{t_\mathrm{f}}\mathbf{R}^\mathsf{T}\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\delta\mathbf{u}(t)dt \tag{4.50}$$

and adjoin (4.50) into (4.49), we obtain

$$\delta J_1+\boldsymbol{\nu}\delta\boldsymbol{\Phi}=\int_{t_0}^{t_\mathrm{f}}\left[\frac{\partial l_1}{\partial\mathbf{u}}+(\boldsymbol{\rho}+\mathbf{R}\boldsymbol{\nu})^\mathsf{T}\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\right]\delta\mathbf{u}(t)dt \tag{4.51}$$

where $\boldsymbol{\nu}$ is also a Lagrange multiplier. The equation (4.51) represents the addjoin cost of the OCP. With the solution of $\delta\mathbf{u}(t)$ in (4.15), we have

$$\delta J_1+\boldsymbol{\nu}\delta\boldsymbol{\Phi}=-\mathbf{W}^{-1}\int_{t_0}^{t_\mathrm{f}}\left\|\frac{\partial l_1}{\partial\mathbf{u}}+(\boldsymbol{\rho}+\mathbf{R}\boldsymbol{\nu})^\mathsf{T}\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\right\|^2dt<0 \tag{4.52}$$

which is negative unless the integrand vanishes over the whole integration interval. The Lagrange multiplier $\boldsymbol{\nu}$ is computed as

$$\boldsymbol{\nu}=-\mathbf{I}^{-1}_{\boldsymbol{\Phi}\boldsymbol{\Phi}}(\delta\boldsymbol{\Phi}+\mathbf{I}_{\boldsymbol{\Phi}\mathbf{J}}) \tag{4.53}$$

with $\delta\boldsymbol{\Phi}=-\epsilon\boldsymbol{\Phi}[\mathbf{x}(t_\mathrm{f})]$ and a constant $\epsilon\in(0,1]$. The choice of $\delta\boldsymbol{\Phi}$ guarantees that the constructed control signal $\mathbf{u}(t)$ drives the system close to the desired states at the final time t_f. Thus the terminal constraints can be satisfied after a finite amount of iterations. Since

the system is fully controllable, $\mathbf{I}_{\boldsymbol{\Phi}\boldsymbol{\Phi}}^{-1}$ exists thus there always exists $\boldsymbol{\nu}$ and $\delta\boldsymbol{u}$ such that the cost function is decreased and the terminal constraints are satisfied. As the optimal solution is approached and $\delta\boldsymbol{\Phi} = 0$, it is clear that

$$\boldsymbol{\nu} \to -\mathbf{I}_{\boldsymbol{\Phi}\boldsymbol{\Phi}}^{-1}\mathbf{I}_{\boldsymbol{\Phi}\mathbf{J}} \tag{4.54}$$

$$\frac{\partial l_1}{\partial \mathbf{u}} + (\boldsymbol{\rho} + \mathbf{R}\boldsymbol{\nu})^\top \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \to 0 \tag{4.55}$$

FOGA is therefore able to construct an optimal control signal $\mathbf{u}(t)$ that minimizes the cost and satisfies terminal constraints. Hence, Assumption 1 is satisfied.

A problem arises here is that Assumption 2 uses the optimal cost $\hat{V}(\mathbf{x}; T)$ as the Lyapunov function to establish the closed-loop stability. This requires FOGA to be performed repeatedly until the optimal solution is found, which is not practical in our case since we want to have fast computation for online capability. However, this optimal solution is actually not necessary. Indeed, we will show that, with a proper "warm start", the closed-loop system is still stable even the cost is not optimal.

Definition 1. (*Warm start*) *An admissible warm start* $\tilde{\mathbf{u}}$*, must steer the current state* $\mathbf{x}$ *to the origin, i.e. satisfy terminal constraint* $\mathbf{x}(T) = 0$.

A warm start needs to satisfy the constraints but does not have to be optimal, hence can be acquired much faster. Further more, warm start only need to be computed once and can be done offline. In our approach, warm start is achieved by performing FOGA a couple of iterations. The process can be speed up with a proper choice of ϵ to calculate the Lagrange multiplier $\boldsymbol{\nu}$ in (4.53). A controller algorithm using warm start is as follow

Controller algorithm with warm start

Data: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\delta \in (0, \infty)$ where δ is the sampling interval

Initialization: At time $t_0 = 0$, if $\boldsymbol{x}_0$ is at the origin, i.e. $\boldsymbol{x}_0 = \mathbf{0}$, meaning the system is already at the equilibrium then employ $\boldsymbol{u} = \mathbf{0}$ to maintain the current state. Else, perform FOGA a couple of iterations to compute a feasible warm start $\mathbf{u}_0$ for the OCP problem in (4.46). Apply the control $\mathbf{u}_0$ to the real system over the interval $[t_0, t_0 + \delta]$.

Repeat:

1. At any time t, if $\mathbf{x}(t) = \mathbf{0}$, meaning the system reaches the origin, employ $\boldsymbol{u} = \mathbf{0}$. Else:
2. At any time $t_i \triangleq i\delta, i \in \mathbb{N}$:
   - Obtain an admissible control $\boldsymbol{u}_i'$ as an initial guess

$$\boldsymbol{u}_i' = \begin{cases} \boldsymbol{u}_{i-1} & \text{for} \quad t \in [t_i, t_i + T - \delta] \\ \mathbf{0} & \text{for} \quad t \in [t_i + T - \delta, t_i + T] \end{cases} \tag{4.56}$$

   - Compute an admissible control horizon which is better than the preceding control horizon in the sense that

$$V(\boldsymbol{x}_i, t_i, \boldsymbol{u}_i) \leq V(\boldsymbol{x}_i, t_i, \boldsymbol{u}_i') \tag{4.57}$$

   - Apply the control $\boldsymbol{u}_i$ to the real system over the interval $[t_i, t_i + \delta]$

The stability proof of MPC with warm start is presented in [MM93]. With a choice of admissible warm start, the controller is asymptotically stable. The terminal constraints are satisfied from the beginning and the cost is improved over time. Even in the situation where the cost increases due to numerical errors, we simply just go back to the warm start of the previous iteration and continue from there.

4.6.2 Stability of TC-SAC

As mentioned, TC-SAC can be interpreted as a combination of FOGA in the first step and the extended SAC in the consequence. Since FOGA is proven to be stable in Section 4.6.1, we only need to confirm that the extended SAC in the second step does not violate the stability property of FOGA. In other words, we need to fulfill two conditions: 1) the performance cost (4.46a) is decreased when applying the extended SAC and 2) the terminal constraints (4.46c) is not violated.

It can be easily seen that both of the conditions are guaranteed naturally by the procedure of TC-SAC presented in Section 4.4. From the methodology point of view, TC-SAC only apply the control signal computed by the extended SAC if it results in a smaller value in both performance cost $V(\mathbf{x}, t; \mathbf{u})$ and terminal cost $l_c = \frac{1}{2}\boldsymbol{\Phi}^\mathsf{T}\mathbf{Q}_c\boldsymbol{\Phi}$, when comparing to FOGA. The reduction in $V(\mathbf{x}, t; \mathbf{u})$ guarantees that condition 1 is satisfied. Similarly, the reduction in terminal cost l_c means that the extended SAC drives the system closer to the origin at time T, thus fulfills condition 2. If any of the conditions is not met, the control signal computed by FOGA is used instead. In both cases, we assure that TC-SAC inherits the stability property of FOGA and therefore, we can conclude that TC-SAC is asymptotically stable.

4.7 Discussion

In this Chapter, we have presented the TC-SAC method as a fast and close to optimal control approach for trajectory generation in dynamic environments. The evaluation of the proposed method in various scenarios has highlighted its advantages in terms of computational efficiency, constraint handling, and real-time applicability.

One of the significant advantages of the TC-SAC method is its ability to generate near-optimal trajectories while maintaining fast computation times. By combining a first-order gradient approach with the sequential action control, TC-SAC achieves a balance between optimality and computational efficiency. The first-order gradient approach provides an initial guess for the control sequence, while the subsequent sequential action control step refines the trajectory based on the cost function. This combination allows TC-SAC to generate trajectories that closely approximate the optimal solution, making it suitable for real-time applications where quick and efficient trajectory planning is essential. The results obtained in this study demonstrate the remarkable performance of TC-SAC in fulfilling reaching/tracking tasks, even in challenging scenarios like the Lissajous trajectory. TC-SAC exhibits higher precision and faster computation times compared to other related methods, underscoring its potential for real-life applications and scenarios that demand optimal control input calculations.

Another advantage of TC-SAC is its capability to handle target constraints effectively. The method incorporates target constraints into the trajectory planning process, ensuring that the generated trajectories satisfy the specified constraints. This is a crucial feature in many practical scenarios where the motion of the robot needs to adhere to certain constraints, such as avoiding obstacles, maintaining safety margins, or achieving specific target positions. By considering these constraints during trajectory generation, TC-SAC provides a reliable solution that guarantees the adherence of the robot to the desired behavior while optimizing the performance criteria. The experimental results demonstrate the efficient obstacle avoidance capabilities of TC-SAC, indicating its ability to handle the highly dynamic nature of the environment.

Furthermore, TC-SAC demonstrates versatility in its application to different systems and tasks. The method has been successfully deployed in diverse contexts, including robotic arms

and car-like systems, showcasing its adaptability to different linear and nonlinear systems beyond the applications presented in this paper. This adaptability highlights the broad potential of TC-SAC in addressing trajectory planning and control challenges across different domains. Whether it is in robotic manipulation tasks, autonomous driving scenarios, or human-robot interaction settings, TC-SAC can be tailored to suit the specific requirements and constraints of the system under consideration.

Moreover, the stability proof presented in this work contributes to the robustness and reliability of the TC-SAC method, particularly in terms of safety considerations. The stability proof establishes that the TC-SAC controller guarantees asymptotic stability, ensuring that the state of the system converge to a desired equilibrium point or a desired trajectory over time. The combination of computational efficiency, constraint handling, versatility, and stability makes TC-SAC a promising trajectory planning and control algorithm that can enhance the safety and reliability of robots and other dynamic systems in trajectory tracking tasks.

It is important to note that TC-SAC does not explicitly handle inequality constraints like MPC does. This limitation means that TC-SAC may not be suitable for systems that heavily rely on strict inequality constraints. However, it is possible to incorporate these constraints as soft constraints in TC-SAC by introducing additional terms in the objective function that penalize violations of these constraints. This allows for some consideration of inequality constraints such as joint angles, velocity limits, or force/torque limits, although not as strictly enforced as in MPC. By balancing the penalty terms, the importance of these constraints can be adjusted in the optimization process, offering a flexible approach to handling inequality constraints in TC-SAC.

