

# From Acting to Interacting: Allowing Robots to Collaborate in the Presence of Uncertainty

Volker Stefan Gabler

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktors der Ingenieurwissenschaften (Dr.-Ing.)**

genehmigten Dissertation.

**Vorsitz:**

Priv.-Doz. Dr. Gabriele Schrag

**Prüfer\*innen der Dissertation:**

1. Priv.-Doz. Dr.-Ing. Dirk Wollherr
2. Prof. Dr. Wataru Takano

Die Dissertation wurde am 05.09.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 27.03.2023 angenommen.



# Preface

## Funding Acknowledgments

This thesis received funding from two distinct research projects over the full lifespan. The first period of this thesis received funding from an industrial research collaboration with the SIEMENS AG focusing on the area of *human-robot collaboration in industrial contexts*. This project has thus allowed for the research results presented in Chapter 3, 4 and 5, and initiated the work from Chapter 6.

The final period of this thesis has received funding from the Horizon 2020 research and innovation programme under grant agreement №820742 of the project "*HR-Recycler - Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment*". Eventually, this project provided the basis for results presented in Chapter 7, 8 and 9.

## Personal Acknowledgments

This dissertation summarizes my work during the roller coaster ride as a research associate with the Chair of Automatic Control Engineering (LSR) at the Technical University of Munich (TUM). Before outlining the technical insights of this dissertation, I would like to express my gratitude to a collection of people that joined me on this journey.

First, I would like to thank my doctoral advisor PD. habil. Dr.-Ing. Dirk Wollherr, not only for believing in my proficiencies but also for allowing to participate in research projects about *human-robot collaboration* with SIEMENS AG as well as the *HR-RECYCLER* EU-Project. Having faced such a wide range of challenges allowed me to establish a new level of skills that I could have not imagined beforehand. Further, I would like to thank Prof. Dr. Wataru Takano for taking over this dissertation's second examiner role. Eventually, I would like to thank Prof. Dr.-Ing./Univ. Tokio Martin Buss for the unique scientific freedom across an impressive range of research topics.

A general thank you belongs to all the people who have accompanied me during this thesis and whom I may have forgotten to list explicitly. Karen, Miruna, and Larissa for the coffee brakes and reliable support in the bureaucratic jungle. Tobias, Kilian, Christian, Thomas and Wolfgang for providing technical support in a reliable and enlightening fashion. The collection of students that I could supervise and work with over the time of this thesis, where a special thanks belong to Tim St., Johannes, Korbinian, Huangjang, and Dominik K.. Further, I would like to thank my project companions, namely Oz, Khoi, Gerold, Zengjie, Sebastian K., Salman, and Annalena, as well as my remaining Ph.D. colleagues from LSR and ITR, for easing this journey. Out of these, a special thanks goes to Khoi for the friendship and for sharing the office – a.k.a. *CoI* – with me for many years, as well as to Jakob, Tim and Alex C. for the distraction from Ph.D.-life. Eventually, sincere gratitude belongs to Gerold and Stefan for their friendship and support, especially when I realized that the *F* in Ph.D. stands for fun. In hindsight, I have to admit that without Gerold joining the SIEMENS project, this journey would have ended as a six-month internship.

I sincerely thank all my friends who have supported me over the last few years. Those back in my hometown Malsch – Christoph, Laura, Maddin, Mascha, Phil, Simon and the *gang*, as well as Timo and the other *Deppen* from Karlsruhe. Those spread across Germany – Chris, Domi, Fex, Vero and across Europe – Flo, Geiger, Lars.

My deepest and sincere gratitude belongs to my family. First and foremost, my sister Uli for her unconditional support, but also for the Doctor, h.c. title I received to end my complaints finally. Secondly, to my parents Helga and Richard, for literally everything they have done and their everlasting support, which deserves more words than I could ever add here. But due to their Swabian upbringing, I'd rather save the printing costs.<sup>1</sup> Eventually, I am most thankful for my beloved little Alex. She has supported me without exceptions or hesitance throughout the final part of this thesis and made me realize what matters in life once Mimi entered our little family.

Munich, May 2023

Doctor, h.c. Volker Gabler of Metaphysis (CCU/USA)

---

<sup>1</sup>Nonetheless, I have to explicitly highlight the proofreading services of my father throughout the past years here, which may make him the only person on planet Earth who reads a footnote in the preface of this wall of text.

# Abstract

Allowing robots to handle and solve complex tasks in arbitrary environments forms a promising goal for research, industry and eventually global society. Recalling the advances and impressive performance that robots possess within current application areas, such as industrial assembly or lab environments, the question arises, why the number of these applications is still comparably low. As long as the number of possible environment states is small, the applications do not diverge drastically from the characteristic of their core application areas as stated above. Unfortunately, real-world applications rarely meet this requirement, so there is still a lot of research to be done to close the gap between the lab and the open world. Thus, a key-challenge is given by the handling of imperfect knowledge and the perception of the environment.

This work therefore specifically addresses the question of how to improve the ability of artificial agents to interact in domains where model knowledge is insufficient or it is not possible to obtain an accurate model. As this research question contains an infinite amount of sub-challenges to be solved due the complexity of robotic systems, the research conducted in this thesis splits across three particular sub-fields of robotic applications: The interaction of humans and robots, which has been established under the term human-robot collaboration (HRC), the interaction of multiple – but individual – autonomous agents under the aspect of multi-agent reinforcement learning (MARL) and eventually the ability of robots to manipulate unknown objects with imperfect perception of the environment.

In the context of HRC this thesis proposes novel concepts to incorporate insights from cognitive science into autonomous decision making by robots for applications where a human and a robot are required to perform a joint task. Specifically, novel methods are presented that model the human performance as partially observable states that are estimated online to obtain team-optimal action assignments for the interactive HRC-process. In addition, concepts from interactive game-theory are incorporated into the action selection method, which unlike related work, allows team-optimal strategies to be found while taking human decision-making ability into account. In contrast to existing work, the proposed methods allow for the direct incorporation of human objectives rather than modeling humans as stochastic black boxes or optimal automata. The proposed methods are evaluated in human-robot studies, which highlight the improvements of the proposed methods in terms of comprehensibility of the selected strategies to the human subjects. Overall, the methods presented form the basis for improving natural and seamless interaction between humans and robots, enabling the use of robots outside of the usually caged environments.

The challenge of solving a task jointly with a fleet of artificial agents holds the possibility of improving the scaling of robotic applications through the insights gathered. Special emphasis is placed on learning new tasks and challenges more quickly and efficiently. In this context, this area has mostly been approached from the machine-learning community by regressing a joint task solely from data, where recent finding on deep-reinforcement learning has drastically boosted development and progress in recent years. In control-theory on the other hand, the majority of approaches imposes strong model and system assumptions to achieve workable results. Motivated by the fact, that hierarchical learning has allowed to

## *Abstract*

embed model knowledge or controllers, this thesis explores novel methods on how to obtain a hierarchical MARL-framework. This framework aims to allow artificial agents to exploit (partial) knowledge about the internal system-dynamics, without adding further constraints on the environment. Therefore, a decentralized learning scheme is presented and evaluated against the state-of-the-art. Eventually, this part of the thesis introduces a novel hierarchical MARL-approach that provides the potential on transitioning from pure model-free end-to-end learning to a hybrid model-free and model-based method.

The final part of this thesis analyses the challenge of manipulation tasks, where the perception of the environment is imprecise. Specifically, three sub-applications are evaluated: how can a robot refine knowledge about an unknown object in the presence of inaccurate visual prior assumptions by collecting only haptic measurements? How can a robot autonomously grasp fragile objects if neither the robot's control modalities provide an interface to directly compensate interaction wrenches, e.g., an impedance controller, nor is a correct goal-pose specified. The final question analyzed in this thesis is how to regress the optimal task parameterization of a task to be learned from just a handful of data trials. To answer the first question, a novel state estimation is proposed that incorporates insights from Bayesian filter theory, as commonly applied in robot navigation or observer design, but depends only on the acquisition of haptic data. The presented method is evaluated within a simulated environment, where physical interaction data is collected and the material properties can be specifically adjusted by the operator. In the second question, a novel gripping strategy is proposed, which can be applied to industrial robots in order to allow compliant end-effector-pose correction without the explicit need for additional force-torque sensors. Eventually, we propose a novel learning system for an industrial robot that learns tasks from only a handful of samples. This method extends existing work by proposing not only to graphically describe the objective to be learned to obtain a reduced parameter-space of the original problem, but also to specifically model the constraints, i.e., feasible state-space of the parameters, in order to directly account for the failure of the task. Both these approaches are finally evaluated on experimental lab evidence, where the collected data underlines the improved performance compared to the baseline solutions and existing methods. Thus, the presented methods allow robots to improve their capabilities of manipulating unknown objects when perception suffers from imprecision.

In summary, this thesis contains a broad collection of novel techniques that improve the capabilities and skills of robots, laying the foundation for future robotic applications. Since research in itself is something that will never be finished, this thesis concludes with a brief outlook on future research questions and aspects.

# Zusammenfassung

Die Fähigkeit von Robotern, komplexe Aufgaben in beliebigen Umgebungen zu bewältigen und zu lösen, ist ein vielversprechendes Ziel für Forschung, Industrie und schließlich die Gesellschaft. In Anbetracht der Fortschritte und der beeindruckenden Leistung, die Roboter in Anwendungsbereichen wie z.B. in der industriellen Montage oder in Laborumgebungen erzielen, stellt sich die Frage, warum die Zahl dieser Anwendungen noch vergleichsweise gering ist. Solange die Zahl der möglichen Zustände der Umwelt gering ist, weichen die Anwendungen nicht drastisch von der oben genannten Charakteristik ihrer Kernanwendungsbereiche ab. Leider erfüllen reale Anwendungen diese Anforderung nur sehr selten, so dass noch viel Forschungsarbeit geleistet werden muss, um die Lücke zwischen Labor und tatsächlicher Endanwendung zu schließen. Eine Schlüsselherausforderung dabei ist der Umgang mit unvollständigem Wissen sowie mit Ungenauigkeiten in der Wahrnehmung der Umgebung.

Diese Arbeit befasst sich daher speziell mit der Frage, wie die Fähigkeiten künstlicher Agenten zur Interaktion in Anwendungsbereichen verbessert werden können, in denen das Modellwissen unzureichend ist oder es nicht möglich ist, ein präzises Modell zu entwerfen. Da diese Forschungsfrage aufgrund der Komplexität von Robotersystemen unzählige Teilprobleme beinhaltet, die es zu lösen gilt, verteilt sich die Forschung in dieser Arbeit auf drei spezielle Teilbereiche von Roboteranwendungen: die Interaktion von Menschen und Robotern, die sich unter dem Begriff der Mensch-Roboter-Kollaboration (MRK) etabliert hat, die Interaktion mehrerer - jedoch individueller - autonomer Agenten unter dem Aspekt des bestärkenden Lernens für Multi-Agenten-Systeme (MARL) und schließlich die Fähigkeit von Robotern, unbekannte Objekte unter ungenauer Wahrnehmung der Umgebung zu manipulieren.

Im Kontext der MRK werden in dieser Arbeit neuartige Konzepte vorgeschlagen, um Erkenntnisse aus der Kognitionswissenschaft in die autonome Entscheidungsfindung von Robotern für Anwendungen einzubeziehen, bei denen ein Mensch und ein Roboter eine gemeinsame Aufgabe zu bewältigen haben. Konkret werden neuartige Methoden vorgestellt, die die menschliche Leistung als teilweise beobachtbare Zustände modellieren, die online geschätzt werden, um eine teamoptimale Aktionszuweisung für den interaktiven MRK-Prozess zu erhalten. Darüber hinaus werden Konzepte aus der interaktiven Spieltheorie in die Handlungsauswahlmethode integriert, die es im Gegensatz zu verwandten Arbeiten erlaubt, teamoptimale Strategien zu finden und dabei die menschliche Entscheidungsfähigkeit zu berücksichtigen. Im Gegensatz zu bestehenden Arbeiten ermöglichen die vorgeschlagenen Methoden den direkten Einbezug menschlicher Ziele, anstatt den Menschen als stochastische Blackbox oder optimalen Automaten zu modellieren. Die vorgeschlagenen Methoden werden in Mensch-Roboter-Studien evaluiert, die die Verbesserungen der vorgeschlagenen Methoden in Bezug auf die Verständlichkeit der ausgewählten Strategien für die menschlichen Probanden unterstreichen. Insgesamt bilden die vorgestellten Methoden die Grundlage für die Verbesserung der natürlichen und nahtlosen Interaktion zwischen Menschen und Robotern, die den Einsatz von Robotern außerhalb der üblichen Labor- oder Käfigumgebungen ermöglicht.

Die Herausforderung, eine Aufgabe gemeinsam mit einer Vielzahl künstlicher Agenten zu lösen, birgt die Möglichkeit, Roboteranwendungen hinsichtlich der Skalierbarkeit durch die dezentral gesammelten Erkenntnisse zu verbessern. Dabei wird besonderer Wert darauf gelegt, neue Aufgaben und Herausforderungen schneller und effizienter zu erlernen. Dieser

Forschungsbereich wurde in erster Linie im Bereich von MARL behandelt, indem eine gemeinsame Aufgabe ausschließlich aus gesammelten Daten erlernt wird, wobei die jüngsten Erkenntnisse im Bereich des Deep Reinforcement Learning die Entwicklung und den Fortschritt in den letzten Jahren dramatisch vorangetrieben haben. In der Regelungstechnik hingegen setzt die Mehrheit der Ansätze starke Modell- und Systemannahmen voraus, um praktikable Ergebnisse zu erzielen. Motiviert durch die Tatsache, dass im Gegensatz zum direkten End-zu-End Lernen das hierarchische Lernen erlaubt, vorhandenes Modellwissen in den Lernprozess mit einzubeziehen, erforscht diese Arbeit neuartige Methoden, um ein hierarchisches Multi-Agenten-System für das Bestärkende Lernen zu erhalten. Dies soll es künstlichen Agenten ermöglichen, (partielles) Wissen über die interne Systemdynamik auszunutzen, ohne dabei zusätzliche Einschränkungen bezüglich der Umgebung vorauszusetzen. Daher wird ein dezentrales Lernverfahren vorgestellt und gegen den aktuellen Stand der Technik evaluiert. Schließlich wird darauf aufbauend ein neuartiges hierarchisches MARL-Verfahren vorgestellt, das eine Möglichkeit zum Übergang vom rein modellfreien Lernen zum hybriden Lernen, bestehend aus modellfreiem und modellbasiertem Lernen, bietet.

Der letzte Teil dieser Arbeit analysiert die Herausforderung von Manipulationsaufgaben, bei denen die Wahrnehmung der Umgebung ungenau ist. Konkret werden drei Teilanwendungen evaluiert: Wie kann ein Roboter das Wissen über ein unbekanntes Objekt bei ungenauen visuellen Vorannahmen verfeinern, indem er ausschließlich haptische Messungen sammelt? Wie kann ein Roboter zerbrechliche Objekte autonom greifen, wenn weder die Steuerungsmodalitäten des Roboters eine Schnittstelle bieten, um Interaktionsfehler direkt zu kompensieren, z.B. einen Impedanzregler, noch eine korrekte Zielposition vorgegeben ist. Die letzte Frage, die in dieser Arbeit analysiert wird, ist die Schätzung der optimalen Aufgabenparametrisierung einer zu lernenden Aufgabe aus lediglich einer Handvoll von Daten. Zur Beantwortung der ersten Frage wird eine neuartige Zustandsschätzung vorgeschlagen, die Erkenntnisse aus der Bayes-Filter-Theorie berücksichtigt, wie sie üblicherweise in der Roboter-navigation oder im Beobachterentwurf angewandt wird, aber nur von der Erfassung haptischer Daten abhängt. Die vorgestellte Methode wird in einer simulierten Umgebung evaluiert, in der physikalische Interaktionsdaten gesammelt werden und die Materialeigenschaften durch den Bediener gezielt eingestellt werden können. In der zweiten Frage wird eine neuartige Greifstrategie entworfen, die auf Industrierobotern angewandt werden kann, um eine Korrektur der Endeffektor-Lage zu ermöglichen, ohne dass zusätzliche Kraft-Drehmoment-Sensoren zwingend erforderlich sind. Schließlich wird ein neuartiges Lernsystem für einen Industrieroboter eingeführt, der Aufgaben nur aus einer Handvoll von Beispielen lernt. Diese Methode erweitert bestehende Arbeiten, indem sie nicht nur vorsieht, das zu erlernende Ziel grafisch zu beschreiben, um so nicht nur einen reduzierten Parameterraum des ursprünglichen Problems zu erhalten, sondern auch den zulässigen Zustandsraum der Parameter spezifisch zu modellieren, um so das Scheitern der Aufgabe direkt zu berücksichtigen. Beide Ansätze werden abschließend anhand von Laborversuchen evaluiert, wobei die gesammelten Daten die verbesserte Leistung im Vergleich zu den Basislösungen oder bestehenden Methoden verdeutlichen. Die vorgestellten Methoden ermöglichen es Robotern daher, ihre Fähigkeiten zur Manipulation unbekannter Objekte zu verbessern, wenn die Wahrnehmung unter Ungenauigkeit leidet.

Zusammenfassend enthält diese Arbeit eine breite Sammlung neuer Methoden, welche die Fertigkeiten und Fähigkeiten von Robotern verbessern und damit die Grundlage für zukünftige Roboteranwendungen schaffen. Da Forschung an sich nichts darstellt, was jemals zu Ende gedacht werden kann, schließt diese Arbeit mit einem kurzen Ausblick auf zukünftige Forschungsfragen und -aspekte.



# Contents

<b>Abstract</b>	<b>v</b>
<b>Zusammenfassung</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Dyadic HRC . . . . .	2
1.2 Multi-Agent Systems . . . . .	3
1.3 Manipulation of Unknown Objects . . . . .	4
1.4 Notation . . . . .	6
1.5 Thesis Outline and Summary of Research Contributions . . . . .	7
<b>Part I Interactive Action-Selection within HRC</b>	<b>11</b>
<b>2 Preliminaries and Background</b>	<b>13</b>
2.1 State-of-the-Art in Mutually Adaptive HRC . . . . .	15
2.1.1 Human Behavior Modelling . . . . .	15
2.1.2 Human-Aware Motion Planning . . . . .	16
2.1.3 Human-Centered Task Allocation and Planning . . . . .	17
2.1.4 Game-Theory within HRI . . . . .	18
2.2 Preliminaries . . . . .	20
2.2.1 Game-Theory . . . . .	20
2.2.2 Solving Games . . . . .	21
2.2.3 Environment Model and High-Level Planning . . . . .	23
<b>3 Legible Action Selection in HRC</b>	<b>25</b>
3.1 Introduction . . . . .	26
3.2 Nonverbal Legible Assembly Problem . . . . .	27
3.2.1 Model Overview . . . . .	28
3.2.2 Reward of Legibility . . . . .	29
3.2.3 Goal Inference . . . . .	30
3.2.4 Myopic Heuristic . . . . .	31
3.3 Feature-Based State Abstraction . . . . .	32
3.4 Experiments . . . . .	33
3.4.1 Experimental Setup . . . . .	33
3.4.2 Experimental Procedure . . . . .	34
3.4.3 Hypotheses . . . . .	35
3.4.4 Measures and Analysis . . . . .	35
3.5 Conclusion . . . . .	37
<b>4 Adaptive Action Selection in HRC using Game Theory</b>	<b>39</b>
4.1 Introduction . . . . .	40
4.2 Related Work . . . . .	41
4.3 Human Robot Collaborative Manipulation as a Game . . . . .	42
4.3.1 Interactive Action Selection Strategy . . . . .	42

4.3.2	Definition of Applied Utility Functions . . . . .	44
4.3.3	Solving the HRC-Game . . . . .	44
4.4	Application on the Robot . . . . .	45
4.4.1	Player-Specific Action Spaces . . . . .	45
4.4.2	Generation of Expected Trajectories . . . . .	45
4.4.3	Applied Utility Functions . . . . .	46
4.5	Experimental Evaluation . . . . .	49
4.5.1	Hypotheses . . . . .	50
4.5.2	Experimental Setup . . . . .	50
4.5.3	Results and Discussion . . . . .	51
4.6	Conclusion . . . . .	52
<b>5</b>	<b>A Conceptual Design for Interactive TAMP within HRC</b>	<b>55</b>
5.1	Modeling Sequential Decision-Making by Means of MGs . . . . .	56
5.2	Game-Theory Inspired TAMP . . . . .	57
5.2.1	Incorporate Human Preferences and Suboptimal Behavior . . . . .	58
5.2.2	Obtaining Robot Policies for the Markov game . . . . .	59
5.2.3	Updating Human States . . . . .	61
5.2.4	Generate and Execute Robot Motions . . . . .	62
5.3	Future Work . . . . .	62
<b>Part II</b>	<b>Learning Behavior-Policies in Groups of Artificial Agents</b>	<b>65</b>
<b>6</b>	<b>Multi-Robot Hierarchical Actor-Critic</b>	<b>67</b>
6.1	Introduction . . . . .	68
6.1.1	Related Work . . . . .	68
6.1.2	Contribution . . . . .	71
6.2	Preliminaries . . . . .	71
6.2.1	Markov-Games . . . . .	72
6.2.2	Multi-Agent reinforcement learning-problem . . . . .	73
6.2.3	Policy Gradient Methods . . . . .	74
6.2.4	Actor-Critic Methods . . . . .	74
6.2.5	Multi-Agent Actor-Critic Algorithms . . . . .	75
6.2.6	Hierarchical Actor-Critic . . . . .	76
6.3	Technical Approach . . . . .	76
6.3.1	Decentralized MARL Based on Stackelberg-equilibria . . . . .	77
6.3.2	Multi-Robot Hierarchical Actor Critic . . . . .	79
6.4	Materials and Methods . . . . .	82
6.5	Results . . . . .	83
6.6	Possible Extensions . . . . .	86
6.6.1	Applying Best-Response Policies on Competitive Environments . . . . .	86
6.6.2	Improving Convergence Behavior by Partially Centralized Learning . . . . .	86
6.6.3	Incorporating Model-Knowledge about Agent State-Dynamics . . . . .	87
6.7	Conclusion . . . . .	88

<b>Part III</b>	<b>Advanced Manipulation Tasks with Unknown Objects</b>	<b>91</b>
<b>7</b>	<b>Haptic Object Identification</b>	<b>93</b>
7.1	Introduction . . . . .	94
7.2	Problem Formulation . . . . .	94
7.3	Related Work . . . . .	95
7.4	Haptic Exploration . . . . .	96
7.4.1	Grid-Based Exploration . . . . .	96
7.4.2	Shape-Based Exploration . . . . .	98
7.5	Object Identification . . . . .	99
7.6	Evaluation . . . . .	100
7.6.1	Grid-Based Exploration . . . . .	102
7.6.2	Shape-Based Identification . . . . .	103
7.6.3	Material Parameter Estimation Accuracy . . . . .	104
7.7	Conclusion . . . . .	105
<b>8</b>	<b>Tactile Finger Grasping</b>	<b>107</b>
8.1	Introduction . . . . .	108
8.1.1	Related Work . . . . .	109
8.1.2	Contribution . . . . .	109
8.2	Problem Formulation . . . . .	110
8.3	Technical Approach . . . . .	110
8.3.1	Alignment Error Estimation Using Tactile Sensor Arrays . . . . .	110
8.3.2	Controller Design for Adaptive Grasping with an Industrial Robot . . . . .	113
8.3.3	Implementation Details . . . . .	115
8.4	Experiment . . . . .	116
8.4.1	Evaluating Communication Speed . . . . .	116
8.4.2	Evaluation of Proposed Grasping Strategies . . . . .	117
8.5	Conclusion . . . . .	120
<b>9</b>	<b>BOC in Graphical Skill-Models for Compliant Manipulation Tasks</b>	<b>123</b>
9.1	Introduction . . . . .	124
9.1.1	Terminology . . . . .	125
9.1.2	Related Work . . . . .	125
9.1.3	Contribution . . . . .	128
9.2	Problem Formulation . . . . .	129
9.3	Preliminaries and Background . . . . .	130
9.3.1	Meta Learning for Robots Using Graphical Skill-Formalisms . . . . .	130
9.3.2	Bayesian Optimization with Unknown Constraints . . . . .	131
9.4	Technical Approach . . . . .	132
9.4.1	Compliant Controller Design for an Industrial Robot . . . . .	133
9.4.2	Applying BOC on Graphical Skill-Representations . . . . .	134
9.4.3	BOC-Model and Acquisition Function . . . . .	136
9.4.4	Exploit Conditional Dependencies for Collected Samples . . . . .	138
9.4.5	Complexity Analysis . . . . .	139
9.5	Application Example - Screw Insertion . . . . .	141
9.6	Experimental Results . . . . .	146
9.7	Conclusion . . . . .	150

<b>10 Conclusion</b>	<b>153</b>
10.1 Interactive Action-Selection within HRC . . . . .	153
10.2 Learning Behavior-Policies in Groups of Artificial Agents . . . . .	154
10.3 Advanced Manipulation Tasks with Unknown Objects . . . . .	155
10.4 Recommendations for Future Research . . . . .	156
<b>A Summary of Software Modules</b>	<b>161</b>
<b>B Hyper-Parameters and Implementation</b>	<b>163</b>
<b>Glossary</b>	<b>167</b>
Acronyms . . . . .	167
Notation . . . . .	170
List of Symbols . . . . .	173
Automated Planning . . . . .	173
Control-Theory and System-Theory . . . . .	173
Graph-Theory . . . . .	174
Markov Game Variables . . . . .	174
Machine-Learning and Stochastics . . . . .	174
Constant Numbers . . . . .	175
Robot-Specific Variables . . . . .	176
State Spaces and Finite Sets . . . . .	176
SE(3)-Variables . . . . .	177
Variables for Part I . . . . .	178
Variables for Part II . . . . .	178
Variables for Part III . . . . .	178
List of Indices . . . . .	179
List of Operators and Functions . . . . .	182
<b>Lists of Algorithms, Figures and Tables</b>	<b>185</b>
<b>References</b>	<b>187</b>
<b>Own Thesis-Related Publications</b>	<b>218</b>

# 1

## Introduction

Over the last decades, robotic systems have become a core-component of a broad range of applications. While robots have become a standard component of industrial assembly halls ever since the third industrial revolution, the application range has been increasing ever since. As robots are able to be exposed to harmful conditions, where humans could not survive without proper equipment or tools, these applications have already exceeded the limits of our hemisphere (Yoshida and Wilcox, 2008). Eventually, plenty of research has been done in order to allow robots to support our everyday-life (Broekens et al., 2009, Kachouie et al., 2014). Nonetheless, the application of robot systems is directly dependent on the ability of robotic systems to cope with unforeseen systems and environments. While tremendous progress has been outlined in well-defined lab-environments, and even industrial robots perform tasks accurately in a repetitive, yet reliable, manner, the transition to generic tasks and full autonomy are still far from production readiness. Specifically, this performance and reliability is directly bound to the strong assumption on acting in well-defined environments, where environment perception uncertainties can be neglected.

Within generic practical applications it is merely impossible to guarantee for such an assumption without explicitly accounting for every possible scenario. Unfortunately, robotic systems are further constrained by computational and storage budgets as well as the cognitive limits of experts who were to define all these scenarios. Thus, the only reasonable path to follow is to allow robots to account for uncertainties and equip robots with proper methods to handle the stochasticity of nature during interaction with the environment.

Eventually, the sources, models and effects of uncertainty in complex systems such as robotic applications are almost infinite. Starting from the rudimentary perception-cognition-action-loop from Thrun et al. (2005) as broadly applied in robotic systems, the cognitive process of robotic systems is defined as a closed-loop where the sensory inputs perceived from the environments are directly subject to the actions taken by the autonomous agents. Thus, in order to properly interact with uncertain environments a robot system has to use appropriate representations – i.e., models – of the environment.

Therefore, this thesis tackles three main application areas in modeling the environment for autonomous agents: the interaction of a human and a robot, which has been established as human-robot interaction (HRI) and human-robot collaboration (HRC) in research, the interaction with other autonomous agents – usually denoted as multi-agent systems and the manipulation of unknown objects.

With these specific sub-applications in mind, we outline the dedicated research content and the derived research questions that are analyzed in the remainder of this thesis in the following subsections.

## 1.1 Dyadic Human-Robot Collaboration

The idea of humans and robots coexisting is directly embedded in the history of robots. In particular, the first usage of the term *robot* by Karel Capek in 1921 introduced robots as human-like machines (Harkins, 1962). Similarly, the first appearance of robots in the movie *Metropolis* in 1927 introduced a robot as the *Maschinenmensch* (Hall, 2021), namely the machine-human. While the pop-culture has mainly established robots as a threatening machine, researchers have started to investigate the safe interaction of humans and robots ever since the 1950s. Early approaches have focused on theoretical concepts such as the Turing-test (French, 2000) or Asimov’s Laws (Asimov, 1941). In contrast, the majority of robots have been applied as advanced machines in industrial production halls, until the robot *Shakey* was built as the first mobile robot that could reason about its surroundings in 1970 (Kuipers et al., 2017). A few decades later, the term HRC rose interest in research as a special subclass of HRI, which involves general interaction (Bauer et al., 2008, Grosz, 1996). In contrast, HRC describes the phenomena of humans and robots forming a team and thus combining their (often) complementary skill-set in order to accomplish a common task. This directly imposes a multitude of cognitive challenges that an artificial agent – often denoted as *cobot* in literature (Peshkin et al., 2001) – has to face. Besides the general challenges an autonomous agent has to solve, such as perceiving and estimating a proper model of the current environment given the various – but noisy – sensor readings, robots need to understand the human coworker and adjust the interaction process accordingly. For brevity, we omit further insights on recent research advances in the area of HRC in this chapter and forward the interested reader to Chapter 2, which closes this gap. Concisely spoken, the focus of this part of the thesis is the development of novel methods to obtain optimal action-assignments for the human and robot coworkers while accomplishing a joint task. This involves research aspects of autonomous planning, human-aware motion planning, human behavior modeling as well as decision-making concepts that range from stochastic Markov-models to game-theory. Within this general research area, the following research questions are evaluated.

### Research Questions

**How can robots estimate and track human suboptimal behavior within HRC?** In the context of HRC the majority of research focuses on learning a task or skill from human input or behavior. Nonetheless, human behavior is also subject to sub-optimality, especially if a task is subject from repetitive routines. For such tasks, robots have been proven to be reliable and high-performant. In the context of HRC, there may exist parts of a joint task, where a robot can profit from these advantages. This eventually results in the research questions, on how a robot can model task ambiguity that a human may be a cognitive burden for the human, while tracking the human state-of-mind w.r.t. this task ambiguity throughout the interaction. Eventually, a suitable decision framework is needed that allows a robot to distinguish between efficient execution and supportive behavior whenever the human coworker is in need of such.

**How can mutually interactive game-theory be applied for autonomous decision-making on robotic systems within HRC?** While a majority of research has focused on modeling the human system as either a static or stochastic black-box system, the direct

question arises, if it were beneficial to account for human decisiveness when selecting an action during the interaction with a human coworker. This involves the question on finding suitable objectives that describe human behavior. Given these objectives, the core question is given by finding the optimal policy for the autonomous agent that does not only optimize over the internal objective of the robotic agent, but also over the objective of the human counterpart.

### **How can robots account for incorrect models during the interaction with humans?**

This question builds upon the former that by obtaining a team-optimal policy for the human-robot team, the policy directly relies on the accuracy of the underlying objective for the human and robot, i.e., the applied interaction-model. Thus, the final question evaluated within this part of the thesis is given by the issue of evaluating and handling sub-optimal human performance, while the term *sub-optimal* directly relates to or depends on the current interaction-model.

These research questions are evaluated from a conceptual and application point of view in Part I. A fine-grained overview of the individual chapters is provided in Section 1.5.

## **1.2 Multi-Agent Systems**

The concept of multi-agent systems has risen interest within robotics, control theory, machine learning and economics for many years. Plenty of this research builds upon early theoretical concepts that were established on mathematical models that in return are motivated from cognitive science. In economics, the monetary flow has been analyzed as multi-agent systems by applying e.g., graphical games (Kearns, 2007). Within control-theory (CT) the main focus is set on stabilizing multi-agent behavior or system states (Lewis et al., 2013, Ma et al., 2017). In robotics most approaches have focused on finding optimal routing strategies for artificial agents (Hausman et al., 2015) or solving decentralized partially observable problems (Dibangoye et al., 2016). Eventually, the machine-learning community has seen a tremendous rise of interest in the aspect of multi-agent reinforcement learning (MARL), which intends to extend the results from single-agent reinforcement learning (RL) to the multi-agent domain. As a result, plenty of researchers proposed novel concepts on how to improve the performance of data-driven multi-agent learning. In contrast to the current trend in this area, this thesis proposes a novel schematic for MARL by introducing hierarchical learning and proposing to explicitly differentiate between internal agent states and external observations. While the actual incorporation of model-based controllers or model-based RL into MARL is beyond the scope of this work, the presented results fulfill the main requirements to further improve MARL by incorporating findings from other fields, such as robotics or CT. In short, this part of the thesis tackles the following research questions.

### **Research Questions**

**How can reinforcement learning in multi-agent systems be embedded into a hierarchical framework without relying on overly restrictive assumptions such as fully synchronous decisions and centralized learning?** Within recent MARL-research, the majority of approaches has applied decentralized execution with centralized learning. This

becomes non-trivial if a hierarchical system acts temporarily relaxed as the collected experience is prone to act asynchronously. As this directly contradicts the Markov-assumption of higher level actors, this part of the thesis evaluates a novel MARL-framework that allows for a decentralized execution and learning.

**How can the effect of hierarchical performance be evaluated against joint task performance?** Within a hierarchical framework an agent is usually asked to achieve an intermediate sub-goal in order to achieve a common sub-goal. While current approaches usually average these rewards by a joint – and often centralized – critic, this thesis evaluates new schemes on how to explicitly differentiate between joint rewards and internal agent costs. As this scheme is applied throughout a variety of robotic applications, this thesis evaluates the applicability within MARL-systems.

**How can basic model-knowledge about the individual agent-dynamics be embedded into model-free MARL without adding overly restrictive model and system assumptions?** Within MARL, the majority of approaches has focused on learning policies in a purely model-free manner. Even though some approaches have used model-knowledge during training, these models are most often solely used to train a parametric policy – most often a deep-neural network – from them. Therefore, this thesis evaluates the possibility of differentiating between agent-based and external observations in order to generate a hierarchical MARL-framework, that allows future work to directly incorporate model-knowledge during exploration or learning as needed.

These research questions are evaluated from a conceptual point of view using a simulated multi-agent environment to benchmark the performance of the proposed methods against recent state-of-the-art in Part II. A short summary in relation to the rest of this thesis and preliminary work is provided in Section 1.5.

### 1.3 Manipulation of Unknown Objects

Handling unknown environments has been a key-challenge for autonomous agents that has raised interest from CT, computer-vision, machine learning and information-theory. Within CT the majority of research has focused on allowing a stable interaction with unknown environments, e.g., the unknown shape of an object. In the context of computer-vision and machine learning the major focus is on detecting suitable features that can be mapped to the goal-space of a robot system to eventually solve the manipulation task. Famous examples for this are e.g., visual servoing (Espiau et al., 1992) or the peg-in-hole (Bruyninckx et al., 1995) task. In order to account for a stochastic environment, plenty of research has been outlined to improve robotic skill-sets using concepts from information-theory (Thrun et al., 2005). As the cognitive process within robotic systems is usually designed as a complex system, the handling of manipulation tasks in unknown environments is increasingly solved by means of end-to-end learning (Silver et al., 2017). Here, the suitable control-modalities or policies – as a direct mapping of visual sensor-data to robot control inputs – are directly regressed from tremendous amounts of data.



Even though these approaches have reached promising results, this thesis proposes alternative novel methods and instead explicitly incorporates model-knowledge as far as possible. Given the remaining uncertainties and model-insufficiencies, the representation of the environment needs to be refined from data. Eventually, this thesis also proposes novel concepts to apply a suitable control-strategy to accomplish manipulation tasks within partially observable domains. In order to manipulate unknown environments, a suitable robot controller is crucial. While plenty of research has resulted in various control-designs that are capable to compensate for unknown environments, such as hybrid force-position-control (Khatib and Burdick, 1986), force-control (Zeng and Hemami, 1997), impedance-control (Hogan, 1984), or computed torque (Middleton and Goodwin, 1986), industrial robots usually do not allow to command the required control-inputs to the robot. While for a high-frequent joint-velocity or joint-position control-interface, concepts such as admittance-control (Seraji, 1994) is a suitable alternative, this thesis explicitly focuses on proposing novel concepts for industrial robot platforms that do not possess such interfaces. In particular, this part of the thesis depicts three research questions that are motivated from specific robot-applications, which we outline below.

## Research Questions

### **How can robots refine their model knowledge about the characteristics of unknown objects if there is no vision data or insufficient vision data available?**

While tremendous progress has been outlined in visual perception of unknown environments (Dellaert and Kaess, 2017, Elfes, 1989), these works usually limit the sensory data-input of robotic systems to contact-independent sensors such as cameras or depth-sensors. In contrast, humans possess incredible cognitive skills in identifying and analyzing object properties by only obtaining haptic cues. Thus, this evaluates this research question by composing a novel method on how artificial agents can explore unknown objects by collecting haptic feedback in the form of contact forces.

### **How can industrial robots execute sensitive grasping skills if neither the robot hardware provides compliant control interfaces nor a force-torque-sensor is available to account for undesired interaction wrenches?**

Robot grasping in unknown environments is another line of research that has risen broad interest over many decades. In here, many approaches have focused on grasp synthesis (Bohg et al., 2014, Shimoga, 1996) and gripper design (MacKenzie and Iberall, 1994), while especially recently data-driven concepts such as from visual servoing (Pedersen et al., 2020) or grasping pose detection (Qian et al., 2020) have become the most prominent line of research. In contrast to these works, this thesis evaluates the question how an industrial robot, that usually does not allow for a compensation of faulty perception of the environment, can be used to successfully grasp an object by specifically evaluating the haptic sensory feedback obtained in the gripper fingers.

### **How can industrial robots efficiently learn compliant manipulation tasks within reasonable time, i.e., only from a handful of experimental trials?**

When interacting in unknown environments, robots are required to provide not only a suitable control-interface that allows for contact-tooling skills, but also to regress unknown task parameters from collected empirical evidence autonomously. Again, plenty of researchers apply purely data-driven

controllers (Silver et al., 2017), which inherits the necessity of tremendous data to be collected. As this is impractical for most real robot applications, this thesis proposes a novel concept that first introduces suitable control-modalities for an industrial robot that allow for compliant manipulation tasks, and secondly outlines a parameter-regression framework that can be used to learn a manipulation task given only a handful of experimental samples.

These research questions are evaluated in Part III, where a special emphasis is laid upon applying the proposed methods on industrial robots in order to increase the applicability and skill-set of existing robot platforms for future applications. Before we present the detailed outline of this thesis, we summarize the notation used for the remainder of this thesis.

## 1.4 Notation

A detailed list of the used acronyms, notation, symbols, indices and operators can be found at the very end of this thesis – starting from page 167. Individual symbols and functions are introduced in the remainder of this thesis and are unique. An exception is given by hyper-parameters  $\kappa$ , thresholds  $\zeta$ , as well as upper- and lower-bound limits  $\mathbf{lb}$  and  $\mathbf{ub}$ , which are solely defined by their indices. Similarly, indexing variables  $i, j, k$ , and  $m, n$  are only used to denote iterations or specific values of other containers.