Although HRC scenarios have not been explicitly evaluated in the experiments conducted thus far, the evaluation of TC-SAC in this work suggests its high potential for integration and performance in HRC experiments. First and foremost, TC-SAC demonstrates its capability to accurately track the given trajectories with high precision. This is a crucial requirement in HRC scenarios where robots need to closely follow predefined trajectories while collaborating with human operators. Additionally, TC-SAC exhibits remarkable adaptability to dynamic changes in the environment and trajectory deviations. For example, in HRC settings where tasks may need to be switched or the environment may undergo unexpected changes, TC-SAC can update the trajectory input in the next iteration of the control sequence, ensuring smooth and responsive robot behavior. While the results presented in this work are based on simulations, we are confident in the transferability of TC-SAC to real systems and HRC scenarios due to its inherent potential. The real-time execution capability of TC-SAC is partially demonstrated in the experiments conducted with car-like robots, represented by a complex single-track model with 7 states and 2 control inputs. Notably, the computation time of TC-SAC is found to be smaller than the simulation time allocated for the experiment when an appropriate prediction horizon is chosen. This indicates that the computation time required for each step of TC-SAC is shorter than the duration of a single interval. It is important to note that the code used in these experiments has not been fully optimized for execution time, leaving room for further improvements. With proper implementation in a system-level programming language such as C, we anticipate that the computation time of TC-SAC can be further reduced. Thus, TC-SAC possesses the potential to be applied in real-time scenarios and integrated into the motion planning framework developed in Chapter 3, serving as the controller that executes given trajectories for the robot while being able to adapt to changes in tasks or environments on-the-fly.

5 Conclusion and Outlook

In this concluding chapter, we undertake a comprehensive exploration to synthesize the main findings, implications, and contributions of this thesis. We also discuss on the limitations of the research, discuss the broader implications of the findings, and outline possible directions for future research as well as address unresolved questions.

5.1 Summary and Discussion

In this thesis, our primary objective was to enhance human-robot collaboration in shared workspaces through the development of novel motion planning and control methods. We pursued three specific research directions to achieve this goal. Firstly, we devised a method to improve the responsiveness of the robot when operating in close proximity to human partners. By incorporating human motion prediction into the behavior of the robot, we enable it to anticipate the future movements of humans and adjust its behavior accordingly, ensuring a safe and efficient collaboration. Secondly, we introduced a new framework that leverages the feedback from human partners to generate predictable robot motion. This framework enables the robot to adapt its movements based on the intentions and actions of the human, fostering a more intuitive and coordinated collaboration. Moreover, we explored the transferability of the learned motion properties to new scenarios and tasks, allowing the robot to generalize its behavior and seamlessly adapt to different collaborative contexts. Lastly, we developed a fast and optimal-based controller that enables the robot to execute the generated trajectories precisely, even in highly dynamic environments. By efficiently solving the optimal control problem, the controller ensures that the movements of the robot align with the desired trajectories, enhancing its performance in complex collaborative tasks.

In Chapter 2, we proposed an innovative approach that integrated human motion prediction into the obstacle avoidance behavior of the robot to enhance its responsiveness and improve safety during close human-robot collaborations, particularly in confined workspaces. Our method utilized a minimum-jerk model for human motion prediction, and we developed an efficient online optimization procedure to solve the reverse problem of predicting the full trajectory based on a given set of current human motion. This enabled real-time implementation of the human prediction method, making it suitable for highly dynamic human-robot collaboration scenarios that require fast computation. The obstacle avoidance behavior was designed based on a potential field approach that created a repulsive force between the robot and human, leveraging the predicted human motion to effectively push the robot away when the human approached. Our extensive verification through real-world human-robot collaborative experiments demonstrated the efficacy of our approach, as the robot successfully avoided collisions with the human partner in all cases and exhibited greater responsiveness compared to scenarios without prediction. The results of our case study highlighted the potential of our method for enhancing human-robot collaboration, as the human partners reported feeling

more comfortable working with the robot. Furthermore, the simplicity of our method facilitated easy integration into different applications without requiring significant modifications. We also extended the applicability of our approach by integrating it into our developed framework in Chapter 3, which aimed to enhance safety and facilitate interaction between humans and robots. In this subsequent chapter, we adopted a slight deviation approach to human motion prediction by utilizing probabilistic movement primitives to represent individual-specific motion characteristics, allowing us to capture the variations in how different individuals perform tasks. Although we did not reiterate the results regarding the responsiveness and avoidance behaviors of the robot in Chapter 3, they were found to be similar to those reported in Chapter 2. Overall, our approach enables the robot to avoid potential collisions and maintain a safe distance from humans more effectively, reducing the risk of accidents or injuries during close interactions. The integration of human motion prediction also enhances the naturalness of the movements of the robot, as it can better adapt to the anticipated motions of humans, thereby fostering smoother and more intuitive human-robot collaboration.

Despite the strengths of our approach, such as its simplicity and suitability for online scenarios, there are limitations that hinder its applicability to a broader range of HRC scenarios. The human motion prediction approach employed in our work was specifically designed for point-to-point motions of the human arm, making it most effective for tasks that predominantly involve this type of motion, such as pick-and-place tasks. This limitation restricts the ability of our method to accurately predict a wide variety of human actions. For instance, when the human partner changes the manner in which they use their arm during collaboration or incorporates whole-body movements for task execution, the prediction strategy may become less effective and require further investigation to address these types of human actions. Additionally, while the prediction strategy can account for regular patterns, it may struggle to handle unexpected or erratic human movements. Unforeseen changes in human behavior or sudden deviations from anticipated trajectories can lead to incorrect predictions. Therefore, further research is needed to enhance the prediction capability of the robot in such cases. It is important to note that even when the prediction is not as effective, the robot can still rely on the potential field to avoid the human partner, thereby ensuring safety. However, the reduced responsiveness of the robot may diminish the comfort experienced by the human partner during the collaborative task.

Another important aspect to consider is the expanding scope of collaboration between humans and robots. Currently, our approach treats the crossing between humans and robots as interference that the robot needs to avoid to ensure the safety of the human partner. However, collaboration can extend beyond this notion. In real-life scenarios, there are situations where the human partner intentionally approaches the robot to engage in collaborative tasks. For instance, the human may approach the robot with the intention of teaching or guiding it through specific motions, or assisting the robot in handling tools or objects. In such cases, the robot should not actively avoid the human; instead, it should respond with corresponding actions to indicate its understanding of the human's intention and readiness to cooperate. Recognizing these intentional human motions and distinguishing them from previous interfering motions present an interesting challenge that merits further investigation. By developing methods to differentiate intentional collaborative gestures from unintentional interference, we can enhance the fluidity and effectiveness of human-robot collaboration in various scenarios.

In Chapter 3, we presented a comprehensive framework for generating predictable motion in human-robot collaboration (HRC) scenarios. Our framework addresses the challenge of improving the predictability of robot trajectories and enhancing the safety and comfort levels of human partners during close interactions. The advantage of our framework, which is our

main contribution, lies in the integration of learning and interaction into a unified approach. By incorporating the approach presented in Chapter 2, we ensure the safety of the human partner while enabling the robot to learn and adapt its motion based on the feedback received from the human partner. In our framework, we utilize DMPs to generate robot trajectories that can be modified through a set of parameters. These parameters play a crucial role in adapting the motion of the robot to be more predictable and natural to the human partner. By allowing different parameter settings, we can represent a variety of trajectories while maintaining the same start and goal positions. To optimize these parameters, we introduce a learning process that involves direct collaboration between the robot and the human partner. The human partner's perception of the robots trajectory is evaluated using a cost function that considers predictability and smoothness. Through an iterative process, the framework updates the DMP parameters based on the human partner's feedback, favoring trajectories that result in lower cost. This adaptive learning process enables the robot to gradually converge towards more predictable and human-friendly motions. One of the key contributions of our work is the development of a task generalization method within the framework. This method allows the policies learned from specific tasks to be transferred and applied to new tasks. By leveraging the knowledge gained from previous tasks, the robot can accelerate the convergence process and reduce the training time required for new tasks. This aspect of our framework demonstrates its versatility and scalability, as it enables the robot to quickly adapt to new collaborative scenarios without extensive retraining. We conducted extensive experiments in both VR simulations and real-world HRC scenarios to validate the effectiveness of our framework. In the VR experiments, we observed significant improvements in the predictability of trajectories of the robot over time. The ability to generate more predictable motions enhanced the overall collaboration experience and fostered a sense of trust and comfort in the human partners. Additionally, we compared our method with a non-adaptive robot that performs the same motion repeatedly for each task. The comparison highlights that the predictability of the motions improves due to the adaptability of the robot and not solely due to the human's adaptation. Furthermore, we implemented our framework on a physical *KUKA* 7-DOF robot in a real HRC scenario. The results from these experiments reinforced the findings from the VR simulations, demonstrating that the trajectories of the robot remained predictable over time. Importantly, we conducted a case study to assess the safety and comfort levels experienced by the human partner when collaborating with the physical robot. The feedback received from the human partners indicated a positive perception of safety and comfort during the collaborative tasks, further affirming the effectiveness of our framework in real-world applications.

While our framework has demonstrated promising results, there are areas that warrant further investigation.Firstly, it is important to expand the range of tasks and scenarios in which the framework is evaluated. The current experiments focused on a limited set of tasks, which provided a simple and intuitive environment for the human partner to predict the motions of the robot. However, to fully assess the capabilities and limitations of the framework, it would be valuable to explore more complex and diverse tasks. This would allow for a comprehensive understanding of how the predictability of robot motions varies across different types of tasks, such as reaching, grasping, or object manipulation. Additionally, the mounting position of our setup, with the base on the table, resulted in a robot shape that does not exactly mimic the movement patterns of a human arm. While the framework successfully incorporated human feedback and generated predictable robot motions, it would be interesting to investigate how the framework performs when the configuration of the robot more closely resembles that of a human arm. For example, conducting experiments with the

base of the robot suspended above, allowing for elbow movements, could provide insights into the impact of additional degrees of freedom on the predictability of robot motions. This configuration would better capture the natural movement characteristics of a human arm and could potentially improve the human partner's ability to anticipate and interact with the robot. Furthermore, investigating how the task generation method can be extended in this scenario would provide valuable insights into adapting the framework to different robot designs and configurations, opening up new possibilities for motion policy adaptation and transfer.