In order to outline the notation or generic relations, we use an arbitrarily chosen placeholder variable  $\mathbf{p}$ . Given this placeholder variable as a scalar term, we denote vectors as  $\mathbf{p} \in \mathbb{R}^n$  and matrices as  $\mathfrak{P} \in \mathbb{R}^{m \times n}$ .<sup>1</sup> Explicit elements of matrices or vectors are denoted as  $[\mathfrak{P}]_{(i,j)}$ , while the transpose is denoted as  $\mathfrak{P}^\top, \mathbf{p}^\top$ . We denote norms as  $\|\mathbf{p}\|_i$ , i.e.,  $\|\mathbf{p}\|_2$  represents the euclidean norm. The identity vector and identity matrix are denoted as  $\mathbb{1}^p$  and  $\mathbb{1}^{p \times p}$ , and similarly as  $\mathbb{0}^p$  and  $\mathbb{0}^{p \times p}$  as the zero vector and zero matrix.

A temporal sequence of vectors  $\mathbf{p}$  over time is described as a trajectory  $\vec{\mathbf{p}} := (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_T)$ , using the time-index convention  $\mathbf{p}_t$ , where  $t$  represents the time. The first-order time derivative is denoted as  $\dot{\mathbf{p}}$ . In order to increase readability, the time indexing may be omitted and every variable is expected to be denoted as  $\mathbf{p}_t$ . For these cases, the temporal successor and predecessor are emphasized as  $\mathbf{p}' := \mathbf{p}_{t+1}$  and  $\mathbf{p}^- := \mathbf{p}_{t-1}$ , where  $:=$  expresses equal by definition in the scope of this thesis. Similarly, the posterior as e.g., in Bayesian inference is denoted as  $\mathbf{p}^+$ . In case an algorithm is run in a cyclic manner, the current iteration is indexed as  $\mathbf{p}_{[k]}$ .

If we refer to a member of containers such as sets, lists, vector, etc. we denote  $\mathbf{p}_i$  as a specific scalar value of the former. In contrast to vectors, lists are only used within algorithms, while we denote sets as  $\bar{\mathbf{p}}$ . For sets, we further denote the union as  $\cup$ , the intersection as  $\cap$ , the set-equality as  $\equiv$ , the difference as  $\setminus$  and the empty set as  $\emptyset$ . In order to represent the size of vectors, we use  $|\mathbf{p}|$ . If  $|\mathbf{p}|$  is applied on sets, the cardinality is used. Eventually, we denote hierarchical systems by denoting layer  $k$  as  $\{^k\} \mathbf{p}$ .

In the context of multi-agent settings,  $\mathbf{p}$  is the scalar variant that is not assigned to any specific agent. In contrast to that,  $\mathbf{p}^{(i)}$  represents a variable explicitly assigned to agent  $i$ , while  $(\mathbf{p})_{i \in N_{\mathfrak{A}}}$  denotes the *joint* team-analogue of said variable over all agents. For the sake of brevity,  $(\mathbf{p})_{i \in N_{\mathfrak{A}}}$  is most commonly denoted as  $\underline{\mathbf{p}}$ . Similarly,  $\underline{\mathbf{p}}^{(-i)}$  wraps all elements of  $(\mathbf{p})_{i \in N_{\mathfrak{A}}}$

---

<sup>1</sup>This convention does not hold for typed letters

except  $\mathbf{p}^{(i)}$ . Within dyadic HRC,  $\mathbf{p}^{(h)}$  expresses the variable being assigned to the human and  $\mathbf{p}^{(r)}$  to the robot. State-spaces are denoted in calligraphic letters, e.g., the action-space  $\mathcal{A} := \{\mathcal{A}^{(h)}, \mathcal{A}^{(r)}\}$ , with the exception of  $\mathcal{N}_{\mathbf{p}}$  expressing a Gaussian distribution over  $\mathbf{p}$ . In case multiple variables need to be indexed identically, we denote this by wrapping the dedicated variables in tuples, e.g., we simplify

$$(a, b, c)^{(-i)} := a^{(-i)}, b^{(-i)}, c^{(-i)} \quad .$$

In the context of stochastic variables, we denote probability density functions (PDFs) as  $\mathbb{P}[\mathbf{p}]$  and conditionally dependent PDFs as  $\mathbb{P}[\mathbf{p}_1 | \mathbf{p}_2]$ . Similarly,  $\mathbb{E}_{\mathbf{p}_1 \sim \rho(\mathbf{p}_2)}$  and  $\mathbb{V}\text{ar}_{\mathbf{p}_1 \sim \rho(\mathbf{p}_2)}$  symbolize the expectation and variance of random variable  $\mathbf{p}_1$ . This variable  $\mathbf{p}_1$  follows a PDF  $\rho(\cdot)$ , which depends on  $\mathbf{p}_2$ ; and where  $(\cdot)$  represents a blank input.

If  $\mathbf{p}$  is used to optimize an objective, the optimal solution is denoted as  $\mathbf{p}^*$ . Within regression or empirical evaluations, for which the actual ground-truth is known, we denote the ground-truth as  $\mathbf{p}^*$ . In case either the true optimum or ground-truth is estimated from collected experience, i.e., evidence, we denote the estimate as  $\hat{\mathbf{p}}$ , the currently best performing sample as  $\mathbf{p}^{\otimes}$  and the worst performing observed data sample as  $\mathbf{p}^{\ominus}$ . If a function is approximated by means of neural networks, we use  $\dagger\mathbf{p}$  to denote a *target network*.

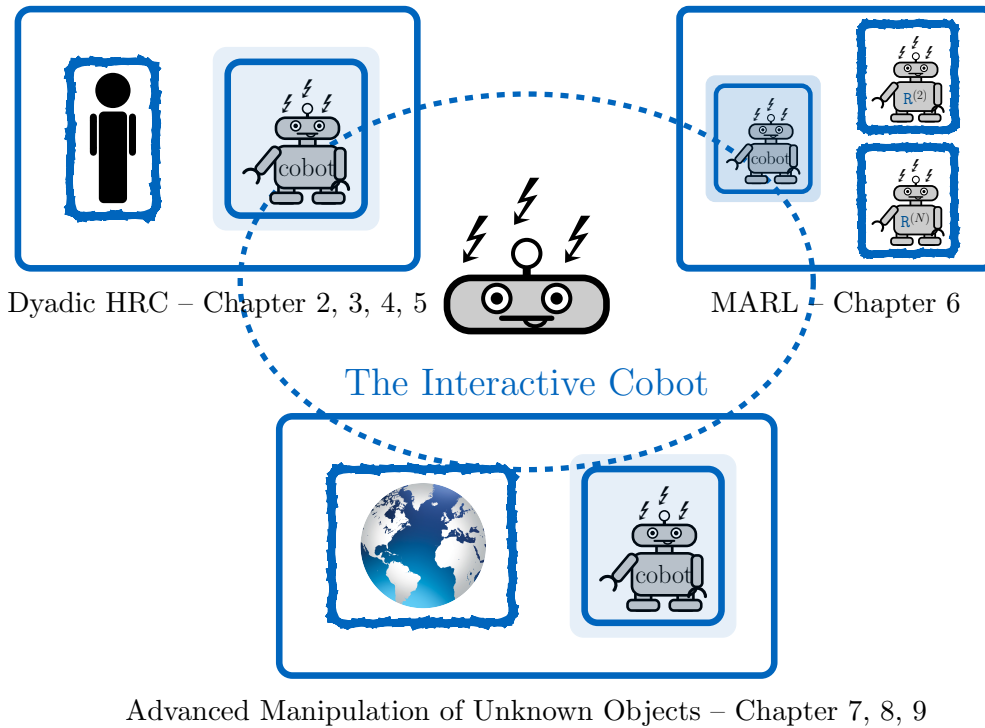
For kinematic robot chains, we refer to the origin of the chain as base  $\mathbf{ba}$ , the end-effector as  $\mathbf{ee}$ , and the tool as  $\mathbf{to}$ . Coordinate transformation matrices are denoted as  ${}^3T_{\eta}$  and rotation matrices as  $\mathbf{R}^{\mathfrak{z}}_{\eta}$ , as a transformation / rotation from  $\eta$  to  $\mathfrak{z}$ . We denote  $\mathbf{R}_{\varphi}, \mathbf{R}_{\theta}$  and  $\mathbf{R}_{\psi}$  as the rotation matrices around coordinate axes  $\mathbf{e}_x, \mathbf{e}_y$  and  $\mathbf{e}_z$ . Regarding translational notations,  ${}^{\mathbf{ba}}\mathbf{p}_{\mathbf{ee}\mathbf{to}}$  describes a vector  $\mathbf{p}$  pointing from the end-effector to the tool, expressed in the base-frame. If no explicit reference frame is provided, the variable is given w.r.t.  $\mathbf{ba}$  for robotic systems and w.r.t. the world-frame for generic settings.

Eventually, for Boolean values we denote *True* and *False* as  $\top$  and  $\perp$ , and denote a logical *AND* as  $\wedge$  and a logical *OR* as  $\vee$ .

## 1.5 Thesis Outline and Summary of Research Contributions

As mentioned above, this thesis tackles three orthogonal research areas in the context of robots interacting in partially observable domains, as depicted in Figure 1.1. As a result, the following chapters are grouped into three individual and self-contained sub-parts. For every chapter, a short abstract is added that summarizes the main content of each chapter while also highlighting any possible pre-publications that may exist for the dedicated chapter. In contrast, any chapter that misses such a reference contains a new contribution that has not yet been published. To summarize the contributions of this thesis w.r.t. scientific pre-publications, that have been collected within this thesis, a concise summary of the individual parts closes this chapter.

First, the advances in interactive HRC are outlined in Part I, as visualized in the upper left hand side of Figure 1.1. As this part contains a collection of individual methods, we first outline the current state-of-the-art in HRC in detail in Chapter 2. Furthermore, Chapter 2 sketches the preliminaries of the presented methods in Part I. Following this, we outline the scientific contributions in the context of dyadic interactive HRC along the subsequent chapters in the presented order:



**Figure 1.1:** Topological overview of the remainder of this thesis. Each box represents a sub-part of the thesis and consists of the dedicated chapters as listed below the dedicated box. For each individual part, the major source of uncertainty is highlighted by a noisy edge.

- a novel concept on incorporating legibility in the decision-making of robots within HRC is outlined in Chapter 3, which has been published in [Zhu et al. \(2017\)](#).
- the first application of game-theory in HRC is outlined in Chapter 4, which builds upon internal work ([Stahl, 2016](#)) and has been published in [Gabler et al. \(2017\)](#), [Ozgun et al. \(2016\)](#).
- finally the findings from Chapter 3 and Chapter 4 are summarized by outlining an extended interaction framework in Chapter 5. In detail, a conceptual framework for interactive HRC is outlined that explicitly takes human decisiveness into account, and bares potential for future extensions – such as task and motion planning.

Building upon the results on dyadic interactive HRC, the concepts of game-theory are transferred to the area of multi-agent interaction. Thus, the thesis enters the field of multi-agent domains – i.e., the upper right hand side of Figure 1.1 – namely MARL in Part II. Within this part, a novel hierarchical learning framework is proposed, that allows to combine model-based with model-free RL by introducing a dynamics-aware hierarchy and representation of the environment. The presented results have been motivated from the findings collected in Part I and [Ackermann et al. \(2019\)](#), and extend preliminary results collected in internal work ([Ackermann, 2018](#), [Krockenberger, 2019](#)).

Followed by the analysis of (multiple) decisive individuals interacting with each-other, the final part of this thesis outlines various – yet independent – aspects of robots interacting with unknown environments or objects, as visualized in the bottom of Figure 1.1. As each chapter

in Part III is independent of the others, regarding their application as well as the underlying method, we omit a separate introduction and conclusion for this chapter and rather classify this part as a collection of individual research projects. As a consequence, the summary and conclusion of Part III is found in the overall conclusion of this thesis in Chapter 10. Briefly summarized, the individual scientific contributions of the individual chapters in the area of robots interacting in stochastic environments are given as:

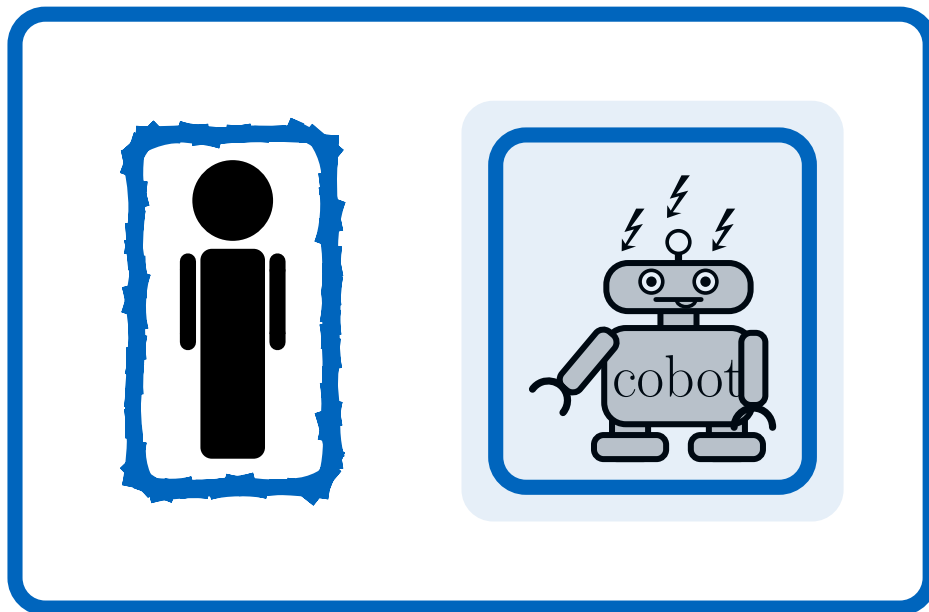
- Chapter 7 proposes a novel method for geometric knowledge refinement from haptic environment feedback. This method uses concepts from autonomous navigation, namely simultaneous localization and mapping. The presented work builds upon internal work (Maier, 2019) and has been published in Gabler et al. (2020b) as well as Gabler et al. (2020a).
- Chapter 8 outlines a novel grasping method, that is specifically tailored to industrial robots, which usually miss the opportunity of compensating for misalignment in a force-sensitive manner. The results have been published in Gabler et al. (2022b).
- Chapter 9 outlines the controller and the applied concepts on tuning the parameters online by means of Bayesian optimization with unknown constraints in a sample-efficient manner. The presented work is also available in Gabler et al. (2022a) and Gabler and Wollherr (2022).

Beyond the contributions to the state-of-the art as outlined in this thesis, a collection of available software-modules has been generated, for which a detailed list is found in Appendix A. Eventually, we close this thesis with a short summary of the results obtained within this thesis followed by a brief outline on future research directions or projects in Chapter 10.



## Part I

# Interactive Action-Selection within Human-Robot Collaboration







# 2

## Preliminaries and Background

### Chapter Abstract

This chapter outlines the research field evaluated within this part of the thesis, namely the interactive decision-making for human-robot collaboration. After introducing the general topic setting, the current state-of-the art in this field is sketched out in detail. Given the presented work, the scientific contributions in this research field will be outlined in the subsequent chapters.

Eventually, this chapter closes with introducing the fundamental methods used in the remainder of this thesis-part: game-theory and autonomous planning.

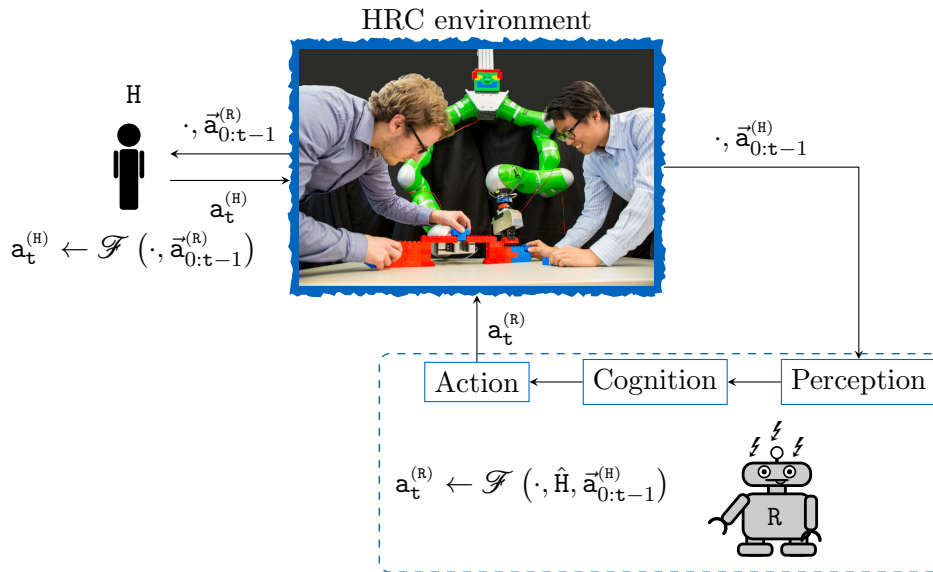
While experts in the field may skip this chapter, our main motivation is to familiarize the reader with the notation and the general problem setting before diving into the insights of our research work.

Within this part of the thesis, a special emphasis is set on the decision-making during the interaction of humans and robots. More precisely, this interaction is evaluated in the context of pursuing a joint task, which has been established in literature under the umbrella of human-robot collaboration (HRC) for industrial assembly (Bauer et al., 2008). Robots are omnipresent in manufacturing halls across a wide range of applications, since the third industrial revolution in the 1970s. Nevertheless, robots are still mostly limited to repetitive tasks and manipulation is error-prone when dexterous tasks are involved. Full automation of an assembly line may either not be possible or come with uneconomically high costs, particularly for small enterprises. The more plausible solution is to introduce HRC into industrial assembly lines to mutually complement human and robot strengths instead.

For a successful collaboration, however, all agents involved have to plan, choose and execute their actions considering the mutual interference of each action taken. The main challenge when interacting with a human is that unlike robots, humans do not necessarily follow the same sequence of actions even when a detailed plan is provided. Most state-of-the-art methods simply adapt the actions of robots to human behavior. Nonetheless, an artificial agent should be able to make use of human adaptivity or take this divergent behavior explicitly into account within an HRC-decision-making framework. We propose that one should not only analyze human behavior but also mutual interference among the human-robot team (HRT). Motivated by this, the following main research hypotheses have been evaluated within this part of the thesis:

- the actual human state is subject to partial observability. Within HRC it is thus beneficial to carry out an estimate over the current human state of mind by means of Bayesian beliefs. Using explicit transition models of the human state can thus improve human-robot interaction (HRI) and provide new capabilities to robotic agents.
- within an interaction of humans and robots, it is beneficial to not only model or predict human behavior as dynamic systems or to consider the worst possible outcome of an interaction. Instead, we propose that it is favorable to model humans as decisive individuals who adjust their decisions to the behavior of the robot.
- humans are acting as decisive individuals, which can be expressed mathematically. Given this, observed human behavior can be compared against expected human behavior to eventually derive the likelihood of predicting human behavior correctly.

As these research hypotheses have been evaluated across a broad variety of research projects, we continue with an extensive overview of the state-of-the-art in HRC and HRI. Based on this, we derive the exact sub-problems that are analyzed in detail in the upcoming chapters. The main emphasis of this part is laid upon the autonomous decision-making – i.e., the autonomous action-selection and task allocation among humans and robots. Nonetheless, the interaction-model visualized in Figure 2.1 highlights that the obtained feedback from humans is also subject to the actual execution of selected actions. Thus, the following literature review recaps recent advancements along the individual components of this interaction scheme. Still, it must be noted that this review does not cover the full range of research fields as prominent research fields e.g., learning from demonstration (LfD) (Argall et al., 2009, Atkeson and Schaal, 1997, Ravichandar et al., 2020), geometrical reasoning (Chen et al., 2011, Garg et al., 2020) or physical human-robot interaction (pHRI) (Ogenyi et al., 2021) are omitted. In addition, crucial aspects such as safety in HRI require to cover aspects that exceed the aspects of this



**Figure 2.1:** Simplified perception-cognition-action loop, cf. [Thrun et al. \(2005\)](#) for a generic HRC-scenario. In here the robot needs to generate a suitable behavioral model of the human coworker, who in return selects new actions based on previous robot behavior at each iteration.

literature review, cf. [Lasota et al. \(2017\)](#). Namely, we omit aspects of handling collisions and refer to [Haddadin et al. \(2017\)](#), [Zhang et al. \(2020b\)](#) in this regard.

## 2.1 State-of-the-Art in Mutually Adaptive Human-Robot Collaboration

The interaction in HRI can be categorized in three categories ([Choudhury et al., 2019](#)):

- myopic robot behavior – i.e., neglecting the influence of humans,
- interacting with a (stochastic) model – either manually designed or obtained from data,
- interaction with a decisive counterpart – using an internal cost-metric or objective.

While the former found broad application in early robot applications, research has spent most effort on improving the latter two. As a result, research projects have found that humans favor robots being adaptive ([Lasota and Shah, 2015](#), [Nikolaidis et al., 2017a](#)) and supportive ([Dragan et al., 2015b](#)) during collaboration. Eventually, some studies stated that the role of robots should go beyond solely being adaptive companions ([Schulz et al., 2018](#)). We continue with an insight into how these ideas have evolved in human behavior modelling.

### 2.1.1 Human Behavior Modelling

Early approaches in human behavior modeling within HRI such as human-motion prediction, human-intention prediction or human-preference learning often model humans as stochastic systems that can be identified from collected experience data ([Koppula and Saxena, 2016](#)).

Using these models, a robot is then able to adjust its own action online, thus distinctly improving the collaboration. Even though human-motion prediction is an imminent problem in this field and closely related to subsequent work of this part of the thesis, we refer to the detailed literature surveys ([Aggarwal and Cai, 1999](#), [Hiatt et al., 2017](#), [Rudenko et al., 2020](#)) for further insights. The work conducted in this thesis also contributed to the publications in [Dinh et al. \(2015\)](#), [Ozgur et al. \(2016\)](#) in this area, for which a subsequent publication has been produced in the meantime ([Oguz et al., 2018b](#)), which also provides a solid overview over said field. Some selective results on fitting stochastic parametric models for human motion models are given as Gaussian mixture models by [Mainprice et al. \(2016\)](#) or probabilistic motion primitives by [Maeda et al. \(2017\)](#). The evaluation of interaction dynamics proposed by [Jarrassé et al. \(2012\)](#) highlighted that even within pHRI human behavior can be classified based on the current interaction scenario.

Aside from human-motion prediction, additional work evaluated human decision-making. Preliminary findings from psychology, such as [Sebanz et al. \(2006\)](#), [Sebanz and Frith \(2004\)](#), indicate that predicting actions can be facilitated if all agents know the task and the collaborative environment. Consequently, humans tend to interpret actions w.r.t. a given goal ([Csibra and Gergely, 2007](#), [Gergely et al., 1995](#)). This is facilitated by the theory of rational actions ([Gergely and Csibra, 2003](#)) in the field of teleological reasoning ([Csibra and Gergely, 2007](#)) within cognitive science, where chosen actions pursue to achieve a common task. Early approaches, such as a collaboration architecture of [Schrempf et al. \(2005\)](#) apply dynamic Bayesian networks as a probabilistic approach to model human uncertainty and improve predictions by triggering task-specific reactions. On the other hand, [Broz et al. \(2013\)](#) provide an HRI system based on time-state aggregated partially observable Markov decision processes (POMDPs), such that robots must estimate the goals of human agents as hidden variables from observations.

Eventually, selective research projects have incorporated the concept of theory of mind to account for human cost-metrics. Initial models thus explicitly model human adaptation. Thus, [Nikolaidis et al. \(2017a\)](#) modeled human adaptivity as a Bayesian belief and obtained a robot policy by solving a mixed observable Markov decision process (MOMDP) ([Ong et al., 2010](#)). The basic analysis of human rationality on the other hand has been discussed in a collection of interesting theoretical work from human behavior studies in psychology and economics. Similarly, [Fridovich-Keil et al. \(2020\)](#) added an uncertainty-metric in the current human-cost metric to evaluate the confidence on predicted human behavior.

### 2.1.2 Human-Aware Motion Planning

Within human-aware motion planning, a key interest has been set on collision-avoidance, which requires a reliable estimation of human behavior. For example, [Kulic and Croft \(2005, 2006, 2007\)](#) proposed collision-aware motion planners, where the effective inertia is adjusted w.r.t. the distance between human and robot. Early approaches outlined collision avoidance by means of cost-maps, cf. [Sisbot et al. \(2007\)](#) and [Hoffman and Breazeal \(2007\)](#), who proposed co-navigation methods that incorporate human-aware safety metrics. This approach has also been applied directly from point-cloud data by [Flacco et al. \(2015\)](#).

Rather than accounting for most likely human motion, another line of research in human-aware motion planning lies in guaranteed safety. Thus, [Pereira and Althoff \(2018\)](#) proposed

a safe motion-controller by applying reachability analysis for human kinematics.

Another area of motion planning research that omits the interaction is given by pure optimization. In here the human behavior is directly optimized and optimal human behavior is assumed. Wang et al. (2018) propose a simplified point-mass model to generate human motion samples in combination with optimization over predefined objectives. In the aspect of optimization-based motion planning, state-of-the-art motion planning algorithms such as covariant Hamiltonian optimization for motion planning (Zucker et al., 2013), *TrajOpt* (Schulman et al., 2014), stochastic trajectory optimization for motion planning (Kalakrishnan et al., 2011) or Gaussian process motion planner (Mukadam et al., 2018) have been applied to HRC applications (Bari et al., 2021, Hayne et al., 2016, Oguz et al., 2017, Pellegrinelli et al., 2016). Closely related, various optimization-based motion planning algorithms assume simple dynamic systems to model human behavior, which eventually allows for dynamic obstacle avoidance (Alonso-Mora et al., 2018). On the other hand, a reverse scheme has been applied, where human collision avoidance behavior has been replicated on robotic manipulators, e.g., Oguz et al. (2018a).

While the approaches from above incorporate a human objective to be optimized, they neglect the aspect of interaction and only partially incorporate the stochasticity. In contrast to that, various approaches have focused on modeling human behavior as a stochastic – yet parameterized – system, that is identified from observed data. Famous examples are probabilistic motion primitives (Paraschos et al., 2018), interaction primitives (Ewerton et al., 2015) or a mixture of interaction primitives (Amor et al., 2014).

Eventually, the concept of theory of mind has been applied within motion-planning approaches, that intend to transfer information between humans and robots, such as legible or predictable motion (Dragan et al., 2013, Dragan and Srinivasa, 2014). Extensions of the work carried out within this thesis also allows to update human preferences from online feedback (Dinh et al., 2019).

### 2.1.3 Human-Centered Task Allocation and Planning

In the context of generating suitable decision-making algorithms for robots in HRC, the most prominent line of research lies in interacting with a (stochastic) model. In here the major challenge is given in predicting human decisions from (noisy) models, that a robot can then react from. To plan actions in the presence of humans, Gombolay et al. (2015) and Hawkins et al. (2014) outline promising methods of exploiting task knowledge in the presence of temporal uncertainty of human actions. Motivated from early results in multi-agent interaction planning (Mausam and Weld, 2008), timing is the major source of uncertainty. Alternative methods are given as timed Petri-nets (Chao and Thomaz, 2016) or Bayesian networks (Baraglia et al., 2017). In return, temporal aspects can also be included in the timing of robot actions to communicate intentions (Zhou et al., 2017). Besides temporal uncertainty Nikolaidis et al. (2015a) propose a method that is inspired by the way humans teach each other new tasks. They use MOMDPs and cross-training to train the underlying reward function for an HRC-scenario that incorporates human preferences. Exceeding the aspect of time, stochastic behavior models – such as human trust and fatigue – have been added to HRC approaches using Petri-nets (Hu and Chen, 2017, Wu et al., 2017). Similarly, Petri-nets have been proposed to cope with finding pareto-optimal allocations for multi-objective tasks (Feng

et al., 2016). Emphasizing the importance of trust (Hoff and Bashir, 2015) has also encouraged to track the human trust as a Bayesian belief (Chen et al., 2018). In this context of explicitly incorporating human uncertainty within an autonomous decision-framework, this thesis has contributed to the community by the results outlined in Chapter 3.

Similar to human-aware motion-planning, a second line of research solves the task-allocation problem for HRT by assuming a cost-metric for all agents and seeking towards the globally optimal team-policy. The approaches incorporate joint objectives in the form of cost-maps (Mainprice et al., 2012, 2011) or individual objectives that are solved by means of mixed-integer linear programs (Chen et al., 2014, Gombolay et al., 2018) or linear temporal logic (Guo and Dimarogonas, 2017).

An additional challenge is given by finding a feasible task allocation among human(s) and robot(s) for complex planning problems in the area of task and motion planning (TAMP). Early approaches of TAMP (Lagriffoul et al., 2014) have focused on bridging early results from the planning community achieved in the early 70s (Fikes and Nilsson, 1971) to the execution level of robots. Even for the single agent systems, coping with perception uncertainty is a key-challenge, often resolved by means of POMDPs (Kaelbling and Lozano-Pérez, 2013). Further adding optimization objectives have improved early results (Hadfield-Menell et al., 2015, 2013, 2016a), such that Toussaint (2015) proposed the first end-to-end optimization TAMP algorithm.

Building upon these results, human-aware planning frameworks have been developed. De Silva et al. (2015) outline the hierarchical agent-based task planner (HATP), which has also been integrated in full HRI frameworks (Pandey, 2012), in order to plan tasks jointly for humans and robots such that actions can be easily allocated among them. A hierarchical planning framework has been outlined by Darvish et al. (2021) that allowed for multiple task sequences to be adopted online by means of hierarchical planning. An initial proposal for an HRC top-down assembly planning architecture has been laid out by Johannsmeier and Haddadin (2017). In their model the human is incorporated based on a predefined cost- and performance metric which remains constant. Similar to the planning framework from Raessa et al. (2020), human actions are assumed to be given as discrete action-set of deterministic goal-poses. Introducing semi-Markov decision processes by Toussaint et al. (2016) in their relational activity process (RAP) allows to combine optimizing over task objectives while accounting for temporal uncertainty. Given this, this method was extended with relational human preference models by Munzer et al. (2017). These methods explicitly assume to have access to a well-defined human cost-metric – such as ergonomics (Busch et al., 2018) – but neglect the influence between robot and human actions.

### 2.1.4 Game-Theory within Human-Robot Interaction

Game-theory describes the mathematical study of decisive individuals interacting with each other. As robots are favored to act autonomously within everyday tasks even if humans are involved, it becomes evident that HRC forms a *game* of human(s) and robot(s). While early applications and formulations of game-theory have been developed for mathematics (Shapley, 1952) and economics, the first applications within robotics were multi-robot systems. In here, concepts like the Bayesian game approximation algorithm, also defined as partially observable Markov game (POMG) have been evaluated (Emery-Montemerlo, 2005). By modeling the

reactive behavior of other agents as stochastic transition functions, POMGs have been replaced by POMDPs (Kumar et al., 2015) or decentralized partially observable Markov decision processes (DEC-POMDPs) (Amato et al., 2007, 2019, Dibangoye et al., 2016, Omidshafiei et al., 2017) in order to achieve task-allocation for large teams of robots. Eventually, the concept of multi-agent task allocation has been using POMGs, POMDPs or DEC-POMDPs is covered intensively by the machine learning-community (Lanctot et al., 2017, Silver et al., 2016), which is covered in detail in Part II. Within control-theory, differential game-theory is mainly applied for robust control and decentralized control (Vamvoudakis and Lewis, 2011, Vamvoudakis et al., 2012). In here, robustness against disturbances (Jiao et al., 2016) or finding a stable team-consensus are the major objectives (Zhang et al., 2017). These approaches require full observability and controllability, which contradicts an application in HRC. Nonetheless, Li et al. (2015, 2016) proposed a human-robot interaction *game* as an application of differential game-theory in pHRI.

Similar to the concept of inverse reinforcement learning for human behavior modeling, the concept of inverse game-theory has been introduced, where inverse-equilibria (Waugh et al., 2011) for inverse games (Kuleshov and Schrijvers, 2015) are regressed from observations. In a dyadic context this is outlined as a cooperative inverse reinforcement learning by Hadfield-Menell et al. (2016b). The authors also outlined that it is non-trivial to design a suitable reward function by hand for interactive tasks (Hadfield-Menell et al., 2017a,b) and thus eventually proposed a factored POMDP solution for the cooperative inverse reinforcement learning problem (Malik et al., 2018).

The major field of robotic applications using game-theory are autonomous driving and mobile robotics. Within urban navigation, Turnwald et al. (2016) analyze the behavior of humans in interactive urban navigation from a game-theoretic perspective and state that the decisions of humans can be depicted as the pareto-optimal equilibrium of an interactive navigation-game. These findings have eventually been validated on empirical subject-experiments in Turnwald and Wollherr (2019). An early application in autonomous driving has been outlined by Bahram et al. (2015), who model autonomous driving on a crowded highway as a *game against nature*, in which an autonomous car plays a game against player *nature*, i.e., traffic, thus retrieving its driving trajectory as a result. Closely related to this, Sadigh et al. (2016a) propose a best-response policy approximation to ease up scaling, while also learning human-cost functions via inverse reinforcement learning. Recent concepts apply the concept of Stackelberg-equilibria and extensive-form games. Li et al. (2018a) apply a  $k$ -level decision framework where levels zero to two are evaluated and actually reflect the models proposed in Choudhury et al. (2019). Fisac et al. (2019) proposed a hierarchical game-theoretic planning framework for autonomous driving that exploited the concept of Stackelberg-equilibria for a strategic maneuver planner in combination with low-level controller that applied a simplified model predictive control.

The work presented in Chapter 4 – and in Gabler et al. (2017), Ozgur et al. (2016) – was the first application of game-theory within online decision-making for robotic manipulation. This idea has taken up by other researchers since then, e.g., Nikolaidis et al. (2017b) have applied a game-theoretic interaction-scheme to communicate a common task to a human. Similar to the experimental task of approach presented in Chapter 3, only the robot is fully aware of the current task.

## 2.2 Preliminaries

Before outlining the individual contributions collected within this thesis in the field of interactive HRC, this section outlines the preliminaries needed for the remainder of this thesis part. First, the examined setting needs to be clarified in detail. As mentioned before, this work aims in particular on well-defined environments, such as industrial HRC. Within these applications, the human and robot are fully aware of the task and the necessary sub-steps to achieve said task. In addition, the number of sub-steps is finite, such that the task can be solved within a finite time, where minimizing the overall time is a core objective of the collaboration. Regarding the human coworker, we expect humans to act partially rational, i.e., that on the one hand there exists an unknown cost-function for a human that allows to regress the human behavior policy, and that said behavior policy is not contradicting the objectives of the robot agent. Last but not least, we limit perception uncertainties of the environment to the behavior of the human subject rather than object or obstacle uncertainties.

### 2.2.1 Game-Theory

Game-theory describes the theory of decisive individuals interacting with each other. The basic representation of game-theory is defined as the normal-form game:

#### Definition 2.1: Normal-form game

Based on *Shoham and Leyton-Brown (2008)*, a finite normal-form game is fully defined by the following components:

- $\underline{\mathcal{A}} = \{ \mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \dots, \mathcal{A}^{(N_{\mathcal{A}})} \}$  is a finite set of  $N_{\mathcal{A}}$  agents.
- $\underline{\mathcal{A}} = \{ \mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \dots, \mathcal{A}^{(N_{\mathcal{A}})} \}$  is a collection of finite action-sets per agent  $i$ .
- $\underline{\pi} : \underline{\mathcal{S}} \times \underline{\mathcal{A}} \mapsto [0, 1]^{N_{\mathcal{A}}}$  is a joint policy that maps the state to an action for each agent.
- $\underline{\mathcal{J}} = \left( \mathcal{J}^{(1)}, \mathcal{J}^{(2)}, \dots, \mathcal{J}^{(N_{\mathcal{A}})} \right)$  are player-specific payoff functions

$$\mathcal{J}^{(i)}(\mathbf{a}^{(i)} \mid \underline{\mathbf{a}}^{(-i)}, \underline{\mathbf{s}}) \mapsto \mathbb{R},$$

that map the current action-profile  $(\mathbf{a}^{(i)}, \underline{\mathbf{a}}^{(-i)})$  at  $\underline{\mathbf{s}}$  to a numeric value for each  $i \in [1, N_{\mathcal{A}}]$ .

A normal-form game can thus be represented by an  $N_{\mathcal{A}}$ -dimensional payoff-matrix. Within the context of dyadic HRC, these components are given as  $N_{\mathcal{A}} = 2$ ,  $\underline{\mathcal{A}} := \{ \mathcal{A}^{(R)}, \mathcal{A}^{(H)} \}$  and  $\underline{\mathcal{A}} := \{ \mathcal{A}^{(H)}, \mathcal{A}^{(R)} \}$ . As a mathematical game can be designed for a variety of tasks, we introduce some key-properties from *Shoham and Leyton-Brown (2008)* below, which are of particular relevance for this thesis-part:

- **Finite Games:** A game of decisive individuals can be characterized as *finite* if and only if in a finite set of players each player can only choose from a finite number of provided actions.



- **Rational Players:** In a game of *rational players*, the players are assumed to maximize their (expected) payoff. This also implies that the player has access to a payoff value for each available action and player.
- **Perfect- and Complete-Information Games:** In a *complete-information game* it is assumed that the applied utility functions are known, but the moves of other players are unknown. In *perfect-information games* everything is known.
- **Zero-Sum Game:** In a *zero-sum game* the sum of all players is always zero. As a result, the payoff of one player is inversely coupled to the payoff of the opponent in a dyadic setting.
- **Cooperative Game:** In a *cooperative game*, players are able to communicate between each other or a selective sub-group of players.

Even though the term is not solely restricted to game-theory, the definition of the team-decomposition is of importance within an interactive context.

### Definition 2.2: Team Decomposition

Let  $\mathcal{A}$  be the set of all available actions to accomplish a given task  $\mathfrak{T}$  for a fixed HRT. If  $\mathcal{A}^{(H)} \equiv \mathcal{A}^{(R)} \equiv \underline{\mathcal{A}}$  holds, the HRT is said to be strictly homogeneous. In contrast to that, the HRT is said to be strictly heterogeneous if  $\mathcal{A}^{(H)} \cap \mathcal{A}^{(R)} \equiv \emptyset$  holds. The HRT is defined as a heterogeneous HRT if  $\mathcal{A}^{(H)} \cap \mathcal{A}^{(R)} \neq \emptyset$  and  $\mathcal{A}^{(H)} \neq \mathcal{A}^{(R)}$  holds.

We continue with sketching common methods on solving games, i.e., how to obtain a suitable team-policy.

## 2.2.2 Solving Games

Solutions for games are optimal strategies in which all agents maximize their payoff functional under the influence of other players. This can be best explained from the concept of best-response (br) and  $\varepsilon$ -best-response (ebr) policies.

### Definition 2.3: ( $\varepsilon$ -)best-response policy

Given a joint policy  $\underline{\pi}^{(-i)}$ , an action  $\mathbf{a}_j^{(i)}$  is called an ebr to  $\underline{\pi}^{(-i)}$  according to [Shoham and Leyton-Brown \(2008\)](#) if and only if

$$\mathcal{J}^{(i)}\left(\mathbf{a}_j^{(i)} \mid \underline{\pi}^{(-i)}\right) \geq \mathcal{J}^{(i)}\left(\mathbf{a}_k^{(i)} \mid \underline{\pi}^{(-i)}\right) - \varepsilon_{\text{br}}, \forall \mathbf{a}_k^{(i)} \in \underline{\mathcal{A}}^{(i)}, k \neq j,$$

i.e.,  $\mathcal{J}^{(i)}$  cannot be increased by more than  $\varepsilon_{\text{br}}$  by deviating from  $\mathbf{a}_j^{(i)}$ . If  $\varepsilon_{\text{br}} = 0$ ,  $\mathbf{a}_j^{(i)}$  is called a br.

[Nash \(1950\)](#) outlined a team-strategy that allows to converge an Markov game (MG) to a stable br-strategy collection, called *Nash-equilibrium (NE)*, which has also been extended to  $\varepsilon$ -Nash-equilibriums (eNEs) as the analogue optimal ebr-strategy by [Kearns \(2007\)](#).

**Definition 2.4: ( $\varepsilon$ -)Nash-equilibrium**

According to [Nash \(1950\)](#), an action profile  $\underline{\pi}^{\text{NE}}$  is a NE if and only if

$$\mathcal{J}^{(i)}(\underline{\pi}^{\text{NE}}) \geq \mathcal{J}^{(i)}(\mathbf{a}_k^{(i)} \mid \underline{\pi}^{(-i)}), \forall \mathbf{a}_k^{(i)} \in \underline{\mathcal{A}}^{(i)}, \mathbf{a}_k^{(i)} \notin \underline{\pi}^{\text{NE}}, \forall i \in \mathbb{N} \wedge i \in [1, N_{\mathfrak{A}}], \quad (2.1)$$

i.e., each player's action is a br to joint policy  $\underline{\pi}^{\text{NE}}$ .

Analogously to that, [Kearns \(2007\)](#) stated that an action profile  $\underline{\pi}^{\text{NE}\varepsilon_{\text{br}}}$  is an eNE if and only if

$$\begin{aligned} \exists \varepsilon_{\text{br}} > 0, \quad \text{s.t. } \mathcal{J}^{(i)}(\underline{\pi}^{\text{NE}\varepsilon_{\text{br}}}) \geq \mathcal{J}^{(i)}(\mathbf{a}_k^{(i)} \mid \underline{\pi}^{(-i)}) - \varepsilon_{\text{br}}, \\ \forall \mathbf{a}_k^{(i)} \in \underline{\mathcal{A}}^{(i)}, \mathbf{a}_k^{(i)} \notin \underline{\pi}^{\text{NE}\varepsilon_{\text{br}}}, \forall i \in \mathbb{N} \wedge i \in [1, N_{\mathfrak{A}}], \end{aligned} \quad (2.2)$$

i.e., each player's action is an ebr to joint policy  $\underline{\pi}^{\text{NE}\varepsilon_{\text{br}}}$ .

[Nash \(1951\)](#) further states that for any game there exists an NE in the space of mixed strategies. By applying Definition 2.1 to pure, i.e., deterministic strategies or policies  $\underline{\pi}$ , there is no guarantee that a NE exists for every game. Nonetheless, as stated by [Kearns \(2007\)](#), there exists an  $\varepsilon_{\text{br}}$  such that an eNE in the space of pure strategies can be found, such that no player is able to improve its expected payoff by more than  $\varepsilon_{\text{br}}$  by deviating from  $\underline{\pi}^{\text{NE}\varepsilon_{\text{br}}}$ .

While the concepts of NE describes a convergence for decision-problems, where all agents are acting identically, the concept of Stackelberg-equilibria introduces a leader-follower schematic at each decision-step.

**Definition 2.5: Stackelberg-equilibrium**

According to [Breton et al. \(1988\)](#), the global br-policy of NEs is simplified by analyzing the response-behavior of agents, if the leader reveals the action-choice. Thus, for each action of the leader, denoted as  $\mathbf{a}^{(j)} = \mathbf{a}^{(\text{leader})}$ , a Stackelberg-equilibrium is given as

$$\begin{aligned} \mathcal{J}^{(i)}(\underline{\pi}^{\text{SE}}) \geq \mathcal{J}^{(i)}(\mathbf{a}_k^{(i)} \mid \mathbf{a}^{(\text{leader})}), \forall \mathbf{a}_k^{(i)} \in \underline{\mathcal{A}}^{(i)}, \mathbf{a}_k^{(i)} \notin \underline{\pi}^{\text{SE}}, \\ \forall i \in \mathbb{N} \cap i \in [1, N_{\mathfrak{A}}] \cap \mathfrak{A}^{(i)} \neq \mathfrak{A}^{(\text{leader})}, \end{aligned} \quad (2.3)$$

i.e., each agent responds with its br-policy.

Eventually, the concept of pareto-optimality describes a joint policy that focuses on team-optimality rather than on consensus stability. The definition of pareto-optimality builds upon the idea of pareto-dominance.

**Definition 2.6: Pareto-dominance**

According to [Shoham and Leyton-Brown \(2008\)](#), pareto-dominance of an action profile is defined as

$$\mathcal{F}^{\text{pareto}}(\underline{\pi}_j, \underline{\pi}_k) := \begin{cases} \top & \text{if } \mathcal{J}^{(i)}(\underline{\pi}_j) \geq \mathcal{J}^{(i)}(\underline{\pi}_k) \quad \forall \mathfrak{A}^{(i)} \in \mathfrak{A} \\ \perp & \exists i, j, k : \mathcal{J}^{(i)}(\underline{\pi}_j) < \mathcal{J}^{(i)}(\underline{\pi}_k) \end{cases}. \quad (2.4)$$

Given this, the concept of pareto-optimality can be defined to obtain a team-optimal strategy within a multi-player game.

### Definition 2.7: Pareto-optimality and pareto-dominance

Given the definition of pareto-dominance from Definition 2.6, a pareto-optimal action profile is formally defined as

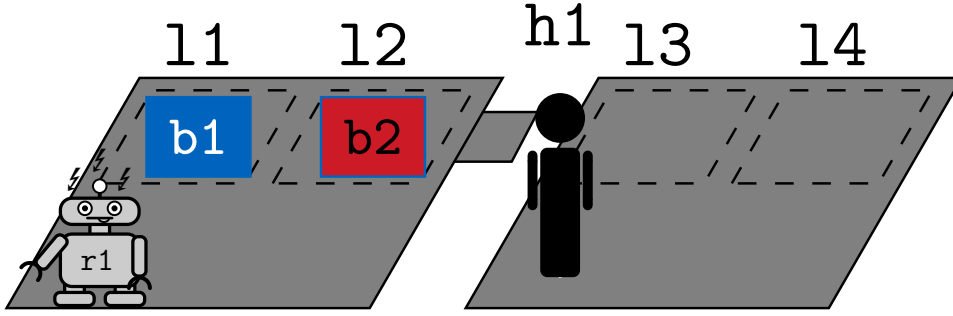
$$\mathcal{F}_{\text{dom}}^{\text{pareto}}(\pi_j) := \begin{cases} \perp & \exists k : \mathcal{F}^{\text{pareto}}(\pi_j, \pi_k) \\ \top & \text{else} \end{cases} . \quad (2.5)$$

### 2.2.3 Environment Model and High-Level Planning

In the following chapters, the term high-level planning is reappearing. As the definitions of terms like *task*, *action* or *skill* varies across literature and research field, we outline the concept of agent-centered planning as applied in the scope of this thesis. The presented concept is closely related to the RAP of Toussaint et al. (2016) and the HATP of De Silva et al. (2015). The RAP allows for concurrent actions – denoted as *activities*, i.e., to include temporal duration models for each action during planning, while the HATP solely applies *action-primitives* using first-order-logic (FOL). In the stochastic planning domain, concepts such as relational Markov processes (Dzeroski et al., 2001) and games (Finzi and Lukasiewicz, 2004) rely on a FOL-representation of the environment state. Thus, we denote the main components of logic programming <sup>1</sup>:

- An *entity*, or a *constant*, is a generic object or an agent within the planning domain, denoted as  $E$ .
- A *variable* is a placeholder for an entity.
- A *term* is either an entity or a variable.
- A *predicate* is a FOL symbol that represents relations or conditions of arbitrarily many entities.
- An *atom* represents either a term or a predicate applied on a tuple of terms in the form  $\mathbf{p}^o(\mathbf{p}^t_1, \mathbf{p}^t_2, \dots, \mathbf{p}^t_n) \leftarrow \top/\perp$  with Boolean value, where  $\mathbf{p}^o$  is a predicate, and each  $\mathbf{p}^t$  a term.
- A *fluent* represents a generalized version of an atom, which allows non-Boolean values to be incorporated.
- *conjunctions* represent sets of atoms and/or fluents.
- *formulas* are conjunctions of fluents or atoms, that can contain free variables.
- *grounds* are terms, formulas, etc. without free variables.

<sup>1</sup>We follow the Prolog notation, i.e., predicate symbols and constants begin with lower case, whereas capital letters express variables.



**Figure 2.2:** This figure shows a toy example for an HRC-planning problem: a collaborative *blocksworld* domain, cf. Nau et al. (2004). In here the knowledge base  $K(\underline{s})$  is used to drive an initial state to a desired goal state, each of which consists of a set of logical *grounds*.

Based on these definitions, the task-progress of an HRC scenario consisting of multiple entities can be represented as relational planning states in the form of logic formula groundings. In addition to logic atomic formulas, fluents contain non-boolean values, e.g., the weight of an object, to meet the requirement of continuous or categorical random variables. In the context of hierarchical planning, a *Planning Problem*<sup>2</sup> is then defined as  $K := (\underline{\mathcal{G}}, \underline{s}_0, D)$ , where  $\underline{\mathcal{G}}$  are predefined goal states,  $\underline{s}_0$  is the initial state and  $D$  describes the *Planning Domain*. The later is defined as  $D := (P_{\text{prim}}, M)$ , where  $P_{\text{prim}}$  is a collection of primitive actions and  $M$  contain abstract methods (Nau et al., 2004). These abstract operators both consist of relational conditions given as formulas, called preconditions and effects. The first one regulates if the operator is applicable at a given relation state, while the later contains the eventual modification of said relational state of the environment, when the operator is applied. The difference of these operators is given by their actual content. While we refer to Lloyd (1987), for further insights into logic computation, we visualize such a planning problem on a toy example. As pictured in the *Hierarchical Task Planning* frame of Figure 2.2, we outline a dyadic pick-and-place HRC-scenario, denoted as *collaborative blocksworld* in related work (Munzer et al. (2017), Natara-jan et al. (2011)). The domain entities consist of agents  $r1, h1$ , blocks  $b1, b2$  and locations  $l1, l2, l3, l4$ . By further defining *super-types*, e.g in here *agent-r1, h1, place-block, location*, the typed atoms  $\text{on}(\text{block}, \text{place}), \text{holds}(\text{agent}, \text{block})$  and fluent  $\text{weight}(\text{block})$ <sup>3</sup> can be used to describe the initial state  $\underline{s}_0$  depicted in Figure 2.2 as

$$\underline{s}_0 = \{ \text{blue}(b3), \text{blue}(b4), \text{on}(b1, l1), \text{on}(b2, l2), \text{weight}(b1)=1.3, \text{weight}(b2)=2.5} \}.$$

All atoms not listed in the state are assumed to have value  $\perp$  and unlisted variables and fluents are assumed to not exist. While a state has to be grounded, a goal state can also be provided as a set of relational formulas, e.g., for the example of Figure 2.2 the goal can be described as  $\text{on}(A, l4), \text{weight}(A) > 1.0$ . This goal subsumes all states in which either  $b1$  or  $b2$  are on  $l4$ . The domain is given by the method  $\text{put}(b\text{-block}, p\text{-place})$  and the activities/primitive  $\text{pick}(a\text{-agent}, b\text{-block})$  and  $\text{place}(a\text{-agent}, b\text{-block}, p\text{-place})$  with a duration of two time steps each. The solution is then either  $\text{put}(b3, b1)$  or  $\text{put}(b4, b1)$ , each of which resulting in the activity-sequence  $\text{pick} \rightarrow \text{place}$ .

<sup>2</sup>In the context of this thesis, the knowledge base symbolizes the planning problem of a task  $\mathfrak{T}$ .

<sup>3</sup>Note that units are neglected in this toy-example to improve readability.

# 3

## Legible Action Selection in Human-Robot Collaboration

### Chapter Abstract

This chapter focuses on the autonomous decision problem, where a robot interacts with a human-coworker that may have imperfect knowledge about the task that needs to be executed. As the robot has access to the actual task being pursued, a robot should choose its actions in such a manner, that these actions help the human coworker to identify the current task without the need of using explicit verbal or textual communication.

Thus, we formalize such problems in interactive assembly tasks as hidden goal Markov decision processes (HGMDPs) to enable the symbiosis of human intention recognition and robot intention expression. In order to avoid the prohibitive computational requirements, we provide a myopic heuristic along with a feature-based state abstraction method for assembly tasks to approximate the solution of the resulting HGMDP.

Eventually we evaluate the presented method in a user study with human subjects in a round-based LEGO assembly. In here, we compare our method against a purely efficient manner, that seeks to achieve the task in a minimum amount of time rather than seeking for supportive actions for the human collaborators. The collected empirical data support our claim that taking supportive actions help humans to identify correct tasks while also decreasing the overall error-rate of the human-robot team.

*Remark:* A majority of this chapter was previously published in [Zhu et al. \(2017\)](#).

## 3.1 Introduction

As sketched out in Chapter 2, there exists a high demand for human-robot collaboration (HRC)-applications. Introducing flexible robots into assembly lines and everyday activities brings in a broad variety of challenges to the application and eventually to the robotic systems. However, wherever there are such flexible applications available, the task description may also suffer from ambiguity of individual sub-tasks. For such scenarios it is not only the robot system that needs to cope with new challenges, but also the imperfect memory of humans. Thus, the robot needs to make its intention clear to the human, ideally without verbal communication, as installing communication modules for the robots can as well be uneconomic.

For example, consider an assembly robot that is limited to a set of nonverbal actions, how should it behave to *tell* the human collaborator which task to carry out? More specifically, given a partially accomplished task and the observed actions of the human collaborator, how can the robot make its next actions *intent-expressive*, or *legible*? To answer this question, we shortly recapitulate how human beings interpret actions of other agents.

Research in psychology suggests that human beings tend to interpret actions as goal-directed (Csibra and Gergely, 2007, Gergely et al., 1995), i.e., humans attribute goals to other agents, including robots (Kamewari et al., 2005), as the causes of their actions. One assumption of action understanding, known as teleological reasoning (Csibra and Gergely, 2007), is based on the principle of rational action (Gergely and Csibra, 2003), which states that actions have the purpose to realize goal-states by the most efficient means available. This suggests a formulation of action understanding as inverse planning or inverse reinforcement learning (Abbeel and Ng, 2004, Hadfield-Menell et al., 2016b, Ziebart et al., 2008), where efficiency is defined as maximizing the reward or minimizing the cost the agent receives in the environment. Taking a probabilistic perspective, Baker et al. (2009) proposed a framework based on Markov decision processes (MDPs) for action understanding and use Bayesian inference to compute the posterior probability of a goal, conditioned on observed actions and the environment.

Based on these research results, legibility as a property of actions can be characterized. Dragan et al. (2013), Dragan and Srinivasa (2013) define a legible motion as one that enables an observer to quickly and confidently infer the correct goal. They point out that while legibility and predictability sometimes can be correlated, they are not the same. A predictable motion is formalized as motion that matches the human collaborator’s expectation given a goal. That is, it is efficient with respect to the given cost or reward function for the goal, but a legible motion can be and is usually inefficient. Stulp et al. (2015) show that legible motions can also be generated using policy improvement through black-box optimization (Stulp and Sigaud, 2012), a model-free reinforcement learning approach, without knowing the underlying cost functions. They improve the robot’s motion through direct trial-and-error interactions with humans to decrease the time the humans need to infer the correct goal.

## Contribution and Outline

In this chapter, we extend the notion of legibility to multi-step human-robot cooperative assembly tasks where the assembly process is viewed as a sequential decision-making problem similar to the ones studied in Hoffman and Breazeal (2007) and Nikolaidis et al. (2015b). The

robot is required to establish a legible policy – a mapping from system states to actions – such that the human collaborator can infer the unknown task goal correctly from the partially built object as early as possible without verbal communication. We will refer to this as the *nonverbal legible assembly problem* in later discussion.

In contrast to motion planning, the *trajectory* of assembly tasks is the building process of the object, which is modeled in a discrete state space and affected not only by the robot but also by the human collaborator. Therefore, it is necessary for the robot to infer the human collaborator’s expectation of the task goal and adjust its policy accordingly. As legible actions can be inefficient, we argue that employing legible policies only when the human collaborator has a wrong expectation of the task goal, can avoid unnecessary inefficiency. Moreover, inference of the human collaborator’s expectation of the task goal is beneficial especially in scenarios where multiple goals are present, as disambiguating multiple goals simultaneously can be hard. A more practical strategy is to compute the probability distribution over the human collaborator’s expectation of the task goal and then choose the legible policy such that only the wrong goal expectation with the highest probability is deviated from.

The contribution of this chapter is to unify human intention recognition and robot intention expression in one framework by modeling the nonverbal legible assembly problem as a hidden goal Markov decision process (HGMDP), a special class of partially observable Markov decision processes (POMDPs) (Fern et al., 2014), where the goal is the only partially observable state variable. On the basis of the underlying task-related cost, or reward, we construct a special form of reward function that promotes legibility, drawing analogy from previous work (Dragan et al., 2013, Dragan and Srinivasa, 2013). The robot then maximizes the total reward of legibility it collects during the assembly process.

As solving a finite-horizon HGMDP is PSPACE-complete even for deterministic dynamics (Fern et al., 2014), another contribution of this chapter is to propose a myopic heuristic: we first learn legible policies offline in reduced fully observable MDPs, and then estimate the current human collaborator’s expectation of the goal online through belief updates in the original HGMDP and adjust the robot’s policy accordingly. In addition, we introduce a systematic way of state abstraction for assembly tasks to further limit the size of the state space.

In the remainder of this chapter, we first illustrate in more detail the proposed framework in Section 3.2 and the state abstraction method in Section 3.3. Then, we describe the human subject experiment and analyze the results in Section 3.4. Finally, we conclude this chapter in Section 3.5.

## 3.2 Nonverbal Legible Assembly Problem

We consider a nonverbal legible assembly problem in which the *robot*,  $R$ , has full knowledge of the task goal, while the *human*,  $H$ , does not. Moreover,  $R$  does not observe  $H$ ’s expectation of the goal directly; rather, it only knows a set of possible goals of  $H$  and has to infer it from  $H$ ’s actions during the assembly process.  $R$  maintains a probability distribution over the possible goal expectations of  $H$  and exploits this information to make the actual goal clear to  $H$  through its actions without verbal communication.

### 3.2.1 Model Overview

Formally, we model the problem as an HGMDP. Using a factored representation similar to Ong et al. (2010), we define it as a tuple  $(\mathcal{X}, \mathcal{Y}, \mathbb{P}[y_0], \mathcal{A}^{(R)}, \mathcal{A}^{(H)}, \mathcal{O}, \mathcal{T}_{\mathcal{X}}, \mathcal{T}_{\mathcal{Y}}, \mathcal{R}^{(R)}, \mathcal{R}^{(H)}, \mathcal{R}_{\text{leg}}, \gamma, \mathbf{y}^*)$ :

- $\mathcal{X}$  is a finite set of fully observable task states  $\mathbf{x} \in \mathcal{X}$ .
- $\mathcal{Y}$  is a finite set of partially observable states  $\mathbf{y} \in \mathcal{Y}$  representing the goal expectation of H, whose prior distribution  $\mathbb{P}[y_0]$  is given.
- $\mathcal{A}^{(R)}$  is a set of actions for R and  $\mathcal{A}^{(H)}$  is a set of actions for H that can be observed by R, i.e., the set of observations  $\mathcal{O} = \mathcal{A}^{(H)}$ .
- $\mathcal{T}_{\mathcal{X}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}') = \mathbb{P}[\mathbf{x}' | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}]$  and  $\mathcal{T}_{\mathcal{Y}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}', \mathbf{y}') = \mathbb{P}[\mathbf{y}' | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}']$  are factored transition probability functions of the system.
- $\mathcal{R}^{(R)}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)})$  and  $\mathcal{R}^{(H)}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(H)})$  denote the reward respectively for R and H taking the action  $\mathbf{a}^{(R)}$  or  $\mathbf{a}^{(H)}$  in state  $\{\mathbf{x}, \mathbf{y}\}$ .
- $\gamma$  is a temporal decay weight that balances long-term versus immediate rewards.
- $\mathbf{y}^* \in \mathcal{Y}$  denotes the actual task goal which is known beforehand only to R and which H and R have to achieve.

A transition in this HGMDP proceeds as follows:

1. given a system state  $\{\mathbf{x}, \mathbf{y}\} \in \mathcal{X} \times \mathcal{Y}$ , R makes an action  $\mathbf{a}^{(R)} \in \mathcal{A}^{(R)}$ , resulting in an intermediate task state  $\mathbf{x}'$ .
2. H makes an action  $\mathbf{a}^{(H)} \in \mathcal{A}^{(H)}$  according to a stochastic policy

$$\pi^{(H)}(\mathbf{x}', \mathbf{y}, \mathbf{a}^{(H)}) = \mathbb{P}[\mathbf{a}^{(H)} | \mathbf{x}', \mathbf{y}] \mapsto [0, 1],$$

and this leads to the next state  $\{\mathbf{x}', \mathbf{y}'\}$ . It has to be noted that in general  $\mathbf{x}' \neq \mathbf{x}$  holds.

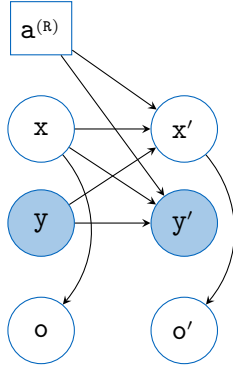
We assume that the transition of task states is deterministic with respect to the actions  $\mathbf{a}^{(R)}$  and  $\mathbf{a}^{(H)}$ . As a result, the stochasticity in the transition solely stems from H's policy.

In modeling the system, we only look at the states where R needs to make a decision; the intermediate states and the effect of H's actions are implicitly modeled in the transition probabilities of the system; hence H is modeled as part of the environment. For simplicity of notation, it is assumed that the task state  $\mathbf{x}'$  also encodes the preceding human action  $\mathbf{a}^{(H)}$ .

Now, R is required to maximize a special form of reward  $\mathcal{R}_{\text{leg}}$  promoting legibility in the dynamics defined above. The total reward is discounted in time by the factor  $\gamma$  to give less weight to rewards collected in the future. To formally define  $\mathcal{R}_{\text{leg}}$ , we first introduce two intrinsic reward functions of the task to characterize rational or efficient actions.

This reward is composed of a high-level reward that promotes similarity towards the goal  $\mathbf{y}$  and a low-level physical reward associated with the specific action. Hence, actions can have different rewards due to their energy consumption, difficulty, or safety, even if they have the same impact on the similarity towards the task goal.





**Figure 3.1:** The HGMDP system structure as a DBN. Shaded nodes are partially observable.

### 3.2.2 Reward of Legibility

Actions with high rewards defined above are greedily efficient; however, we want  $R$  also to take inefficient actions that can make the actual goal clear when  $H$  has a wrong goal expectation. To do that, we derive a reward function of legibility  $\mathcal{R}_{\text{leg}}$  for  $R$  from the principle of rational action, i.e.,  $H$  interprets  $R$ 's action by assuming  $R$  is acting efficiently towards the task goal

$$\mathbb{P}[\mathbf{a}^{(R)} | \mathbf{x}, \mathbf{y}] \propto \exp(\kappa_{\text{rat}}^{(R)} \mathcal{R}^{(R)}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)})), \quad (3.1)$$

where  $\kappa_{\text{rat}}$  is a parameter that  $H$  assumes how strictly  $R$  follows the principle of rational action.

Note that the policy for  $R$  assumed above by  $H$  is only optimal in one step; for  $R$  to achieve maximal accumulated reward till the termination, the corresponding POMDP must be solved. However, it is highly unlikely that  $H$  would have such computational capacity; therefore, we assume that it only considers a greedily efficient policy for  $R$ .

We assume that  $H$  does not infer the unknown goal from the whole trajectory at every time step; rather, it infers only based on the current state-action pair and tends to believe what it already believes, which is known as belief perseverance (Anderson and Ross, 1980) or cognitive inertia (Hodgkinson, 1997) in cognitive science. Thus, the system can be represented as a dynamic Bayesian network (DBN) as depicted in Figure 3.1.

Given the actual task goal  $\mathbf{y}^*$ ,  $R$  should choose an action  $\mathbf{a}^{(R)}$  in state  $\{\mathbf{x}, \mathbf{y}\}$  that increases the probability  $\mathbb{P}[\mathbf{y}^* | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}]$  while decreasing  $\mathbb{P}[\mathbf{y}' | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}], \forall \mathbf{y}' \neq \mathbf{y}^*$ , yielding a reward function of the form

$$\mathcal{R}_{\text{leg}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}) = \mathbb{P}[\mathbf{y}^* | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}] - \kappa_{\text{pnlty}} \sum_{\mathbf{y}' \in \mathcal{Y} \setminus \{\mathbf{y}^*\}} \mathbb{P}[\mathbf{y}' | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}], \quad (3.2)$$

where  $\kappa_{\text{pnlty}}$  is a tuning parameter that determines how much a wrong expectation should be penalized.

Considering the effect of cognitive inertia that  $H$  tends to believe  $\mathbf{y}'$  more if  $\mathbf{y}' = \mathbf{y}$ , we define

$$\mathbb{P}[\mathbf{y}' | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}] \propto \begin{cases} \kappa_{\text{bcnst}} \mathbb{P}[\mathbf{y}' | \mathbf{x}, \mathbf{a}^{(R)}] & \text{if } \mathbf{y} = \mathbf{y}' \\ \frac{1 - \kappa_{\text{bcnst}}}{|\mathcal{Y}| - 1} \mathbb{P}[\mathbf{y}' | \mathbf{x}, \mathbf{a}^{(R)}] & \text{otherwise} \end{cases}, \quad (3.3)$$

where  $\frac{1}{|\mathcal{Y}|} \leq \kappa_{\text{bcnst}} \leq 1$  denotes a coefficient indicating how much the human sticks to its previous belief. The probabilities above are computed using Bayes' theorem

$$\mathbb{P}[\mathbf{y}' | \mathbf{x}, \mathbf{a}^{(R)}] \propto \mathbb{P}[\mathbf{a}^{(R)} | \mathbf{x}, \mathbf{y}'] \mathbb{P}[\mathbf{y}' | \mathbf{x}]. \quad (3.4)$$

### 3.2.3 Goal Inference

The goal inference in HGMDP is achieved by updating the distribution of  $\mathbf{y}$  at each transition according to

$$\mathbf{b}_{\mathbf{y}'} \propto \mathcal{O}(\mathbf{x}', \mathbf{y}', \mathbf{a}^{(R)}, \mathbf{o}) \sum_{\mathbf{y}} \mathcal{T}_{\mathcal{X}, \mathcal{Y}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}', \mathbf{y}') \mathbf{b}_{\mathbf{y}}, \quad (3.5)$$

where

$$\mathcal{T}_{\mathcal{X}, \mathcal{Y}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}', \mathbf{y}') = \mathcal{T}_{\mathcal{X}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}') \mathcal{T}_{\mathcal{Y}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}', \mathbf{y}'), \quad (3.6)$$

and  $\mathcal{O}(\mathbf{x}', \mathbf{y}', \mathbf{a}^{(R)}, \mathbf{o}) = \mathbb{P}[\mathbf{o} | \mathbf{x}', \mathbf{y}', \mathbf{a}^{(R)}]$  is the probability of observing  $\mathbf{o}$  in state  $\{\mathbf{x}', \mathbf{y}'\}$  after  $R$  taking action  $\mathbf{a}^{(R)}$  in state  $\{\mathbf{x}, \mathbf{y}\}$ .

Recall that we encode the preceding human action in  $\mathbf{x}'$ ; hence, the observation function  $\mathcal{O}(\mathbf{x}', \mathbf{y}', \mathbf{a}^{(R)}, \mathbf{o})$  is deterministic

$$\mathcal{O}(\mathbf{x}', \mathbf{y}', \mathbf{a}^{(R)}, \mathbf{o}) = \mathbb{P}[\mathbf{o} | \mathbf{x}', \mathbf{y}', \mathbf{a}^{(R)}] = \begin{cases} 1, & \text{if } \mathbf{o} = \mathbf{a}^{(H)} \\ 0, & \text{otherwise} \end{cases}. \quad (3.7)$$

It can be easily seen from the DBN that  $\mathbf{x}'$  and  $\mathbf{y}'$  are conditionally independent given  $\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}$ . Thus, we obtain the transition probability

$$\mathcal{T}_{\mathcal{Y}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}', \mathbf{y}') = \mathbb{P}[\mathbf{y}' | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}]. \quad (3.8)$$

Furthermore, we assume that  $H$  always acts greedily efficiently according to its goal expectation

$$\pi^{(H)}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(H)}) \propto \exp(\kappa_{\text{rat}}^{(H)} \mathcal{R}^{(H)}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(H)})), \quad (3.9)$$

where  $\kappa_{\text{rat}}^{(H)}$  is a parameter that controls how strictly  $H$  follows the principle of rational action.

Since the uncertainty of the task state transition comes only from  $H$ , the transition probability is simply

$$\mathcal{T}_{\mathcal{X}}(\mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}, \mathbf{x}') = \mathbb{P}[\mathbf{x}' | \mathbf{x}, \mathbf{y}, \mathbf{a}^{(R)}] = \pi^{(H)}(\mathbf{x}', \mathbf{y}, \mathbf{a}^{(H)}), \quad (3.10)$$

where  $\mathbf{x}'$  is the intermediate state reached by  $R$  taking action  $\mathbf{a}^{(R)}$  in state  $\{\mathbf{x}, \mathbf{y}\}$  and  $\mathbf{x}'$  by  $H$  taking action  $\mathbf{a}^{(H)}$  in state  $\{\mathbf{x}', \mathbf{y}\}$ .

### 3.2.4 Myopic Heuristic

An optimal legible policy  $\pi_{\text{leg}}(\mathbf{x}, \mathbf{b}_y, \mathbf{a}^{(R)}) \mapsto [0, 1]$  selects actions for  $R$  to achieve the maximal accumulated reward of legibility. Unfortunately, solving HGMDPs is PSPACE-complete even for deterministic dynamics. Therefore, we will not seek exact solutions of this HGMDP; rather, we employ a myopic heuristic to approximate the legible policies. To that end, we first learn the optimal legible policy under each wrong goal expectation of the human collaborator and then switch between those policies according to the current belief state  $\mathbf{b}_y$  of the original HGMDP.

When  $H$  has a fixed wrong goal expectation  $y_i \neq y^*$ , the HGMDP is reduced to an MDP, i.e., the tuple  $(\mathcal{X}, \mathcal{Y}_i, \mathcal{A}^{(R)}, \mathcal{A}^{(H)}, \mathcal{F}_i, \mathcal{R}_{\text{leg}}, \gamma)$ , where  $\mathcal{Y}_i = \{y^*, y_i\}$  and

$$\mathcal{F}_i = \mathbb{P}[\mathbf{x}', y' | \mathbf{x}, y, \mathbf{a}^{(R)}] = \begin{cases} \pi^{(H)}(\mathbf{x}', y_i, \mathbf{a}^{(H)}) & \text{if } y' = y_i \\ 0 & \text{else} \end{cases}, \quad (3.11)$$

where  $\mathbf{x}'$  is the intermediate task state reached by executing  $\mathbf{a}^{(H)}$  in state  $\mathbf{x}$ .

In defining the reward of legibility, we still assume that  $H$  will virtually change its mind despite our assumption of fixed wrong goal expectation

$$\mathcal{R}_{\text{leg},i}(\mathbf{x}, y_i, \mathbf{a}^{(R)}) = \mathbb{P}[y^* | \mathbf{x}, y_i, \mathbf{a}^{(R)}] - \kappa_{\text{pnlty}} \mathbb{P}[y_i | \mathbf{x}, y_i, \mathbf{a}^{(R)}]. \quad (3.12)$$

We apply a standard Q-learning (Watkins and Dayan, 1992) algorithm to solve the MDP associated with each possible wrong goal expectation. An episode of Q-learning terminates when the probability  $\mathbb{P}[y^* | \mathbf{x}, \mathbf{a}^{(R)}]$  exceeds a threshold  $\zeta_Q$  or the actual goal is achieved. Thus, we obtain a legible policy  $\pi_{\text{leg}}(\mathbf{x}, y_i, \mathbf{a}^{(R)})$  for each wrong goal expectation  $y_i$ .

Recall that the distribution of  $y$  can be updated by (3.5) at each time step, which allows us to adjust the policy accordingly. A simple heuristic can be obtained as

$$\pi_{\text{leg}}(\mathbf{x}, \mathbf{b}_y, \mathbf{a}^{(R)}) = \pi_{\text{leg}} \left( \mathbf{x}, \arg \max_{y \in \mathcal{Y} \setminus \{y^*\}} \mathbf{b}_y, \mathbf{a}^{(R)} \right). \quad (3.13)$$

That is,  $R$  acts under the assumption that  $H$ 's expectation of the task goal is the one with the highest probability. For general POMDPs, such heuristics suffer from poor performance if the uncertainty is high in the belief state (Aberdeen, 2003), as the robot will not actively take *information gathering actions* on the hidden states. To alleviate this, some algorithms (Melo and Ribeiro, 2006, Sadigh et al., 2016b) incorporate entropy information in the reward structure to encourage the POMDP-agent to take actions that decrease the entropy of the belief state. However, our problem involves a special case that the legible actions are in fact *information gathering* in the sense that they increase the probability of the actual goal being inferred by  $H$ .

As legible actions can be inefficient, we let the robot switch to the greedily efficient policy once the probability assigned to the actual goal reaches a certain threshold, so as to prevent unnecessarily inefficient actions.

### 3.3 Feature-Based State Abstraction

In order to alleviate the effect of curse of dimensionality (Bellman, 1957), we provide a feature-based state abstraction method for assembly tasks. An assembly task can be seen as a combination of objects at the corresponding positions. We call a correct object-position pair a component  $\mathbf{t}$  and represent an assembly task  $\mathfrak{T}$  as a set of its components, i.e.,  $\mathfrak{T} = \{\mathbf{t}_1, \mathbf{t}_2, \dots\}$ . In a nonverbal legible assembly problem, the human collaborator is faced with multiple possible tasks  $\{\mathfrak{T}_1, \mathfrak{T}_2, \dots\}$ , from which we obtain the set of all task components  $\tilde{\mathbf{t}} := \bigcup_{i=1}^{|\mathfrak{T}|} \mathfrak{T}_i$ . For each component  $\mathbf{t}_i$ , we can find the set of tasks to which it belongs  $\mathfrak{T}_{\text{prnt},i} := \{\mathfrak{T}_j | \mathbf{t}_i \in \mathfrak{T}_j\}$ , to which we refer as parents of  $\mathbf{t}_i$ . It is not hard to see that different components can have the same parents, i.e.,  $\mathfrak{T}_{\text{prnt},i} = \mathfrak{T}_{\text{prnt},j}, i \neq j$ . We define an equivalence relation for such components

$$\mathcal{F}_{\text{equ}}(\mathbf{t}_i, \mathbf{t}_j) := \begin{cases} \top & \text{if } \mathfrak{T}_{\text{prnt},i} = \mathfrak{T}_{\text{prnt},j} \\ \perp & \text{else} \end{cases} . \quad (3.14)$$

Thus, a partition  $\tilde{\mathbf{t}}_{\text{part}}$  of  $\tilde{\mathbf{t}}$  can then be obtained as

$$\tilde{\mathbf{t}}_{\text{part}} = \{\mathbf{t}_i | \mathcal{F}_{\text{equ}}(\mathbf{t}_i, \mathbf{t}_j) \mapsto \top \quad i \neq j, \forall \mathbf{t}_i, \mathbf{t}_j \in \tilde{\mathbf{t}}_{\text{part}}\} . \quad (3.15)$$

As this set can be obtained for all available sub-tasks, we can rewrite this formulation as

$$\tilde{\mathbf{t}}_{\text{part}} = \{\mathfrak{T}_{\text{sub},i} | i \in \{1, 2, 3, \dots, |\tilde{\mathbf{t}}_{\text{part}}|\}\} , \quad (3.16)$$

where  $\mathfrak{T}_{\text{sub}}$  denotes the subtasks for  $i \in \{1, 2, 3, \dots, |\pi|\}$ .

Given an arbitrary task state  $\mathbf{x} \in \mathcal{X}$  and its corresponding ongoing task  $\mathfrak{T}_{\mathbf{x}}$  as a set of the components built in state  $\mathbf{x}$ , we count the number of built components for each subtask and represent the task state with these numbers. Formally, we define the following features

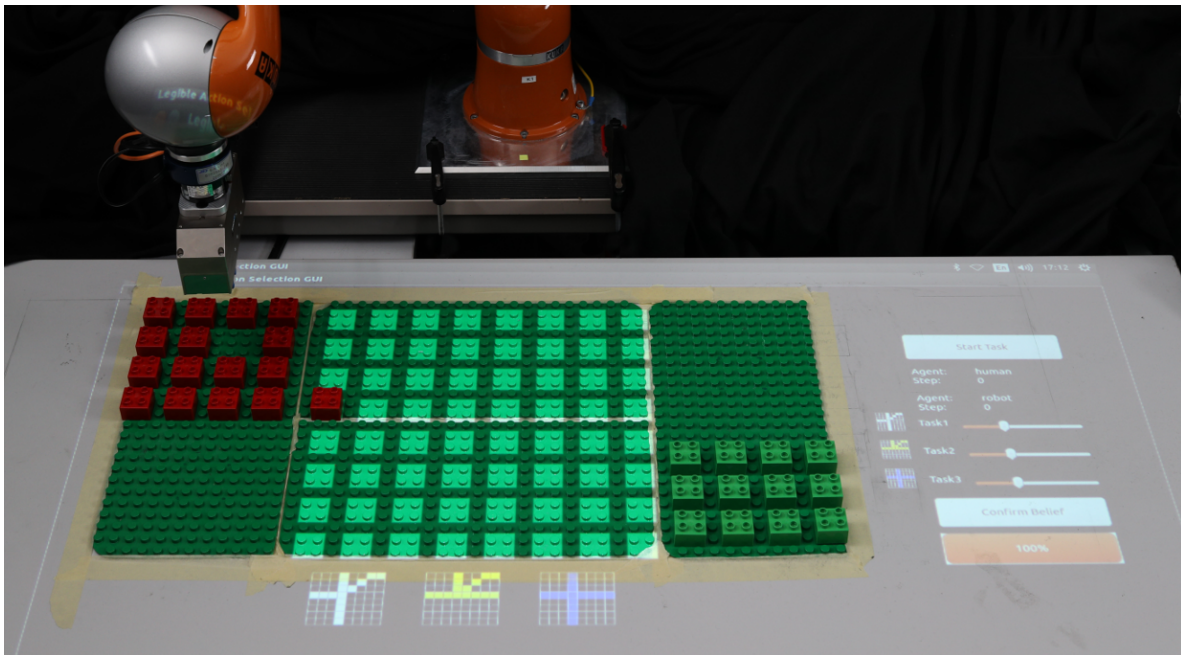
$$\Phi_i : \mathbf{x} \mapsto |\mathfrak{T}_{\text{sub},i} \cap \mathfrak{T}_{\mathbf{x}}| , \quad (3.17)$$

where  $\mathbf{x} \in \mathcal{X}$  and  $i \in \{1, 2, 3, \dots, |\tilde{\mathbf{t}}_{\text{part}}|\}$ .