In Chapter 4, we proposed the TC-SAC approach, which offers a fast and close to optimal motion control solution for generating and executing robot motion. TC-SAC addresses the need for adaptive and responsive control in dynamic environments by combining the receding horizon concept of MPC with the analytical solution provided by the SAC method. By leveraging the analytical solution of SAC, TC-SAC achieves fast computation times compared to traditional optimal control approaches. This computational efficiency is crucial for real-time applications where quick decision-making is essential. Furthermore, TC-SAC integrates the ability to handle target constraints, enabling the robot to perform given tasks accurately. Our TC-SAC approach consists of two main components: the FOGA controller and the SAC-based optimal controller. The FOGA controller takes into account the target constraints and computes an initial control input. This step ensures that the motion of the robot adheres to the specified targets. The optimal controller, based on SAC, further refines the control input obtained from FOGA. The combination of these two controllers allows TC-SAC to generate trajectories that closely approximate the optimal solution while satisfying the target constraints. We conducted simulations with a 2-DOF robot and a car-like system to evaluate the performance of TC-SAC. The results demonstrate the efficacy of TC-SAC in trajectory tracking and obstacle avoidance tasks. The robot successfully achieves high accuracy in following reference trajectories while effectively avoiding both static and dynamic obstacles. The computational time of TC-SAC was found to be faster than FOGA, demonstrating its efficiency in generating control inputs. Although slightly slower than SAC, the computational time difference is negligible considering the advantage of TC-SAC in reaching arbitrary positions. Additionally, TC-SAC offers stability guarantees. We provided a complete stability proof for TC-SAC, ensuring that the controller leads to asymptotic stability, where the state of the system converges to a desired equilibrium or trajectory over time. This stability proof enhances the reliability and robustness of TC-SAC, providing confidence in its application in various motion control scenarios.

As a continuation of this work, one potential direction for further improvement is the optimization of the prediction horizon used in TC-SAC. The prediction horizon determines the time window over which the optimal control problem is solved. Currently, TC-SAC employs a fixed prediction horizon based on trial and error. However, different scenarios may benefit from varying prediction horizons. For example, in dynamic environments or systems with rapidly changing dynamics, a shorter prediction horizon may be necessary for quick responsiveness. Conversely, in more predictable and stable environments, a longer prediction horizon may yield more optimal trajectories. Future research could explore adaptive or online methods to dynamically adjust the prediction horizon based on the real-time context, optimizing the trade-off between computation time and optimality. Alternatively, learning-based approaches could be employed to determine the most suitable prediction horizon for TC-SAC, taking into account the specific characteristics of the system and the environment.

Furthermore, there are opportunities to explore parallelization, approximation methods, or model reduction techniques to address the computational complexity of TC-SAC for nonlinear

and high-dimensional systems. These techniques can help reduce the computational burden and enable real-time implementation of TC-SAC for more complex systems. Additionally, it would be valuable to investigate the impact of model inaccuracies and mismatches between the simulated model and the real system on the performance of TC-SAC. Understanding the effects of model discrepancies on controller performance and exploring methods to mitigate these effects would contribute to the robustness and applicability of TC-SAC in practical settings. Moreover, investigating the robustness of TC-SAC in the presence of uncertainties, noise, and disturbances would provide insights into its performance under realistic operating conditions.

Overall, this thesis has contributed to the body of knowledge on motion planning and control for human-robot collaboration. The developed methods and frameworks offer valuable tools for enhancing safety, responsiveness, and adaptability in shared workspaces. By addressing current challenges and identifying future research directions, this work lays the foundation for further advancements in human-robot collaboration, ultimately promoting the successful integration of robots into various collaborative scenarios.

5.2 Outlook

In our approach to integrate human motion prediction into local obstacle avoidance, we explored two methods, namely the minimum-jerk model and probabilistic motion primitives, to predict human motion. While these methods showed promising results, there are opportunities for improvement and expansion. More complex arm models, such as the minimum commanded torque change model [NIO+99] and minimum variance curvature [FS11], could be studied and integrated to enhance prediction accuracy and increase the flexibility of human motion. Additionally, we can investigate modeling assembly tasks as sequences of human motions and using hidden Markov models for prediction [LW17]. Moreover, predicting human intentions to expand collaboration contexts is an important topic that warrants further study. Approaches like the teaching-learning-prediction model [WLCJ18] and human intention-driven learning [RTD19] can be explored to verify their capability, especially in online HRC scenarios.

In the predictable motion generation framework, an intriguing area to explore is how the relative perspective of the human and robot affects the learned predictable trajectories. Considering that the human's position is typically not fixed in HRC scenarios, understanding this aspect can lead to more adaptive and intuitive robot behavior. For collaborative and complex tasks, utilizing data-driven imitation learning systems [VSG+17] to enable the robot to learn from human-human demonstrations presents an exciting avenue for investigation. The use of DMPs as trajectory generation [Ior23] for these tasks is also worth exploring. Additionally, familiarizing the human with the robot to improve the learning process [DS14] can further enhance the collaboration experience, as humans also learn and adapt during interactions. The task generalization method requires further investigation to identify relevant features and their influence on robot trajectories, enabling the framework to handle a wider range of tasks [SCGHRA22, LLLK20].

For our proposed TC-SAC controller to be applicable in real-time applications, optimizations at the system programming level, such as C programming, are necessary to ensure fast computation. The selection of the prediction horizon should be carefully considered to achieve good results in terms of optimality without sacrificing computational efficiency. Leveraging learning-based approaches [HWMZ20] can be explored to fine-tune the prediction horizon. For complex systems with high degrees of freedom or humanoid robots, simplifying the model

[SSAP12, WLA14] may be necessary to reduce computation time while maintaining adequate control performance. In cases with unknown parameters, an identification procedure [JGJ14, SAS17] should be performed first to estimate the system parameters accurately. Furthermore, investigating the robustness of TC-SAC in the presence of noise, disturbances, and inaccuracies between real system models and simulated ones [GLNR18, AH20] is essential to ensure reliable performance in dynamic and uncertain environments.

A Appendix

Vehicle Dynamic Model

For car-like systems, a wide range of model complexities exists. One of the most complex representations of a car-like system is the multi-body model which can very accurately describe the overall physical behavior of the car [AKM17]. However, this level of complexity does not suite the requirements for online application due to the excessive computational load. The model presented in this work is a simplified single track model [MDRMAF16], which is commonly used in studies of different controllers/methods related to car-like systems due to its simplicity and accurate representation of the dynamic behaviors of a car. The state-space model is as followed:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Psi}_c \\ \dot{v} \\ \dot{\alpha} \\ \ddot{\Psi}_c \end{bmatrix} = \begin{bmatrix} v\cos(\alpha + \Psi_c) \\ v\sin(\alpha + \Psi_c) \\ \dot{\Psi}_c \\ a_{\text{ave}} \\ s_{\text{ave}} \\ \omega_{\text{ave}} \end{bmatrix} \tag{A.1}$$

with the x and y-coordinates X and Y, the orientation Ψ_c, the velocity v, the side slip angle α and the change in orientation $\dot{\Psi}_c$. The control input is given by $\mathbf{u} = \begin{bmatrix} \delta \\ M \end{bmatrix}$, with the steering angle δ and the applied torque M. a_{ave}, s_{ave} and ω_{ave} are given by:

$$\begin{aligned} a_{\text{ave}} = \frac{1}{m}(&F_{f,X}\cos(\alpha - \delta) + F_{r,X}\cos(\alpha) + F_{f,Y}\sin(\alpha - \delta) \\ &+ F_{r,Y}\sin(\alpha) - 0.5 c_w \rho_L A v^2) \end{aligned} \tag{A.2}$$

$$\begin{aligned} s_{\text{ave}} = \frac{1}{mv}(&-F_{f,X}\sin(\alpha - \delta) - F_{r,X}\sin(\alpha) \\ &+ F_{f,Y}\cos(\alpha - \delta) + F_{r,Y}\cos(\alpha) - mv\dot{\Psi}_c) \end{aligned} \tag{A.3}$$

$$\omega_{\text{ave}} = \frac{F_{f,X} l_f \sin(\delta) + F_{f,Y} l_f \cos(\delta) - F_{r,Y} l_r}{\Theta_Z} \tag{A.4}$$

with the drag coefficient c_w, the density of the air ρ_L and the front surface A of the car and the inertia Θ_Z of the car. $F_{f/r,Y}$ in (A.2) - (A.4) are the lateral tire forces obtained by using the Pacejka's magic tire formula

$$F_{f/r,Y} = D_Y \sin\left(C_Y \operatorname{atan}\left(B_Y \alpha_{f/r} - E_Y\left(B_Y \alpha_{f/r} - \operatorname{atan}\left(B_Y \alpha_{f/r}\right)\right)\right)\right) \tag{A.5}$$

with $B_Y = \frac{K_Y F_{f/r,Z}}{C_Y D_Y}$, $D_Y = \mu_Y F_{f/r,Z}$ and C_Y, E_Y, K_Y are the constants. μ_Y is the friction coefficient in lateral direction, $F_{f/r,Z}$ is the tire load on the front and rear axis and $\alpha_{f/r}$ is

the side slip angle on the front and rear axis. $F_{f/r,Z}$ is given by

$$F_{f/r,Z} = m_c g \frac{l_{r/f}}{l_f + l_r} \tag{A.6}$$

For the longitudinal forces $F_{f,X}$ in (A.2) - (A.4), they are simplified as

$$F_{f,X} = F_{r,X} = \frac{M}{r} \tag{A.7}$$

with the torque M as the control input for the system and r as the radius of the tires.

Model Modification: The single track model presented above is not linearly with respect to δ, the steering angle, where it is used in trigonometrical functions as in (A.4). Since TC-SAC requires the system to be in control-affine form as described in (4.4), a simple way to correct it is to change the steering angle from a control input to a state and introduce the derivative of it $\dot{\delta}_{\text{input}}$ as the new control input.
After these changes are applied, the state space model changes to:

$$\mathbf{x} = \begin{bmatrix} X \\ Y \\ \Psi_c \\ v \\ \alpha \\ \dot{\Psi}_c \\ \delta \end{bmatrix}; \dot{\mathbf{x}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Psi}_c \\ \dot{v} \\ \dot{\alpha} \\ \ddot{\Psi}_c \\ \dot{\delta} \end{bmatrix} \tag{A.8}$$

with the control input

$$\mathbf{u} = \begin{bmatrix} \dot{\delta}_{\text{input}} \\ M \end{bmatrix} \tag{A.9}$$

with

$$\dot{\delta} = \dot{\delta}_{\text{input}} \tag{A.10}$$

and the steering angle δ still can be controlled directly. All other equations of the state-space model remain the same.