Recall that we define a component as a correct object-position pair. Hence, a missing component can result either from a wrong object or a wrong position besides solely vacancy. We call such wrong object-position pairs *errors* and denote the number of errors by an extra feature  $\Phi_{\text{err}}$ . Here we assume that the number of errors is bounded by a maximal value  $\text{ub}_{\text{err}}$ . Together, the task state can be aggregated to  $\mathbb{R}^{|\tilde{\mathbf{t}}_{\text{sub}}|+1}$  by mapping the state  $\mathbf{x}$  to the feature vector

$$\mathbf{x} \mapsto \begin{bmatrix} \Phi_{\text{err}}, \\ \Phi_1, \\ \Phi_2, \\ \vdots \\ \Phi_{|\pi|} \end{bmatrix} =: \Phi . \quad (3.18)$$

From the abstract task state, a corresponding abstraction for actions follows naturally: when adding a new object, i.e., a component, this component belongs to a subtask according to the factorization outlined above. Thus, all features are increased for this specific action, while removing an object would decrease the feature counts. In case the action expresses an error, the features remain unchanged and the error-counter is updated.



**Figure 3.2:** HRC LEGO - assembly scenario with the goal being unknown to the human collaborator. Participants are asked to give their belief over the possible task goals via the sliding bar on the projected GUI.

## 3.4 Experiments

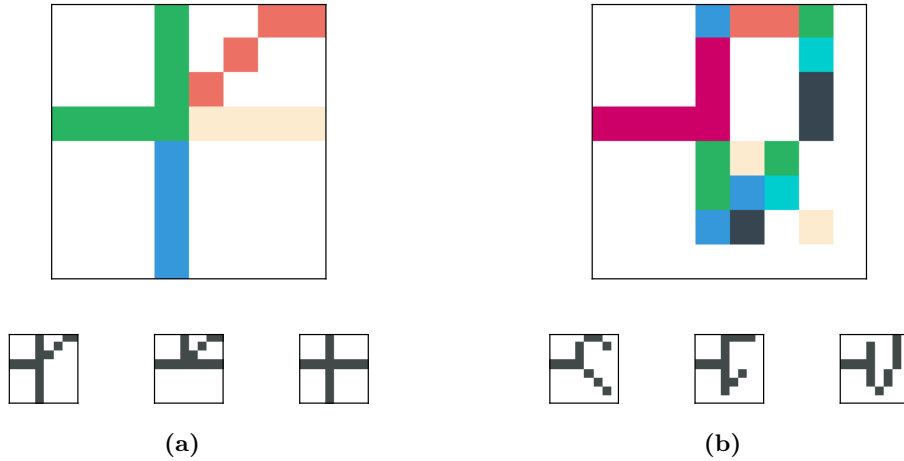
In this section we evaluate the proposed HGMDP in a real HRC-scenario based on an exemplary dyadic pick-and-place experiment with 10 individual subjects ( $\mu_{\text{age}} = 26.47$ ;  $\mu_{\text{bkgd}} = 2.3$  on a three point Likert scale ranging from no to professional robotics background).

### 3.4.1 Experimental Setup

We designed two pick-and-place scenarios in which three different tasks with overlapping subtasks according to (3.16) are given, as depicted in Figure 3.3.

The experimental setup depicted in Figure 3.3b was characterized by having distinct, overlapping and shared subtasks, whereas in the task scenario shown in Figure 3.3a no task had a distinct subtask. Each run was assembled by dyads in a round-based manner with the robot acting first. In order to collect consequent user feedback, a graphical user-interface (GUI) was projected upon the workspace from top as shown in Figure 3.2, which was used to obtain the human action  $\mathbf{a}^{(H)}$  and self-evaluated belief  $y$  over the task goals.

As solely asking for accomplishing the goal would result in barely any difference between the different policies mentioned above, the dyads were asked to assemble the given shape most efficiently, i.e., with the minimum overall travel-distance. This allows the investigation on our claim that a robot can deviate from the efficient policy to decrease the uncertainty of the human collaborator’s belief over the task goals.



**Figure 3.3:** Visualization of three pick-and-place goals for the two task scenarios. The subtasks  $\mathfrak{X}_{\text{sub},i}$  for state abstraction are visualized by color.

We compared three robot decision-making modes:

- *efficient policy* (E) In this mode the robot was acting purely efficiently, regardless of the human collaborator’s belief, thus assembling the closest component at every step.
- *HGMDP-based policy* (L) In this mode the HGMDP was applied as outlined in Section 3.2.
- *partially HGMDP-based policy with feedback* (LF) In this mode the HGMDP was partially applied. In contrast to L, the user-feedback replaced the HGMDP belief estimation.

### 3.4.2 Experimental Procedure

Upon arrival, all participants signed an informed consent form and were surveyed about their background. After this, the experimental setup was explained to the subjects in the form of written text, experimental trials as well as training examples until the subject agreed upon continuation.

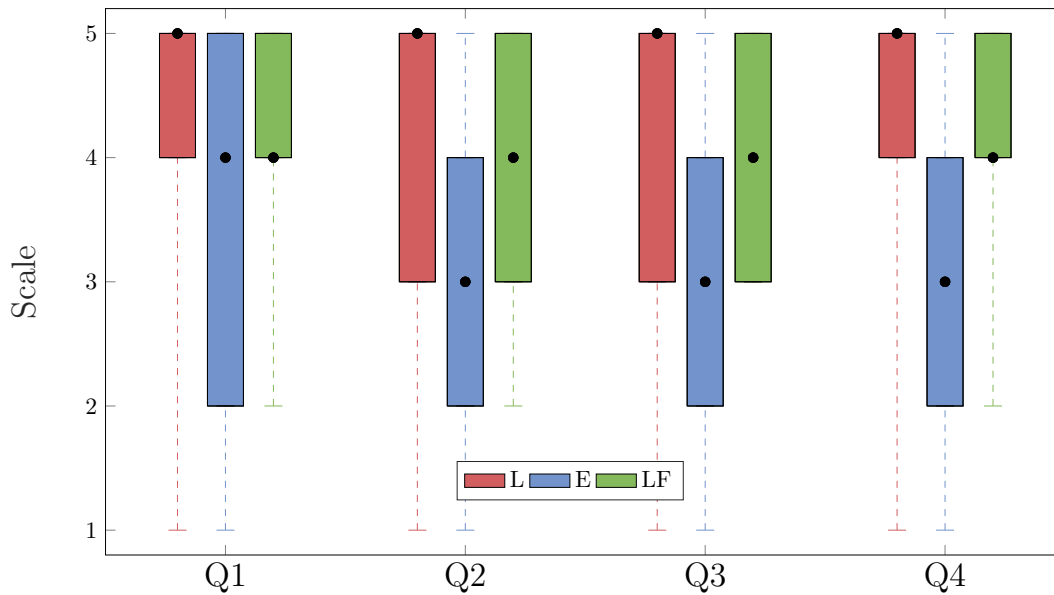
Each participant conducted 18 experimental runs such that each decision-making mode was performed 6 times and each scenario 9 times in no particular order. At the end of every assembly task, the participants were asked to answer the questionnaire shown in Table 3.1 in a five point Likert scale. Additionally, the subjects were asked to rate their belief of the task goals after each robot’s action via the GUI from Figure 3.2.

---

Q1	<i>The robot was acting efficiently.</i>
Q2	<i>The robot adapted the strategy when I was in doubt about the task.</i>
Q3	<i>The robot reacted when I made errors.</i>
Q4	<i>The choice of actions of the robot was helpful.</i>

---

**Table 3.1:** Questionnaire



**Figure 3.4:** Answers for each question are grouped by three different modes: L, E and LF. The upper and lower boundaries of the box represent the interquartile range. Whiskers above and below the box indicate the maximum and minimum value of the data. The median is marked by a black dot.

### 3.4.3 Hypotheses

We propose the following 4 hypotheses (**Hyps**) upon designing our algorithm to point out the performance and potential:

**Hyp1** - *Participants will agree more strongly that the robot’s actions are helpful and efficient in mode L or LF compared to E.* We claim that the efficiency and helpfulness of the robot’s actions perceived by the human collaborator is improved by the robot acting efficiently when possible and only selecting legible but inefficient actions when the human collaborator’s false belief requires it.

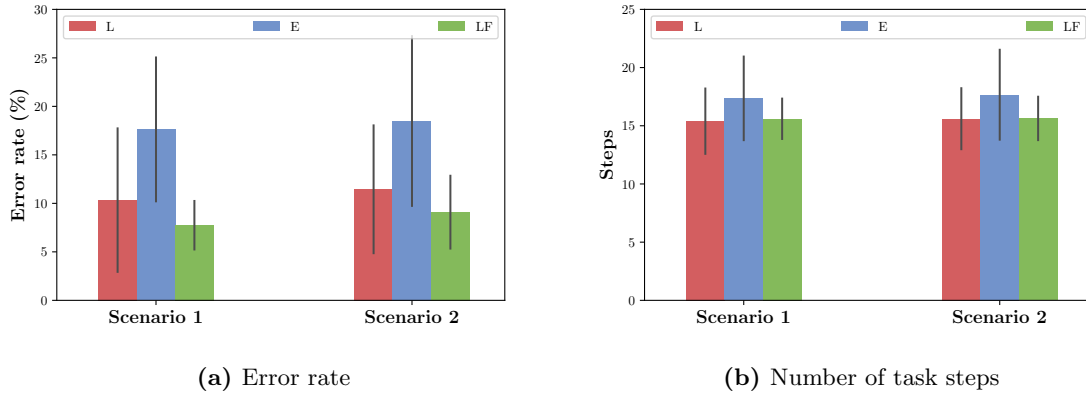
**Hyp2** - *Participants will agree more strongly that the robot’s actions are responsive in mode L or LF compared to E.* We claim that the proposed framework allows the robot to adjust its policy according to the inferred goal expectation of the human collaborator, leading to more responsive actions.

**Hyp3** - *Participants’ belief over the goal will converge faster to the correct goal in mode L or LF compared to E.* We claim that the legible policies applied by our framework enable the participants to infer the actual task goal more quickly.

**Hyp4** - *The overall error rate will be lower in mode L or LF compared to E.* We claim that an early intervention due to the legible policies helps the human collaborator recover from a wrong belief, thus resulting in lower error rates.

### 3.4.4 Measures and Analysis

The results of the participant surveys are reported in Figure 3.4. A Friedman’s test for overall comparison was conducted for each question, where the robot decision-making mode



**Figure 3.5:** Mean and standard deviation of the quantitative measures for L, E and LF.

is the treatment factor in which we are interested and the task scenario is the blocking factor whose effects need to be taken into account but are not of interest. Post-hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied, resulting in a significance level set at  $p < 0.017$ . The  $p$ -values are summarized in Table 3.2.

With a statistically significant difference, the participants agreed more strongly that the robot’s actions were efficient and helpful in mode L or LF, compared to E (Q1 and Q4). This supports **Hyp1**. Interestingly, we observed a higher variance of the answers for Q1 between the subjects in mode E. We attribute this to the possible different definitions of *efficiency* of the participants. While the robot’s actions in mode E were efficient in terms of travel-distance, they failed to convey the robot’s intention clearly and thus resulted in more steps on average to complete the task, which might be perceived as inefficient by some participants.

Furthermore, the participants agreed more strongly that the robot responded when they were in doubt of the task or made errors in mode L or LF, compared to E (Q2 and Q3). This supports **Hyp2** and suggests that the proposed framework was able to estimate the human collaborator’s belief and adjust its policy accordingly. The performance perceived by the participants seems comparable between the mode L and LF, however. To support this claim, an equivalence test is required in future work.

In order to further evaluate the performance of the proposed framework and the hypotheses mentioned above, we also consider the following quantitative measures.

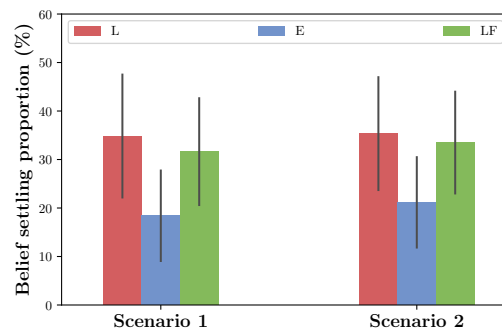
Questions	Overall Comparison	L vs E	L vs LF	E vs LF
Q1	<b>0.0009</b>	<b>0.0013</b>	0.8591	<b>0.0004</b>
Q2	< <b>0.0001</b>	< <b>0.0001</b>	0.2789	<b>0.0002</b>
Q3	< <b>0.0001</b>	< <b>0.0001</b>	0.5525	< <b>0.0001</b>
Q4	< <b>0.0001</b>	< <b>0.0001</b>	0.8552	< <b>0.0001</b>

**Table 3.2:** Subjective evaluation. Each cell holds  $p$ -values for overall & pairwise comparison. Statistically significant values are highlighted in bold.



- *Task completion steps* The total number of steps required by the human-robot team to complete the task is measured for all decision-making modes.
- *Error rate* As a direct measure of a false belief of the human collaborator, the number of errors during the tasks is divided by the number of the task completion steps. We remove the cases across all decision-making modes where the participants guessed the actual goal correctly and thus made no errors.
- *Belief settling proportion* During the experiment, the participants were asked to give their belief over the task goals after every robot action. We count the steps from the task completion where the human continuously has a correct goal expectation, i.e., the probability assigned to the actual goal is higher than 0.5, and divide it by the total steps of the task and refer to it as the belief settling proportion.

The quantitative measures show that compared to working with the robot in mode E, when the participants were working with the robot in mode L or LF, they had a larger belief settling proportion (Figure 3.6) and lower error rates (Figure 3.5a) on average, supporting our hypotheses **Hyp3** and **Hyp4**. As shown in Figure 3.5b, the participants also completed the task within fewer steps during the task on average in mode L or LF compared to E. Moreover, we observed that the variance of the task completion steps between the subjects was lower in mode L or LF, compared to E. This can result through the fact that while participants made more errors in mode E when they had a wrong goal expectation, there was a certain chance that they guessed the goal correctly from the beginning and thus completed the task within very few steps. As this can happen in the other two modes as well, a lower variance of the task completion steps further suggests that the decisions made by the robot in mode L and LF were more helpful in reducing the potential errors when the human collaborator had a wrong expectation of the task goal, as shown in Figure 3.5a.



**Figure 3.6:** Mean and standard deviation for the belief settling proportion for L, E and LF.

## 3.5 Conclusion

In this chapter we extend the concept of legibility in motion planning to the domain of sequential decision-making where continuous trajectories are replaced by discrete action-sequences. With one of the major challenges being the human actions as part of the system trajectory, we propose a framework based on HGMDPs in which the human collaborator’s expectation of the task goal forms the partially observable variable. As solving the resulting HGMDP is PSPACE-complete, policies in reduced fully observable MDPs are obtained offline, and selected according to the online human belief estimation in the original HGMDP.

We evaluated our algorithm through dyadic pick-and-place experiments. In this scenario, the robot deviates from the spatially efficient policy to make the actual task goal more clear

according to the estimated human belief. The experimental results confirm the proposed hypotheses with empirical measurements as well as subjective feedback.

Although our general framework is not limited to a specific task setup, the state abstraction method is only applicable for certain assembly scenarios where the potential tasks can be decomposed into object-position pairs. The belief estimation in the HGMDP could be further improved by incorporating richer observations such as eye gaze and hand gestures. Moreover, as the current algorithm only takes into account the selection of abstract actions, future work may consider the integration of legible motion planning into the execution of those abstract actions.

# 4

## Adaptive Action Selection in Human-Robot Collaboration using Game Theory

### Chapter Abstract

This chapter focuses on the task allocation among human(s) and robot(s) within collaborative tasks. Specifically, a framework based on game theory is presented that allows robots to choose appropriate actions with respect to the action of human coworkers when collaborating in close proximity. The proposed framework models human-robot collaboration (HRC) scenarios as iterative games and selects action-strategies for the human-robot team (HRT) by finding a suitable equilibrium of these games.

In contrast to most common approaches, our proposed HRC-game treats the decision-making behavior equally for all agents involved. Therefore, the concept of game theory is applied to evaluate the mutual interference of all actions on the HRT to obtain pareto-optimal Nash-equilibriums, i.e., team-optimal action-allocations.

The general framework of the proposed HRC-game is applied on an interactive pick-and-place scenario in close proximity. This exemplary HRC-game is tested in a lab-experiment with 30 human subjects. In this task a Kuka robot and a human coworker are asked to jointly assemble toy-bricks in close proximity, where our approach is compared to a non-adaptive baseline method. The experimental measurements and statistically significant improvements in the subjective feedback underline the potential that our proposed HRC-game allows to achieve towards an improved collaborative behavior for robotic applications.

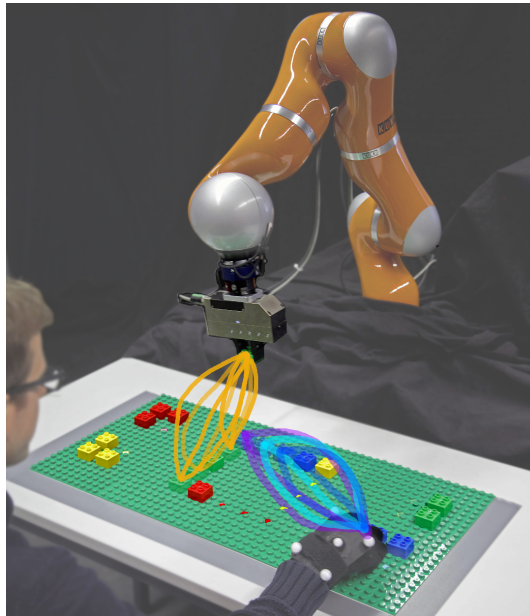
*Remark:* A majority of this chapter was previously published in [Gabler et al. \(2017\)](#) and builds upon internal project work ([Stahl, 2016](#)).

## 4.1 Introduction

As mentioned in Chapter 2, industrial automation has revolutionized manufacturing over the last decades and established robots in industrial assembly across a wide range of applications. Nevertheless, robots are limited to repetitive tasks and are error-prone when dexterous tasks are involved. As a full automation of an assembly line may not be possible or linked with uneconomic high costs in development and research, the more plausible solution is to introduce human-robot collaboration (HRC) into industrial assembly lines to combine human and robot strengths instead.

However, this combination requires both parties to be collaborative. An efficient collaboration requires all agents involved to choose their actions with respect to the mutual interference of each action taken, especially when working in a shared but confined workspace. As outlined in Section 2.2.3, we evaluate *actions* as *action-primitives*, such as `pick(robot, obj)`, where agent `robot` picks up object `obj`.

While humans adapt easily to new tasks and coworkers, a fully autonomous and flexible robot is still far from reality. The main challenge when interacting with a human is that unlike robots, humans do not always follow the same sequence of actions even when a detailed plan is provided. The scenario in Figure 4.1 gives an example of the challenges a shared workspace bears for all agents involved. The more confined a workspace, the harder the challenge of choosing not only the right path but also the best action to reduce mutual disturbance.



**Figure 4.1:** Exemplary illustration of an interactive action-selection process given multiple actions for a human and a robot to choose from. In this pick-and-place scenario the action-space corresponds to a set of reference trajectories across the workspace.

Therefore, it is important to analyze the mutual interference of each action. Thereby, the sequence of actions can be adapted on-the-fly w.r.t. the human coworkers, unlike classic robot planning in which a robot follows a predetermined sequence of actions. As it was outlined by [Lewkowicz et al. \(2013\)](#), humans predict other humans' actions through mental imagery.

Consequently, our framework is based on virtually evaluating the impact of single actions on other team members and vice-versa. Game theory models general decision problems from the perspective of equal and decisive individuals that consider the mutual interference amongst each other. In contrast to most other approaches, the reaction of the human is directly incorporated in the decision-process. Therefore, we propose an autonomous decision framework by applying the concept of game theory to an iterative action-selection algorithm applicable to actual dyadic HRC-scenarios.

An overview about related work in Section 4.2 demonstrates that the autonomous action-selection in an HRC-scenario has not yet been successfully tackled from a game-theoretic perspective. Therefore, the general representation of an HRC-scenario as an interactive game is proposed in Section 4.3. The application of this generic game on an initial HRC-pick-and-place scenario is outlined in Section 4.4. In contrast to the related work, the proposed HRC-game model is additionally evaluated in a real HRC-assembly-scenario in Section 4.5, as depicted in Figure 4.1. The last section concludes this chapter.

## 4.2 Related Work

When working in close proximity with robots, previous research projects have found that humans favor robots being adaptive (Lasota and Shah, 2015) and supportive (Dragan et al., 2015a) during collaboration. In order to achieve such a behavior, a lot of research has been conducted concerning autonomous planning of action-sequences within HRC over the last years.

In order to plan actions in the presence of humans, Gombolay et al. (2015) and Hawkins et al. (2014) outline promising methods of exploiting task knowledge in the presence of temporal uncertainty of human actions. Besides temporal uncertainty Nikolaidis et al. (2015a) use a Mixed Observability Markov Model and cross-training to train the reward function for an HRC-scenario that incorporates human preferences. Another focus is laid upon the incorporation of human motions in order to generate adaptive robot motions respectively. Mainprice and Berenson (2013) use a learned database of Gaussian mixture models (GMMs) to evaluate an executed human motion online. Maeda et al. (2017) and Lioutikov et al. (2017) outline an approach of using probabilistic motion primitives that combines prediction and control of the robot into one framework such that the robot's actions are learned in correlation with a human motion.

Even though all of these approaches improve the collaboration and interpretation of the human coworker distinctly, the action selection is narrowed down to an adaption w.r.t. a predicted yet given human action. In contrast to that, we propose an interactive human-robot decision framework based on game theory, thus taking into account the mutual influence of multiple actions for all agents involved.

Game theory has found a wide range of applications in multi-robot-systems (Emery-Montemerlo, 2005). Lately, game theory has also found its way to HRC scenarios. A general framework to classify and tackle different interaction types within joint manipulation has been proposed by Jarrassé et al. (2012). Similar to the approach of Li et al. (2015), physical human-robot

interaction (pHRI) is outlined as a differential game which depicts a multi-agent optimal control problem with a limited time-horizon that is therefore restricted to immediate dynamic adaptations and fails in reflecting long-term action goals.

First realizations of game theory in an human-robot interaction (HRI) scenario have been depicted by [Bahram et al. \(2015\)](#) and [Turnwald et al. \(2016\)](#). [Bahram et al. \(2015\)](#) propose a framework that models autonomous driving on a crowded highway as "game against nature" in which an autonomous car plays a game against player "nature", i.e., traffic, thus retrieving its driving trajectory as a result. Even though their approach is closely related to ours, their framework limits the players' payoff evaluation to the model of collision probabilities, thus limiting their method to evasive path planning. The same holds for [Turnwald et al. \(2016\)](#) who analyze the behavior of humans in interactive urban navigation from a game theoretic perspective and state that the decisions of humans can be depicted as the pareto-optimal equilibrium of an interactive navigation-game.

We propose an HRC-game model that depicts the decision process for an human-robot team (HRT) and outline a pick-and-place scenario as a first specified HRC-game scenario to highlight its potential in online use-cases. The framework can be used with various human motion prediction methods as analyzed by [Ozgur et al. \(2016\)](#). In contrast to [Ozgur et al. \(2016\)](#), this chapter provides a detailed outline about the decision framework in use. In difference to the approaches mentioned above, the presented approach sticks out in terms of reflecting multiple actions on an abstracted level rather than only adapting the motion towards a fixed goal-point. Additionally, an outline is given on how the proposed model is applied on a real robot system to show its functionality in HRC scenarios.

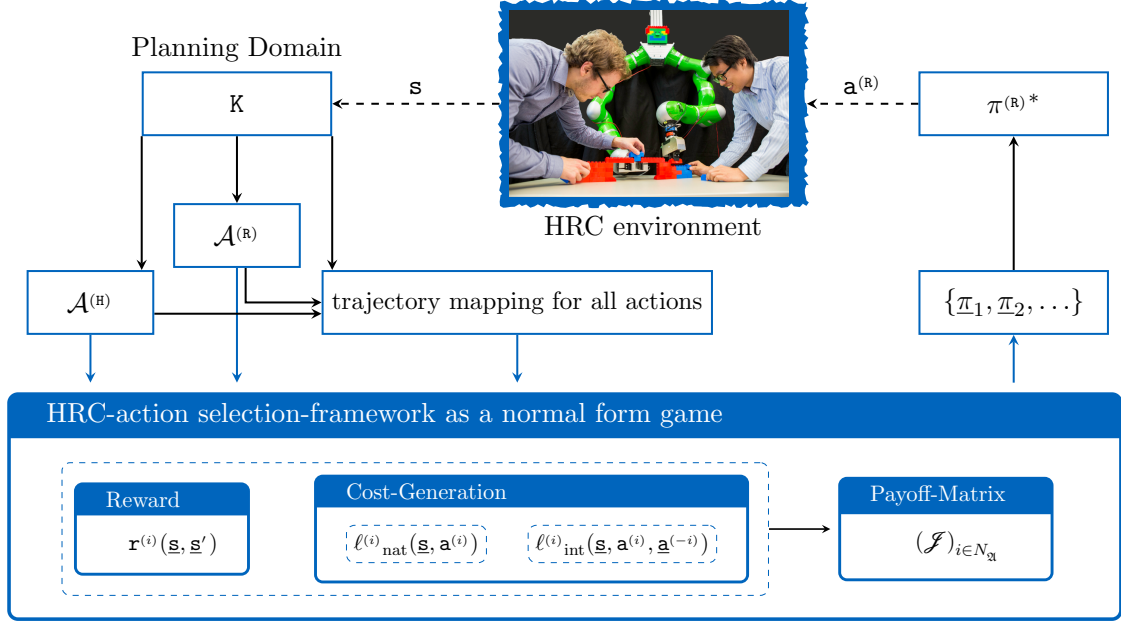
### 4.3 Human Robot Collaborative Manipulation as a Game

In this section an overview of our proposed HRC-game model and the assumptions it is built upon are given. Besides pointing out the core parts of our framework, it is explained how the team-strategy and robot action are obtained in an online collaboration scenario.

The proposed framework focuses on the interaction scenario of  $N_{\mathfrak{A}}$  agents collaborating in close proximity within a shared workspace. The main focus is set on the allocation of action-primitives which require an agent to perform reaching motions across the workspace. As a result, the action-selection needs to evaluate the evolution of each action-primitive across the workspace. In the following, a general way of including the dynamics of each action-primitive of such an HRC-scenario into a game theoretic framework is proposed.

#### 4.3.1 Interactive Action Selection Strategy

We propose to model the interactive action-selection problem from a game-theoretic perspective, that explicitly takes human performance objectives into account. Furthermore, we claim that the order of actions taken should not be fixed for generic HRC-scenarios. Therefore, the HRC-game is modeled in the Normal Form according to the according to Definition 2.1 which allows a simultaneous action-selection for all agents. The resulting action-selection game for dyadic HRC is depicted in the schmatic HRC-decision framework in Figure 4.2. In particular, a dyadic normal-form game consists of:



**Figure 4.2:** Overview of the proposed HRC-game decision framework

- $\underline{\mathcal{A}} := \{\text{H}, \text{R}\}$  as the human and robot *players*.
- $\underline{\mathcal{A}} := \{\mathbf{a}^{(\text{H})}, \mathbf{a}^{(\text{R})}\}$  is a finite set of total actions for H and R. According to Figure 4.2 finite action-sets  $\mathbf{a}^{(\text{H})}, \mathbf{a}^{(\text{R})}$  can be obtained by applying the knowledge base K on the current state  $\underline{\mathbf{s}}$ .
- $\underline{\pi} := \{\pi^{(\text{R})}(\underline{\mathbf{s}}, \mathbf{a}^{(\text{R})}) \mapsto \{0, 1\}, \pi^{(\text{H})}(\underline{\mathbf{s}}, \mathbf{a}^{(\text{H})}) \mapsto \{0, 1\}\}$ , as a joint dyadic policy.
- $\underline{\mathcal{F}} := \{\mathcal{F}^{(\text{H})}(\underline{\mathbf{s}}, \mathbf{a}^{(\text{H})}, \mathbf{a}^{(\text{R})}, \underline{\mathbf{s}}') \mapsto \mathbb{R}, \mathcal{F}^{(\text{R})}(\underline{\mathbf{s}}, \mathbf{a}^{(\text{R})}, \mathbf{a}^{(\text{H})}, \underline{\mathbf{s}}') \mapsto \mathbb{R}\}$  are individual payoff-functions that map the available action-sets and state-transition to a numeric value for H and R.

Furthermore, the normal-form game is characterized by the following properties:

**Finite Game:** Within a joint manipulation scenario, especially a dyadic setup, the composition of team-members  $\underline{\mathcal{A}}$  is known beforehand and thus finite. Furthermore, the current task  $\mathcal{T}$  and thus the planning-problem K result in finite action-primitive sets for all agents at any state. Therefore, the interaction scenario forms a finite game according to Section 2.2.2.

**Rational Players:** According to Section 2.2.2 we assume that each player tries to maximize their (expected) utility throughout the game. Specifically, they are expected to minimize their energy consumption and the likeliness of risky situations, i.e., collisions.

**Complete Information:** The applied utility functions are assumed to be known beforehand by all players whereas the chosen actions are not. This results in a game of complete information according to Section 2.2.2.

**Non-Constant-Sum:** Within a joint manipulation scenario, the agents are expected to work to a common goal, which contradicts the definition of a zero-sum game according to Section 2.2.2.

**Non-Cooperative:** In the HRC-game each player is modeled as an independent individual that performs the actions without communicating with other agents before choosing an action. This implies that our method is not restricted to robotic setups that allow verbal or visual communication. This results in a non-cooperative game according to Section 2.2.2.

### 4.3.2 Definition of Applied Utility Functions

In order to evaluate the total action set  $\underline{\mathcal{A}}$ , the utility functions  $\underline{\mathcal{J}}$  have to be mapped to each  $\mathbf{a}^{(i)}$  and each  $\mathfrak{A}^{(i)}$  respectively. Closely related to the definitions in Rohrmüller (2011), the player-specific payoff-metric

$$\mathcal{J}^{(i)}(\underline{\mathbf{s}}, \underline{\pi}, \underline{\mathbf{s}}') := \mathbf{r}(\underline{\mathbf{s}}, \underline{\mathbf{s}}') - \ell^{(i)}(\underline{\mathbf{s}}, \underline{\pi}) \in \mathbb{R} \quad (4.1)$$

within the HRC-game is modeled as the difference of reward  $\mathbf{r}$  and cost  $\ell^{(i)}$  for each player  $\mathfrak{A}^{(i)}$  in our HRC-game. Rewards result from accomplishing actions or tasks along a given shared plan, while costs are situation-aware penalty-functions which take the current dynamic state  $\mathbf{s}$  of the environment into account. Thus, the system is capable of a clear distinction between semantic planning and situation-aware reaction.

The situation-aware costs are additionally separated into two main components, inspired by the analysis of Turnwald et al. (2016):

**Native Costs**  $\ell^{(i)}_{\text{nat}} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \mapsto \mathbb{R}$  are self-reflective costs, e.g., the effort an agent has to spend.

**Interactive Costs**  $\ell^{(i)}_{\text{int}} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \times \underline{\mathcal{A}}^{(-i)} \mapsto \mathbb{R}$  result from the interference of multiple agents' actions,  $\underline{\mathcal{A}}^{(-i)}$  may contain any subset of agents in  $\{\underline{\mathfrak{A}} \setminus \mathfrak{A}^{(i)}\}$ .

### 4.3.3 Solving the Human-Robot Collaboration-Game

According to the definitions of game theory (Shoham and Leyton-Brown, 2008) the Nash-equilibrium (NE)

$$\pi^{(i)\text{NE}} = \operatorname{argmax}_{\mathbf{a}^{(i)} \in \mathcal{A}^{(i)}} \mathcal{J}^{(i)} = \operatorname{argmax}_{\mathbf{a}^{(j)} \in \mathcal{A}^{(j)}} \mathcal{J}^{(j)} = \pi^{(j)\text{NE}}, \quad \forall \mathfrak{A}^{(j)} \in \underline{\mathfrak{A}}, \quad i \neq j \quad (4.2)$$

formulates a solution in which no player  $\mathfrak{A}^{(i)}$  can improve its own pay-off by changing its action as long as the other players do not deviate from the current action profile.

By definition, a game has at least one mixed NE according to Shoham and Leyton-Brown (2008). In common payoff-games one often obtains not only single  $\underline{\pi}^{\text{NE}}$  but a set  $\{\underline{\pi}_1^{\text{NE}}, \underline{\pi}_2^{\text{NE}}, \dots\}$  of NEs for the underlying interaction scenario. In this case the team policy is chosen from this policy-subset based on further evaluation:

$$\underline{\pi}^* \leftarrow \operatorname{arg max}_{k=\{1,2,\dots\}} \mathcal{J}^{(i)}(\underline{\pi}_k^{\text{NE}}), \quad (4.3a)$$

$$\underline{\pi}^* \leftarrow \operatorname{arg max}_{k=\{1,2,\dots\}} \sum_{i=1}^{N_{\mathfrak{A}}} \mathcal{J}^{(i)}(\underline{\pi}_k^{\text{NE}}), \quad (4.3b)$$

$$\underline{\pi}^* \leftarrow \mathcal{F}_{\text{dom}}^{\text{pareto}}(\underline{\pi}_k^{\text{NE}}), \quad k=\{1,2,\dots\} \quad (4.3c)$$



thus finding either the NE that maximizes the payoff of a specific player (4.3a), the sum over all players' payoffs (4.3b) or the fairest NE as the pareto-optimal (Shoham and Leyton-Brown, 2008) solution (4.3c) from the given set of NEs.

The strategy of the HRC-game framework is visualized at a fixed sample-point  $\mathbf{t}_k$  in Figure 4.2. The game is iteratively replayed to adapt to unforeseen changes within the collaboration. A higher-level planning module is used to obtain a limited action set  $\underline{\mathcal{A}}$  given the current progress of the task  $\mathcal{T}$  for each player to assign the rewards  $\mathbf{r}^{(i)}(\underline{\mathbf{s}}, \underline{\mathbf{s}}')$  for each action primitive in the payoff-generation for each action. In addition to that, the scene evaluation module generates environment-aware cost-values  $\ell(\underline{\mathbf{s}}, \underline{\mathbf{a}})$  for each action. Based on the resulting payoffs, the abstracted HRC is evaluated as a normal-form game resulting in a  $N_{\mathcal{A}} \times N_{\mathcal{A}}$ -payoff matrix. The final policy is then obtained in a consecutive process, by first calculating all NE-candidates  $\underline{\pi}_k^{\text{NE}}$  and then (4.3c) to obtain the fairest team-strategy  $\underline{\pi}^*$ .

Based on the methods mentioned above, an interactive HRC-game setup is created, successfully applied to and tested on a robot platform. In the following section an insight is given into how the schematic overview given in Figure 4.2 is applied onto a robot platform in detail.

## 4.4 Application on the Robot

Even though the application of game theory in human-machine interaction scenarios has been analyzed before, the actual application on a real interaction scenario has not yet been shown. Especially a close proximity scenario such as collaborative pick-and-place exceeds the limitations of the methods proposed in previous approaches (Bahram et al., 2015, Turnwald et al., 2016). Therefore, a two-player game of a pick-and-place task with one robot and one human sharing a confined workspace is outlined in detail in this section.

### 4.4.1 Player-Specific Action Spaces

The semantic complexity for this initial game-realization is restricted on purpose to focus on the evaluation of the environment-aware action-selection of the robot rather than its semantic reasoning capabilities. Consequently, the semantic planning module returns either an equally weighted set of (pick, obj)- or (place, obj)-action-primitives for which the preconditions  $\mathcal{C}^{\text{pre}}(\underline{\mathbf{s}}, \mathbf{t}_k)$  are fulfilled at  $\mathbf{t}_k$ .

Regarding a pick-and-place task,  $\underline{\mathcal{A}}$  is given as a set of goal-points  $\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{|G|}\}$  each of which is matched to a reference trajectory for all players  $\underline{\mathbf{a}}^{(i)}$ . The purpose of the action-selection framework is then to obtain the team-optimal primitive allocation amongst the team  $\underline{\mathcal{A}}$ , thus minimizing the collision risks and effort for each agent involved.

### 4.4.2 Generation of Expected Trajectories

As the purpose of our framework is the allocation of optimal primitives among the HRT, the obtained trajectories only serve as an estimation of the evolution of each action primitive within the shared workspace. Therefore, a database of dynamic movement primitives

(DMPs) (Ijspeert et al., 2013) in Cartesian coordinates is recorded for the robot:

$$\mathfrak{s}_\tau \ddot{\mathbf{p}} = \alpha_{\text{DMP}} (\beta_{\text{DMP}} ((\mathbf{p}_{\text{des}} - \mathbf{p}) - \dot{\mathbf{p}})) + \mathcal{F}_{\text{shp}} \in \mathbb{R}^3. \quad (4.4)$$

Namely, these DMPs are used to obtain a characteristic shape via the excitation terms  $\mathcal{F}_{\text{shp}}$  and a scalable travel speed along the trajectory by alternating  $\mathfrak{s}_\tau$  for every  $\mathbf{p}_{\text{des}} \in \mathcal{G}^{(i)}$ .

By applying this trajectory-matching method given the current set of goal-points  $\mathcal{G}^{(i)}$  from the semantic planning module, a set of trajectories  $\vec{\mathbf{p}}^{(\text{R})}$  is obtained as  $\mathcal{A}^{(\text{R})}$ .

The human motions on the other hand are estimated using a minimum-jerk model for reaching motions towards all available goal-points as outlined in Dinh et al. (2015), resulting in  $\vec{\mathbf{p}}^{(\text{H})}$  as  $\mathcal{A}^{(\text{H})}$  for the human counterpart.

It has to be noted that the framework is not limited to these trajectory generation methods for neither robots nor humans in particular and that they are not evaluated in terms of accuracy within this chapter. Nevertheless, they clearly outline the application of the proposed HRC-game.

### 4.4.3 Applied Utility Functions

As mentioned in Section 4.4.1, the emphasis of our initial framework is set on environment-aware cost-evaluations similar to Hoffman and Breazeal (2007) propose. As a result, (4.1) mainly depends on a linear combination of cost-functions.

Our cost-functions are designed as a set of heuristic measurements as an abstraction from precise motion planning to evaluate the available reaching motions mapped on  $\underline{\mathcal{A}}$ .

**Native Costs** In particular, the native cost for player  $\mathfrak{A}^{(i)}$  in a pick-and-place scenario is proposed as a linear combination of three heuristic components

$$\ell_{\text{nat}}^{(i)} = \kappa_{\text{travel}} \ell_{\text{travel}}^{(i)} + \kappa_{\text{reach}} \ell_{\text{reach}}^{(i)} + \kappa_{\text{pref}} \ell_{\text{pref}}^{(i)}. \quad (4.5)$$

This term evaluates the travel effort  $\ell_{\text{travel}}^{(i)}$  the action requires, a reachability cost  $\ell_{\text{reach}}^{(i)}$  for each player based on the position of the object, and a preference cost  $\ell_{\text{pref}}^{(i)}$  to evaluate the direction of the motion in relation to the corresponding goal-point. The weighting terms  $\kappa_{\text{travel}}, \kappa_{\text{reach}}, \kappa_{\text{pref}}$  serve as linear weights to strengthen the impact of single cost-terms throughout the interaction.