With these changes, $\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{u})$ can be written in a control-affine form with:

$$\mathbf{h}(t, \mathbf{x}(t)) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{\cos(\alpha)+\cos(\alpha-\delta)}{mr} \\ 0 & \frac{-\sin(\alpha)+\sin(\alpha-\delta)}{mrv} \\ 0 & \frac{l_f \sin(\delta)}{r\Theta} \\ 1 & 0 \end{bmatrix} \tag{A.11}$$

and

$$\mathbf{g}(t, \mathbf{x}(t)) = \begin{bmatrix} v\cos(\Psi_c + \alpha) \\ \vdots \\ 0 \end{bmatrix} \tag{A.12}$$

Although the input $\dot{\delta}$ serves mainly for the reason of control-affine form, it also improves the smoothness of the steering angle i.e. a jump in the derivative results into a linear change in the signal. This makes the controller become more realistic when applying on real system, since the steering angle will be applied, not the derivative of it. However, the drawback of the modification is, since the steering angle is not a control input, it cannot be saturated afterwards. In detail, it cannot be limited to realistic values,i.e. $35° - 45°$. To overcome this problem, u_δ is changed as follows:

$$\dot{\delta} = u_\delta - k\delta \tag{A.13}$$

with the constant $k \in \mathbb{R}^+$. Note that now u_δ has no direct physical meaning anymore. Through this change, $\mathbf{h}(t, \mathbf{x}(t))$ is not affected, while $\mathbf{g}(t, \mathbf{x}(t))$ changes to:

$$\mathbf{g}(t, \mathbf{x}(t)) = \begin{bmatrix} v\cos(\Psi_c + \alpha) \\ \vdots \\ -k \end{bmatrix} \tag{A.14}$$

and the control input $\mathbf{u}$ to:

$$\mathbf{u} = \begin{bmatrix} u_\delta \\ M \end{bmatrix} \tag{A.15}$$

For the maximum and minimum steering angle $\delta_{\max} = -\delta_{\min}$ holds, as for the maximum and minimum control input $u_{\delta,\max} = -u_{\delta,\min}$ holds. k therefore, has to be chosen depending on the desired maximum steering angle $\delta_{\max}$ and the maximum control input $u_{\delta,\max}$ for it. So k is determined with:

$$k = \frac{u_{\delta,\max}}{\delta_{\max}} \tag{A.16}$$

Proof of Theorem 1

Here we summarize the proof given in [MM90] for the case of nonlinear systems with terminal constraints. The idea is to show that the optimal cost function $\hat{V}(\mathbf{x}, t; \mathbf{u})$ can be used as a Lyapunov function for the receding horizon control. Let $\mathbf{x}^*$ and $\mathbf{u}^*$ denote the receding horizon strategy when the initial state is $\mathbf{x}_0$ at $t = 0$. We wish to evaluate $(d/dt)\hat{V}(\mathbf{x}^*(t))$ at an arbitrary, fixed instant of time. By definition,

$$\begin{aligned}\hat{V}(\mathbf{x}^*(t)) = \frac{1}{2}\int_t^{t+\Delta t} \left[\hat{\mathbf{x}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{x}}(\tau) + \hat{\mathbf{u}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{u}}(\tau)\right] d\tau \\ + \frac{1}{2}\int_{t+\Delta t}^{t+T} \left[\hat{\mathbf{x}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{x}}(\tau) + \hat{\mathbf{u}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{u}}(\tau)\right] d\tau\end{aligned} \tag{A.17}$$

where $\hat{\mathbf{x}}(\tau) \triangleq \hat{\mathbf{x}}(\tau; \mathbf{x}^*(t), t)$ and $\hat{\mathbf{u}}(\tau) \triangleq \hat{\mathbf{u}}(\tau; \mathbf{x}^*(t), t)$ are the optimal solution for the OCP in (4.46). Consider a control $\tilde{\mathbf{u}} : [t + \Delta t, t + T + \Delta t]$ defined as follows:

$$\tilde{\mathbf{u}}(\tau) \triangleq \begin{cases} \hat{\mathbf{u}}(\tau; \mathbf{x}^*(t), t) & \text{for} \quad \tau \in [t + \Delta t, t + T] \\ 0 & \text{for} \quad \tau \in (t + T, t + T + \Delta t]\,. \end{cases} \tag{A.18}$$

Let $\tilde{\mathbf{x}}(\cdot) = \tilde{\mathbf{x}}(\cdot; \hat{\mathbf{x}}(t + \Delta t), t + \Delta t)$ denote the corresponding trajectory with initial condition $\tilde{\mathbf{x}}(t + \Delta t) = \hat{\mathbf{x}}(t + \Delta t; \mathbf{x}^*(t), t)$. Clearly,

$$\tilde{\mathbf{x}}(\tau) = \begin{cases} \tilde{\mathbf{x}}(\tau; \mathbf{x}^*(t), t) & \text{for} \quad \tau \in [t + \Delta t, t + T] \\ 0 & \text{for} \quad \tau \in (t + T, t + T + \Delta t) \end{cases} \tag{A.19}$$

because $\tilde{\mathbf{x}}(t+T;\hat{\mathbf{x}}(t+\Delta t),t+\Delta t)=\hat{\mathbf{x}}(t+T;\mathbf{x}^*(t),t)=0$ and $\tilde{\mathbf{u}}=0$ for $\tau>t+T$. Since $\tilde{\mathbf{u}}$ is not necessarily optimal, it follows that

$$\begin{aligned}\hat{V}(\mathbf{x}^*(t)) &= \frac{1}{2}\int_t^{t+\Delta t}\left[\hat{\mathbf{x}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{x}}(\tau)+\hat{\mathbf{u}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{u}}(\tau)\right]d\tau + V(\hat{\mathbf{x}}(t+\Delta t),t+\Delta t;\tilde{\mathbf{u}}) \\ &\geq \frac{1}{2}\int_t^{t+\Delta t}\left[\hat{\mathbf{x}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{x}}(\tau)+\hat{\mathbf{u}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{u}}(\tau)\right]d\tau + \hat{V}(\hat{\mathbf{x}}(t+\Delta t)) \quad \text{(A.20)}\end{aligned}$$

so that

$$\hat{V}(\hat{\mathbf{x}}(t+\Delta t))-\hat{V}(\mathbf{x}^*(t)) \leq -\frac{1}{2}\int_t^{t+\Delta t}\left[\hat{\mathbf{x}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{x}}(\tau)+\hat{\mathbf{u}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{u}}(\tau)\right]d\tau \quad \text{(A.21)}$$

Since $\hat{V}$ is continuously differentiable, it follows from the Mean Value Theorem that

$$\begin{aligned}\frac{\hat{V}(\hat{\mathbf{x}}(t+\Delta t))-\hat{V}(\mathbf{x}^*(t))}{\Delta t} &= \nabla_{\mathbf{x}}\hat{V}(\mathbf{x}^*(t) \\ &\quad + \theta(\Delta t)(\hat{\mathbf{x}}(t+\Delta t)-\mathbf{x}^*(t)))\frac{(\hat{\mathbf{x}}(t+\Delta t)-\mathbf{x}^*(t))}{\Delta t} \quad \text{(A.22)}\end{aligned}$$

for $\theta(\Delta t)\in(0,1)$. Since

$$\mathbf{x}^*(t)=\hat{\mathbf{x}}(t;\mathbf{x}^*(t),t)=\hat{\mathbf{x}}(t),\mathbf{u}^*(t)=\hat{\mathbf{u}}(t;\mathbf{u}^*(t),t)=\hat{\mathbf{u}}(t) \quad \text{(A.23)}$$

and $\hat{\mathbf{u}}$ is continuous at t, we have that

$$\lim_{\Delta t\to 0_+}\frac{\hat{\mathbf{x}}(t+\Delta t)-\mathbf{x}^*(t)}{\Delta t}=f(\hat{\mathbf{x}}(t),\hat{\mathbf{u}}(t))=f(\mathbf{x}^*(t),\mathbf{u}^*(t)). \quad \text{(A.24)}$$

Since $\nabla_{\mathbf{x}}\hat{V}$ and $\mathbf{x}$ are continuous, then from (A.20), (A.23) and (A.24), it follows that

$$\lim_{\Delta t\to 0_+}\frac{\hat{V}(\hat{\mathbf{x}}(t+\Delta t))-\hat{V}(\mathbf{x}^*(t))}{\Delta t}=\nabla_{\mathbf{x}}\hat{V}(\mathbf{x}^*(t))f(\mathbf{x}^*(t),\mathbf{u}^*(t)). \quad \text{(A.25)}$$

By continuity of $\hat{\mathbf{x}}$ and $\hat{\mathbf{u}}$ and the Mean Value Theorem for integrals,we also have

$$\begin{aligned}&\lim_{\Delta t\to 0_+} -\frac{1}{2\Delta t}\int_t^{t+\Delta t}\left[\hat{\mathbf{x}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{x}}(\tau)+\hat{\mathbf{u}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{u}}(\tau)\right]d\tau \\ &\qquad \leq \lim_{\Delta t\to 0_+} -\frac{1}{2\Delta t}\int_t^{t+\Delta t}\left[\hat{\mathbf{x}}^\mathsf{T}(\tau)\mathbf{Q}\hat{\mathbf{x}}(\tau)\right]d\tau = -\frac{1}{2}\left[\mathbf{x}^*(t)^\mathsf{T}\mathbf{Q}\mathbf{x}^*(t)\right]. \quad \text{(A.26)}\end{aligned}$$

Dividing both sides of (A.21) by $\Delta t>0$ and taking the limit as $\Delta t\to 0_+$ yields

$$\nabla_{\mathbf{x}}\hat{V}(\mathbf{x}^*(t))f(\mathbf{x}^*(t),\mathbf{u}^*(t)) \leq -\frac{1}{2}\left[\mathbf{x}^*(t)^\mathsf{T}\mathbf{Q}\mathbf{x}^*(t)\right]<0. \quad \text{(A.27)}$$

Hence,

$$(d/dt)\hat{V}(\mathbf{x}^*(t))=\nabla_{\mathbf{x}}\hat{V}(\mathbf{x}^*(t))f(\mathbf{x}^*(t),\mathbf{u}^*(t))<0 \quad \text{(A.28)}$$

unless $\mathbf{x}^*(t)=0$. Hence, the system is asymptotically stable.