1. The travel cost function

$$\ell_{\text{travel}}^{(i)} = \kappa_{\text{len}} \int_{\mathfrak{t}_k}^{T_{\text{max}}} \dot{\mathbf{x}}^{(i)} \, d\mathfrak{t} + \kappa_{\text{time}} \int_{\mathfrak{t}_k}^{T_{\text{max}}} d\mathfrak{t} + \kappa_{\text{dst}} \|\mathbf{x}_{\text{des}}^{(i)} - \mathbf{x}^{(i)}\|_2 \quad (4.6)$$

consists of the trajectory length from the current hand or tool center point (TCP)  $\mathbf{x}^{(i)}$  at the current time step  $\mathfrak{t}_k$  to the final time step  $T_{\text{max}}$ , the travel-time difference

from  $\mathbf{t}_k$  to  $T_{\max}$ , as well as the Euclidean distance between  $\mathbf{x}^{(i)}$  and the desired goal-point  $\mathbf{x}_{\text{des}}^{(i)}$ . The additional weighting terms  $\kappa_{\text{len}}$ ,  $\kappa_{\text{time}}$  and  $\kappa_{\text{dst}}$  are used to include a reliability-metric of estimated trajectories into our framework, such that

$$\sum_{p \in \{\text{len}, \text{time}, \text{dst}\}} \kappa_p = 1 \quad (4.7)$$

is satisfied.

2. The limited range of each player is modeled as a Gaussian-shaped penalty-function

$$\ell^{(i)}_{\text{reach}} = \begin{cases} \infty & \text{if } d^{(i)} > d_{\max} \\ \exp\left(\kappa_{\text{shp}} \left(\frac{d^{(i)} - d_{\max}}{d_{\max}}\right)^2\right) & \text{else} \end{cases}, \quad (4.8)$$

that respects the distance  $d^{(i)}$  from the agents shoulder or base  $\mathbf{x}_{\text{ba}}^{(i)}$  to  $\mathbf{x}_{\text{des}}^{(i)}$ . The maximum distance  $d_{\max}$  signifies the range limit of each player. The decline of the Gaussian shape factor  $\kappa_{\text{shp}}$  can be fit by exemplary human-human recordings.

3. Additionally, the preference cost for an agent to change the currently executed action to an alternative one is introduced. Therefore, the angular difference

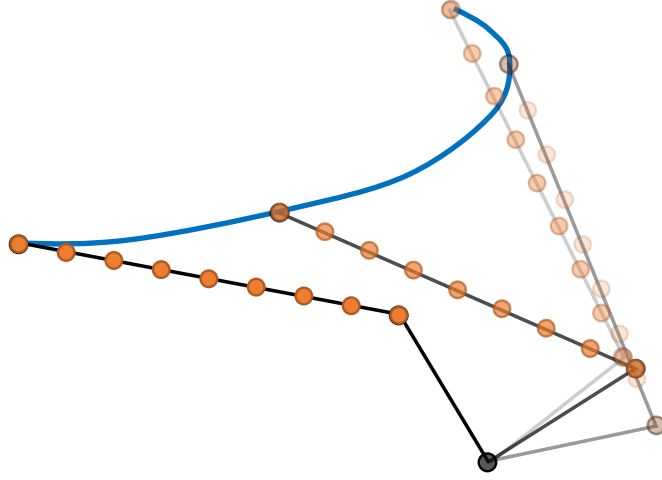
$$\varepsilon_{\text{ang}} = \left| \cos^{-1} \left( \frac{\langle \mathbf{x}_{\text{des}}^{(i)} - \mathbf{x}^{(i)}, \dot{\mathbf{x}}^{(i)} \rangle}{\|\mathbf{x}_{\text{des}}^{(i)} - \mathbf{x}^{(i)}\|_2 \|\dot{\mathbf{x}}^{(i)}\|_2} \right) \right| \quad (4.9)$$

of the current hand/end-effector-velocity  $\dot{\mathbf{x}}$  and a direct line from the end-effector to the desired goal-point  $\mathbf{x}_{\text{des}}^{(i)}$ , i.e.,  $\mathbf{x}_{\text{des}}^{(i)} - \mathbf{x}^{(i)}$ , at  $\mathbf{t}_k$  is obtained. We model a preference heuristic cost-function

$$\ell^{(i)}_{\text{pref}} = \begin{cases} 0 & \text{if } \|\dot{\mathbf{x}}^{(i)}\|_2 \leq \zeta_{\text{vel}} \\ \varepsilon_{\text{ang}} \|\dot{\mathbf{x}}^{(i)}\| & \text{else} \end{cases} \quad (4.10)$$

as a simplification of motor-dynamic models such as the Minimum-Jerk-Model ([Flash and Hogan, 1985](#)). This term prioritizes straight motions towards a goal over actions, that require a distinct change of perpendicular velocities. As this function is sensitive to measurement noise, a velocity-threshold  $\zeta_{\text{vel}}$  is defined, such that small disturbances in the velocity calculation do not affect the overall method.

**Interactive Costs** The interactive costs model the impact of multiple players executing different actions at the same time. Within the given HRC-scenario, interactive costs evaluate the collision risks of motions across the workspace. Therefore, a predefined number of  $N_{\text{spl}}$  way-points with matching time samplings  $T_{\text{spl}}$  is sampled along each trajectory. For each of these way-points the inverse kinematics are calculated for each player as shown in the representative bird-eye view in Figure 4.3 for one player. As a result, the reference trajectory shown in blue in Figure 4.3 is extended to  $N_{\text{link}}$  trajectories spread over the virtual *forearm* link with  $N_{\text{spl}}$  way-points each. For two actions being compared, pairwise distances between the virtual human and robot forearm for all way-points of the reference trajectories hold the basic information for the interactive costs. For most of the scenarios it is expedient to not only compare equivalent sectors in temporal progress, but rather to cross-evaluate the temporal shift of sampled pairs in both directions. Regarding the example in Figure 4.3, a temporal



**Figure 4.3:** Exemplary sketch of the trajectory-link sampling in the  $x - y$ -plane with the hand of a player following the blue reference trajectory. Given  $N_{\text{spl}} = 4$  way-points and  $N_{\text{link}} = 9$  samples along the link obtained from the inverse-kinematics, the resulting sample points for the pairwise distance of one player is shown orange.

shift by  $k = 1$ , i.e.,  $\tau_k = T_{\text{spl}}$ , would result in a shift of the way-points for 1 sample towards the trajectories goal point.

In general, given a temporal shift of  $k$  time samples results in the minimum distance  $d_{\tau k}$  from the pairwise distances of the robot and human forearm-trajectories. Based on that, the interaction-cost term,

$$\ell^{(i)}_{\text{int},k} = \begin{cases} \ell^{(i)}_{\text{col}} & \text{if } d_{\tau k} < \mathbf{1b}_d \\ \mathcal{F}_{\mathcal{N}}(d_{\tau k}) & \text{if } \mathbf{1b}_d < d_{\tau k} < \mathbf{ub}_d, \\ 0 & \text{else} \end{cases}, \quad (4.11)$$

i.e., collision-cost is calculated. If  $d_{\tau k}$  is smaller than a certain collision threshold  $\mathbf{1b}_d$ , which is given by the dimensions of the hand of the player, a high collision penalty  $\ell^{(i)}_{\text{col}}$  is applied. For larger distances the cost-function follows a Gaussian-shaped transition  $\mathcal{F}_{\mathcal{N}}(d_{\tau k})$  between the collision threshold  $\mathbf{1b}_d$  and a threshold  $\mathbf{ub}_d$ , indicating unaffected movement.

In order to handle the temporal evolution of minimum distances along the human and robot trajectories, (4.11) is extended by altering the temporal shift  $k$ . Therefore, a weighted sum and the maximum - respecting single collisions - over the temporally shifted trajectories are obtained:

$$\ell^{(i)}_{\text{int,avg}} = \frac{\sum_{k=1}^{N_{\text{spl}}} \kappa_{\text{temp},k} \ell^{(i)}_{\text{int},k}}{\sum_{k=1}^{N_{\text{spl}}} \kappa_{\text{temp},k}} \quad (4.12)$$

$$\ell^{(i)}_{\text{int,max}} = \max_{k \in \{1, 2, \dots, N_{\text{spl}}\}} (\kappa_{\text{temp},k} \ell^{(i)}_{\text{int},k})$$

The individual temporal weighting factor  $\kappa_{\text{temp},k} = (1 - \kappa_{\text{temp}})^k$  with a constant decline factor  $\kappa_{\text{temp}}$  limits the impact of large temporal differences on the cost calculation. The overall interaction cost value is obtained by

$$\ell^{(i)}_{\text{int}} = \max(\ell^{(i)}_{\text{int,avg}}, \ell^{(i)}_{\text{int,max}}), \quad (4.13)$$

as the maximum of temporally averaged cost values  $\ell^{(i)}_{\text{int,avg}}$  and the maximum cost value over all compared samples  $\ell^{(i)}_{\text{int,max}}$ . This brings the advantage of being less sensitive to false predictions by including the temporally increasing uncertainty of the estimated trajectories which is especially critical when evaluating human trajectories.

## 4.5 Experimental Evaluation

Finally, we conducted an experiment in which two different realizations of the proposed action-selection framework were compared to an a priori fixed policy. The main purpose of this experiment is to point out the improvements of obtaining an adaptive action-selection online by applying the HRC-game from Section 4.4 instead of following a predetermined policy within assembly processes. In contrast to the related work from Section 4.2 in which the focus was set on adapting the motion given a fixed policy, this experiment is designed to point out the advantages of also changing the action-primitives on-the-fly. Therefore we applied the following three behavior strategies:

- fixed behavior policy (*Fixed*) – where the robot follows a fixed policy chosen beforehand.
- spline game-policy (*Spline*) and line game-policy (*Line*) as two instances of the proposed HRC-game that differ in the applied human-motion prediction.

The robot is controlled with a database of pre-learned DMPs throughout all runs to diminish the influence of a trajectory generation module. Even though the *Fixed* method does not reflect the human actions, an additional underlying obstacle avoidance as outlined in [Dinh et al. \(2015\)](#) assures the safety of subjects throughout the experiments and creates an evasive behavior of the robot rather than blindly crashing into the human subject. The method applied uses an underlying compliance control altering the current executed motion as a reaction to virtual repellent force. This force is excited by the human hand – which is tracked by a motion capture system throughout the experiment – entering a predefined ball shaped safety region. Within our experiments, the radius of this safety region was set to 2.5 cm around the TCP of the robot.

The presented action-selection strategies differ in terms of integrating the human trajectory estimation. In the *Spline* method a minimum-jerk-based trajectory prediction from [Dinh et al. \(2015\)](#) and simple straight lines with trapezoidal velocity shapes for *Line* respectively are used.

---

Q1	<i>How would you grade the collaboration with the robot?</i>
Q2	<i>How would you grade the robot as a helpful coworker?</i>
Q3	<i>How would you grade the motion reaction of the robot?</i>
Q4	<i>How would you grade the action selection of the robot?</i>

---

**Table 4.1:** Questionnaire

### 4.5.1 Hypotheses

As an initial result to highlight the applicability of the proposed framework in HRC, the following hypotheses are claimed:

**Hyp1** - *Participants prefer the robot's action-selection when working in the Spline mode or the Line mode over the decisions in Fixed mode.* We assume that the cross-evaluation of action-pairs in our framework enables the robot to choose actions pro-actively such that its decisions will improve the collaboration.

**Hyp2** - *The decisions of the robot increase the safety for the human in Spline mode or the Line mode, compared to the Fixed mode.* By taking into account the effects of actions for each player, our framework enables the robot to detect risks at an early stage and adapt its decisions accordingly.

**Hyp3** - *The robot's decisions adapt to the human and therefore decrease the overall completion time in the Spline mode or the Line mode, compared to the Fixed mode.* With the solution of the proposed game being an NE the required effort of all players involved is minimized, which decreases the overall assembly time.

### 4.5.2 Experimental Setup

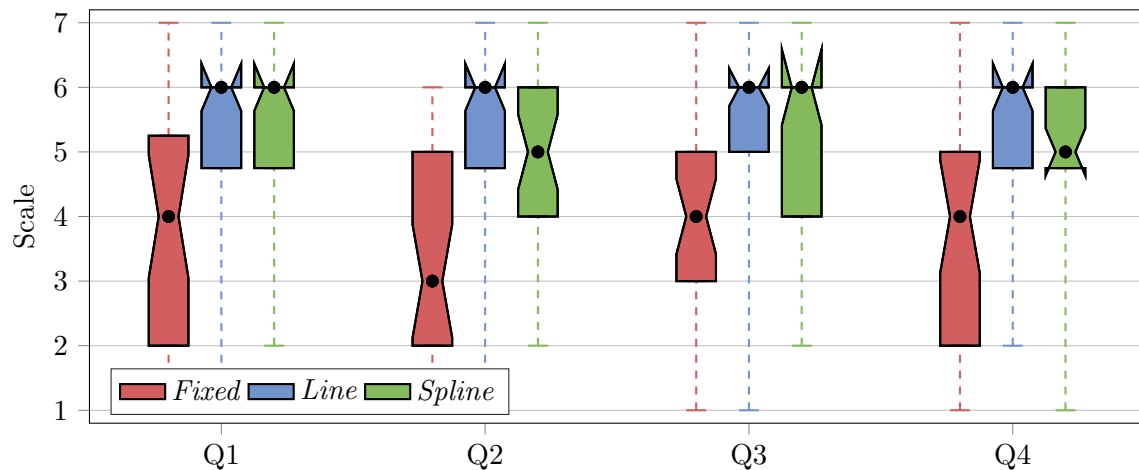
In our experiment 30 persons were asked to assemble 16 bricks of four different colors in a quadratic arrangement in cooperation with one Kuka-robot as depicted in Figure 4.1. The human and the robot were sitting face-to-face acting simultaneously. The scenario was designed in such a way that each pick-and-place sequence required the players to cross the shared workspace, thus a pro-active action-selection is of importance. In addition, this scenario fits our initial HRC-game outline as no action primitive can be favored over the other in terms of its impact towards accomplishing the goal.

Assuming that the underlying compliance controller reacts as a safety fall-back solution, the alteration of the executed motion is considered as a safety violation. Therefore, the integration of the virtual repellent force was taken as a measurement of safety violations throughout the experimental runs.

The human hand was tracked by six markers with eight motion capture cameras throughout the experiment in order to provide enough tracking redundancy to assure no participant was harmed.

The subjective perception of the collaboration with the human was evaluated based on a short questionnaire after each run. The answers to the ordered questions from Table 4.1 are mapped into a seven-point Likert-scale in which lower values depict negative feedback, and positive feedback is represented by large values.

The order of the different modes tested throughout the experiment was randomized after every full run to minimize the learning effect in the experimental evaluation results.



**Figure 4.4:** Box-plot results of feedback to Table 4.1, where the median is shown as a black dot. Each question had a seven-point Likert-scale which ranged from denoting the collaboration from *disturbing* to *helpful* for Q1 and the evaluation of the robot as a colleague for Q2 from *disturbing* to *equal team-member*. Q3 pointed out if the robot was from *not at all* to *all the time* reactive to the human coworker. Q4 evaluated the action selection from *very bad* to *very good*.

### 4.5.3 Results and Discussion

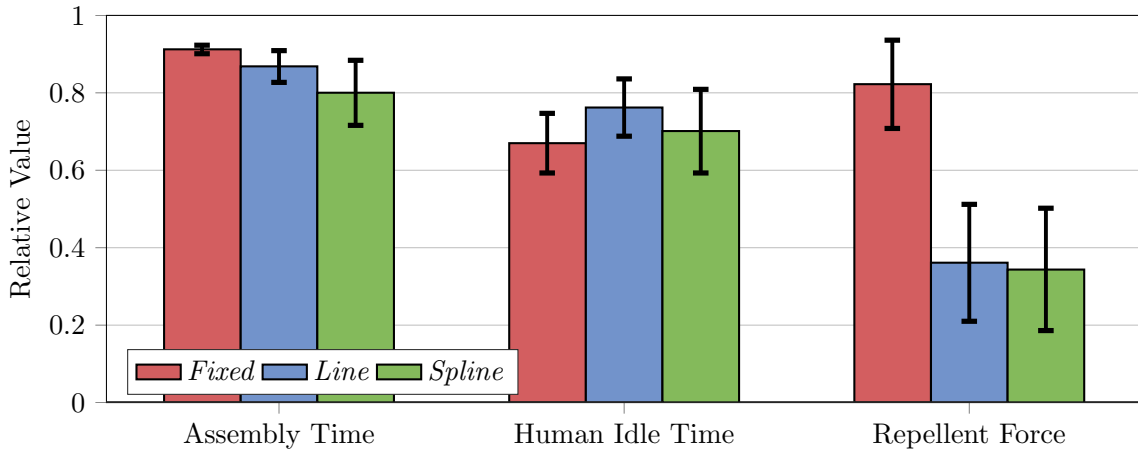
The participants' subjective feedback is depicted in Figure 4.4. It shows that the *Spline* method outperforms the *Fixed* method in all investigated aspects.

In order to evaluate the statistical impact of the questionnaire, a post hoc analysis on a three-way Friedman's test was run. The statistically significant difference between the *Spline* / *Line* and *Fixed* method depicted in Table 4.2 confirms hypothesis **Hyp1**.

In addition to the participants' subjective feedback, the three runs are compared in terms of human idle time, completion time and safety awareness. As mentioned before, the repellent force of the underlying local obstacle avoidance was used as a measure of safety. As these values strongly depend on each participant's behavior, each full run is normalized for each individual, thus reflecting the relative change over the runs per participant. The results shown in Figure 4.5 are given as the averaged relative measurement of all participants.

Question	Overall Comparison	<i>Line</i> vs <i>Spline</i>	<i>Line</i> vs <i>Fixed</i>	<i>Spline</i> vs <i>Fixed</i>
Q1	<b>0.0016</b>	0.6670	<b>0.0034</b>	<b>0.0014</b>
Q2	<b>0.0004</b>	0.9324	<b>0.0009</b>	<b>0.0005</b>
Q3	<b>0.0032</b>	0.9327	<b>0.0025</b>	<b>0.0052</b>
Q4	<b>0.0021</b>	0.7709	<b>0.0023</b>	<b>0.0030</b>

**Table 4.2:** Evaluated questionnaire data. Each cell holds  $p$ -values for a three-way (overall comparison) or pairwise Friedman's test. Statistically significant values ( $p < 0.01$ ) are shown in bold.



**Figure 4.5:** Direct evaluation of the overall assembly time, the human idle time and the activation of the repellent virtual force of the local obstacle avoidance around the TCP of the robot over all experimental runs.

The results of the repellent force of the local obstacle avoidance in Figure 4.5 show significant improvements by applying our framework in the outlined pick-and-place scenario due to the ability of our framework to detect collisions at an early stage and to adapt the actions accordingly. This result confirms hypothesis **Hyp2**.

Comparing the results of the overall assembly time, hypothesis **Hyp3** can only be confirmed for the *Spline* method due to the variance overlap of the results for the *Line* and *Fixed* methods.

However, the variance of the completion time is still distinctly high and the improvements of human idle-time are not yet satisfying. The decreased human idle time during the *Fixed* runs on the other hand mainly results from the humans evading the robot with the payoff of increased overall assembly time.

Last but not least it has to be noted that the trajectory mapping of our initial HRC-game needs further improvements for an increased overall system performance. As our framework mainly obtains the allocation of action-primitives among the dyads, additional trajectory planning has to be integrated. Nonetheless, the results of this initial HRC-game in which our framework was tested without such an additional trajectory planner, holds as a proof-of-concept of the method proposed in this chapter.

## 4.6 Conclusion

In this paper an HRC action selection algorithm based on game theory is proposed. The general structure of a game in an HRC context is explained. The idea is furthermore outlined on a manipulation scenario regarding the action selection in collaborative pick-and-place tasks. An insight is given into how the action space is obtained and matched to a set of reference-trajectories for the players involved. In addition to that, this initial model is tested in an HRC-human user study. Based on the results obtained in the experiment, including the



subjective feedback from the participants involved, the potential of the proposed framework to improve HRC is shown.

While the results shown in this chapter hold as proof-of-concept of the general framework as such, suggestions for future work are discussed in the next chapter.



# 5

## A Conceptual Design to Realize Interactive Task-and-Motion-Planning within Human-Robot Collaboration by Means of Repeated Markov Games

### Chapter Abstract

This chapter outlines a conceptual extension of the methods presented in the previous chapters and also serves as a joint conclusion and outlook into future work that results from the work presented in this thesis.

Specifically, we outline how the concept of normal form games can be extended to generic Markov games (MGs) to meet the requirements of an interactive human-robot collaboration-process. Thus, we outline how the concepts of a normal-form game can be extended to depict a MG-analogue of the previously presented method. Given the basics of MG, we outline how the joint planning problem of human(s) and robot(s) can be fit into an autonomous decision-making framework and elaborate the necessity for extending the action-selection method to joint motion-planning.

On the one hand, the motion planning problem cannot be simply ignored in order to select team-optimal action assignments, on the other hand, there exist no suitable motion planner that would allow the required features for such an interactive task and motion planning (TAMP)-framework. As this is both subject-aggravated by not having control over the human decisions and motions, while also having imprecise human behavior models, we outline which requirements a conceptual game-theory-inspired TAMP framework must fulfill, but also propose initial concepts how such a method is achievable.

Eventually, we conclude this chapter and this part of the thesis by summarizing the collected findings and sketching how future work can build upon the work presented in this part.

Given the empirical evidence and proposed methods from Chapter 3 and Chapter 4, this chapter highlights a combination of the former and the latter to achieve interactive task and motion planning (TAMP) for heterogeneous and homogeneous human-robot teams (HRTs). While Markov-models such as Markov decision processes (MDPs) or partially observable Markov decision processes (POMDPs) are powerful tools in modeling stochastic decision problems, they can only treat human and robot agents jointly. To respect individual objectives at each transition, iterative Markov games (MGs) have been established in literature that allow to combine both approaches. As the presented extension builds upon this, we briefly sketch the theoretical background and dedicated Bellmann equations below.

## 5.1 Modeling Sequential Decision-Making by Means of Markov Games

In contrast to classical games, Markov games form the combination of MDPs and sequential games, as stated by [Shapley \(1953\)](#). Within an MDP an agent can choose an action  $\mathbf{a} \in \mathcal{A}$  at a state  $\mathbf{s} \in \mathcal{S}$ , that fully describes the environment, such that the environment transitions to a new state according to the Markovian transition probability rate  $\mathcal{T} = \mathbb{P}[\underline{\mathbf{s}}' | \underline{\mathbf{s}}, \underline{\mathbf{a}}]$ . At each state, the agent obtains a reward  $\mathbf{r} \leftarrow \mathcal{J} := \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ . The solution of an MDP is given as the optimal stationary Markovian policy  $\pi^*$ , that forms a mapping  $\pi := \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ . The policy  $\pi$  is called *stationary*, if it only depends on the state, but not time, i.e.,  $\pi(\mathbf{s}_k, \mathbf{a}, \mathbf{t}_k) = \pi(\mathbf{s}_k, \mathbf{a}, \mathbf{t}_j) \neq \pi(\mathbf{s}_j, \mathbf{a}, \mathbf{t}_k), \forall j \neq k$ . Furthermore, a policy is *Markovian* if and only if it solely depends on the current state  $\mathbf{s}$ . In contrast to MDPs, an MG is defined by the following components

- $\underline{\mathcal{S}}$  is a finite set of states  $\underline{\mathbf{s}} = (\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \dots, \mathbf{s}^{(N_{\mathfrak{A}})}) \in \underline{\mathcal{S}}$  resembling the environment.
- $\underline{\mathfrak{A}} = \{\mathfrak{A}^{(1)}, \mathfrak{A}^{(2)}, \dots, \mathfrak{A}^{(N_{\mathfrak{A}})}\}$  is a finite set of  $N_{\mathfrak{A}}$  agents.
- $\underline{\mathcal{A}} = \{\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \dots, \mathcal{A}^{(N_{\mathfrak{A}})}\}$  is collection of finite action-sets per agent  $i$ .
- $\underline{\pi} : \underline{\mathcal{S}} \times \underline{\mathcal{A}} \mapsto [0, 1]^{N_{\mathfrak{A}}}$  is a joint Markovian policy that maps the state to an action for each agent.
- $\mathcal{T} = \mathbb{P}[\underline{\mathbf{s}}' | \underline{\mathbf{s}}, \underline{\mathbf{a}}]$  is the probability of reaching state  $\underline{\mathbf{s}}'$  from  $\underline{\mathbf{s}}$  when drawing a joint action  $\underline{\mathbf{a}}$  from  $\underline{\pi}$ .
- $\underline{\mathcal{J}} = (\mathcal{J}^{(1)}, \mathcal{J}^{(2)}, \dots, \mathcal{J}^{(N_{\mathfrak{A}})})$  are player-specific payoff functions

$$\mathcal{J}^{(i)}(\mathbf{a}^{(i)} | \underline{\mathbf{a}}^{(-i)}, \underline{\mathbf{s}}) \mapsto \mathbb{R},$$

that map the current action-profile  $(\mathbf{a}^{(i)}, \underline{\mathbf{a}}^{(-i)})$  at  $\underline{\mathbf{s}}$  to a numeric value for each agent.

- $\gamma \in [0, 1]$  is a discount factor, which weights the temporal impact of future versus imminent payoffs.

Given the existence of a joint goal, one can further introduce the common goal space  $\underline{\mathcal{G}} \in \underline{\mathcal{S}}$  as a subset of the state-space.

## Bellmann Equations for Markov Games

By defining the expected accumulated reward per agent on a strategic level

$$\mathcal{R}^{(i)} = \int_{t=0}^{\infty} \gamma^t \mathbb{E}_{\underline{\mathbf{a}}^{(-i)}} [\mathbf{r}(\underline{\mathbf{s}}_t, \mathbf{a}_t^{(i)}, \underline{\mathbf{a}}_t^{(-i)})], \quad (5.1)$$

one obtains the objective that each agent of the MG tries to maximize. By introducing the state-value function

$$\mathbf{V}^{(i)}(\underline{\mathbf{s}}) = \mathbb{E}_{\underline{\mathbf{a}}^{(-i)}} [\mathbf{r}(\underline{\mathbf{s}}_t, \mathbf{a}_t^{(i)}, \underline{\mathbf{a}}_t^{(-i)}) + \mathbf{V}(\underline{\mathbf{s}}^{(i)'})], \quad (5.2)$$

that approximates the optimal accumulated reward for each agent according to the recursive Bellmann equation (5.2). The state-value function can also be defined by means of the state-action-value function or Q-function

$$\mathbf{V}^{(i)}(\underline{\mathbf{s}}) = \max_{\mathbf{a}^{(i)}} \mathbf{Q}^{(i)}(\underline{\mathbf{s}}, \mathbf{a}^{(i)}), \quad (5.3)$$

which can be used to define the advantage function

$$\mathbf{A}^{(i)}(\underline{\mathbf{s}}, \mathbf{a}^{(i)}) = \mathbf{Q}^{(i)}(\underline{\mathbf{s}}, \mathbf{a}^{(i)}) - \mathbf{V}^{(i)}(\underline{\mathbf{s}}), \quad (5.4)$$

that defines the expected payoff win or loss for each agent when choosing an action at state  $\underline{\mathbf{s}}$ . Referring to the definition of  $\varepsilon$ -Nash-equilibrium (eNE), the advantage function for each agent is bounded by  $\mathbf{A}^{(i)}(\underline{\mathbf{s}}, \mathbf{a}^{(i)}) \geq -\varepsilon_{\text{br}}$ .

## 5.2 Game-Theory Inspired Task and Motion Planning

As laid out in Section 2.1.3, multiple research projects have proposed TAMP-frameworks to achieve optimal task allocation for HRT while also generating and executing suitable trajectories. In brief, the main challenges for a human-robot collaboration (HRC)-TAMP framework are given as:

- find a feasible trajectory of high-level actions, that transition the current state  $\underline{\mathbf{s}}$  to the goal-space  $\underline{\mathcal{G}}$ .
- allocate feasible actions among the HRT w.r.t. a predefined objective, e.g., runtime.
- account for temporal duration of concurrency during allocation.
- account for feasible and optimal trajectories in the generated planning setting.
- estimate the human behavior and adjust robot behavior.

In the scope of this work, plenty of effort was laid in including the interaction behavior of humans and robots across all the domains stated above, to achieve an interactive TAMP framework that allows to incorporate the concepts of MG. Specifically, this also involves to account for suboptimal human behavior and imprecise human objective estimates. Introducing such a human response behavior imposes a stochastic behavior across multiple layers of such a TAMP-framework. This quickly violates any Markovian assumption on a dedicated higher-level planning model, such as the ones presented in Section 2.2.3. While early planning work was solely relying on first-order-logic or emphasized on stochastic representations

of the environment, introducing temporal uncertainty, planning uncertainty and stochastic trajectories to represent a human, requires the robot to either neglect all stochastic behavior – resulting in existing work – or to regress an optimal policy from an unpredictable stochastic black-box.

Nonetheless, there remain extensions and realizations of TAMP to account for interactive HRC. As most approaches within TAMP still rely either on simulated environments (Hartmann et al., 2020) or single-agent systems (Garrett et al., 2021), we propose that there is potential room to extend the online execution of these methods by incorporating concepts from MGs. It is well known that solving large-scale MGs is NP-hard. They scale exponentially in action-spaces and agents, while each transition depends on the policy of other agents, which in return may eventually violate the underlying Markov-property. By the definition of HRC, the individuals have a joint goal that needs to be reached. It is thus legitimate to assume optimal behavior and neglecting mutual influences for the planning process. In contrast to existing work, we propose that not only a valid plan should be obtained, but also the individual state-value (5.2) and state-action-value (5.3) for the human and robot. Depending on the complexity of the problem, this may either be achieved by Q-learning (Watkins and Dayan, 1992), relational regression (Munzer et al., 2017) or even recent findings from multi-agent reinforcement learning, for which we provide a more detailed insight in Part II. Obtaining such a baseline solution for a given task is preferably obtained offline, as proposed in related work that proposed similar methods to our extension by explicitly accounting for suboptimal human behavior (Fisac et al., 2019, Fridovich-Keil et al., 2020, Malik et al., 2018). Thus, we outline possible extensions, given the baseline solutions  $\mathbf{Q}^{(R)}$ ,  $\mathbf{Q}^{(H)}$  and  $\mathbf{V}$  obtained from simulation. Similar to the relational activity process (RAP) (Toussaint et al., 2016), we assume that the dedicated state-action functions contain the actions of the other agent(s) in their dedicated state-description, i.e., that the objective of each agent is directly conditioned on the action of the other agent.

### 5.2.1 Incorporate Human Preferences and Suboptimal Behavior

During the actual execution, a robot needs to take into account that the assumed human objective as approximated by  $\mathbf{Q}^{(H)}$  diverges from the actual one. Thus, the fully observable state-space<sup>1</sup> is extended by two artificial states

$$\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathbf{x}_{\text{rat}}, \mathbf{x}_{\text{pref}}\} \mapsto [0, 1]^2, \quad (5.5)$$

where  $\mathbf{x}_{\text{rat}}$  denotes how *rationaly* the human is following the current objective, while  $\mathbf{x}_{\text{pref}}$  denotes how well the assumed human objective describes the actual human behavior. As result  $\mathbf{x}_{\text{rat}} \mapsto 1$  and  $\mathbf{x}_{\text{pref}} \mapsto 1$  results in both agents following the assumed cost-metric. As each human behaves differently, an online objective  $\mathbf{Q}_{\text{pref}}(\mathbf{s}, \mathbf{a}^{(H)})$  for H is introduced besides the latter of the two partially observable states. This function evaluates additional human preferences and may either be realized as a neural network if plenty of data can be collected or may have been pre-recorded, relational preference learning (Munzer et al., 2017), visual affordances (Koppula and Saxena, 2016) or a weighted convex-optimization function, for which the weights need to be regressed (Oguz et al., 2018b).

---

<sup>1</sup>As stated in Chapter 2 this work neglects perceptual ambiguity, which would result in partial observability. Given that this part of the thesis solely models humans as a source for stochastic system behavior, results in a fully observable state-space.

Given the extended state-space, we propose to model the online decision-making as a  $k$ -level sequential game, with the robot taking the first initiative. Similar to Fridovich-Keil et al. (2020), the TAMP-problem transforms into a two-level hierarchical system, where the upper layer handles the strategic interaction of the artificial agents, whereas the lower layer executes the action by means of trajectory optimization. Given collected evidence from cognitive science (Baker et al., 2007), we propose to use a Boltzmann-distribution to approximate the human policy on the strategic layer:

$$\hat{\pi}^{(H)}(\mathbf{s}, \mathbf{a}^{(H)} | \mathbf{a}^{(R)}) \approx \frac{1}{\nu} \exp(\beta(Q^{(H)}(\mathbf{s}, \mathbf{a}^{(H)} | \mathbf{a}^{(R)}) + Q_{\text{pref}}(\mathbf{s}, \mathbf{a}^{(H)}))), \quad (5.6)$$

using a normalizer  $\nu$  to generate a probability density function over the state-space of the human as the current policy, while  $\beta$  expresses the likelihood of the human following the currently assumed objective metric.

### 5.2.2 Obtaining Robot Policies for an Online HRC-Process

Fridovich-Keil et al. (2020) proposed a Q-learning alternative to approximate the human response as a Stackelberg-equilibrium, which is legitimate due to the simplicity of state-space and system-dynamics within autonomous driving. Within HRC such a solution can be incorporated into existing TAMP approaches, to improve the estimates of  $Q^{(R)}$ ,  $Q^{(H)}$  and  $V$ . Nonetheless, in our model, we propose a quantified representation of the artificial states  $\{\mathbf{x}_{\text{rat}}, \mathbf{x}_{\text{pref}}\}$ , where each quantified value maps to a dedicated  $\beta$  and human-preference  $Q_{\text{pref}}(\underline{\mathbf{s}}, \mathbf{a}^{(H)})$  function. Thus, given a sufficiently high value for  $\beta$  if  $\mathbf{x}_{\text{rat}} \mapsto 1$  and  $\mathbf{x}_{\text{pref}} \mapsto 1$  – i.e., a complete rational agent – the Boltzmann-distribution converges to a dirac-impulse. Given this mapping, the strategic layer can be represented as a mixed observable Markov decision processes (MOMDPs) similar to Chapter 3 by tracking a belief over the quantified values of human behavior metrics.

Alternatively, we propose to approximate the stochastic robot policy by means of Monte-Carlo tree search (MCTS) for the limited  $k$ -level extensive game. Given a limited computational budget, we thus propose to run an MCTS algorithm, where the nodes contain states of the environment that takes as inputs the current state  $\mathbf{s}$ , robot cost objective  $Q^{(H)}(\underline{\mathbf{s}}, \mathbf{a}^{(R)} | \mathbf{a}^{(H)})$  human cost objective  $Q^{(H)}(\underline{\mathbf{s}}, \mathbf{a}^{(H)} | \mathbf{a}^{(R)})$ , human rationality estimate  $\mathbf{x}_{\text{rat}}$ , human preference estimate  $\mathbf{x}_{\text{pref}}$ , as well as the available computational budget  $N_{\text{iter}}$  and the depth-level of the  $k$ -level decision problem  $N_k$ . As Algorithm 5.1 is intended to be run iteratively, it also returns and accepts the currently explored tree. Namely, before executing Algorithm 5.1, the previously explored sub-tree can be reused by defining the node related to the current state as the new root of the tree in Line 2. Similarly, the explored tree is returned once the computational budget is used.

At the beginning of each run of Algorithm 5.1, the rationality, i.e., Boltzmann constant  $\beta$  and current human preference is obtained from the current virtual state values  $\mathbf{x}_{\text{rat}}, \mathbf{x}_{\text{pref}}$  in Lines 3 and 4, which are kept constant for the whole iteration to reduce the stochasticity for the current decision-problem. Given this, the feasible action-set is obtained from the offline agent-objectives  $Q^{(R)}, Q^{(H)}$  and  $V$  in Line 6. Again, we propose a dependency on the assumed human rationality metric. By normalizing the state-action-function and value-function by the maximum state-action-function value at the current state, the current advantage-function is also normalized. Recalling the concept of eNE, instead of querying a knowledge base to

---

**Algorithm 5.1:** Obtain an approximated solution for the strategic robot using a limited computational budget and MCTS.

---

**Input:**  $\mathbf{s}$ ,  $Q^{(R)}(\mathbf{s}, \mathbf{a}^{(R)} \mid \mathbf{a}^{(H)})$ ,  $Q^{(H)}(\mathbf{s}, \mathbf{a}^{(H)} \mid \mathbf{a}^{(R)})$ ,  $\mathbf{x}_{\text{rat}}$ ,  $\mathbf{x}_{\text{pref}}$ ,  $N_{\text{iter}}$ ,  $N_k$ ,  $\mathcal{V}$

**Output:** Robot Policy  $\pi^{(R)}$ ,  $\mathcal{V}$

---

```

1 Function GameEpisodeMCTS:
2    $\mathbf{v}_0, \mathcal{V} \leftarrow \text{init}(\mathbf{s}, \mathcal{V})$  ▷ initialize root node and tree
3    $\beta \leftarrow \text{initRational}(\mathbf{x}_{\text{rat}}, \mathbf{x}_{\text{pref}})$  ▷ initialize rationality
4    $Q_{\text{pref}} \leftarrow \text{initPreference}(\mathbf{x}_{\text{pref}})$  ▷ initialize preference
5   for  $n = 0$  to  $N_{\text{iter}}$  do
6     // approximate human policy via (5.6)
7      $\mathcal{A}_{[n]}^{(R)}, \mathcal{A}_{[n]}^{(H)} \leftarrow \text{getActions}(\mathbf{v}_0, \mathbf{x}_{\text{rat}})$  ▷ get action sets for H and R
8      $\hat{\pi}^{(H)}(\underline{\mathbf{s}}, \mathbf{a}^{(H)} \mid \mathbf{a}^{(R)}) \leftarrow \text{approxHumanPolicy}(Q^{(H)}(\mathbf{s}, \mathbf{a}^{(H)} \mid \mathbf{a}^{(R)}), Q_{\text{pref}}, \mathcal{A}_{[n]}^{(H)})$ 
9      $\mathbf{a}^{(R)} \leftarrow \arg \max_{\mathcal{A}_{[n]}^{(R)}} \mathbb{E}_{\pi^{(H)}} [Q^{(R)}(\mathbf{s}, \mathbf{a}^{(R)} \mid \mathbf{a}^{(H)}), \mathbf{v}_0]$  ▷ select greedy robot action
10     $\mathbf{v}_1 \leftarrow \text{stateTransition}(\mathbf{s}, \mathbf{a}^{(R)}, \hat{\pi}^{(H)})$  ▷ sample next state from  $\hat{\pi}^{(H)}$ 
11    for  $k = 1$  to  $N_k$  do
12       $\mathcal{V} \leftarrow \mathcal{V} \cup \mathbf{v}_k$  ▷ add node to tree
13       $\mathbf{v}_{k+1} \leftarrow \text{sample}(\mathbf{v}_k)$  ▷ simulate consecutive steps by sampling
14       $\text{updateTree}(\mathcal{V})$  ▷ update tree
15   $\pi^{(H)} \leftarrow \text{getPolicy}(\mathcal{V}, \mathbf{x}_{\text{rat}})$ 

```

---

check for valid actions, it is possible to directly evaluate the advantage-function to obtain feasible action-sets. By limiting feasible actions to hold  $\mathbf{A}^{(i)}(\underline{\mathbf{s}}, \mathbf{a}^{(i)}) \geq -\mathbf{x}_{\text{rat}}$ , only candidates for an eNE are drawn from the available actions. As a direct consequence, the size of the actual action-space directly depends on the assumed rationality.

Nonetheless, as solely applying this limit may also result in proceeding cyclic or useless robot actions, the tree is always checked against infeasible edges, i.e., robot actions, at the end of the algorithm in Line 13, that includes applying the usual back-propagation of collected objective data as well as pruning infeasible edges from the tree. Given the resulting action-sets, the human responses are evaluated for all available robotic actions according to (5.6). Implementation-wise it is thus preferable to introduce intermediate nodes after a robotic action, from which the human response can be realized as a stochastic transition. It must further be noted that once the human policy has been evaluated, the transition probability remains unchanged, such that (5.6) is not required to be evaluated at every iteration. Given the human response behavior, the robot action is assumed in a greedy manner, while also accounting for the number of visits for each sample. Therefore, we propose to use upper confidence bound applied to trees (UCT) as the utility to obtain the current robot action:

$$\mathcal{U}(\mathbf{a}^{(R)}) := Q^{(R)}(\mathbf{s}, \mathbf{a}^{(R)} \mid \cdot)^{\otimes} + \kappa_{\text{UCT}} \sqrt{\frac{n}{N_{\text{cnt}}(\mathbf{a}^{(R)})}}, \quad (5.7)$$

where  $\kappa_{\text{UCT}}$  weighs the impact of exploration across the robot action-space,  $Q^{(R)}(\mathbf{s}, \mathbf{a}^{(R)} \mid \cdot)^{\otimes}$  denotes the best observed robot objective, while  $n$  denotes the current iteration of Algorithm 5.1 and  $N_{\text{cnt}}(\mathbf{a}^{(R)})$  represents how often the dedicated robot action has been explored. Given the dedicated action, the human action is sampled via (5.6) in Line 9. Depending on



the strategic depth, the remaining steps are evaluated by drawing samples for the human and robot policies in Line 12. During this sampling procedure, we propose to repeat the process from Line 6 to Line 9 with a minor adjustment. Rather than applying a greedy-action using the utility from (5.7), the expected  $Q^{(R)}$ -values are collected from the approximated human policy to obtain a Boltzmann-distribution with  $\beta = 1$ , from which samples can be drawn at each iteration, once the node is added to the graph.

Eventually, the robot policy is returned to allow to generate suitable trajectories. At each iteration, the observed cumulative objective for the robot is obtained and saved in the intermediate robot action-nodes. These values are on the one hand used to calculate (5.7), but also the final policy. Again, we incorporate the current rationality estimate to obtain the robot policy. When interacting with a rational agent towards a joint goal, it is preferable to seek for the possible optimum value, while the expected value is a preferable metric for the interaction with a random, i.e., irrational human counterpart. Thus, the final robot policy in Line 14 is obtained as

$$\pi^{(R)}(\mathbf{s}, \mathbf{a}^{(R)}) := \frac{1}{\nu} \mathbf{x}_{\text{rat}} Q^{(R)}(\mathbf{s}, \mathbf{a}^{(R)} | \cdot)^{\otimes} + (1 - \mathbf{x}_{\text{rat}}) \frac{1}{N_{\text{cnt}}(\mathbf{a}^{(R)})} \sum_{i=0} N_{\text{cnt}}(\mathbf{a}^{(R)}) Q^{(R)}(\mathbf{s}, \mathbf{a}^{(R)} | \mathbf{a}^{(H)}). \quad (5.8)$$

Given this policy, suitable robot trajectories can be obtained as we will briefly propose in Section 5.2.4. The resulting policy on the other hand only solves the decision-problem on finding the robot policy for a single action, i.e., assigning an action-primitive to the robot. In order to solve the full manipulation task, a robot is thus required to execute the obtained policy from Algorithm 5.1, observe the human performance in the meantime, update the virtual human performance states and rerun Algorithm 5.1 before the next allocation is needed. Before outlining possibilities of executing the robot policy, we continue with the human performance observation and how this impacts the applied human behavior states.

### 5.2.3 Updating Human States

In contrast to MOMDPs or POMDPs, our presented method does not model the human metric states as partially observable states and instead keeps the values constant for each iteration in Algorithm 5.1. This diminishes the need for a Markovian transition function. As a linear Markovian update also may result in cyclic updates of the assumed human states, which ultimately results in cyclic robot behavior, we instead propose to record the human performance metric in a cyclic buffer  $\mathcal{D}^{(H)}$ . In here, all state, action tuples are stored, where the actions of the human and robot are stored. This data can then be used to update the current performance metric by applying the regression function that belongs to the current preference metric. Assuming normalized preference-values, comparing the absolutes of the preference values in  $\mathcal{D}^{(H)}$  before and after running the regression can then be used as a reliability metric for the preference function, similar to the internal convergence of a parametric optimization objective.

Given this updated preference-metric, the advantage-function for the human and the collected data in  $\mathcal{D}^{(H)}$  can directly be calculated from (5.4). The assumed human rationality can then simply be set to

$$\mathbf{x}'_{\text{rat}} := 1 - \sum_{i=0}^{|\mathcal{D}^{(H)}|} A^{(H)}(\mathbf{s}_i, \mathbf{a}_i^{(H)}, \mathbf{a}_i^{(R)}). \quad (5.9)$$

Further, we propose to initialize an empty buffer, such that divergent human impact during early stages has a dedicated stronger impact, while later errors may also be just subject to fatigue, which resembles partially rational behavior rather than unpredictable behavior. Similarly, the size of  $\mathcal{D}^{(H)}$  needs to be limited as otherwise the system ignores human suboptimal behavior at some stage. As this chapter only introduces conceptual work, this value needs to be evaluated from empirical data and is thus left for future work.

### 5.2.4 Generate and Execute Robot Motions

Eventually, the obtained robot policy needs to be executed on the robot in the form of obtaining suitable trajectories. As the strategic layer assumes to have access to the optimal allocation metrics from simulation, the optimal policies are usually expected to result in kinematically feasible solutions. Nonetheless, a major benefit of the proposed method lies in the possibility of not only obtaining a valid discrete action-assignment, but rather a distribution over possible actions, weighted by their expected objective. This allows to solve multiple trajectory optimization problems, depending on the current robot action-space and the available computational budget. As each of these motion planning problems can be solved independently, the number of parallel runs can thus be run concurrently. We further propose that it is beneficial to solve the resulting motion generation similar to [Toussaint \(2015\)](#) by modeling each robot action-edge of the strategic layer as an underlying non-linear program. This may be especially of interest, if iterative motion planners for HRC, such as [Bari et al. \(2021\)](#), are used. Eventually, the incorporation of multi-modal trajectory optimization ([Osa, 2020](#)) as well as ideas to account for the human objective gradient in the optimization problem ([Sadigh et al., 2016a](#)) may further improve the overall performance of the proposed interaction concept.

## 5.3 Future Work

To conclude this part of the thesis, we briefly outline future work that results from the presented work and methods. First, the presented concept from this chapter – even though it represents a novel concept – requires experimental evaluation and verification or even falsification given empirical data and subjective evaluations. Besides the work presented in this chapter, this would also require designing a suitable motion planner that fits the requirements of the proposed online-decision making algorithm. This implies the ability to produce fast results, while also accounting for multiple solutions to e.g., reach a desired goal pose.

Furthermore, a solid cross-comparison is needed on which prior model is best to be used to obtain the offline solutions for  $\mathbf{Q}^{(R)}$ ,  $\mathbf{Q}^{(H)}$  and  $\mathbf{V}$ . This implies not only a comparison in terms of precisely predicting the value of the true optimum, but also the required computation time to evaluate the dedicated functions. As these will be used in [Algorithm 5.1](#), it is important to reduce the computation time upon every call to allow for a thorough evaluation.

Eventually, the current approach still assumes full state-observability, which remains a challenging and often unjustified assumption for real-world applications. Thus, a major line of research stemming from the presented work is the incorporation of accounting for uncertainty in the states but also the observations, especially for the currently executed and previously observed human actions. Recent research in classifying human actions produces reliable results, but even reliable estimators cannot guarantee absolute accuracy. If such imprecise

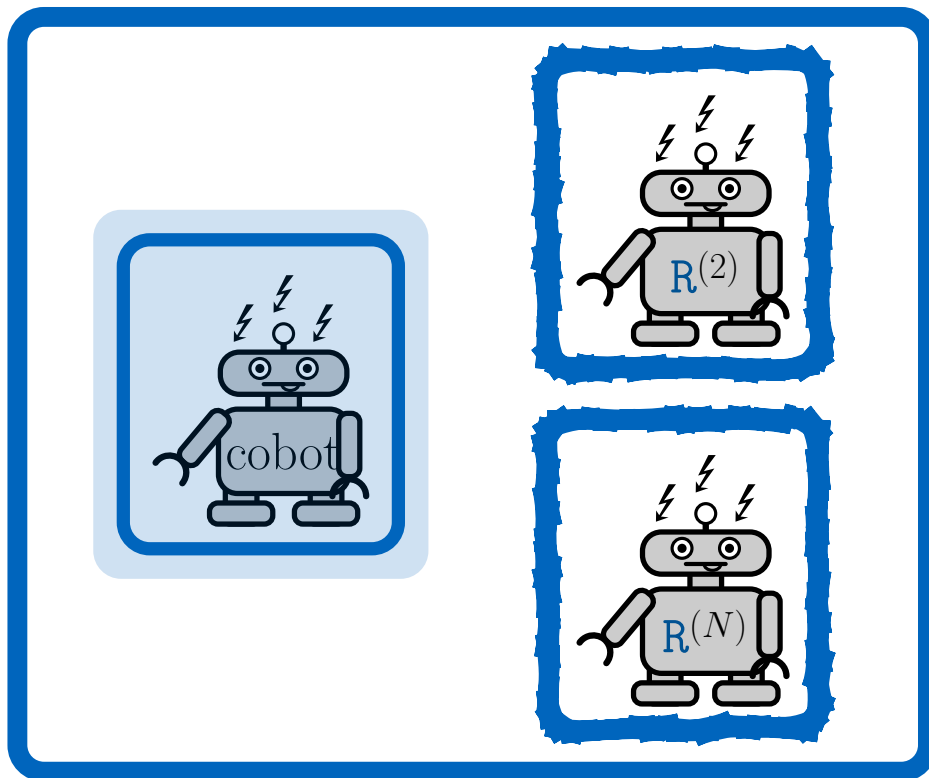
estimates are used to extract the human policy and eventually the decisions for the robot, this imprecision may have drastic consequences.

Similarly, this holds for the actual execution, where collision avoidance and human behavior model strongly rely on accurate measurements. Thus, the major line of research in joint HRC is given by guaranteeing safe interaction, even though data is solely obtained from noisy sensors.



## Part II

# Learning Behavior-Policies in Groups of Artificial Agents





# 6

## Multi-Robot Hierarchical Actor-Critic

### Chapter Abstract

Multi-agent systems are an interdisciplinary research field that describe the concept of multiple decisive individual interacting with a usually partially observable environment. Given the recent advances in single-agent reinforcement learning, the area of multi-agent reinforcement learning (MARL) has gained tremendous interest within recent years. Even though various research work has been proposed that allows artificial agents to discover team-optimal policies, the majority of these approaches still rely on pure end-to-end learning. It is well known that these approaches still suffer from the necessity of big data in order to achieve useful results. If the agents are only rewarded sparsely, this issue becomes inherently worse. On the other hand, aside of training a policy from demonstrated data, there is rare potential on decoupling the multi-agent interaction or even incorporate model-knowledge online.

Nonetheless, recent research has shown that hierarchical concepts allow to improve scaling in sparse environments, but also provide the possibility of directly embedding model-knowledge or robotic controllers. As these approaches are still limited to single-agent applications, and the existing solutions mainly focus on fully synchronized settings, this chapter outlines a novel actor-critic (AC)-approach that decouples the MARL-problem into a set of agents modeling the other agents as responsive entities. We further propose to estimate two separate critics per agent in order to distinguish between the joint task reward and agent-based cost-metrics as commonly applied within multi-robot planning.

Finally, we outline how this AC-framework can be embedded into a hierarchical MARL-approach as the decentralized learning allows for asynchronous decisions along hierarchical layer-agents. Within the presented hierarchical reinforcement learning-framework we further propose to impose structured observations for each agent, i.e., to explicitly distinguish between internal agent-states and environmental observations. This allows to neglect the observations of other agents within the native critic due to the conditional independence. This critic evaluates the objective of reaching a proposed hierarchical sub-goal, which eventually improves the learning speed.

We evaluate our presented methods within a sparsely rewarded simulated multi-agent environment. While our approach already outperforms the state-of-the art learners, we close this chapter by outlining possible extensions that are expected to further improve the overall performance and learning speed.

*Remark:* A majority of this chapter is also available in [Gabler and Wollherr \(2023\)](#).

## 6.1 Introduction

Based on the recent advances in robotics research over the last decades, automated robotic systems have been established in our every-day life even beyond industrial applications. Motivated by the wide range of applications, that have been presented within well-defined environments such as labs or production halls, it is favorable to allow robots to easily learn and adjust to new tasks without the need of re-programming them from scratch every time. Thus, bridging concepts from machine learning (ML) and control-theory (CT) has been omnipresent in robotics research over the last couple of years and has brought up promising results, especially for single-robot systems. In the context of reinforcement learning (RL) the core motivation is to equip robots with the ability of simultaneously exploring and learning unknown tasks. Building upon this, the concept of MARL has risen interest to improve scalability by executing tasks by a fleet of robots rather than a single autonomous (centralized) unit. In such settings it is desirable to handle each robot as an independent individual, such that the overall system still provides a sufficient performance-level even when a single robot may suffer from malfunction.

In this chapter we are focusing on MARL in the context of robotic systems, where a fixed set of  $N_{\mathfrak{q}}$  robots, which we denote as (artificial) agents in the remainder of this chapter, interact with an unknown environment in order to solve a joint task. As the design of a global reward-function itself is a challenging task that can doom the final performance of an RL or optimization algorithm, it is favorable to design an algorithm that is capable to obtain satisfactory results even in sparse reward environments. In contrast to most pure ML-based approaches, we claim that it is further not favorable to treat robotic systems, and thus also multi-agent systems as pure black-box systems that shall be solely learned from data. As there exist no usable framework that allows for an appropriate usage of model-based and model-free RL for multi-agent systems yet, this chapter evaluates the possibility of outlining a hierarchical MARL-framework. In order to allow future research projects to embed basic pre-knowledge about the individual agents, such a framework should allow a fully decentralized execution and learning of the individual agents involved. Therefore, this article evaluates on how to incorporate findings from applied robotic research and game-theory into MARL to achieve a fully decentralized learning, that may eventually be embedded into a hierarchical MARL.

In the remainder of this section we briefly sketch our contributions w.r.t. the state-of-the-art, followed by a summary of the technical foundations of this chapter and the technical problem in Section 6.2. The core concept of our proposed framework is outlined in Section 6.3. In order to evaluate the presented method we summarize our simulation environment in Section 6.4 and present the results of our method against state-of-the-art MARL methods in Section 6.5. Eventually, we propose promising extensions of our presented methods for future research projects in Section 6.6 and conclude this chapter in Section 6.7.

### 6.1.1 Related Work

Even though early applications of RL on robotic systems have shown promising results (Kolter and Ng, 2009, Ng et al., 2004), it was the success of outperforming humans in computer-games via deep-RL (Mnih et al., 2015, Silver et al., 2016, Vinyals et al., 2019) without suffering from catastrophic interference (McCloskey and Cohen, 1989) problems, that has opened the



door for RL-applications within complex, real-world environments. Given the computational power of modern graphics processing units (GPUs), policy gradients such as the stochastic policy gradient from [Sutton et al. \(1999a\)](#) or deterministic policy gradient (DPG) from [Silver et al. \(2014\)](#) have been realized via function approximators, such as neural networks (NNs). A famous example is given as the deep deterministic policy gradient (DDPG) from [Lillicrap et al. \(2016\)](#). DDPG has shown that deep RL can also be applied on continuous action-spaces such that the applicability of RL within robotic systems has been boosted drastically ever since. Even though further policy gradient (PG) methods have been developed in order to improve the variance sensitivity issue, such as trust region policy optimization ([Schulman et al., 2015a](#)), proximal policy optimization ([Schulman et al., 2017](#)) or maximum a-posteriori policy-optimization ([Song et al., 2020](#)), the majority of algorithms relies on an AC-architecture, where an additional critic reduces the variance drastically, such as the soft actor-critic (SAC) ([Haarnoja et al., 2018](#)). As an intense outline of advances in single-agent RL is beyond the scope of this chapter, we forward the interested reader to available literature survey papers ([Arulkumaran et al., 2017](#), [Kaelbling et al., 1996](#), [Kober et al., 2013](#)). Besides single-agent RL, the concepts of hierarchical reinforcement learning (HRL) and MARL have found great interest over the last decades and are thus outlined separately in the following.

#### 6.1.1.1 Hierarchical Reinforcement Learning

HRL follows the intuitive principle of divide and rule to split up problems into relaxed sub-problems. The concept of options as introduced by [Sutton et al. \(1999b\)](#) introduces hand-crafted *options* as temporally abstracted versions of actions. As a manual selection of options hinders the generalization of such approaches, [Bacon et al. \(2017\)](#) have introduced the option-critic architecture that simultaneously learns intra-option policies, termination functions and a policy over options without prior knowledge except the number of options to be learned. Alternatively, identifying options from data has been proposed by [Arulkumaran et al. \(2016\)](#).

Besides option-based methods, [Schaul et al. \(2015\)](#) have introduced the concept of universal value function approximation (UVFA), where promising sub-goals are identified. They propose to add current sub-goals to the input of the value-function to evaluate their value given the current state. As outlined by [Andrychowicz et al. \(2017\)](#) UVFAs allow to embed hindsight experience replay (HER) in order to increase the learning speed by altering the sub-goals in hindsight depending on the outcome of an episode. The hierarchical actor-critic (HAC) from [Levy et al. \(2019\)](#) combines HER and DDPG in a hierarchical architecture. In HAC, high-level actors or policies send sub-goals to the underlying policy and only the lowest layer chooses primitive actions that are executed on the environment. Due to its hierarchical nature it has an improved learning speed and allows application in sparse reward environments where related approaches often fail due to numeric sparseness. Introducing goal-conditioned observations also encouraged the research area on applying optimization and / or planning on collected data ([Eysenbach et al., 2019](#)).

The extension to MARL has already proposed by [Kulkarni et al. \(2016\)](#) and [Tang et al. \(2018\)](#), where concurrent HER for synchronous and asynchronous hierarchies was outlined, while [Ryu et al. \(2020\)](#) introduced hierarchical abstractions by clustering agents into individual groups. Similarly, the factorization of centralized critics via approaches like *QMIX* ([Rashid et al., 2018](#)) or *QTRAN* ([Son et al., 2019](#)) use a hierarchical critic ensemble to improve scaling

that is specifically tailored to multi-agent settings. This directly transitions to the current state-of-the-art in (deep-)MARL.

### 6.1.1.2 (Deep) Multi-Agent Reinforcement Learning

Besides solving complex Markov decision process (MDP) problems, the decentralized extension of Markov game (MG) has gained attention in the context of MARL (Littman, 1994, van der Wal, 1980). The naive approach of extending Q-Learning to a set of  $N_{\mathfrak{a}}$  independent learners (Tan, 1993) works well for small-scaled problems or selective applications. Similar to deep RL, initial results on MARL have been found on discrete action-sets, such as the Differentiable Inter-Agent Learning from Foerster et al. (2016) or explicit communication learning in Havrylov and Titov (2017), Mordatch and Abbeel (2017). In general though, independent learners violate the Markov assumption (Laurent et al., 2011).

Multi-agent deep deterministic policy gradient (MADDPG) is an extension of DDPG to MARL (Lowe et al., 2017), that also applies an AC architecture. During training, a centralized critic uses additional information about the other agents' states and actions to approximate the Q-function. Given this centralized critic, each agent updates a policy that is only conditioned on the local observations of each agent. Thus, the actor only relies on local observations during execution. MADDPG has achieved very robust results in simulated benchmark environments (Mordatch and Abbeel, 2017) for cooperative and competitive scenarios. Various extensions to MADDPG have been proposed. In Li et al. (2019), an extension to MADDPG has been introduced that used the minimax concept of game-theory to make decisions under uncertainty. The idea is to take the best action under the worst possible case.

As pointed out in Ackermann et al. (2019), the overestimation bias is also present in MARL. Some initial works have proposed to bridge concepts from the single-agent domain (van Hasselt, 2010) to MARL (Sun et al., 2020). Thus, SAC has been adjusted to the multi-agent domain in Wei et al. (2018), for which further extensions have been outlined, e.g., Zhang et al. (2020a) propose a Lyapunov-based penalty-term to the policy update, to stabilize the policy gradient. As a centralized learning is inherently suffering from poor scaling, Iqbal and Sha (2019) introduced attention-mechanisms in the multi-actor-attention-critic (MAAC). In order to cope for large-scale MARL, Sheikh and Bölöni (2020) explicitly differentiate between local and global reward metrics that each agent obtains from the environment.

In contrast to single-agent systems, the critic also suffers from the non-stationarity of the policies of other agents. This initiated the research of explicitly modeling the learning behavior of other agents, such as Foerster et al. (2018). Alternatively, Tian et al. (2019) proposed to model the MARL problem as an inference problem, i.e., to estimate the most likely action of the other agents and respond with the best-response (br).

As a full survey of MARL is beyond the scope of this chapter, we refer to Hernandez-Leal et al. (2019, 2020), Nguyen et al. (2020), Yang and Wang (2020), Zhang et al. (2021a) for a more detailed literature review. In order to illustrate the relevance of MARL from an application-driven perspective, there exists a variety of recent examples such as logistics (Tang et al., 2021), internet-of-things (Wu et al., 2021) or motion-planning for robots (He et al., 2021).

### 6.1.2 Contribution

While the majority of ML research follows the trend of enabling robots to learn tasks in an end-to-end manner, this chapter seeks towards leveraging this principle and proposes that existing model knowledge should be used in the context of multi-agent robotic systems. Concepts such as HRL have found great results in combining lower-level controllers or model knowledge in order to improve the overall performance for robotic systems. Within MARL there only exist rare applications of HRL, as the decoupled hierarchical actions often lead to asynchronous behavior, which eventually violates the Markov assumption, that existing AC-approaches rely upon.

Therefore, this thesis introduces a novel AC-method for MARL that allows to fully decouple the learning among the agents, while achieving comparable performance to current state-of-the-art MARL-approaches. This approach models the other agent by modeling the reactions of other agents to a selected action of each individual agent. This decision-theoretic principle stems from Stackelberg-equilibria from game-theory and is tailored to non-zero sum games in the scope of this chapter.

Our proposed method furthermore introduces another concept of multi-robot planning and game-theory by explicitly differentiating between joint task rewards and agent-specific, i.e., *native* costs. Namely, each agent estimates the performance of the joint policy w.r.t. the current task to be learned, but also a cost critic that evaluates the agent-specific cost.

Eventually, we propose a novel hierarchical AC-method that exploits the possibility of the br-based model to act asynchronously. In addition, we claim that factored observation representations are well-suited to boost the performance of MARL, but especially within HRL. In detail, we claim that it is beneficial to differentiate between internal agent states and external observations.

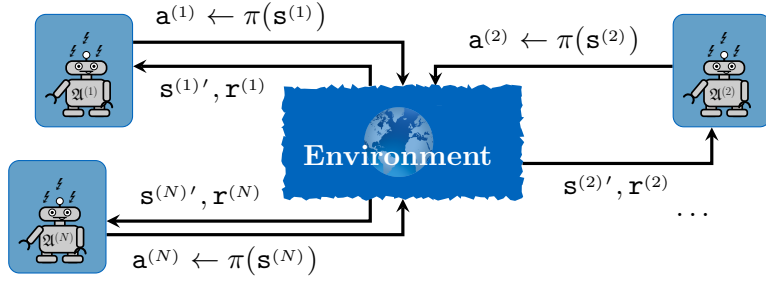
This allows to embed the presented br-ACs directly in the schematic of a hierarchical-AC. In contrast to our br-AC, this model evaluates the cost-critic w.r.t. to a virtual sub-goal that the agent is provided by an upper layer. By averaging the policy-gradient over the task and the hierarchical cost, the agent learns to optimize over the task and to prioritize to reach the current sub-goal.

As each agent acts independently, the presented method can directly exploit hierarchical learning concepts such as HER, which usually requires full synchronicity across the applied hierarchies.

As the presented framework opens a variety of further research that has been left blank within MARL, our last contribution is given by a short outline on possible extensions, that have a great potential to improve the performance of MARL.

## 6.2 Preliminaries

As the methods presented in this chapter build upon various findings from literature, we give an insight into these methods. We continue with introducing MGs.



**Figure 6.1:** Sketch of a general MARL problem, where  $N_{\mathfrak{A}}$  agents interact with each other in an unknown environment. Each agent has access to the individual state-observation  $\mathbf{s}^{(i)}$ , from which an action  $\mathbf{a}^{(i)}$  using the current policy  $\pi^{(i)}$  in such a manner that the expected individual return  $\mathbf{r}^{(i)}$  is maximized.

### 6.2.1 Markov-Games

An MG is an extension of an MDP to the multi-agent domain, that is fully described by the tuple  $(\mathfrak{A}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$ , where  $N_{\mathfrak{A}}$  agents  $\mathfrak{A} = (\mathfrak{A}^{(1)}, \mathfrak{A}^{(2)}, \dots, \mathfrak{A}^{(N_{\mathfrak{A}})}) = (\mathfrak{A})_{i \in N_{\mathfrak{A}}}$  interact with each other in a stochastic environment (Shapley, 1952, 1953) as shown in Figure 6.1. The state  $\underline{\mathbf{s}} = (\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \dots, \mathbf{s}^{(N_{\mathfrak{A}})}) \in \mathcal{S}$  of the environment with state-space  $\mathcal{S}$  is perceived as individual state-observations  $\mathbf{s}^{(i)}$  for each agent. Due to the Markov-property, the dynamics of an MG is given by each individual choosing an action  $\mathbf{a}^{(i)} \in \mathcal{A}^{(i)} \subset \mathcal{A}$  out of an agent-specific action-space  $\mathcal{A}^{(i)}$ , thus forming a joint action  $\underline{\mathbf{a}}$  that transitions  $\underline{\mathbf{s}}$  to  $\underline{\mathbf{s}}'$  according to a transition probability function  $\mathcal{T} := \mathbb{P}[\underline{\mathbf{s}}' | \underline{\mathbf{s}}, \underline{\mathbf{a}}]$ , and  $\mathbb{P}[\underline{\mathbf{s}}' | \underline{\mathbf{s}}, \underline{\mathbf{a}}]$  as the conditional probability for  $\underline{\mathbf{s}}'$  given  $\underline{\mathbf{s}}$  and  $\underline{\mathbf{a}}$ . The individual reward functions  $\mathcal{R} = (\mathcal{R}^{(i)} : \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \times \mathcal{A}^{(-i)} \times \mathcal{S}^{(i)} \rightarrow \mathbb{R})_{i \in N_{\mathfrak{A}}}$  map a transition from  $\underline{\mathbf{s}}$  to  $\underline{\mathbf{s}}'$  given  $\underline{\mathbf{a}}$ , to a numeric value for each agent  $\mathfrak{A}^{(i)}$ , denoted as  $\mathbf{r}^{(i)} := \mathcal{R}^{(i)}(\underline{\mathbf{s}}, \mathbf{a}^{(i)}, \underline{\mathbf{a}}^{(-i)}, \mathbf{s}^{(i)})$ . Given this, each agent  $\mathfrak{A}^{(i)}$  is following stochastic behavior policy  $\mathbf{a}^{(i)} \sim \pi^{(i)}(\mathbf{s}^{(i)})$  that intends to maximize the objective for each agent

$$\begin{aligned} \mathcal{J}^{(i)} &:= \sum_{t=0}^{\infty} \gamma^t \int_{\mathcal{A}} \pi(\underline{\mathbf{a}}_t | \underline{\mathbf{s}}_t) \int_{\mathcal{S}} \mathcal{T}(\underline{\mathbf{s}}_{t+1} | \underline{\mathbf{s}}_t, \underline{\mathbf{a}}_t) \mathbf{r}^{(i)} d\underline{\mathbf{s}}_{t+1} d\underline{\mathbf{a}}_t \\ &= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\underline{\mathbf{a}}^{(-i)} \sim \pi^{(-i)}(\mathbf{s}_t^{(i)})} \left[ \int_{\mathcal{A}} \pi(\mathbf{a}_t^{(i)} | \mathbf{s}_t^{(i)}) \mathbb{P}[\underline{\mathbf{a}}^{(-i)}] \int_{\mathcal{S}} \mathcal{T}(\underline{\mathbf{s}}_{t+1} | \underline{\mathbf{s}}_t, \underline{\mathbf{a}}_t) \mathbf{r}^{(i)} d\underline{\mathbf{s}}_{t+1} d\underline{\mathbf{a}}_t \right], \end{aligned} \quad (6.1)$$

where the hyperparameter  $\gamma \in (0, 1]$  is a temporal decay weight that scales short-term versus long-term impact.

In order to solve (6.1), the state-value function

$$\mathbf{V}_{\pi}^{(i)}(\underline{\mathbf{s}}) = \sum_{t=0}^{\infty} \mathbb{E}_{\underline{\mathbf{a}}_t \sim \rho(\pi), (\underline{\mathbf{s}}_t, \underline{\mathbf{s}}_{t+1}) \sim \rho(\mathcal{T}, \pi)} [\gamma^t \mathbf{r}^{(i)} | \underline{\mathbf{s}}_0], \quad (6.2)$$

state-action-function

$$\mathbf{Q}_{\pi}^{(i)}(\underline{\mathbf{s}}, \underline{\mathbf{a}}) = \mathbf{r}^{(i)}(\underline{\mathbf{s}}, \underline{\mathbf{a}}) + \gamma \mathbb{E}_{\underline{\mathbf{s}}' \sim \rho(\mathcal{T}, \pi)} [\mathbf{V}_{\pi}^{(i)}(\underline{\mathbf{s}}')], \quad (6.3)$$

and advantage-function

$$\mathbf{A}_{\pi}^{(i)}(\underline{\mathbf{s}}, \underline{\mathbf{a}}) = \mathbf{Q}_{\pi}^{(i)}(\underline{\mathbf{s}}, \underline{\mathbf{a}}) - \mathbf{V}_{\pi}^{(i)}(\underline{\mathbf{s}}), \quad (6.4)$$

have been introduced as the multi-agent version of the Bellman-backup operator for MDPs (Bellman, 1957). Given that the agents follow a fixed and optimal policy  $\pi^*$ , the dynamic program-problem eventually solves (6.1) as the global optimum of the MG as shown by Littman (1994). Given the optimal  $\mathbf{Q}_{\pi^*}$ -function, the optimal policies for each agent can be obtained as

$$\pi^{(i)}(\underline{\mathbf{s}})^* \leftarrow \arg \max_{\pi^{(i)}} \mathbf{Q}_{\pi^*}^{(i)}(\underline{\mathbf{s}}, \underline{\mathbf{a}}^{(-i)}, \mathbf{a}) \Big|_{\mathbf{a} \leftarrow \pi^{(i)}(\underline{\mathbf{s}})}. \quad (6.5)$$

As solving (6.5) requires each agent to follow an optimal policy, the definition of a best-response policy is of importance in MGs.

#### Definition 6.1: best-response policy

Given a joint policy  $\pi^{(-i)}$  for the neighboring agents of agent  $\mathfrak{A}^{(i)}$ , a policy  $\text{br}_{\pi^{(i)}}$  is called a br to  $\pi^{(-i)}$  if and only if

$$\mathcal{J}^{(i)}\left(\text{br}_{\pi^{(i)}} \Big| \pi^{(-i)}\right) \geq \mathcal{J}^{(i)}\left(\mathbf{a}^{(i)} \neq \text{br}_{\pi^{(i)}} \Big| \pi^{(-i)}\right),$$

i.e., agent  $\mathfrak{A}^{(i)}$  cannot improve the individual payoff-return  $\mathcal{J}^{(i)}$  by deviating from  $\text{br}_{\pi^{(i)}}$  (Shoham and Leyton-Brown, 2008).

Within an MG the optimal policy requires that the policies of the individual agents are a br to the policies of the surrounding agents, leading to the definition of a Nash-equilibrium (NE).

#### Definition 6.2: Nash-equilibrium

According to Nash (1950), a policy  $\text{NE}_{\pi} := (\text{NE}_{\pi}^{(i)})_{i \in N_{\mathfrak{A}}}$  is a NE if and only if each agent following  $\text{NE}_{\pi}^{(i)} \in \text{NE}_{\pi}$  results in each policy being a br-policy according to Definition 6.1. Replacing the objectives  $\mathcal{J}^{(i)}$  by the state-action values  $\mathbf{Q}^{(i)}$  this requires

$$\begin{aligned} \mathbf{Q}_{\text{NE}_{\pi}^{(i)}, \text{NE}_{\pi}^{(-i)}}^{(i)} &\geq \mathbf{Q}_{\tilde{\pi}^{(i)}, \text{NE}_{\pi}^{(-i)}}^{(i)} \\ \mathbf{Q}_{\text{NE}_{\pi}^{(i)}, \text{NE}_{\pi}^{(-i)}}^{(i)} &\geq \mathbf{Q}_{\text{NE}_{\pi}^{(i)}, \tilde{\pi}^{(-i)}}^{(i)}, \end{aligned}$$

with  $\tilde{\pi} \neq \text{NE}_{\pi}$ ,  $\forall \mathfrak{A}^{(i)} \in \mathfrak{A}, \forall \underline{\mathbf{s}} \in \mathcal{S}$

to hold on the global state-space  $\mathcal{S}$ .

Nonetheless, in real-world problems, neither  $\pi^*$  nor the value-functions are known. In addition, the environment is characterized by multiple learners, whose policies and thus actions vary over time and cannot directly be controlled by an individual agent in an MG. Which results in the problem formulation of this chapter.

### 6.2.2 Multi-Agent RL-problem

Given a set of agents  $(\mathfrak{A})_{i \in N_{\mathfrak{A}}}$ , that try to optimize their individual accumulated discounted reward according to Section 6.2.1, an optimal policy for each agent has to be found, that

fulfills:

- $(\pi \leftarrow \arg \max_{i \in N_{\mathfrak{q}}} Q_{\pi})$  according to (6.5).
- The joint action  $\underline{\mathbf{a}} = (\mathbf{a} \leftarrow \pi^*)_{i \in N_{\mathfrak{q}}}$  is an NE of the MG according to Definition 6.2.

We will continue with a short overview of RL methods that have been established as current state-of-the-art methods within single-agent RL and MARL.

### 6.2.3 Policy Gradient Methods

Obtaining an optimal policy  $\pi_{\Pi}$ , parameterized by  $\Pi$ , has been tackled by generating PGs (Sutton et al., 1999b), that estimate the stochastic gradient over  $\Pi$  of a policy the policy-loss function as

$$\nabla_{\Pi} \mathcal{J}(\pi_{\Pi}) = \mathbb{E}_{\mathbf{s} \sim \pi(\mathbf{s})} \left[ \sum_{t=0}^{\infty} \nabla_{\Pi} \log \pi_{\Pi}(\mathbf{a}_t | \mathbf{s}_t) \chi_t \right], \quad (6.6)$$

where  $\chi_t$  may for example be the single agent version of (6.3) or (6.4), i.e.,  $Q_{\pi}$  or  $A_{\pi}$ . If one can obtain the gradient  $\nabla_{\mathbf{a}} \chi_t$  directly, i.e., the action-space is continuous and the environment is stationary, it is also possible to obtain the DPG from (6.6) as

$$\nabla_{\Pi} \mathcal{J}(\pi_{\Pi}) = \mathbb{E}_{\mathbf{s} \sim \mathcal{D}} \left[ \nabla_{\Pi} \pi_{\Pi}(\mathbf{a} | \mathbf{s}) \nabla_{\mathbf{a}} \chi |_{\mathbf{a} \leftarrow \pi(\mathbf{s})} \right], \quad (6.7)$$

where the expectation is approximated by drawing samples from an experience replay buffer  $\mathcal{D}$ , that contains observed environment transitions. Exemplary DDPG uses  $\chi_t := Q_{\pi}$  in order to obtain the gradient of the state-action-value in (6.7). As it can be seen in (6.6) and (6.7), PGs and DPGs are in general highly sensitive to the variance of  $\chi_t$ . As a consequence, AC-methods have been outlined that add a policy evaluation metric to the policy update of PG methods.

### 6.2.4 Actor-Critic Methods

As the accumulated reward does in general suffer from high variance over repeated episodes, AC-algorithms simultaneously estimate  $A_{\pi}$  or  $Q_{\pi}$  alongside of the PGs in (6.6). Deep Q-network presented by Mnih et al. (2015) use NNs as function approximators, thus approximating  $Q_{\pi}$  by  $Q_{\Theta}$  and  $\dagger Q_{\Theta}$ , parametrized by  $\Theta$ , where  $\dagger Q$  denotes the *target-net* of  $Q$ . These two function approximators are then used to learn  $Q_{\pi}$  via off-policy temporal-difference learning, which is obtained via iteratively minimizing the loss-function

$$\begin{aligned} \mathcal{L}_Q(\Theta) &:= \mathbb{E}_{(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}') \sim \mathcal{D}} \left[ \frac{1}{2} (\mathbf{p} - Q_{\Theta}(\mathbf{s}, \mathbf{a}))^2 \right] \\ &\text{with } \mathbf{p} = \mathbf{r}(\mathbf{s}, \mathbf{a}, \mathbf{s}') + \gamma(1 - \mathbf{d}) \dagger v_{\Theta}(\mathbf{s}') \quad , \\ \dagger v_{\Theta}(\mathbf{s}') &= \mathbb{E}_{\mathbf{a}' \leftarrow \pi(\mathbf{s}')} \left[ \dagger Q_{\Theta}(\mathbf{s}', \mathbf{a}') \right] \end{aligned} \quad (6.8)$$

where  $\mathcal{D}$  is again a replay buffer that stores experienced transitions from the environment during the exploration process. Each sample contains the state  $\mathbf{s}$ , action  $\mathbf{a}$ , next state  $\mathbf{s}'$ , as well as the experienced reward  $\mathbf{r}$  and termination flag  $\mathbf{d}$ . The term  $(1 - \mathbf{d})$  thus ignores the value of the successor-state in the Bellmann-backup operator in (6.3) at terminal states.