References

[AAG+19] P. Aivaliotis, S. Aivaliotis, C. Gkournelos, K. Kokkalis, G. Michalos, and S. Makris. Power and force limiting on industrial robots for human-robot collaboration. *Robotics and Computer-Integrated Manufacturing*, 59:346–360, 2019. doi:10.1016/j.rcim.2019.05.001.

[AC99] J. K. Aggarwal and Q. Cai. Human motion analysis: A Review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999. doi:10.1006/cviu.1998.0744.

[ACM+05] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila. Task planning for human-robot interaction. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*, pages 81–85, 2005. doi:10.1145/1107548.1107574.

[ADRVM13] F. Abu-Dakka, F. Rubio, F. Valero, and V. Mata. Evolutionary indirect approach to solving trajectory planning problem for industrial robots operating in workspaces with obstacles. *European Journal of Mechanics - A/Solids*, 42, 2013. doi:10.1016/j.euromechsol.2013.05.007.

[AH20] A. Altan and R. Hacioglu. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. *Mechanical Systems and Signal Processing*, 138:106548, 2020. doi:10.1016/j.ymssp.2019.106548.

[AKM17] M. Althoff, M. Koschi, and S. Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–726, June 2017. doi:10.1109/IVS.2017.7995802.

[ALB16] S. Apostolopoulos, M. Leibold, and M. Buss. Online motion planning over uneven terrain with walking primitives and regression. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3799–3805, May 2016. doi:10.1109/ICRA.2016.7487568.

[All05] F. Allgöwer. Nonlinear model predictive control. *Control Theory and Applications, IEE Proceedings* -, 152:257–258, June 2005. doi:10.1049/ip-cta:20059060.

[AM16] A. R. Ansari and T. D. Murphey. Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems. *IEEE Transactions on Robotics*, 32(5):1196–1214, 2016. doi:10.1109/TRO.2016.2596768.

[AS17] A. Asma and B. Sadok. Dynamic distributed pso joints elites in multiple robot path planning systems: theoretical and practical review of new ideas. *Procedia Computer Science*, 112:1082–1091, 2017. doi:10.1016/j.procs.2017.08.128.

[BBB13] D. Bortot, M. Born, and K. Bengler. Directly or on detours? how should industrial robots approximate humans? In *2013 8th ACM/IEEE International*

Conference on Human-Robot Interaction (HRI), pages 89–90, 2013. doi:10.1109/HRI.2013.6483515.

[BBSF14] A. Byravan, B. Boots, S. S. Srinivasa, and D. Fox. Space-time functional gradient optimization for motion planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6499–6506, 2014. doi:10.1109/ICRA.2014.6907818.

[BEL17] F. Z. Baghli, L. Elbakkali, and Y. Lakhal. Optimization of arm manipulator trajectory planning in the presence of obstacles by ant colony algorithm. *Procedia Engineering*, 181:560–567, 2017. doi:10.1016/j.proeng.2017.02.434.

[BGLS17] B. Busch, J. Grizou, M. Lopes, and F. Stulp. Learning legible motion from human-robot interactions. *International Journal of Social Robotics*, 9(5):765–779, 2017. doi:10.1007/s12369-017-0400-4.

[BHFK+19] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin. A scalable framework for real-time multi-robot, multi-human collision avoidance. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 936–943, 2019. doi:10.1109/ICRA.2019.8794457.

[BHS79] A. Bryson, Y. C. Ho, and G. Siouris. Applied optimal control: Optimization, estimation, and control. *Systems, Man and Cybernetics, IEEE Transactions on*, 9:366–367, 07 1979. doi:10.1109/TSMC.1979.4310229.

[BK02] O. Brock and O. Khatib. Elastic Strips: A framework for motion generation in human environments. *The International Journal of Robotics Research*, 21(12):1031–1052, 2002. doi:10.1177/027836490202101200 2.

[BLOD17] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan. Learning robot objectives from physical human interaction. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78, pages 217–226, November 2017. URL: https://proceedings.mlr.press/v78/bajcsy17a.html.

[BOvdS99] V. Boor, M. H. Overmars, and A. F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proceedings 1999 IEEE International Conference on Robotics and Automation*, volume 2, pages 1018–1023, 1999. doi:10.1109/ROBOT.1999.772447.

[BPC08] A. Bicchi, M. A. Peshkin, and J. E. Colgate. *Safety for Physical Human–Robot Interaction*, pages 1335–1348. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-30301-5_58.

[BWB08] A. Bauer, D. Wollherr, and M. Buss. Human-robot collaboration: a survey. *I. J. Humanoid Robotics*, 5:47–66, 03 2008. doi:10.1142/S0219843608001303.

[CEA16] Z. Chen, N. L. Elyaaqoubi, and G. Abba. Optimized 3d stable walking of a bipedal robot with line-shaped massless feet and sagittal underactuation. *Robotics and Autonomous Systems*, 83:203–213, 2016. doi:10.1016/j.robot.2016.05.003.

[CMW+17] H. Chu, L. Ma, K. Wang, Z. Shao, and Z. Song. Trajectory optimization for lunar soft landing with complex constraints. *Advances in Space Research*, 60(9):2060–2076, 2017. doi:10.1016/j.asr.2017.07.024.

[CR87] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *28th Annual Symposium on Foundations of Computer Science*, pages 49–60, 1987. doi:10.1109/SFCS.1987.42.

[CRR91] J. Canny, A. Rege, and J. Reif. An exact algorithm for kinodynamic planning in the plane. *Discrete & computational geometry*, 6(5):461–484, 1991. doi:10.1007/BF02574702.

[CS82] C. C. Chen and L. Shaw. On receding horizon feedback control. *Automatica*, 18(3):349–352, 1982. doi:10.1016/0005-1098(82)90096-6.

[CS18] J. H. Chen and K. T. Song. Collision-free motion planning for human-robot collaborative safety under cartesian constraint. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4348–4354, 2018. doi:10.1109/ICRA.2018.8460185.

[CSZT16] H. Cao, S. Sun, K. Zhang, and Z. Tang. Visualized trajectory planning of flexible redundant robotic arm using a novel hybrid algorithm. *Optik*, 127(20):9974–9983, 2016. doi:10.1016/j.ijleo.2016.07.078.

[DBDW07] M. Diehl, H. Bock, H. Diedam, and P. B. Wieber. Fast direct multiple shooting algorithms for optimal robot control. *Lecture Notes in Control and Information Sciences*, 340, July 2007. doi:10.1007/978-3-540-36119-0_4.

[DBP16] P. K. Das, H. S. Behera, and B. K. Panigrahi. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm and Evolutionary Computation*, 28:14–28, 2016. doi:10.1016/j.swevo.2015.10.011.

[Dij59] E. W. Dijkstra. A note on two problems in connection with graphs. *Numer. Math.*, 1(1):269–271, December 1959. doi:10.1007/BF01386390.

[DLF12] A. De Luca and F. Flacco. Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration. In *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 288–295, 2012. doi:10.1109/BioRob.2012.6290917.

[DLS13] A. D. Dragan, K. C. T. Lee, and S. Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308, 2013. doi:10.1109/HRI.2013.6483603.

[DMDB16] J. Dong, M. Mukadam, F. Dellaert, and B. Boots. Motion planning as probabilistic inference using gaussian processes and factor graphs. In *Robotics: Science and Systems 2016*, 06 2016. doi:10.15607/RSS.2016.XII.001.

[Dra15] A. D. Dragan. *Legible robot motion planning.* PhD thesis, Carnegie Mellon University, 2015.

[DS14] A. Dragan and S. Srinivasa. Familiarization to robot motion. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, pages 366–373, 2014. doi:10.1145/2559636.2559674.

[DSAC11] F. Dehais, E. A. Sisbot, R. Alami, and M. Causse. Physiological and subjective evaluation of a human–robot object hand-over task. *Applied Ergonomics*, 42(6):785 – 791, 2011. doi:10.1016/j.apergo.2010.12.005.

[DWK+05] K. Dautenhahn, S. Woods, C. Kaouri, M. L. Walters, K. L. Koay, and I. Werry. What is a robot companion - friend, assistant or butler? In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1192–1197, Aug 2005. doi:10.1109/IROS.2005.1545189.

[EMA17] M.Y.B. Elshabasy, K. T. Mohamed, and A. A. Ata. Power optimization of planar redundant manipulator moving along constrained-end trajectory using hybrid techniques. *Alexandria Engineering Journal*, 56(4):439–447, 2017. doi:10.1016/j.aej.2017.01.040.

[EMCB07] W. Erlhagen, A. Mukovskiy, F. Chersi, and E. Bicho. On the development of intention understanding for joint action tasks. In *2007 IEEE 6th International Conference on Development and Learning*, pages 140–145, 2007. doi:10.1109/DEVLRN.2007.4354022.

[EWA06] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-time optimization for switched-mode dynamical systems. *IEEE Transactions on Automatic Control*, 51(1):110–115, Jan 2006. doi:10.1109/TAC.2005.861711.

[EZMLU19] S. El Zaatari, M. Marei, W. Li, and Z. Usman. Cobot programming for collaborative industrial tasks: An overview. *Robot. Auton. Syst.*, 116(C):162–180, June 2019. doi:10.1016/j.robot.2019.03.003.

[FAMJ15] S. A. Fadzli, S. I. Abdulkadir, M. Makhtar, and A. A. Jamal. Robotic indoor path planning using dijkstra's algorithm with multi-layer dictionaries. In *2015 2nd International Conference on Information Science and Security (ICISS)*, pages 1–4, 2015. doi:10.1109/ICISSEC.2015.7371031.

[FE16] C. Feller and C. Ebenbauer. A stabilizing iteration scheme for model predictive control based on relaxed barrier functions. *Automatica*, 80, March 2016. doi:10.1016/j.automatica.2017.02.001.

[FFLS18] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza. Pampc: Perception-aware model predictive control for quadrotors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2018. doi:10.1109/IROS.2018.8593739.

[FH85] T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985. doi:10.1523/JNEUROSCI.05-07-01688.1985.

[FJ09] Z. Fuhao and L. Jiping. An algorithm of shortest path based on dijkstra for huge data. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 4, pages 244–247, 2009. doi:10.1109/FSKD.2009.848.

[FJS+17] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli. Real-time motion planning of legged robots: A model predictive control approach. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 577–584, 2017. doi:10.1109/HUMANOIDS.2017.8246930.

[FKBF+20] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin. Confidence-aware motion prediction for real-time collision avoidance1. *The International Journal of Robotics Research*, 39(2-3):250–265, 2020. doi:10.1177/0278364919859436.