The SAC (Haarnoja et al., 2018) is an extension of the general AC that approximates the solution of (6.1) via a maximum entropy objective by introducing a soft-value function, thus replacing (6.2) by

$$V_\pi(\mathbf{s}) := \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \sim \mathcal{D}} \left[ \sum_{t=0}^{\infty} \gamma^t \mathbf{r}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) + \alpha \mathbb{H}(\pi(\cdot | \mathbf{s}_t)) \right], \quad (6.9)$$

where  $\mathbb{H}(\cdot)$  denotes the policy entropy at a given state, and  $\alpha$  is a temperature parameter that weighs the impact of the entropy against the environment reward. In contrast to (6.2), this objective explicitly encourages exploration in regions of high rewards, thus decreasing the chance of converging to local minima. Further, two function approximators are used for the critic as in twin delayed deep deterministic policy gradient (TD3), such that the target value-function in (6.8) is obtained as

$$\dagger V_\Theta(\mathbf{s}') = \mathbb{E}_{\mathbf{a}' \sim \pi(\mathbf{s}')} \left[ \min_{j=1,2} \dagger Q_{\Theta,j}(\mathbf{s}', \mathbf{a}') - \alpha \log \pi_\Pi(\mathbf{a}' | \mathbf{s}') \right], \quad (6.10)$$

where  $\mathbf{a}$  is obtained from  $\pi(\mathbf{s}')$ , whereas  $\mathbf{s}'$  is drawn from  $\mathcal{D}$ . In contrast to this, the actual policy loss is obtained by applying the *reparameterization trick*

$$\mathcal{L}_\pi^{\text{SAC}}(\Pi) := \mathbb{E}_{\mathbf{s} \sim \mathcal{D}} \left[ \underbrace{\min_{j=1,2} \dagger Q_{\Theta,j}(\mathbf{s}, f_\varphi(\mathbf{s}, \mathbf{n}))}_{\chi} - \alpha \log \pi_\Pi(f_\varphi(\mathbf{s}, \mathbf{n}) | \mathbf{s}) \right], \quad (6.11)$$

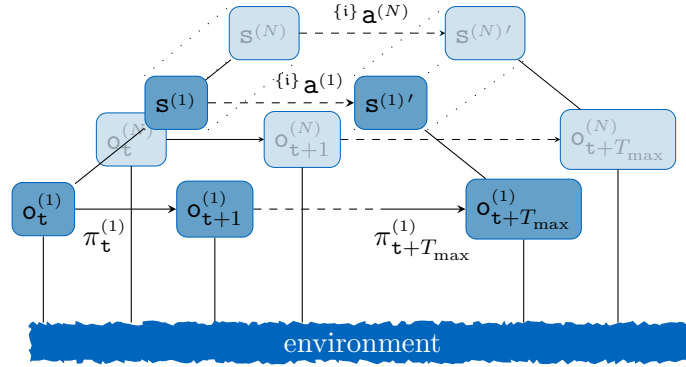
that computes a deterministic function  $f_\varphi(\mathbf{s}, \mathbf{n})$  that depends on the state  $\mathbf{s}$ , policy parameters  $\Pi$  and independent noise vector  $\mathbf{n}$  drawn from a fixed distribution, e.g., mean-free Gaussian noise. In contrast to e.g., DDPG, this parameterized policy is also squashed via a tanh function to the bounds of the action space, thus resulting in valid samples that can be used to generate a stochastic policy for the stochastic policy gradient update step.

### 6.2.5 Multi-Agent Actor-Critic Algorithms

The methods mentioned above have been recently extended to the multi-agent domain. The MADDPG extends AC with DDPG by proposing the schematic of decentralized execution in combination with centralized learning. As such, each  $\mathfrak{A}^{(i)}$  learns an individual (deterministic) policy  $\pi := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \mapsto \{0, 1\}$ , while setting  $\chi_t := Q^{(i)}(\underline{\mathbf{s}}, \underline{\mathbf{a}})$  in (6.7) that has access to the observations  $\mathbf{s}^{(i)}$ , actions  $\mathbf{a}^{(i)}$  and policies of all agents such that (6.8) can be directly applied on the multi-agent domain. This requires to have access to all policies during learning in order to calculate the target values of (6.8). Similar approaches have been proposed by MAAC and counterfactual multi-agent (COMA)-PG, that additionally incorporate a baseline value-function for the policy update and thus use the multi-agent advantage function

$$\chi := A^{(i)}(\underline{\mathbf{s}}, \underline{\mathbf{a}}) = Q^{(i)}(\underline{\mathbf{s}}, \mathbf{a}^{(i)}, \underline{\mathbf{a}}^{(-i)}) - V_b(\underline{\mathbf{s}}, \underline{\mathbf{a}}^{(-i)}) \quad (6.12)$$

for their policy loss declarations. The baseline  $V_b(\underline{\mathbf{s}}, \underline{\mathbf{a}}^{(-i)})$  estimates the value of a current state and the opponents current actions, such that optimizing (6.12) leads to a best-response action of agent  $\mathfrak{A}^{(i)}$  according to Definition 6.1. While COMA inserts (6.12) into (6.6), MAAC uses SAC, thus inserting (6.12) into (6.11). Furthermore, MAAC improves the centralized critic by adding an attention-mechanism that explicitly learns, which parts of the observations have an actual impact on the values of the critic.



**Figure 6.2:** Exemplary step of a two-level hierarchical MARL-step where each low-level step represents an interaction with the environment from Figure 6.1. For brevity, only selective nodes and edges are labeled. In here, the upper layer acts synchronously, such that the observed transition would qualify for centralized learning for all layers, which is emphasized via the dashed lines for the upper layer. Due to the hierarchical structure, the agents of the upper layer access the environmental observations  $\mathbf{s}$  to obtain a higher level action. In contrast, the agents in the lower level rely on the current observation and the latest sub-goal from the upper layer.

### 6.2.6 Hierarchical Actor-Critic

The HAC solves universal Markov decision processes (UMDPs), that extend MDPs by an additional goal-space  $\mathcal{G}$  the agent needs to reach from the current state or observation  $\mathbf{s}_t$ . In HAC, the observation  $\mathbf{s}$  of an agent is thus explicitly extended by a goal  $\mathbf{g} \in \mathcal{G}$ , generated from a  $K$ -level hierarchical policy set  $\{\{1\}\pi, \{2\}\pi, \dots, \{K\}\pi\}$ . This set consists of hierarchically decomposed sub-UMDPs, where  $\{k\}$  denotes the hierarchy-level the dedicated policy is learned for. While the lowest layer  $k = 1$  interacts with the environment, the goal-space of the highest layer is equal to the original UMDP goal-space. The coupling for the intermediate layers is given in the form of the action of an upper layer defining the goals of the lower layer. As a consequence,  $\{k\}\mathcal{A} = \mathcal{S}$  holds for all upper layers  $k \geq 2$ . Given that, UVFA and HER are applied in order to generate a goal-conditioned policy for each layer. HER allows to alter the observed data before storing it in  $\mathcal{D}$  in the form of setting reached states by lower layer policies as the actions of the upper layer and the goal of a current layer in hindsight. This has shown to boost the learning speed for single agent domains with sparse reward metrics, where pure sampling might fail to converge at all. As learning multiple coupled policies in parallel is prone to suffer from non-stationarity, they apply subgoal-testing in the rollouts that prohibits lower levels to draw exploration samples but rather follow their current policy metric in a deterministic, i.e., greedy manner.

## 6.3 Technical Approach

In the context of this chapter, we seek to incorporate findings from MARL and HRL. In the context of HRL for multi-agent systems, a key-challenge is given by handling asynchronous decisions, which directly results in decentralized learning and execution. This contradicts the paradigm of decentralized execution with centralized learning as commonly applied within MARL (Lowe et al., 2017). In HRL, each layer is modeled as a MDP, where a step of the higher layer consists of multiple steps of the layer below. For simplicity, we visualize this



scheme based on a synchronized step for  $N_{\mathfrak{A}}$  agents and a two-layered hierarchy in Figure 6.2. As it can be seen from this figure, centralized learning would not only require synchronous updates along all agents and layers, it also would require to know the current sub-goals of each agent.

In order to leverage these constraints, we propose a novel decentralized MARL concept that builds upon the concept of best-response-policies and separates joint rewards from internal agent objectives.

### 6.3.1 Decentralized MARL Based on Stackelberg-equilibria

In order to achieve a decentralized model for MARL-problems, previous work has evaluated the application of predicting the br-policy to the inferred action of an opponent (Tian et al., 2019) or assume overly restrictive access to the environment feedback of other agents. The latter is always fulfilled for centralized learning. In order to decouple the decentralized learning procedure, we propose a similar idea to Tian et al. (2019) and instead reformulate their inference-based policy by modeling the br-policy of other agents. In detail, we apply the concept of Stackelberg-equilibria. A Stackelberg-equilibrium evaluates the br of an agent, if the opponent has unveiled the current actions. Therefore, each agent regresses not only a policy  $\pi^{(i)}_{\Pi} := \mathcal{S}^{(i)} \mapsto \mathcal{A}^{(i)}$  – parameterized by  $\Pi$ – that intends to optimize the player-individual agent-objective, but also a br-policy  $\pi_{\text{br}}^{(-i)}_{\Xi} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \mapsto \underline{\mathcal{A}}^{(-i)}$  – parameterized by  $\Xi$ – that represents the reactions of the other agent(s) at each step.

In addition to regressing the br-policy of the other agent, we further claim that it is beneficial to distinguish between joint task rewards and individual / native cost-terms. In general, we assume that the individual reward for a cooperative MARL-problem is given in the form of

$$\mathcal{F}^{(i)}(\underline{\mathbf{s}}, \underline{\pi}, \underline{\mathbf{s}}') = \mathbf{r}^{(i)}(\underline{\mathbf{s}}, \underline{\pi}, \underline{\mathbf{s}}') - \hat{\ell}_{\text{nat}}^{(i)}(\underline{\mathbf{s}}, \pi^{(i)}, \underline{\mathbf{s}}') \in \mathbb{R}, \quad (6.13)$$

i.e., as a joint or cooperative task-reward that depends on the joint action or policy, as well as an interactive cost-component that only affects each player. While some existing work assumes to directly have access to local and global rewards, i.e., to obtain  $\mathbf{r}(\underline{\mathbf{s}}, \underline{\pi}, \underline{\mathbf{s}}')$  and  $\hat{\ell}_{\text{nat}}^{(i)}$  directly (Sheikh and Bölöni, 2020), we propose a model that only has access to the agent-reward as well as the averaged joint task reward of all agents. Thus, the cost of the agents needs to be estimated from this joint-reward at each transition. Thus, we apply

$$\begin{aligned} \mathbf{r}(\underline{\mathbf{s}}, \underline{\pi}, \underline{\mathbf{s}}') &\leftarrow \mathbf{r}^{(i)}(\underline{\mathbf{s}}^{(i)}, \underline{\mathbf{a}}, \underline{\mathbf{s}}^{(i)'}) \\ \hat{\ell}_{\text{nat}}^{(i)} &\approx \min \left( -\mathbf{r}^{(i)}(\underline{\mathbf{s}}^{(i)}, \underline{\mathbf{a}}, \underline{\mathbf{s}}^{(i)'}) + \frac{1}{N_{\mathfrak{A}}} \sum_{j=1}^{N_{\mathfrak{A}}} \mathbf{r}^{(j)}(\underline{\mathbf{s}}^{(i)}, \underline{\mathbf{a}}, \underline{\mathbf{s}}^{(i)'}), 0 \right), \end{aligned} \quad (6.14)$$

and implicitly assume that an agent only receives individual penalties, i.e., no negative cost values. Finally, our br-AC approach approximates:

- the (interactive) task critic  $\mathbf{Q}_{\text{int}}^{(i)} := \mathcal{S}^{(i)} \times \underline{\mathcal{A}} \mapsto \mathbb{R}$ , that intends to maximize the accumulated task reward.
- the (native) agent critic  $\mathbf{Q}_{\text{nat}}^{(i)} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \mapsto \mathbb{R}$ , that intends to minimize the agent-specific cost or penalties.

The agent-policy and critics can then be regressed by means of existing AC-methods, such as SAC or TD3. In contrast to the default methods, an averaged gradient over the critics above is required. As the cost needs to be minimized, the difference of the two critics provides the final critic that is used for the policy gradient of the current actor. Eventually, the br-policy needs to be updated as well. In contrast to the agent policy, the native critic is independent of the br-policy. As we emphasize on cooperative MARL, the br-policies intend to optimize the joint task-critic as well. Thus, the br-policy is given by obtaining the gradient of the joint task critic w.r.t. the policy of the other agents, after applying the current agent-policy. Denoting the cost-estimation from (6.14) as `GetCost`, a single update step for agent  $i$  is sketched in Algorithm 6.1. The dedicated critic losses – denoted as `CriticLoss` and

---

**Algorithm 6.1:** Decentralized br-policy based MARL-update step for agent  $i$ . Due to the decentralized learning, the update step can be run in a fully parallelized procedure.

---

```

1 Decentral update step for agent i:
2    $(\mathbf{s}, \mathbf{a}^{(i)}, \underline{\mathbf{a}}^{(-i)}, \underline{\mathbf{r}}, \mathbf{d}, \mathbf{s}') \sim \mathcal{D}^{(i)}$  ▷ sample agent experience batch
3   /* update (interactive) task critic  $Q_{\text{int}}^{(i)} := \mathcal{S}^{(i)} \times \mathcal{A} \mapsto \mathbb{R}$ , cf. (6.8) or (6.10) */
4    $\mathbf{a}^{(i)'} \leftarrow \dagger \pi(\mathbf{s}^{(i)'})$  ▷ get next action
5    $\underline{\mathbf{a}}^{(-i)'} \leftarrow \dagger \pi_{\text{br}}^{(-i)}(\mathbf{s}^{(i)'} | \mathbf{a}^{(i)'})$  ▷ get best-response to next action
6    $\mathcal{L}_{Q_{\text{int}}^{(i)}} \leftarrow \text{CriticLoss}(\mathbf{s}, \underline{\mathbf{a}}, \mathbf{r}^{(i)}, \mathbf{d}, \mathbf{s}', \underline{\mathbf{a}}')$ 
7   /* update (native) agent critic  $Q_{\text{nat}}^{(i)} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \mapsto \mathbb{R}$ , cf. (6.8) or (6.10) */
8    $\hat{\ell}_{\text{nat}}^{(i)} \leftarrow \text{CostCriticUpdate}(\underline{\mathbf{r}})$  ▷ estimate step-cost
9    $\mathcal{L}_{Q_{\text{nat}}^{(i)}} \leftarrow \text{CriticLoss}(\mathbf{s}^{(i)}, \mathbf{a}^{(i)}, \hat{\ell}_{\text{nat}}^{(i)}, \mathbf{d}, \mathbf{s}^{(i)'}, \mathbf{a}^{(i)'})$ 
10  /* update agent policy  $(\pi^{(i)} := \mathcal{S}^{(i)} \mapsto \mathcal{A}^{(i)})$  from critics, cf. (6.6), (6.7) or (6.11) */
11   $\mathcal{J}(\pi_{\text{br}}^{(-i)} | \Xi) = \mathbb{E}_{(\mathbf{s}^{(i)}, \underline{\mathbf{a}}^{(-i)})} \left[ \left( \nabla_{\mathbf{a}^{(i)}} Q_{\text{int}}^{(i)}(\mathbf{s}^{(i)}, \underline{\mathbf{a}}) - \nabla_{\mathbf{a}^{(i)}} Q_{\text{nat}}^{(i)}(\mathbf{s}^{(i)}, \mathbf{a}^{(i)}) \right) \Big|_{\mathbf{a}^{(i)} \leftarrow \pi^{(i)}_{\Pi}(\mathbf{s}^{(i)})} \right]$ 
12  /* update br-policy  $(\pi_{\text{br}}^{(-i)} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \mapsto \mathcal{A}^{(-i)})$  from task-critic, cf. (6.7) */
13   $\mathcal{J}() = \mathbb{E}_{(\mathbf{s}^{(i)}, \mathbf{a}^{(i)})} \left[ \nabla_{\underline{\mathbf{a}}^{(-i)}} Q_{\text{int}}^{(i)}(\mathbf{s}^{(i)}, \underline{\mathbf{a}}) \Big|_{\underline{\mathbf{a}}^{(-i)} \leftarrow \pi_{\text{br}}^{(-i)} | \Xi(\mathbf{s}^{(i)}, \mathbf{a}^{(i)})} \right]$ 

```

---

`CostCriticUpdate` in Algorithm 6.1 – are calculated by setting

$$\begin{aligned}
\mathbf{p}_{\text{int}} &= \mathbf{r}^{(i)}(\underline{\mathbf{s}}^{(i)}, \underline{\mathbf{a}}, \underline{\mathbf{s}}^{(i)'}) + \gamma(1 - \mathbf{d}) \left( \min_{j=1,2} \dagger Q_{\text{int},j}^{(i)}(\mathbf{s}^{(i)'}, \underline{\mathbf{a}}') + \mathbf{p}_{\text{SAC}} \right) \\
\mathbf{p}_{\text{nat}} &= \hat{\ell}_{\text{nat}}^{(i)} + \gamma^2(1 - \mathbf{d}) \left( \min_{j=1,2} \dagger Q_{\text{nat},j}^{(i)}(\mathbf{s}^{(i)}, \mathbf{a}^{(i)'}) \right) \\
\mathbf{a}^{(i)'} &\leftarrow \dagger \pi^{(i)}_{\Pi}(\mathbf{s}^{(i)'}) \\
\mathbf{a}^{(-i)'} &\leftarrow \dagger \pi_{\text{br}}^{(-i)} | \Xi(\mathbf{s}^{(i)'}, \mathbf{a}^{(i)'}) \\
\mathbf{p}_{\text{SAC}} &= \begin{cases} -\alpha \log \pi^{(i)}(\mathbf{a}^{(i)' | \mathbf{s}^{(i)'})} & \text{for an SAC model} \\ 0 & \text{else} \end{cases}
\end{aligned} \tag{6.15}$$

in (6.8). We explicitly do not apply SAC for the cost critic, as exploration should be emphasized on the task to be learned rather than exploring accumulated costs. Further cost-feedbacks are usually experienced locally. Therefore, we increase the temporal decay effects by squaring the decay-parameter for the cost-critic. Eventually, we distinguish between two interaction schemes in order to model  $\pi^{(-i)}_{\Pi, \text{br}}$ . First, the other agents can be modeled as an

unknown black-box system, usually denoted as a *game against nature* within game-theory. Thus, a single policy is tracked

$$\pi_{\text{br}}^{(-i)} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \mapsto (\underline{\mathcal{A}})_{j \in -i}, \quad (6.16)$$

that models an interaction with the current agent and the *responsive* nature. Our second approach uses a dyadic interaction scheme and models the br-policy of each agent to the current agent individually

$$\pi_{\text{br}}^{(j)} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \mapsto \mathcal{A}^{(j)}. \quad (6.17)$$

Both policies leverage the effect of mutual interaction among the other agents to diminish the combinatorial explosion. Given this decentralized learning scheme, we now continue with outlining a hierarchical MARL-framework.

### 6.3.2 Multi-Robot Hierarchical Actor Critic

Having outlined a decentralized MARL framework for flat hierarchies, we proposed to use findings from HRL to improve scaling in environments with sparse rewards. Our extensions for this hierarchies MARL framework rely on the a collection of assumptions, which is also assumed in existing work (Levy et al., 2019):

- There exists an agent-specific state-space  $\mathcal{X}^{(i)} \in \mathcal{S}^{(i)}$ , and  $\mathbf{x}^{(i)} \in \mathbf{s}^{(i)}$  always holds.<sup>1</sup>
- There exist deterministic mapping functions  $\mathcal{F}_g := \mathcal{X}^{(i)} \times \{\text{p}\} \mathcal{A}^{(i)} \mapsto \{\text{p-1}\} \mathcal{G}^{(i)} \in \mathcal{X}^{(i)}$  and  $\mathcal{F}_g^{-1} := \mathcal{X}^{(i)} \times \{\text{p-1}\} \mathcal{G}^{(i)} \mapsto \{\text{p}\} \mathcal{A}^{(i)}$ , that map the actions of the upper layer to the goal-space of the lower layer and vice-versa for each agent.
- There exists a deterministic evaluation-metric  $\{\text{p}\} \mathcal{S} := \{\text{p}\} \mathcal{G} \times \mathcal{X}^{(i)} \mapsto [0, 1]$ , that evaluates the achievement of a goal  $\{\text{p}\} \mathbf{g}^{(i)}$  given the current agent state  $\mathcal{X}^{(i)}$ .

This differs from the original assumption from Levy et al. (2019) by the fact, that we propose to explicitly distinguish between the internal agent-state  $\mathbf{x}^{(i)}$  and the full environment observation  $\mathbf{s}^{(i)}$ .

In fact, we claim that within multi-agent HRL it is specifically beneficial to distinguish between internal and external observations. Therefore, we use structured observations as

$$\mathbf{s}^{(i)} := (\mathbf{x}, \mathbf{y}_{\text{e}}, \mathbf{y}_{(-i)})_{i \in N_{\mathfrak{A}}}, \quad (6.18)$$

where  $\mathbf{x}^{(i)}$  reflects the internal state of an agent, e.g., current position, velocity, etc., and environmental observations  $\mathbf{y}_{\text{e}}^{(i)}$  from  $\mathfrak{A}^{(i)}$ 's perspective, e.g., images or laser range data, as well as observations of the other agents  $\mathbf{y}_{(-i)}^{(i)}$ .

Given this representation, we propose a two-layered hierarchy, where the upper layer proposes sub-goals to the lower layer agents. This lower-level is denoted as the *environment-layer* or  $\{\text{p}\} \mathbf{p}$  in the following, while the upper layer is denoted as the *team-coordination-layer* or  $\{\text{i}\} \mathbf{p}$ . On the lowest layer, we apply the br-policies from Section 6.3.1 using a dyadic interaction scheme, where the individual components per agent are given as:

- the joint task critic  $\mathbf{Q}_{\text{int}}^{(i)}(\mathbf{s}^{(i)}, \underline{\mathbf{a}})$ .

<sup>1</sup>The assumption of  $\mathbf{x}^{(i)} \in \mathbf{s}^{(i)}$  emphasizes that we do not expect full state-observability, and that the intrinsic observations do not provide additional knowledge to the robotic agents.

- the native hierarchical critic  $Q_{\text{nat}}^{(i)}(\mathbf{x}^{(i)}, \mathbf{y}_{\mathbf{e}}^{(i)}, \mathbf{g}^{(i)})$ .
- a goal-conditioned action-policy for the current agent  $\pi^{(i)}(\mathbf{s}^{(i)}, \mathbf{g}^{(i)})$ .
- the dyadic br-policies  $\left(\pi^{(j)}\left(\mathbf{x}, \mathbf{y}_{\mathbf{e}}, \mathbf{y}^{(-i)}, \mathbf{a}_{i \in N_{\mathfrak{A}}}\right)\right)_{j \in -i}$ .

As the individually agents are provided with a sub-task that is to be reached by the dedicated agents alone, the hierarchical native critic preferably drops the dependency on the observation of other agents. Namely, this native hierarchical critic evaluates the hierarchically imposed rewards instead of estimating the current step-cost from the deviation w.r.t. the average reward. As a result, the update-step of the lower layer follows Algorithm 6.1 but replaces the difference in Line 9 by an average over the two critics. Furthermore, the native critic does not only evaluate the environmental task-success  $\mathbf{d}$ , but also if the update-step has accomplished the current sub-goal. As the rest remains identical to Algorithm 6.1 and (6.15), we omit repeating the same equations.

In contrast, the upper layer only tracks a single critic, as infeasible sub-goals are also resulting in unpredictable task-performance. Unfortunately, the agents do not have access to the goal-mapping of other agents, such that it is impossible to directly impose their policies or higher-level actions in the critic within decentralized settings. Further, the upper layer usually suffers from asynchronous decisions, which would require to add the decision-epochs to the state of the critic to allow sampling from the experience buffer. Therefore, we propose to apply an observation-oracle instead of a br-policy

$$\{i\}\pi_{\text{br}}^{(j)} := (\mathcal{X} \times \mathcal{Y}_{\text{env}} \times \mathcal{Y}^{(j)} \times \{i\}\mathbf{a}_{i \in N_{\mathfrak{A}}}) \mapsto \mathcal{Y}^{(j)}, \quad (6.19)$$

i.e., instead of predicting the agent-action on the upper layer, the next observation is predicted. In case, a (partially) centralized learning scheme is applied, this observation-oracle can also be replaced with

$$\{i\}\pi_{\text{br}}^{(j)} := (\mathcal{X} \times \mathcal{Y}_{\text{env}} \times \mathcal{X}^{(j)} \times \{i\}\mathbf{a}_{i \in N_{\mathfrak{A}}}) \mapsto \mathcal{X}^{(j)}, \quad (6.20)$$

thus predicting the next internal state of agent  $j$ . These opponent models allow to use data from an experience buffer independent of the higher-level policies or decision-epochs during execution. As a result, the (interactive) task critic of the upper layers are regressed in one of the following representations:

$$\{i\}Q_{\text{int}}^{(i)} := \mathcal{S}^{(i)} \times \mathcal{A}^{(i)} \times \left(\mathcal{Y}_{(j)}^{(i)}\right)_{j \in -i} \mapsto \mathbb{R} \quad (6.21)$$

$$\{i\}Q_{\text{int}}^{(i)} := \mathbf{s} \times \mathbf{a}^{(i)} \times \left(\mathcal{X}_{(j)}^{(i)}\right)_{j \in -i} \mapsto \mathbb{R} \quad (6.22)$$

Given these models, the overall hierarchical MARL-framework is summarized by the algorithmic skeleton in Algorithm 6.2. For insights about HER, we refer to existing work (Levy et al., 2019) but also to the listed extensions at the end of this chapter, cf. Section 6.6. At the beginning of each update-step, the environment layer has to check for necessity of new sub-goals, i.e., if the upper layer has to draw a new action. Selecting the layer-testing is omitted for brevity, so we refer to Levy et al. (2019).

---

**Algorithm 6.2:** Proposed hierarchical MARL algorithmic episode step. In here, the model-parameters are expected to be initialized. Further, the function expects the following inputs: the initial observation, a hierarchical replay buffer  $\mathcal{D}_{\text{HER}}$ .

---

**Input:**  $\mathbf{s}_0, \mathcal{D}_{\text{HER}}$

```

1 for  $t = 0$  to  $T_{\text{eps}}$  do
2    $(\{i\} \mathbf{g}, \{i\} \mathbf{a} \leftarrow \text{RequestGoal}(\mathbf{s}))_{i \in N_{\mathfrak{A}}}$            ▷ request new sub-goal as needed
3    $(\mathbf{a} \leftarrow \{i\} \pi(\mathbf{s}, \mathbf{g}))_{i \in N_{\mathfrak{A}}}$                        ▷ get environment action
4    $\underline{\mathbf{s}}', \underline{\mathbf{r}}, \underline{\mathbf{d}} \leftarrow \text{EnvStep}(\underline{\mathbf{a}})$                  ▷ update environment
5   if  $\underline{\mathbf{d}} \vee (\mathbf{g} \in \underline{\mathbf{s}}')_{i \in N_{\mathfrak{A}}}$  then
6     ApplyHER( $\mathcal{D}_{\text{HER}}$ )                                       ▷ apply HER
7     if  $\underline{\mathbf{d}}$  then
8       break                                               ▷ stop exploration if task is done
9    $\mathcal{D}_{\text{HER}} \leftarrow \mathcal{D}_{\text{HER}} \cup (\underline{\mathbf{s}}, \underline{\mathbf{a}}, \underline{\mathbf{g}}, \underline{\mathbf{r}}, \underline{\mathbf{d}}, \underline{\mathbf{s}}')$ ,  $\underline{\mathbf{s}} \leftarrow \underline{\mathbf{s}}'$    ▷ add data to buffer and update state
10 for  $n_{\text{train}} = 1$  to  $N_{\text{train}}$  do
11   (TrainLayer( $\{i\} \mathcal{D}^{(i)}$ )) $_{i \in N_{\mathfrak{A}}}$                  ▷ update lower-level, cf. Algorithm 6.1
12   (TrainLayer( $\{i\} \mathcal{D}^{(i)}$ )) $_{i \in N_{\mathfrak{A}}}$                  ▷ update higher-level

```

---

While the lower layer is updated similarly to Algorithm 6.1 as stated in Line 11, the lower layer critic update is given by calculating

$$\begin{aligned}
\mathbf{p} &= \frac{1}{\{i\} T_{\text{max}}} \sum_{n=1}^{\{i\} T_{\text{max}}} \gamma^{\{i\} T_{\text{max}} - n} \mathbf{r}^{(i)}(\mathbf{s}_{t+n}, \underline{\mathbf{a}}_{t+n}, \mathbf{s}_{t+n+1}) \\
&+ (1 - \mathbf{d}) \begin{cases} 0 & \text{if } \mathcal{S}(\mathbf{x}^{(i)'}, \mathcal{F}_g(\mathbf{x}^{(i)}, \{i\} \mathbf{a}^{(i)})) \mapsto \top \\ -(T_{\text{max}} - n) & \text{if } \exists n : \mathcal{S}(\mathbf{x}^{(i)}_{t+n+1}, \mathcal{F}_g(\mathbf{x}^{(i)}, \{i\} \mathbf{a}^{(i)})) \mapsto \top \\ -T_{\text{max}} & \text{else} \end{cases}, \quad (6.23) \\
&+ \gamma(1 - \mathbf{d}) \left( \min_{k=1,2} \{i\} \dagger \mathbf{Q}_{\text{int},k}^{(i)} \left( \mathbf{s}^{(i)'}, \{i\} \mathbf{a}^{(i)'}, \left( \mathbf{y}_{(j)}^{(i)'} \right)_{j \in -i} \right) \right) \\
&\quad \{i\} \mathbf{a}^{(i)'} \leftarrow \{i\} \dagger \pi_{\Pi}^{(i)}(\mathbf{s}^{(i)'}) \\
&\quad \left( \mathbf{y}' \leftarrow \{i\} \dagger \pi_{\text{br}\Xi}^{(j)}(\mathbf{s}^{(i)'}, \{i\} \mathbf{a}^{(i)'}) \right)_{j \in -i}
\end{aligned}$$

where  $\{i\} T_{\text{max}}$  denotes the number of maximum sub-steps for a hierarchical transition, and  $\mathbf{s}' := \mathbf{s}_{t+\{i\} T_{\text{max}}}$  represents the observation of agent  $i$  after a hierarchical update step. The first term averages the environmental reward, while the second adds the hierarchical penalty-term depending on whether the lower layer could achieve the current action or respective sub-goal. Eventually, the value-function is approximated via querying the current higher-level policy and predicting the observations of the other agents.

To conclude the overall algorithm, the br-policies can again be learned by treating the state-predictor or observation predictor as an additive policy to the current state, or simply applying behavioral cloning. For the latter, it is beneficial to add exploration noise on the observed agent-observations and evaluate the sampled data on the current critic. Instead of learning

the exact behavior, the cloning is then to be obtained on the maximum, i.e., best performing samples.

## 6.4 Materials and Methods

The proposed algorithm has been evaluated on the multi-agent particle environment (MPE) that has been extended from previous work (Lowe et al., 2017, Mordatch and Abbeel, 2017) to fit the scope of this chapter. The source code of the benchmark scenarios<sup>2</sup> and the presented work<sup>3</sup> can be found online, while the hyper-parameters and further implementation details leading to the results are listed in Appendix B.

In order to meet the assumptions stated in Section 6.3, the original simulation environment has been adjusted such that the agents are able to differentiate between internal, external agent-related and external environment-based observations. Thus, we introduce structured observations for our adjusted version of the MPE. Further, the goal-mapping and evaluation metrics stated in Section 6.3 have been handcrafted and embedded into the dedicated environments similar to the original work from Levy et al. (2019). As claimed in Section 6.3 our approach tackles cooperative multi-robot RL tasks, such that only environments with pure continuous action-spaces have been tested. Before outlining the experimental findings collected, we shortly highlight the adjustments that were added to the default gym-environment and the MPE.

### Structured Observations in the Multi-Agent Particle Environment

As stated in Section 6.3, the observation of each agent is obtained as  $\mathbf{s}^{(i)} := (\mathbf{x}^{(i)}, \mathbf{y}_c^{(i)}, \mathbf{y}_{(-i)}^{(i)})$ . The MPE is characterized by  $N_{\mathfrak{A}}$  agents moving in a  $xy$ -planar surface by applying a force on their body center. Thus, each agent is implemented as a point-mass, where the internal state and action are defined as

$$\mathbf{x}^{(i)} := \begin{bmatrix} \mathbf{x}^{(i)} \\ \mathbf{y}^{(i)} \\ \dot{\mathbf{x}}^{(i)} \\ \dot{\mathbf{y}}^{(i)} \end{bmatrix} \quad \mathbf{a}^{(i)} := \begin{bmatrix} f_x^{(i)} \\ f_y^{(i)} \end{bmatrix}, \quad (6.24)$$

where the action is a planar force<sup>4</sup> actuated on the individual point-masses, which then follow the linear point-mass dynamics

$$\mathbf{x}^{(i)'} := \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & \mathbf{v} & 0 \\ 0 & 0 & 0 & \mathbf{v} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(i)} \\ \mathbf{y}^{(i)} \\ \dot{\mathbf{x}}^{(i)} \\ \dot{\mathbf{y}}^{(i)} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \end{bmatrix} \mathbf{a}^{(i)}, \quad (6.25)$$

<sup>2</sup>Source code: [https://gitlab.com/vg\\_tum/multi-agent-gym.git](https://gitlab.com/vg_tum/multi-agent-gym.git)

<sup>3</sup>Source code: [https://gitlab.com/vg\\_tum/mahac\\_rl.git](https://gitlab.com/vg_tum/mahac_rl.git)

<sup>4</sup>Various implementations available online realize the action input as the difference of two positive force terms, as this eases the comparison to discrete action spaces, where the result equals learning an optimal bang-bang controller. As our framework explicitly highlights continuous applications, we kept this implementation for the comparison to the state-of-the-art methods, but used the interfaces from (6.24) for our method.

using the mass of the entity  $\mathbf{m}$  and a damping-term  $\mathbf{v} \in [0, 1]$  in free-space. In case a particle collides with an object or an agent, a simple point-mass collision is applied. Even though the actual observation highly depends on the actual scenario or task to be solved, all our implementations contain the internal agent-state in the observation of the agents.

As a result, the mapping-functions for the MPE are then given as:

$$\begin{aligned} \{c\} \mathbf{g}^{(i)} \leftarrow \mathcal{F}_{g_{\text{MPE}}}(\mathbf{x}^{(i)}, \{i\} \mathbf{a}^{(i)}) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}^{(i)} + \{i\} \mathbf{a}^{(i)} \\ \{i\} \mathbf{a}^{(i)} \leftarrow \mathcal{F}_g^{-1}(\mathbf{x}^{(i)}, \{c\} \mathbf{g}^{(i)}) &= \{i\} \mathbf{g}^{(i)} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}^{(i)}, \end{aligned} \quad (6.26)$$

while the success-metric is simply given as

$$\{i\} \mathcal{S}_{\text{MPE}}(\mathbf{x}^{(i)}, \{c\} \mathbf{g}^{(i)}) := \left\| \{c\} \mathbf{g}^{(i)} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}^{(i)} \right\|_2 \leq \zeta_{\mathbf{g}, \text{MPE}}. \quad (6.27)$$

The threshold-parameter is thus given as another hyper-parameter that is listed in Appendix B.

As claimed in Section 6.3, our approach tackles cooperative multi-robot RL tasks, such that only environments with pure continuous action-spaces have been tested. Besides *cooperative navigation*, we evaluated our approaches on the *cooperative collection* task, in which  $N_{\mathfrak{A}}$  agents are asked to reach  $N_{\mathfrak{A}}$  goal-locations. The reward-signal is provided sparsely:

$$\mathbf{r}^{(i)} \leftarrow \sum_{k=0}^{N_{\mathfrak{A}}} \begin{cases} 0 & \text{if } \text{visited}(\mathbf{g}_k) \vee \exists j : \|\mathbf{x}^{(j)} - \mathbf{g}_k\|_2 \leq \zeta_{\mathbf{g}, \text{MPE}} \\ -1 & \text{else} \end{cases}. \quad (6.28)$$

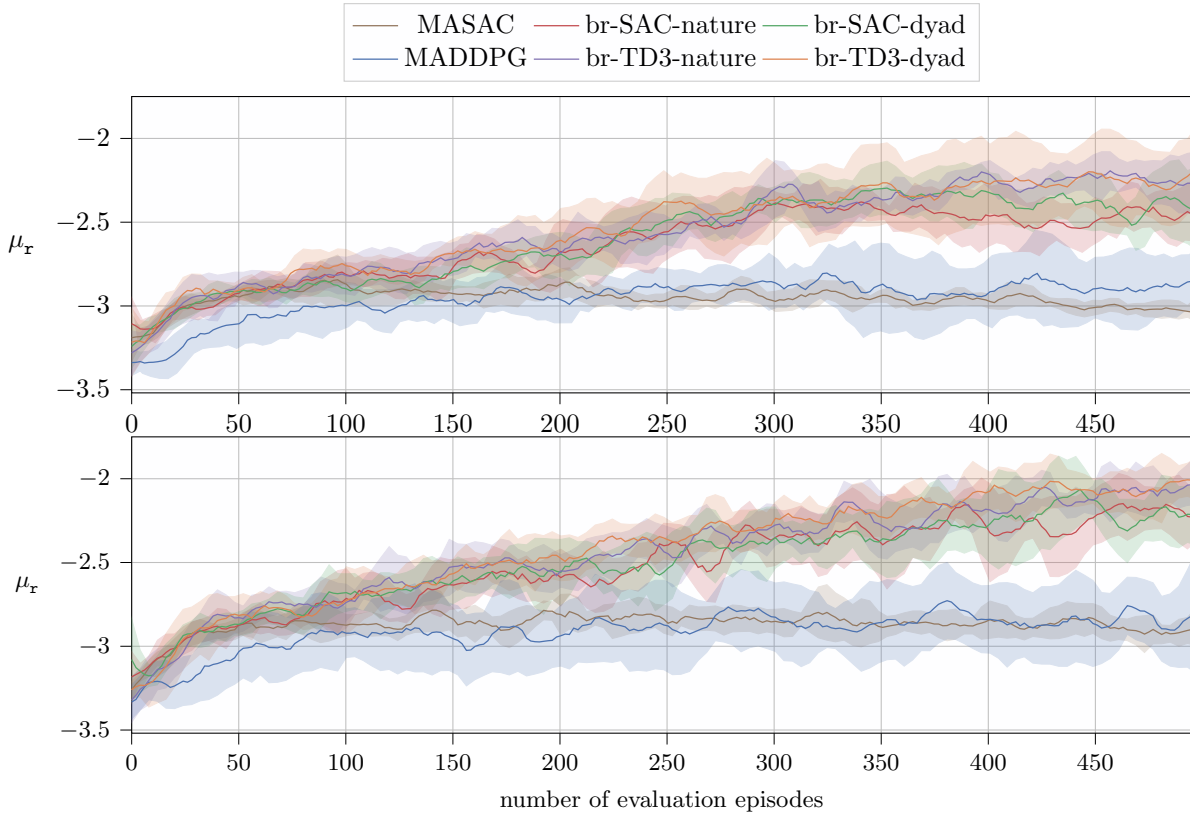
In addition, each agent is penalized with a direct cost-value of  $-1$  every time a collision with the environment or another agent is encountered.

## 6.5 Results

In this section we evaluate the performance of our decentralized br-policy MARL-framework within the simulation environment from Section 6.4. Within this environment, we evaluated our algorithm against state-of-the-art algorithms within MARL, namely MADDPG and multi-agent soft actor-critic (MASAC).