[FKDLK12] F. Flacco, T. Kröger, A. De Luca, and O. Khatib. A depth space approach to human-robot collision avoidance. In *2012 IEEE International Conference on Robotics and Automation*, pages 338–345, 2012. doi:10.1109/ICRA.2012.6225245.

[FKDZ19] L. P. Fröhlich, E. D. Klenske, C. Daniel, and M. N. Zeilinger. Bayesian optimization for policy search in high-dimensional systems via automatic domain selection. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 757–764, 2019.

[FLS05] D. Ferguson, M. Likhachev, and A. Stentz. A guide to heuristic-based path planning. In *Proceedings of ICAPS '05 Workshop on Planning under Uncertainty for Autonomous Systems*, June 2005.

[FPM+23] M. N. Finean, L. Petrovic, W. Merkt, I. Markovic, and I. Havoutis. Motion planning in dynamic environments using context-aware human trajectory prediction. *Robotics and Autonomous Systems*, 166:104450, 2023. doi:10.1016/j.robot.2023.104450.

[FS11] G. Ferrer and A. Sanfeliu. Comparative analysis of human motion trajectory prediction using minimum variance curvature. In *Proceedings of the 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 135–136, March 2011. doi:10.1145/1957656.1957698.

[GA20] O. A. Gbadamosi and D. R. Aremu. Design of a modified dijkstra's algorithm for finding alternate routes for shortest-path problems with huge costs. In *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, pages 1–6, 2020. doi:10.1109/ICMCECS47690.2020.240873.

[GLNR18] R. S. Gesser, D. M. Lima, and J. E. Normey-Rico. Robust model predictive control: Implementation issues with comparative analysis. *IFAC-PapersOnLine*, 51(25):478–483, 2018. 9th IFAC Symposium on Robust Control Design ROCOND 2018. doi:10.1016/j.ifacol.2018.11.183.

[GOS12] J. Gregory, A. Olivares, and E. Staffetti. Energy-optimal trajectory planning for robot manipulators with holonomic constraints. *Systems Control Letters*, 61(2):279–291, 2012. doi:10.1016/j.sysconle.2011.11.005.

[GS17] P. Gierlak and M. Szuster. Adaptive position/force control for robot manipulator in contact with a flexible environment. *Robotics and Autonomous Systems*, 95:80–101, 2017. doi:10.1016/j.robot.2017.05.015.

[GSK15] S. Gupta, S. T. Sarkar, and A. Kumar. Design optimization of minimally invasive surgical robot. *Applied Soft Computing*, 32:241–249, 2015. doi:10.1016/j.asoc.2015.03.032.

[GZ07] A. Gasparetto and V. Zanotto. A new method for smooth trajectory planning of robot manipulators. *Mechanism and Machine Theory*, 42(4):455–471, 2007. doi:10.1016/j.mechmachtheory.2006.04.002.

[HAMA19] A. Hentout, M. Aouache, A. Maoudj, and I. Akli. Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017. *Advanced Robotics*, 33(15-16):764–799, 2019. doi:10.1080/01691864.2019.1636714.

[HASH07] S. Haddadin, A. Albu-Schäeffer, and G. Hirzinger. Safe physical human-robot interaction: Measurements, analysis new insights. *International Symposium on Robotics Research (ISRR2007)*, pages 439–450, January 2007.

[HBA14] C. Hernández, J. Baier, and R. Achá. Making A* run faster than D*-lite for path-planning in partially known terrain. *Proceedings of the International Conference on Automated Planning and Scheduling*, 24:504–508, May 2014. doi:10.1609/icaps.v24i1.13675.

[HC16] S. Haddadin and E. Croft. *Physical Human–Robot Interaction*, pages 1835–1874. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-32552-1_69.

[HC22] S. Hjorth and D. Chrysostomou. Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly. *Robotics and Computer-Integrated Manufacturing*, 73:102208, 2022. doi:10.1016/j.rcim.2021.102208.

[HDLW22] K. Hoang-Dinh, M. Leibold, and D. Wollherr. A fast and close-to-optimal receding horizon control for trajectory generation in dynamic environments. *Robotics*, 11(4), 2022. doi:10.3390/robotics11040072.

[HDOEW19] K. Hoang-Dinh, O. S. Oguz, M. Elsayed, and D. Wollherr. Adaptation and transfer of robot motion policies for close proximity human-robot interaction. *Frontiers in Robotics and AI*, 6, 2019. doi:10.3389/frobt.2019.00069.

[HDOH+15] K. Hoang-Dinh, O. S. Oguz, G. Huber, V. Gabler, and D. Wollherr. An approach to integrate human motion prediction into local obstacle avoidance in close human-robot collaboration. In *International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 1–6. IEEE, 2015. doi:10.1109/ARSO.2015.7428221.

[HDWLW17] K. Hoang-Dinh, P. Weiler, M. Leibold, and D. Wollherr. Fast and close to optimal trajectory generation for articulated robots in reaching motions. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1221–1227, 2017. doi:10.1109/AIM.2017.8014185.

[HFD11] B. Houska, J. Ferreau, and M. Diehl. ACADO toolkit–an open source framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32:298 – 312, May 2011. doi:10.1002/oca.939.

[HHK+12] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer. On making robots understand safety: Embedding injury knowledge into control. *The International Journal of Robotics Research*, 31(13):1578–1602, 2012. doi:10.1177/0278364912462256.

[HMH09] S. Yan H. Mu, Y. Zhou and A. Han. Third-order trajectory planning for high-accuracy point-to-point motion. *Frontiers of Electrical and Electronic Engineering in China*, 4:83–87, March 2009. doi:10.1007/s11460-009-0017-y.

[HNR68] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi:10.1109/TSSC.1968.300136.

[Hog85] N. Hogan. Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications. *ASME Journal of Dynamic Systems and Measurement Control*, 107:1–24, March 1985. doi:10.1115/1.3140701.

[HUP+10] S. Haddadin, H. Urbanek, S. Parusel, D. Burschka, J. Roßmann, A. Albu-Schäffer, and G. Hirzinger. Real-time reactive motion generation based on variable attractor dynamics and shaped velocities. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3109–3116, 2010. doi:10.1109/IROS.2010.5650246.

[HW98] C. Harris and D. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, 09 1998. doi:10.1038/29528.

[HWMZ20] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020. doi:10.1146/annurev-control-090419-075625.

[HWR08] R. Haschke, E. Weitnauer, and H. Ritter. On-line planning of time-optimal, jerk-limited trajectories. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3248–3253, 2008. doi:10.1109/IROS.2008.4650924.

[INH+13] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Computation*, 25(2):328–373, February 2013. doi:10.1162/NECO_a_00393.

[Ior23] Perovic G. Cini F. et al. Iori, F. DMP-based reactive robot-to-human handover in perturbed scenarios. *Int J of Soc Robotics*, 15:233–248, 2023. doi:10.1007/s12369-022-00960-4.

[JGJ14] A. Jubien, M. Gautier, and A. Janot. Dynamic identification of the Kuka LWR robot using motor torques and joint torque sensors data. *IFAC Proceedings Volumes*, 47(3):8391–8396, 2014. doi:10.3182/20140824-6-ZA-1003.01079.

[K.17] Serdar K. Optimal trajectory generation algorithm for serial and parallel manipulators. *Robotics and Computer-Integrated Manufacturing*, 48:219–232, 2017. doi:10.1016/j.rcim.2017.04.006.

[KAAS15] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart. Fast nonlinear model predictive control for multicopter attitude tracking on so(3). In *2015 IEEE Conference on Control Applications (CCA)*, pages 1160–1166, 2015. doi:10.1109/CCA.2015.7320769.

[KBTK18] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause. Learning-based model predictive control for safe exploration. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6059–6066, 2018. doi:10.1109/CDC.2018.8619572.

[KCT+11] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal. STOMP: Stochastic trajectory optimization for motion planning. In *2011 IEEE International Conference on Robotics and Automation*, pages 4569–4574, 2011. doi:10.1109/ICRA.2011.5980280.

[KFT+08] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How. Motion planning for urban driving using rrt. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1681–1686, 2008. doi:10.1109/IROS.2008.4651075.

[Kha90] O. Khatib. *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*, pages 396–404. Springer New York, 1990. doi:10.1007/978-1-4613-8997-2_29.

[KK22] C. Khazoom and S. Kim. Humanoid arm motion planning for improved disturbance recovery using model hierarchy predictive control. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6607–6613, 2022. doi:10.1109/ICRA46639.2022.9811878.

[KLL19] A. Kolbeinsson, E. Lagerstedt, and J. Lindblom. Foundation for a classification of collaboration levels for human-robot cooperation in manufacturing. *Production & Manufacturing Research*, 7(1):448–471, 2019. doi:10.1080/21693277.2019.1645628.

[KMUS90] M. Kawato, Y. Maeda, Y. Uno, and R. Suzuki. Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion. *Biological Cybernetics*, 62(4):275–288, 1990. doi:10.1007/BF00201442.

[KNGD02] T. Kalmar-Nagy, P. Ganguly, and R. D'Andrea. Real-time trajectory generation for omnidirectional vehicles. In *Proceedings of the American Control Conference*, volume 1, pages 286–291, February 2002. doi:10.1109/ACC.2002.1024818.

[KSLO96] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996. doi:10.1109/70.508439.

[KSS+07] K. Koay, E. Sisbot, D. Syrdal, M. Walters, K. Dautenhahn, and R. Alami. Exploratory study of a robot approaching a person in the context of handing

over an object. *AAAI spring symposium: multidisciplinary collaboration for socially assistive robotics*, 2007. URL: https://hal.laas.fr/hal-01979353.

[KTS+10] A. Kirsch, K. Thibault, A. Sisbot, R. Alami, M. Lawitzky, S. Hirche, P. Basili, and S. Glasauer. Plan-based control of joint human-robot activities. *KI - Künstliche Intelligenz*, 24(3):223–231, 2010. doi:10.1007/s13218-010-0043-1.

[KW10] T. Kröger and F. M. Wahl. Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *IEEE Transactions on Robotics*, 26(1):94–111, 2010. doi:10.1109/TRO.2009.2035744.

[KWP+11] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime Motion Planning using the RRT*. In *2011 IEEE International Conference on Robotics and Automation*, pages 1478–1483, 2011. doi:10.1109/ICRA.2011.5980479.

[LaV98] S. M. LaValle. Rapidly-exploring random trees : a new tool for path planning. *The annual research report*, page TR 98–11, 1998.

[LBOD22] D. P. Losey, A. Bajcsy, M. K. O'Malley, and A. D. Dragan. Physical interaction as communication: Learning robot objectives online from human corrections. *The International Journal of Robotics Research*, 41(1):20–44, 2022. doi:10.1177/02783649211050958.

[Lee11] J. Lee. Model Predictive Control: Review of the Three Decades of Development. *International Journal of Control, Automation and Systems*, 9:415–424, June 2011. doi:10.1007/s12555-011-0300-6.

[LFS17a] P. Lasota, T. Fong, and J. Shah. A survey of methods for safe human-robot interaction. *Foundations and Trends in Robotics*, 5:261–349, January 2017. doi:10.1561/2300000052.

[LFS17b] P. A. Lasota, T. Fong, and J. A. Shah. A survey of methods for safe Human-Robot Interaction. *Foundations and Trends® in Robotics*, 5(4):261–349, 2017. doi:10.1561/2300000052.

[LG10] A. Lazaric and M. Ghavamzadeh. Bayesian Multi-Task Reinforcement Learning. In *ICML - 27th International Conference on Machine Learning*, pages 599–606, June 2010. URL: https://hal.inria.fr/inria-00475214.

[LHS+13] S. Lallée, K. Hamann, J. Steinwender, F. Warneken, U. Martienz, H. B. Gonzales, U. Pattacini, I. Gori, M. Petit, G. Metta, P. Verschure, and P. F. Dominey. Cooperative human robot interaction systems: Iv. communication of shared plans with naïve humans using gaze and speech. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 129–136, 2013. doi:10.1109/IROS.2013.6696343.

[LJJK01] S. M. LaValle and Jr. J. J. Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001. doi:10.1177/02783640122067453.

[LK16] C. Lichtenthäler and A. Kirsch. Legibility of Robot Behavior : A Literature Review. April 2016. URL: https://hal.archives-ouvertes.fr/hal-01306977.

[LL22] S. Liu and P. Liu. Benchmarking and optimization of robot motion planning with motion planning pipeline. *The International Journal of Advanced Manufacturing Technology*, 118, January 2022. doi:10.1007/s00170-021-07985-5.

[LLH09] A. C. Leite, F. Lizarralde, and L. Hsu. Hybrid adaptive vision—force control for robot manipulators interacting with unknown surfaces. *The International Journal of Robotics Research*, 28(7):911–926, 2009. doi:10.1177/0278364909101932.

[LLK11] C. Lichtenthäler, T. Lorenz, and A. Kirsch. Towards a Legibility Metric: How to Measure the Perceived Value of a Robot. In *International Conference on Social Robotics, ICSR 2011*, November 2011. URL: https://hal.archives-ouvertes.fr/hal-01571938.

[LLLK20] Y. Liu, Z. Li, H. Liu, and Z. Kan. Skill transfer learning for autonomous robots and human–robot cooperation: A survey. *Robotics and Autonomous Systems*, 128:103515, 2020. doi:doi.org/10.1016/j.robot.2020.103515.

[LLLW17] W. Liu, Z. Li, L. Li, and F. Y. Wang. Parking like a human: A direct trajectory planning solution. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3388–3397, 2017. doi:10.1109/TITS.2017.2687047.

[LNTC+11] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters. Trajectory planning for optimal robot catching in real-time. In *2011 IEEE International Conference on Robotics and Automation*, pages 3719–3726, May 2011. doi:10.1109/ICRA.2011.5980114.

[LRS14] P. A. Lasota, G. F. Rossano, and J. A. Shah. Toward safe close-proximity human-robot interaction with standard industrial robots. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 339–344, 2014. doi:10.1109/CoASE.2014.6899348.

[LSZL14] X. Li, Z. Sun, Q. Zhu, and D. Liu. A unified approach to local trajectory planning and control for autonomous driving along a reference path. In *2014 IEEE International Conference on Mechatronics and Automation*, pages 1716–1721, August 2014. doi:10.1109/ICMA.2014.6885959.

[LVS20] C. E. Luis, M. Vukosavljev, and A. P. Schoellig. Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robotics and Automation Letters*, 5(2):604–611, 2020. doi:10.1109/LRA.2020.2964159.

[LW17] H. Liu and L. Wang. Human motion prediction for human-robot collaboration. *Journal of Manufacturing Systems*, 44:287–294, 2017. doi:10.1016/j.jmsy.2017.04.009.

[May14] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014. doi:10.1016/j.automatica.2014.10.128.

[MCTK17] T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. *Applied Soft Computing*, 59:68–76, 2017. doi:10.1016/j.asoc.2017.05.012.

[MDRMAF16] A. Mendes, D. De Rizzo Meneghetti, M. Ackermann, and A. T. Fleury. Vehicle dynamics - lateral: Open source simulation package for matlab. In *Congresso SAE Brasil 2016*, October 2016. doi:10.4271/2016-36-0115.

[MHB15] J. Mainprice, R. Hayne, and D. Berenson. 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)*, pages 885–892, 2015. doi:10.1109/ICRA.2015.7139282.

[MM90] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990. doi:10.1109/9.57020.

[MM93] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993. doi:10.1109/9.262032.

[MMM00] M.B. Milam, K. Mushambi, and R.M. Murray. A new computational approach to real-time trajectory generation for constrained mechanical systems. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 1, pages 845–851, 2000. doi:10.1109/CDC.2000.912875.

[MNK+13] D. Madås, M. Nosratinia, M. Keshavarz, P. Sundström, R. Philippsen, A. Eidehall, and K. M. Dahlén. On path planning methods for automotive collision avoidance. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 931–937, June 2013. doi:10.1109/IVS.2013.6629586.

[MPMK17] G. Magistris, A. Pajon, S. Miossec, and A. Kheddar. Optimized humanoid walking with soft soles. *Robotics and Autonomous Systems*, 95:52–63, 09 2017. doi:10.1016/j.robot.2017.05.006.

[MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. doi:10.1016/S0005-1098(99)00214-9.

[MSV+23] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti. Biconmp: A nonlinear model predictive control framework for whole body motion planning. *IEEE Transactions on Robotics*, 39(2):905–922, 2023. doi:10.1109/TRO.2022.3228390.

[NdCF+16] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli. Fast nonlinear model predictive control for unified

trajectory optimization and tracking. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1398–1404, 2016. doi:10.1109/ICRA.2016.7487274.

[NIO+99] E. Nakano, H. Imamizu, R. Osu, Y. Uno, H. Gomi, T. Yoshioka, and M. Kawato. Quantitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model. *Journal of Neurophysiology*, 81(5):2140–2155, 1999. doi:10.1152/jn.1999.81.5.2140.

[NRGS15] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, page 189–196, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2696454.2696455.

[NSG+18] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters*, 3(3):1458–1465, 2018. doi:10.1109/LRA.2018.2800124.

[Oht04] T. Ohtsuka. A continuation/gmres method for fast computation of nonlinear receding horizon control. *Automatica*, 40:563–574, April 2004. doi:10.1016/j.automatica.2003.11.005.

[OSHDW17] O. Oguz, O. Sari, K. Hoang-Dinh, and D. Wollherr. Progressive stochastic motion planning for human-robot interaction. *IEEE International Symposium on Robot and Human Interactive Communication*, 2017. doi:10.1109/ROMAN.2017.8172456.

[OUT16] M. Obayashi, K. Uto, and G. Takano. Appropriate overtaking motion generating method using predictive control with suitable car dynamics. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 4992–4997, 2016. doi:10.1109/CDC.2016.7799032.

[PCN13] Q. C. Pham, S. Caron, and Y. Nakamura. Kinodynamic Planning in the configuration space via admissible velocity propagation. In *Robotics: Science and System*, June 2013. doi:10.15607/RSS.2013.IX.052.

[PCY+16] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, March 2016. doi:10.1109/TIV.2016.2578706.

[PDE+14] K. Paes, W. Dewulf, K. V. Elst, K. Kellens, and P. Slaets. Energy efficient trajectories for an industrial abb robot. *Procedia CIRP*, 15:105–110, 2014. doi:10.1016/j.procir.2014.06.043.

[PDPN13] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Probabilistic Movement Primitives. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 2616–2624, 2013. URL: http://dl.acm.org/citation.cfm?id=2999792.2999904.

[Pen55] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955. doi:10.1017/S0305004100030401.

[PHPS08] D. H. Park, H. Hoffmann, P. Pastor, and S. Schaal. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pages 91–98, 2008. doi:10.1109/ICHR.2008.4755937.

[PNS+15] A. Pekarovskiy, T. Nierhoff, J. Schenek, Y. Nakamura, S. Hirche, and M. Buss. Online deformation of optimal trajectories for constrained nonprehensile manipulation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2481–2487, 2015. doi:10.1109/ICRA.2015.7139531.

[Pon87] L. S. Pontryagin. *Mathematical Theory of Optimal Processes*. Classics of Soviet Mathematics. CRC Press, 1987.

[QB00] S. J. Qin and T. A. Badgwell. An overview of nonlinear model predictive control applications. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 369–392, Basel, 2000. Birkhäuser Basel. doi:10.1007/978-3-0348-8407-5_21.

[Qua19] QualisysAB. Qualisys track manager (qtm), 2019. URL: https://www.qualisys.com/.

[Qui95] S. Quinlan. *Real-Time Modification of Collision-Free Paths*. PhD thesis, Stanford, CA, USA, 1995. UMI Order No. GAX95-16902.

[RA13] D. Rubin and S. Arogeti. Vehicle yaw stability control using rear active differential via sliding mode control methods. In *21st Mediterranean Conference on Control and Automation*, pages 317–322, 2013. doi:10.1109/MED.2013.6608740.

[Rao10] A. Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135:497–528, January 2010.

[RB20] U. Rosolia and F. Borrelli. Learning how to autonomously race a car: A predictive control approach. *IEEE Transactions on Control Systems Technology*, 28(6):2713–2719, 2020. doi:10.1109/TCST.2019.2948135.

[RBL+17] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pérez-Oria. Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access*, 5, 2017. doi:10.1109/ACCESS.2017.2773127.

[RIS13] Z. Rymansaib, P. Iravani, and M. N. Sahinkaya. Exponential trajectory generation for point to point motions. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 906–911, 2013. doi:10.1109/AIM.2013.6584209.

[RMD17] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model Predictive Control: Theory,Computation and Design*. Nob Hill Publishing,LLC, January 2017.

[RNH10] A. Rucco, G. Notarstefano, and J. Hauser. Dynamics exploration of a single-track rigid car model with load transfer. In *49th IEEE Conference on Decision and Control (CDC)*, pages 4934–4939, 2010. doi:10.1109/CDC.2010.5718133.

[RPH+20] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras. Human motion trajectory prediction: a survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. doi:10.1177/0278364920917446.

[RTD19] H. C. Ravichandar, D. Trombetta, and A. P. Dani. Human intention-driven learning control for trajectory synchronization in human-robot collaborative tasks. *IFAC-PapersOnLine*, 51(34):1–7, 2019. doi:10.1016/j.ifacol.2019.01.001.

[Ryo11] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *2011 International Conference on Computer Vision*, pages 1036–1043, 2011. doi:10.1109/ICCV.2011.6126349.

[SA12] E. A. Sisbot and R. Alami. A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(5):1045–1057, October 2012. doi:10.1109/TRO.2012.2196303.

[SAS17] Y. R. Stürz, L. M. Affolter, and R. S. Smith. Parameter identification of the KUKA LBR IIWA Robot including constraints on physical feasibility. *IFAC-PapersOnLine*, 50(1):6863–6868, 2017. doi:10.1016/j.ifacol.2017.08.1208.

[SCAR08] E. A. Sisbot, A. Clodic, R. Alami, and M. Ransan. Supervision and motion planning for a mobile manipulator interacting with humans. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 327–334, March 2008. doi:10.1145/1349822.1349865.

[SCGHRA22] A. D. Sosa-Ceron, H. G. Gonzalez-Hernandez, and J. A. Reyes-Avendano. Learning from demonstrations in humanndash;robot collaborative scenarios: A survey. *Robotics*, 11(6), 2022. doi:10.3390/robotics11060126.

[Sch06] S. Schaal. Dynamic Movement Primitives–A Framework for Motor Control in Humans and Humanoid Robotics. *Adaptive Motion of Animals and Machines*, January 2006. doi:10.1007/4-431-31381-8_23.

[SCM15] A. St. Clair and M. Mataric. How robot verbal feedback can improve team performance in human-robot task collaborations. 2015:213–220, March 2015. doi:10.1145/2696454.2696491.

[SD15] A. Sarkar and A. Dutta. 8-DOF biped robot with compliant-links. *Robotics and Autonomous Systems*, 63:57–67, 2015. doi:10.1016/j.robot.2014.09.014.

[SDH+14] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014. doi:10.1177/0278364914528132.

[SGBL15] F. Stulp, J. Grizou, B. Busch, and M. Lopes. Facilitating intention prediction for humans by optimizing robot motions. *Intelligent Robots and Systems*, pages 1249–1255, October 2015. doi:10.1109/IROS.2015.7353529.

[SLCS04] T. Siméon, J. P. Laumond, J. Cortés, and A. Sahbani. Manipulation planning with probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7–8):729–746, 2004. doi:10.1177/0278364904045471.

[SMAS07] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, Oct 2007.

[SMTA18] R. Soloperto, M. Müller, S. Trimpe, and F. Allgöwer. Learning-based robust model predictive control with state-dependent uncertainty. *IFAC-PapersOnLine*, 51:442–447, January 2018. doi:10.1016/j.ifacol.2018.11.052.

[SMUB+10] E. A. Sisbot, L. F. Marin-Urias, X. Broquère, D. Sidobre, and R. Alami. Synthesizing robot motions adapted to human presence. *International Journal of Social Robotics*, 2(3):329–343, September 2010. doi:10.1007/s12369-010-0059-6.

[SNB19] M. Safeea, P. Neto, and R. Béarée. A quest towards safe human robot collaboration. In *Towards Autonomous Robotic Systems*, pages 493–495, July 2019. doi:10.1007/978-3-030-25332-5_48.

[SS12] F. Stulp and O. Sigaud. Policy Improvement Methods: Between Black-Box Optimization and Episodic Reinforcement Learning. October 2012. doi:hal-00738463.

[SSAP12] P. Sánchez-Sánchez and M. A. Arteaga-Pérez. Simplied methodology for obtaining the dynamic model of robot manipulators. *International Journal of Advanced Robotic Systems*, 9(5):170, 2012. doi:10.5772/51305.

[SSD14] S. Stevo, I. Sekaj, and M. Dekan. Optimization of robotic arm trajectory using genetic algorithm. *IFAC Proceedings Volumes*, 47(3):1748–1753, 2014. doi:10.3182/20140824-6-ZA-1003.01073.

[TDJ11] L. Takayama, D. Dooley, and W. Ju. Expressing thought: Improving robot readability with animation principles. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 69–76, 2011. doi:10.1145/1957656.1957674.

[TFFS19] E. Terzi, L. Fagiano, M. Farina, and R. Scattolini. Learning-based predictive control for linear systems: A unitary approach. *Automatica*, 108:108473, 2019. doi:10.1016/j.automatica.2019.06.025.

[TFG19] M. Tanaskovic, L. Fagiano, and V. Gligorovski. Adaptive model predictive control for linear time varying mimo systems. *Automatica*, 105:237–245, 2019. doi:10.1016/j.automatica.2019.03.030.

[TIN11] W. Takano, H. Imagawa, and Y. Nakamura. Prediction of human behaviors in the future through symbolic inference. In *2011 IEEE International Conference on Robotics and Automation*, pages 1970–1975, 2011. doi:10.1109/ICRA.2011.5980422.

[Tod04] E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9), 2004. doi:10.1038/nn1309.

[VPLS18] V. Villani, F. Pini, F. Leali, and C. Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248–266, 2018. doi:10.1016/j.mechatronics.2018.02.009.

[VSB92] O. V. Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357–373, December 1992. doi:10.1007/BF02071065.

[VSG+17] D. Vogt, S. Stepputtis, S. Grehl, B. Jung, and H. B. Amor. A system for learning continuous human-robot interactions from human-human demonstrations. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2882–2889, 2017. doi:10.1109/ICRA.2017.7989334.

[vSS94] O. von Stryk and M. Schlemmer. *Optimal Control of the Industrial Robot Manutec r3*, pages 367–382. Birkhäuser Basel, 1994. doi:10.1007/978-3-0348-8497-6_30.

[VWE13] C. Vogel, C. Walter, and N. Elkmann. A projection-based sensor system for safe physical human-robot collaboration. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5359–5364, 2013. doi:10.1109/IROS.2013.6697132.

[VZQD16] R. Verschueren, M. Zanon, R. Quirynen, and M. Diehl. Time-optimal race car driving using an online exact hessian based nonlinear mpc algorithm. In *Control Conference (ECC), 2016 European*, pages 141–147, 2016. doi:10.1109/ECC.2016.7810277.

[WHCZ21] K. Wabersich, L. Hewing, A. Carron, and M. Zeilinger. Probabilistic model predictive safety certification for learning-based control. *IEEE Transactions on Automatic Control*, January 2021. doi:10.1109/TAC.2021.3049335.

[WHHAS13] R. Weitschat, S. Haddadin, F. Huber, and A. Albu-Schaeffer. Dynamic optimality in real-time: A learning framework for near-optimal robot motions. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5636–5643, November 2013. doi:10.1109/IROS.2013.6697173.

[WLA14] K. Wang, F. Léonard, and G. Abba. A novel approach for simplification of industrial robot dynamic model using interval method. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1697–1703, 2014. doi:10.1109/AIM.2014.6878328.

[WLCJ18] W. Wang, R. Li, Y. Chen, and Y. Jia. Human intention prediction in human-robot collaborative tasks. pages 279–280, March 2018. doi:10.1145/3173386.3177025.

[WLLW20] L. Wang, S. Liu, H. Liu, and X. Wang. Overview of human-robot collaboration in manufacturing. pages 15–58, June 2020. doi:10.1007/978-3-030-46212-3_2.

[WTLL15] A. Werner, D. Trautmann, D. Lee, and R. Lampariello. Generalization of optimal motion trajectories for bipedal walking. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1571–1577, September 2015. doi:10.1109/IROS.2015.7353577.

[WvdB13] D. J. Webb and J. van den Berg. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics. In *2013 IEEE International Conference on Robotics and Automation*, pages 5054–5061, 2013. doi:10.1109/ICRA.2013.6631299.

[XWF15] S. Xiao, Z. Wang, and J. Folkesson. Unsupervised robot learning to predict person motion. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 691–696, 2015. doi:10.1109/ICRA.2015.7139254.

[YC16] M. C. Yip and D. B. Camarillo. Model-less hybrid position/force control: A minimalist approach for continuum manipulators in unknown, constrained environments. *IEEE Robotics and Automation Letters*, 1(2):844–851, 2016. doi:10.1109/LRA.2016.2526062.

[YPW19] Y. Yang, J. Pan, and W. Wan. A survey of optimal motion planning. *IET Cyber-Systems and Robotics*, 1, June 2019. doi:10.1049/iet-csr.2018.0003.

[ZHFD18] A. Zanelli, G. Horn, G. Frison, and M. Diehl. Nonlinear model predictive control of a human-sized quadrotor. In *2018 European Control Conference (ECC)*, pages 1542–1547, 2018. doi:10.23919/ECC.2018.8550530.

[ZRD+13] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa. CHOMP: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10):1164–1193, 2013. doi:10.1177/0278364913488805.

[ZTC12] Y. Zhang, G. Tang, and L. Chen. Improved A* algorithm for time-dependent vehicle routing problem. In *Proceedings of the 2012 International Conference on Computer Application and System Modeling (ICCASM 2012)*, pages 1341–1344, August 2012. doi:10.2991/iccasm.2012.343.

Own Thesis-Related Publications

[HDLW22] K. Hoang-Dinh, M. Leibold, and D. Wollherr. A fast and close-to-optimal receding horizon control for trajectory generation in dynamic environments. *Robotics*, 11(4), 2022. doi:10.3390/robotics11040072.

[HDOEW19] K. Hoang-Dinh, O. S. Oguz, M. Elsayed, and D. Wollherr. Adaptation and transfer of robot motion policies for close proximity human-robot interaction. *Frontiers in Robotics and AI*, 6, 2019. doi:10.3389/frobt.2019.00069.

[HDOH+15] K. Hoang-Dinh, O. S. Oguz, G. Huber, V. Gabler, and D. Wollherr. An approach to integrate human motion prediction into local obstacle avoidance in close human-robot collaboration. In *International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 1–6. IEEE, 2015. doi:10.1109/ARSO.2015.7428221.

[HDWLW17] K. Hoang-Dinh, P. Weiler, M. Leibold, and D. Wollherr. Fast and close to optimal trajectory generation for articulated robots in reaching motions. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1221–1227, 2017. doi:10.1109/AIM.2017.8014185.