Given our adjusted MPE benchmark environment, we ran a decentralized version of TD3 (Ackermann et al., 2019) and the multi-agent version of Haarnoja (2018) for the joint critic in our algorithm. Given the dyadic and game against nature variants, we use the following notations:

- The state-of-the-art algorithm are directly denoted as commonly known in literature, i.e., MADDPG and MASAC.
- Our extension of TD3 is denoted as br-TD3-dyad/nature.
- Our extension of SAC is denoted as br-SAC-dyad/nature.



**Figure 6.3:** Results of the decentralized br-based algorithms for the cooperative collection task using sparse rewards. The figures present averaged rewards of all agents over 8 learning-runs per algorithm and environment. The shaded areas highlight a confidence-interval (CI) of 70%. The upper figure shows the performance of the collection task with static goal-locations, whereas the environment on the bottom samples new goal-locations upon every reset. The  $x$ -axis denotes the evaluation steps, which are run after 10 exploration episodes to evaluate the current performance.

As stated above, our main emphasis is set on improving the performance in sparsely rewarded environments, Further, we explicitly tailor our approach to continuous action-spaces in cooperative sections. Therefore, we applied the cooperative collection task according to the parameterization in Appendix B.

For brevity, we present the evaluation-performance of the individual algorithms based on the average rewards of all agents in Figure 6.3. In here, the term *evaluation* refers to the agents following their current policies in a greedy manner rather than drawing samples from it. The collected results show a static – i.e., using fixed goal-locations – in the lower figure and a non-static version in the upper one. In this environment three agents are updating their policies over 5000 exploration episodes. As the evaluation is only run every tenth episode, the number of evaluation steps is lower than the actual explorations. In addition, it has to be mentioned that the averaged reward per evaluation run is logged, which in return strongly depends on a randomly sampled starting state of the agent – and also the goal-locations in the non-static environment. As a result, the collected data suffers from high noise, which is reduced by smoothing the collected reward-values using a Savitzky-Golay filter (Savitzky and Golay, 1964) and the implementation from Virtanen et al. (2020).



As it can be seen, our algorithms outperform the current state-of-the-art algorithms in terms of final performance but also in terms of convergence speed for both scenarios. Unsurprisingly, our method performs best in static-environment that requires the agents to reach static goal locations. In these scenarios, there is a direct relation between the agent states and the actual value-functions, which leads to an improved learning rate.

	static	MADDPG	MASAC	br-TD3-d	br-TD3-n	br-SAC-d	br-SAC-n
$\mathbf{r}^{(1)}$		$-2.92 \pm 0.38$	$-2.95 \pm 0.24$	<b><math>-2.54 \pm 0.48</math></b>	$-2.55 \pm 0.47$	$-2.6 \pm 0.43$	$-2.64 \pm 0.42$
$\mathbf{r}^{(2)}$		$-2.94 \pm 0.4$	$-2.96 \pm 0.23$	<b><math>-2.54 \pm 0.48</math></b>	$-2.57 \pm 0.47$	$-2.6 \pm 0.43$	$-2.64 \pm 0.43$
$\mathbf{r}^{(3)}$		$-2.99 \pm 0.44$	$-2.95 \pm 0.23$	<b><math>-2.54 \pm 0.48</math></b>	$-2.55 \pm 0.46$	$-2.61 \pm 0.44$	$-2.64 \pm 0.43$
$\mathbf{d}_{\text{ev}}$		3	0	<b>35</b>	31	12	10
$\mathbf{d}_{\text{ex}}$		86	62	<b>1121</b>	901	256	246
$\mathbf{r}^{(1)}$	✓	$-2.89 \pm 0.46$	$-2.87 \pm 0.3$	<b><math>-2.4 \pm 0.51</math></b>	$-2.41 \pm 0.51$	$-2.5 \pm 0.52$	$-2.51 \pm 0.53$
$\mathbf{r}^{(2)}$	✓	$-2.97 \pm 0.5$	$-2.87 \pm 0.3$	<b><math>-2.42 \pm 0.51</math></b>	$-2.44 \pm 0.52$	$-2.5 \pm 0.52$	$-2.51 \pm 0.53$
$\mathbf{r}^{(3)}$	✓	$-2.88 \pm 0.44$	$-2.87 \pm 0.3$	<b><math>-2.41 \pm 0.51</math></b>	$-2.44 \pm 0.52$	$-2.5 \pm 0.52$	$-2.52 \pm 0.53$
$\mathbf{d}_{\text{ev}}$	✓	7	0	50	<b>52</b>	31	41
$\mathbf{d}_{\text{ex}}$	✓	289	68	<b>1391</b>	1284	554	481

**Table 6.1:** Detailed performance metrics for evaluated environments. Again the results of the static environment are listed on the bottom. The best performing values are highlighted in bold. The values show the averaged results with the optional standard-deviation appended by  $\pm$ . The terms dyadic and nature are abbreviated by their first letter for brevity. Similarly, the number of successful trials success of the exploration and evaluation runs are denoted as  $\mathbf{d}_{\text{ex}}$  and  $\mathbf{d}_{\text{ev}}$ .

For a closer evaluation of our presented algorithms, the per-agent rewards-metrics are listed in Table 6.1. Furthermore, the number of total successful trials per algorithm during exploration and evaluation are listed. An exploration is not only run distinctly more often, it also contains double the amount of steps per run. As a consequence, the number of successful exploration runs is distinctly higher compared to the evaluation numbers.

Nonetheless, the collected numbers underline that our presented method outperforms current state-of-the-art methods distinctly, not only in terms of averaged accumulated reward as shown in Figure 6.3, but also for each individual agent involved. The performance increase becomes evident by comparing the success rates of the algorithms, where MASAC even failed completely to find a successful policy.

Comparing the overall results, the TD3-agents outperformed not only the state-of-the-art methods, but also our SAC-variants. Furthermore, the dyadic setup resulted in improved performance for all evaluation metrics compared to the game against nature schematic. This confirms our initial statement that it is preferable to handle interactions individually, rather than regressing interaction schemes fully from a NN.

Regarding the standard-deviations of our proposed methods, it also becomes evident that our methods suffer from higher variance in the accumulated rewards. Even though this may seem as a disadvantage of our approaches compared to the existing algorithms, it has to be kept in mind that a successful episode usually distinctly differs from an unsuccessful episode, thus directly resulting in an increased variance. Regarding the number of successful samples per algorithm, this directly relates to the increased variance.

In summary, it can be stated that our presented algorithms outperform the existing methods within our simulated environments and are thus valuable extensions that can be used within

a hierarchical MARL-framework as outlined in Section 6.3.2. In order to further improve the performance of such a hierarchical MARL-framework, we outline explicit extensions to the presented methods in the following chapter to close this part of the thesis.

## 6.6 Possible Extensions

In this section, we elaborate possible extensions of the presented approach in order to increase the applicability and further enhance the performance.

### 6.6.1 Applying Best-Response Policies on Competitive Environments

If the agents have access to all reward values during learning, an additional critic for the objective of the other agent can be added to the presented algorithm. This results in applying the gradient-step for the br-policy not only over the joint task critic for the current agent, but also the agent-specific agent critic. If applying this metric, it is strongly recommended to apply the dyadic interaction scheme from above, as our algorithm is restricted to optimizing the average reward over all agents otherwise.

Another extension is given by modeling non-cooperative agent(s). In order to model this procedure, it is best to condition non-cooperative agents on the joint team-policy of all cooperative agents, thus leading to the conditional interaction policy:

$$\underline{\pi}(\underline{\mathbf{s}}) \approx \underline{\pi}^{(-i,-i)}(\mathbf{s}^{(i)} | \underline{\mathbf{a}}^{(-i,i)}, \mathbf{a}^{(i)}) \underline{\pi}^{(-i,i)}(\mathbf{s}^{(i)} | \mathbf{a}^{(i)}) \underline{\pi}^{(i)}(\mathbf{s}^{(i)}), \quad (6.29)$$

where the cooperative policy is denoted as  $\underline{\pi}^{(-i,i)}(\mathbf{s}^{(i)} | \mathbf{a}^{(i)})$  and the non-cooperative policy is denoted as  $\underline{\pi}^{(-i,-i)}(\mathbf{s}^{(i)} | \underline{\mathbf{a}}^{(-i,i)}, \mathbf{a}^{(i)})$ . Alternatively, on-policy-based approaches, such as proximal policy optimization (PPO) or trust region policy optimization are worth an investigation to model the behavior of other agents. In here, a direct approach is given by conditioning the policy estimate on the average over all estimators. A more promising approach would be given by averaging over all agent-advantages, and thus applying a gradient-step. This bares the potential on stabilizing the estimated opponent models and thus the overall task-critic updates, which eventually increases the likelihood of converging to the team-optimal policy.

### 6.6.2 Improving Convergence Behavior by Partially Centralized Learning

The presented method fully decouples learning by learning opponent models without applying centralized learning schemes. This is endangered to lead to divergent agent behavior and thus converging to suboptimal team-behavior. Therefore, our current method could be further enhanced by introducing centralized learning without adding restrictive full observability assumptions. Rather than sharing the full observations, the individual opponent policy predictions can be shared during learning, such that the policy gradient can be conditioned on the Kullback–Leibler (KL)-divergence of the predicted opponent policies

$$\begin{aligned} \mathbf{a}^{(j)} &\leftarrow \pi^{(j)}_{\Pi, \text{br}}(\mathbf{s}^{(i)}, \mathbf{a}^{(i)}) \\ \mathcal{J}(\pi^{(j)}_{\Pi, \text{br}}) &= \mathbb{E}_{\mathbf{s}^{(i)}, \mathbf{a}^{(i)}} \left[ \nabla_{\mathbf{a}^{(-i)}} \mathbf{Q}_{\text{int}}^{(i)}(\mathbf{s}^{(i)}, \underline{\mathbf{a}}) \Big|_{\mathbf{a}^{(j)}} - \frac{1}{N_{\mathfrak{A}} - 2} \sum_{k=1, k \neq j, k \neq i}^{N_{\mathfrak{A}}} D_{\text{KL}}(\pi^{(j)} || \pi^{(k)}) \right]. \quad (6.30) \end{aligned}$$

### 6.6.3 Incorporating Model-Knowledge about Agent State-Dynamics

Eventually, the major advance of the presented hierarchical method lies in the ability in mixing model-based with model-free RL. Especially in multi-agent systems, the overall learning can be improved by incorporating knowledge about the individual agents, while learning a state-predictor oracle of the other agents from data. The existing hierarchy allows to apply safe state-space regions, in which the agents can explore for data-driven policies, cf. [Zhou et al. \(2021\)](#). Alternatively, the presented framework can embed model-based controllers on the lower level, and focus on proposing suitable task goals by the upper layer, which can be regressed from data without the necessity of respecting the current layer test-mode. A major benefit of this approach is then given by being able to directly impose constraints within the selected controller. On the other hand, solely relying on a single controller may lead to undesirable or overly restrictive behavior. Therefore, the investigation of obtaining suitable controllers by decomposing a set of controllers from data ([Sharma et al., 2020](#)), but also to improve their performance by advanced model predictors ([Saxena et al., 2021](#)) bares great potential to improve the overall learning behavior. Similarly, control or model-based priors may allow to further improve the performance rather than obtaining data from random exploration ([Rana et al., 2021](#)). Eventually, it is preferable to identify regions of the state-space – or observation-space – in which it is not only safe to apply RL, but also where it is actually needed. As a consequence, rather than replacing well-established methods such as model predictive control or iterative linear-quadratic regulator by data-driven policies, a classifier that maps the accuracy of the dedicated models bares a distinctly higher potential towards improving overall system performance. Regarding the hierarchical aspect, concepts such as reachability analysis ([Althoff, 2010](#)) can be used to induce hierarchical rewards on infeasible goal-states, but also to account for invalid sub-goals online. Thus eventually, the learning and intense data-collection is only of relevance in the areas where the assumed dynamic models differ from the experienced data.

#### 6.6.3.1 Applying HER Based on Current Agent Performance Across Hierarchies

In contrast to existing work HER cannot only be applied by evaluating the current episode or step reward, but also by directly evaluating the current advantage(s) at each step. A promising approach is given by directly applying the generalized advantage estimator (GAE), similar to PPO. In order to differentiate between team-average and agent-based rewards, we recommend to regress two value-estimates of the current state:

$$\begin{aligned} V_{\text{int}}(\mathbf{s}^{(i)}, \mathbf{a}^{(-i)}) &\approx \sum_{t=t}^{\infty} \frac{1}{N_{\mathfrak{A}}} \sum_{j=1}^{N_{\mathfrak{A}}} \mathbf{r}^{(j)}_{\mathbf{t}}, \\ V_{\text{nat}}(\mathbf{s}^{(i)}, \mathbf{a}^{(-i)}) &\approx \sum_{t=t}^{\infty} \mathbf{r}^{(i)}_{\mathbf{t}} \end{aligned}, \tag{6.31}$$

which represents the multi-agent baseline performance as also proposed in [Foerster et al. \(2018\)](#) and [Iqbal and Sha \(2019\)](#). As this value-estimate depends only on the current state, it can be evaluated throughout the hierarchies to estimate the advantage of a (hierarchical) action, using the (accumulated) GAE – cf. [Schulman et al. \(2015b\)](#). Recalling the the temporal

difference (TD)-residual of the single-agent GAE

$$\delta_{\text{TD}} := r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) + \gamma V_{\pi}(\mathbf{s}_{t+1}) - V_{\pi}(\mathbf{s}_t), \quad (6.32)$$

it is straightforward to see, that by simply adding the observed state-action values from the experience buffer, the agent can evaluate the advantage of each step w.r.t. the average team effort and personal reward. Using a weighted average of the step-advantages along a hierarchical step can then be used to directly account for environment-sensitive HER-rules:

- In case a lower layer step has a positive native advantage, map the next observation to the current sub-goal.
- In case an averaged update step has a positive native and interactive advantage, adjust the action of the upper layer such that the agent would have reached the dedicated sub-goal.

## 6.7 Conclusion

Within this chapter we have proposed a novel MARL-framework that allows for decentralized learning while also differentiating between agent-based native costs and joint task rewards. Even though our method relies on estimates of the agent-based costs, it outperforms recent state-of-the-art methods in terms of convergence speed within sparsely rewarded environments.

As the presented framework allows for a fully decentralized execution and learning, this chapter presented a hierarchical MARL-framework that allows each agent to regress the original task-critic without increasing the state-space, but simultaneously estimating the objective of reaching self-imposed sub-goals via a separate critic. Due to the decentralized execution and learning principle, the proposed MARL-framework is not affected by asynchronous decisions along the hierarchy. This allows each agent to specifically learn the task but also the hierarchical nature, which usually distinctly improves learning performance in sparsely rewarded environments.

Eventually, our method differs from existing MARL-methods by directly imposing structured observations and thus to impose different learning schemes for internal agent-states and external observations. This gives room for future research to impose minimal pre-knowledge, e.g., assuming knowledge about the individual dynamic models of the agent, while regressing the evolution of the environment from data. Given the amount of conceptual extensions presented in this chapter, we close this part of the thesis with a short outlook into future work.

## Future Work

In order to conclude this part of the thesis, we shortly present directions for future work. Starting from the possible extension that we have outlined within this chapter, the first line of research is given by extending the presented method to competitive domains, i.e., zero-sum games. Even though zero-sum games bare additional challenges on increased impact of individual agent-policies, we propose a conceptual extension of our br-policies to a team-based decision stage. In detail, the actions of the opponents are regressed by minimizing instead of

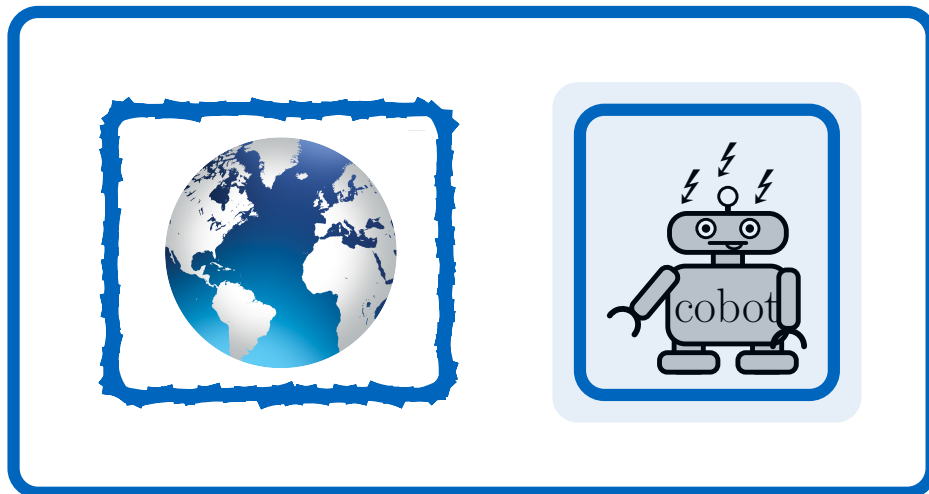
maximizing the agent task critic. This encourages the agents to take behaviors, where the vulnerability towards opponent policies is minimized.

Another line of research for future projects is the analysis of stabilizing the convergence behavior by sharing the predicted br-policies among the individual agent-models. Nonetheless, the most promising line of research of the presented approach lies in the extension of the hierarchical MARL-framework by explicitly incorporating available model knowledge and diminish the idea of pure model-free RL. In fact, the hierarchical structure of the proposed learning-framework allows to directly impose available model-knowledge of the individual agents and apply low-level controllers or primitive behavioral models. As a result, a suitable control-policy can then either be found by proposing suitable goal-states to the lower level, or by composing suitable control-policies by a set of controller candidates. As this framework would still allow to embed a model-free policy as an alternative controller, the presented method allows to bridge findings from control-theory to model-free RL.



## Part III

# Advanced Manipulation Tasks with Unknown Objects







# 7

## Haptic Object Identification for Advanced Manipulation Skills

### Chapter Abstract

This chapter focuses on the aspect of coping with unknown objects, namely with the identification of the shape and material properties of objects in the environment of a robot manipulator. Motivated from the concept how humans improve their visual prior information by further exploiting their sensory and motoric abilities, the research field of haptic perception evolves.

While recent research has focused on estimating either the geometry or material properties, this chapter strives to combine these aspects by outlining a probabilistic framework that efficiently refines initial knowledge from visual sensors by generating a belief state over the object shape while simultaneously regressing the material parameters. Specifically, we present a grid-based and a shape-based exploration strategy, that both apply the concepts of Bayesian-Filter theory in order to decrease the uncertainty by optimizing the expected information-gain at each step. Furthermore, the presented framework is able to learn about the geometry as well as to distinguish areas of different material types by applying unsupervised machine learning methods, namely density-based-spatial-clustering for applications with noise to cluster individual materials.

We evaluate the presented haptic exploration framework within a simulated environment using a simplified robot, that allows to collect haptic feedback via a force-torque-sensor. The collected data highlights the potential of the presented methods towards enabling robots to autonomously explore unknown objects, yielding information about shape and structure of the underlying object and thus, opening doors to robotic applications where environmental knowledge is limited.

*Remark:* A majority of this chapter was previously published in [Gabler et al. \(2020b\)](#) and builds upon internal project work ([Maier, 2019](#)).

## 7.1 Introduction

From research projects within well-defined lab environments to future applications in everyday life and industry, future robot applications are favored to handle arbitrary objects without requiring a perfect model of the environment. This skill is a key-requirement in order to achieve long-term autonomy. As a result, object identification and knowledge acquisition are crucial – yet important – assets for robots. While a rough estimation of the shape of an object can be obtained from visual data, the exact material decomposition remains in general unknown. Nonetheless, these material properties have a distinct effect on the selection of the subsequent manipulation tasks, e.g., the choice of material-dependent cutting tools. In order to allow robots to autonomously identify these object characteristics, one approach is to mimic human behavior in applying haptic data acquisition methods, i.e., actively interacting with the unknown object. This approach, known as tactile and haptic exploration, enables robots to significantly increase and extend the results of visual object identification methods. In contrast to recent research in haptics, this chapter presents an online inference algorithm which is capable of acquiring information not only about the geometry but also about the material parameters of an unknown object.

The remainder of this chapter is structured as follows: the next section outlines the mathematical problem tackled in this chapter, followed by an outline on how this chapter is positioned compared to related work in Section 7.3. The concepts of the proposed grid-based and the shape-based exploration strategies are sketched in Section 7.4. The idea of classifying individual components by their material types is shown in Section 7.5, whereas Section 7.6 outlines the evaluation of the proposed methods in a simulated environment. The summary in Section 7.7 concludes this chapter.

## 7.2 Problem Formulation

The task of haptic object identification consists of two main challenges. First, the geometric shape of an object, denoted as  $\mathbf{S}$  in the context of this chapter, is in general unknown. Second, the object is characterized by an undefined parameterization  $\boldsymbol{\xi}$ , that describes the material properties of an object, e.g., a material classifier that maps each component of  $\mathbf{S}$  to a finite set of materials. Given the state, control inputs and measurements of a robot as  $\vec{\mathbf{x}}_{0:t} := (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t)$ ,  $\vec{\mathbf{u}}_{0:t} := (\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_t)$  and  $\vec{\mathbf{z}}_{0:t} := (\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_t)$  from time step 0 to the current time step  $t$ , the problem is given by finding proper mapping functions

$$\mathbf{S} \leftarrow \mathcal{F}_{\mathbf{S}}(\vec{\mathbf{x}}_{0:t}, \vec{\mathbf{u}}_{0:t}, \vec{\mathbf{z}}_{0:t}, \boldsymbol{\xi}), \quad (7.1)$$

$$\boldsymbol{\xi} \leftarrow \mathcal{F}_{\text{prm}}(\vec{\mathbf{x}}_{0:t}, \vec{\mathbf{u}}_{0:t}, \vec{\mathbf{z}}_{0:t}, \mathbf{S}). \quad (7.2)$$

Finding proper mappings  $\mathcal{F}_{\mathbf{S}}$  and  $\mathcal{F}_{\text{prm}}$  is non-trivial as they are in general dependent on each other. Nonetheless, when analyzing the problem individually, one can relax these problems and focus on finding these mappings for a fixed  $\boldsymbol{\xi}$  or  $\mathbf{S}$ . As a variety of promising methods on solving these problems individually exists in literature, we continue with an overview of related work.

## 7.3 Related Work

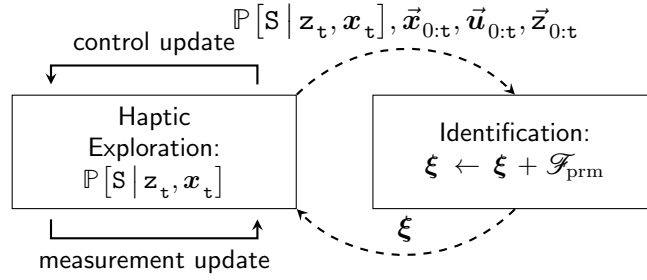
Although vision has been established as the backbone of robotic perception, haptic information acquisition has been used to understand or recognize shapes of objects for years (Allen and Roberts, 1989). Navarro et al. (2012) present an approach for haptic object recognition based on extracting key features of tactile and kinesthetic data using a clustering algorithm, where a tactile sensor performs haptic sensation tasks using different robotic hands. Behbahani et al. (2016, 2015) have introduced haptic simultaneous localization and mapping (SLAM) into the field of haptic exploration, which is inspired by visual SLAM techniques (Durrant-Whyte and Bailey, 2006) and occupancy grid methods (Elfes, 1989). Through adaption of the FastSLAM (Montemerlo et al., 2002) algorithm, a novel method is proposed to iteratively learn the shape of the surface of objects. The same approach is used to mimic haptic perceptual algorithms from neuroscience in (Behbahani, 2016). Another method using haptic SLAM is presented by Schaeffer and Okamura (2003), although their algorithms require knowledge about the underlying object shape in advance. Further on, there are methods to detect objects and especially edges of geometries through clever exploration strategies. Pezzementi et al. (2011) extract features based on data from a tactile sensor array using methods inspired by computer vision techniques. This concept is extended in Martinez-Hernandez et al. (2013) by actively following contours based on tactile sensor data. Nonetheless, this approach heavily relies on distinct edges and sharp angles in the contour of the object. Another approach is represented in Hegazy and Denzler (2009), where range data and 2D images are combined to a generic object recognition algorithm. These techniques succeed in solving the geometric shape estimation task, but fail in providing any further information about the underlying material decomposition.

The problem of finding a dedicated choice of control actions that can help to maximize the accuracy of the available information is tackled by Bourgault et al. (2002), who use an information-theoretic approach to select actions with a high information gain. Similarly Julian et al. (2012) use sequential Bayesian filters to increase the information gain for a joint state exploration task with multiple robotic agents.

Regarding the aspect of identifying material parameters based on haptic cues promising results have been found in literature. Luo et al. (2017) provide a detailed review of tactile perception using surface and texture-based information to find material properties and material types. Friedl et al. (2016) identify textures using recurrent spiking neural networks. Decherchi et al. (2011) classify material types using methods from computational intelligence from contact forces. Xu et al. (2013) propose a classification algorithm based on texture and propose a Bayesian exploration algorithm which seeks to minimize the uncertainty in the underlying belief (Fishel and Loeb, 2012). These methods allow to distinguish between different material properties, but fail to simultaneously refine shape estimation and material classification.

### Contribution

In contrast to the state-of-the-art, this chapter outlines a haptic object identification framework that allows to simultaneously refine the shape estimation and regress the underlying material parameters as visualized in Figure 7.1. In order to obtain  $\mathcal{F}_S$ , we incorporate findings from haptic SLAM (Behbahani et al., 2015). This encourages to maximize the information gain at every step by applying concepts of Bayesian Filter theory (Thrun et al., 2005) in an



**Figure 7.1:** Proposed framework components. The haptic exploration iteratively decreases model uncertainty, while the identification allows to batch-process a collection of data measurements in order to refine the object parameters  $\xi$ .

iterative cycle of control and measurement updates. In order to decouple the simultaneous parameter estimation problem,  $\xi$  is assumed to be fixed for  $N_{\text{eps}}$  steps and only updated once collective data batches of update steps have been obtained. In contrast to the cyclic nature of Bayesian Filters, this module has access to a collection of data measures and can thus run nonlinear regression techniques to regress the material parameters  $\xi$ . In the remainder of this chapter,  $\xi$  describes the boundaries of a classifier that maps individual components of an object to a set of available material types. Given this,  $\mathcal{F}_{\text{prm}}$  is realized by applying unsupervised clustering and model-fitting techniques.<sup>1</sup>

## 7.4 Haptic Exploration

Before being able to extract information about the objects in the workspace, the robot has to collect data through exploration. In order to gather this sensor data in an efficient manner, we design a control flow for exploring our environment based on a grid-based and a shape-based representation. Given initial data from e.g., computer vision, an initial belief can be obtained, that can be iteratively updated.

### 7.4.1 Grid-Based Exploration

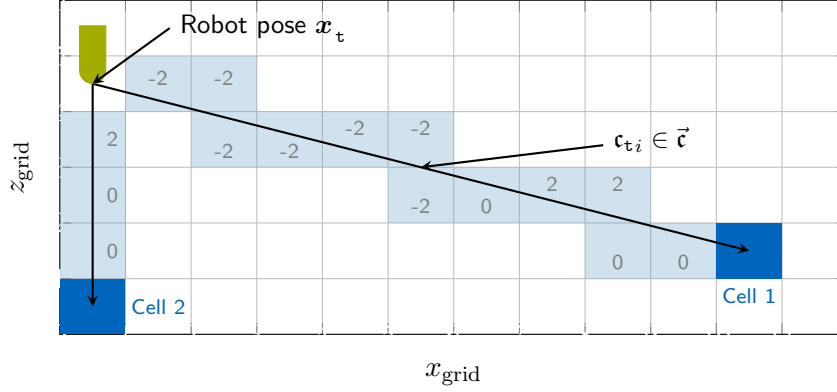
We incorporate the findings from Behbahani et al. (2015), where the belief of the geometry is stored in an occupancy grid consisting of individual cells  $\mathbf{c} \in \mathcal{S}$ . We extend this to

$$\mathcal{G}^{\mathcal{S}} = (\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_{N_{\text{mtrl}}}),$$

as an inference grid consisting of binary classifier layers  $\mathbf{S}_{t_i}$  for  $N_{\text{mtrl}}$  material types, and an occupancy grid for  $i = 0$ , where each grid  $\mathbf{S}_{t_i}$  assigns a class-membership probability to each cell  $\mathbf{c}$ . However, in contrast to storing actual probability values in the grid as in Elfes (1989), we use the log-odds-notation

$$\mathbb{P}[\mathbf{c} | \mathbf{z}, \mathbf{x}] \propto \frac{\exp \mathbf{S}_{t_i}(\mathbf{c})}{1 + \exp \mathbf{S}_{t_i}(\mathbf{c})}$$

<sup>1</sup>The framework outlined in this chapter is not restricted to the presented identification method. Nonetheless, this specific example serves as a proof of concept.



**Figure 7.2:** Basis for utility and accessibility calculations in a 2D grid for two different goal cells, shown in blue. The black arrows show the direct connection from  $\hat{\mathbf{x}}$  to the goal cells, while the light blue cells indicate the result of the line discretization and the content of the set  $(\mathbf{c}_{t_1}, \mathbf{c}_{t_2}, \dots, \mathbf{c}_{t_{N_{\text{cell}}}})$ . The gray numbers in the cells show the respective belief stored in one layer of the inference grid, e.g., the occupancy layer  $\mathbf{S}_{t_0}$ .

from Behbahani (2016) to store the current belief of each cell and layer. With all layers being binary classifiers both measurements and states can only take values in  $\{0, 1\}$ . As the haptic exploration seeks to maximize the expected information gain, a utility metric needs to be defined that encourages to maximize information gain upon choosing the next cell to explore. We incorporate findings from Julian et al. (2012), that map the prior belief of a cell to all possible measurements. Predicting the posterior given the current belief and models, the utility of a cell  $\mathbf{c}$  results in

$$\mathcal{U}_{\mathbf{c}}(\mathbf{c}_i) = \frac{1}{N_{\text{mtrl}}} \sum_{i=1}^{N_{\text{mtrl}}} \sum_{\mathbf{z} \in \{0,1\}} \sum_{\mathbf{s}_{\mathbf{c}} \in \mathcal{S}} \mathbb{P}[\mathbf{z}^+ = \mathbf{z} | \mathbf{s}] \mathbb{P}[\mathbf{s}_{\mathbf{c}}] \ln \left( \frac{\mathbb{P}[\mathbf{s}_{\mathbf{c}} | \mathbf{z}^+ = \mathbf{z}]}{\mathbb{P}[\mathbf{s}_{\mathbf{c}}]} \right), \quad (7.3)$$

where  $\mathbf{z}$  stands for the possible results of the measurement, which are again given as binary mapping  $\{0, 1\}$  for all layers for a given cell in the inference grid, and  $\mathbf{s}_{\mathbf{c}}$  iterates over the possible state of cell  $\mathbf{c}$  in the dedicated layer of the inference grid. In order to evaluate the uncertainty over all classes, we finally average over all classes  $N_{\text{mtrl}}$  and obtain a utility score for each cell in the inference grid. Given the prior belief  $\mathbb{P}[\mathbf{s}_{\mathbf{c}}]$  and the sensor model  $\mathbb{P}[\mathbf{z} | \mathbf{s}_{\mathbf{c}}]$ , the probability  $\mathbb{P}[\mathbf{s}_{\mathbf{c}} | \mathbf{z}]$  can be directly inferred by Bayes' law. However, there are cells with a high utility which may be unreachable for the robot in realistic scenarios, e.g., the inside of rigid bodies or geometries with cavities. Hence, we introduce an accessibility metric evaluating how well cell  $\mathbf{c}$  is accessible from the current pose of the robot  $\mathbf{x}_t$ , given a cell-trajectory as visualized in Figure 7.2. Denoting the cell-trajectory as  $\vec{\mathbf{c}} := (\mathbf{c}_{t_1}, \mathbf{c}_{t_2}, \dots, \mathbf{c}_{t_{N_{\text{cell}}}})$  of length  $N_{\text{cell}}$ , we exploit the log-odds-notation by accumulation of signs of the occupancy layer for each cell:

$$\mathcal{U}_{\text{reach}}(\mathbf{c}_i) = \begin{cases} \frac{1}{N_{\text{cell}}} & \text{if } \mathbf{S}_{\mathbf{c}_i}^0 = 0 \forall \mathbf{c}_i \in \vec{\mathbf{c}} \\ \left\| \frac{1}{N_{\text{cell}}} \sum_{\mathbf{c}_i \in \vec{\mathbf{c}}} -\text{sign}(\mathbf{S}_{\mathbf{c}_i}^0) \right\|_2 & \text{otherwise} \end{cases}, \quad (7.4)$$

where the upper case simply avoids a reachability of 0 for all cells, if the occupancy grid is empty for all cells evaluated. Hence, the final rank and thus the criteria for selecting the next

exploration cell is obtained as

$$\mathbf{c}^* \leftarrow \arg \max_{\mathbf{c} \in (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{N_{\text{mtrl}}})} \mathcal{U}_{\text{reach}}(\mathbf{c}) \mathcal{U}_{\mathbf{c}}(\mathbf{c}). \quad (7.5)$$

A single exploration step is thus given by selecting cell  $\mathbf{c}^*$ , approaching this cell, starting a measurement by e.g., applying a force upon the object and updating the inference grid based on the new measurement.

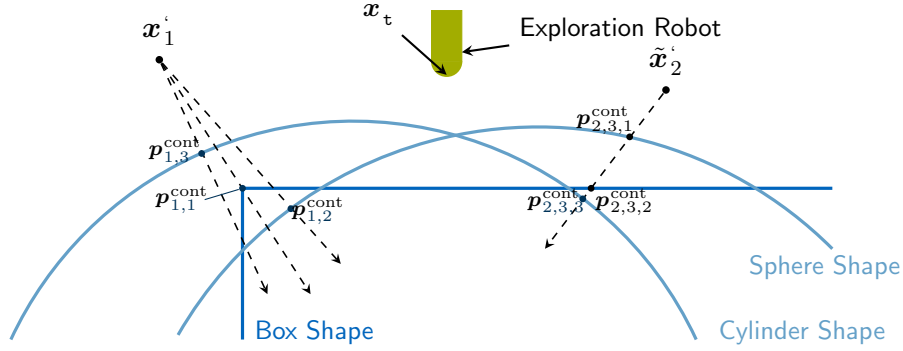
#### 7.4.2 Shape-Based Exploration

A major drawback of the grid-based framework is that the resolution of the grid inevitably leads to uncertainties due to the discretization of the grid. Therefore, we propose a second exploration approach which uses analytic shape representations of geometric primitives, namely spheres, cylinders, boxes and planes. Provided initial data points from e.g., computer vision, we fit initial models that generate a hypothesis for each model shape. As each model-fit is conditionally independent from the others, each hypothesis forms a shape-particle  $\mathfrak{R}^{\mathbf{S}_i}$ . All shape-particles denoted as  $\overline{\mathfrak{R}^{\mathbf{S}}}$  form a Particle-Filter, where each particle is associated with a belief, initialized by a uniform distribution over all particles.

In order to explore the environment efficiently, a utility metric is needed that minimizes the uncertainty at each step, i.e., to distinguish between various shape candidates. This selection boils down to finding the optimal contact point from a set of candidates  $\overline{\mathbf{p}^{\text{cnd}}} := \{\mathbf{p}^{\text{cnd}}_1, \mathbf{p}^{\text{cnd}}_2, \dots, \mathbf{p}^{\text{cnd}}_{N_{\text{cnd}}}\}$  representing possible contact points on a surface of a shape-particle. Similar to (7.3), the utility metric is based on Julian et al. (2012) and the concept of mutual information (Elfes, 1995). We further assume a multivariate normal-distributed sensor-model with covariance  $\Sigma$ , such that the utility results in

$$\mathcal{U}_{\mathbf{S}}(\overline{\mathbf{p}^{\text{cont}}}, \overline{\mathfrak{R}^{\mathbf{S}}}) = \sum_{\mathfrak{R}^{\mathbf{S}_i} \in \overline{\mathfrak{R}^{\mathbf{S}}}} \sum_{\mathbf{p}_i \in \overline{\mathbf{p}^{\text{cnd}}}} \mathcal{N}(\mathbf{p}_i | \hat{\mathbf{p}}_i, \Sigma) \mathbb{P}[\mathfrak{R}^{\mathbf{S}_i} | \mathbf{x}_t, \mathbf{z}_t] \ln \left( \frac{\mathcal{N}_{\mathbf{p}_i}(\hat{\mathbf{p}}_i, \Sigma)}{\mathbb{P}[\mathfrak{R}^{\mathbf{S}_i} | \mathbf{x}_t, \mathbf{z}_t]} \right), \quad (7.6)$$

where  $\mathbb{P}[\mathfrak{R}^{\mathbf{S}_i} | \mathbf{x}_t, \mathbf{z}_t]$  is the prior belief of shape  $S^j$  within the current particle filter, and  $\hat{\mathbf{p}}_i$  denotes the expected contact point for shape  $S^j$  on the particular axis. This utility combines the knowledge of the prior belief with the influence of expected measurements, and therefore allows to estimate the expected impact of these measurements. The utility only depends on prior belief of the shapes and the set of possible contact points  $\overline{\mathbf{p}^{\text{cnd}}}$ . The selection of an optimal contact point forms the initiation of a single exploration step of the shape-based strategy and is visualized in Figure 7.3 with three shape particles. In order to obtain contact points and simultaneously explore the workspace, a set of intermediate positions  $\mathbf{x}'$  are sampled in the near vicinity of the robot  $\mathbf{x}_t$ . Each of these points is evaluated in parallel by drawing a line to the closest point of each shape. Given these lines, the intersection points of the remaining shapes and the connection lines as well as the closest point define the set  $\overline{\mathbf{p}^{\text{cont}}}$ , e.g.,  $\{\mathbf{p}_{2,3,1}^{\text{cont}}, \mathbf{p}_{2,3,2}^{\text{cont}}, \mathbf{p}_{2,3,3}^{\text{cont}}\}$  in Figure 7.3, from which the utility of testing the selected shape hypothesis, given the sampled starting position, can be obtained. The algorithm then chooses the intermediate starting position that returns the optimal expected utility. In contrast to the grid-based strategy, not only a fixed goal point is chosen, but instead all possible contact points along the selected line are sequentially checked until a measurement can be obtained



**Figure 7.3:** Choice of next action with 3 shapes shown in 2D. The points  $\mathbf{x}_1^i$  and  $\mathbf{x}_2^i$  are randomly sampled, each  $\mathbf{p}_{i,j}^{\text{cont}}$  depicts the possible contact points. On  $\mathbf{x}_1^i$ , the closest contact points  $\mathbf{p}_{1,j}^{\text{cont}}$  with each shape are displayed through the dashed lines. For  $\mathbf{x}_2^i$  on the right, the contact  $\mathbf{p}_{2,3}^{\text{cont}}$  with the cylinder shape is shown exemplarily with the respective  $\mathbf{p}_{2,3,1}^{\text{cont}}$ . The points  $\mathbf{p}_{2,3,1}^{\text{cont}}$ ,  $\mathbf{p}_{2,3,2}^{\text{cont}}$  and  $\mathbf{p}_{2,3,3}^{\text{cont}}$  are used to calculate the utility for the exploration axis  $\mathbf{x}_2^i \rightarrow \mathbf{p}_{2,3,3}^{\text{cont}}$ .

or a constraint is violated. The exploration step ends with updating the particle beliefs that have been tested with a predefined update weight.

## 7.5 Object Identification

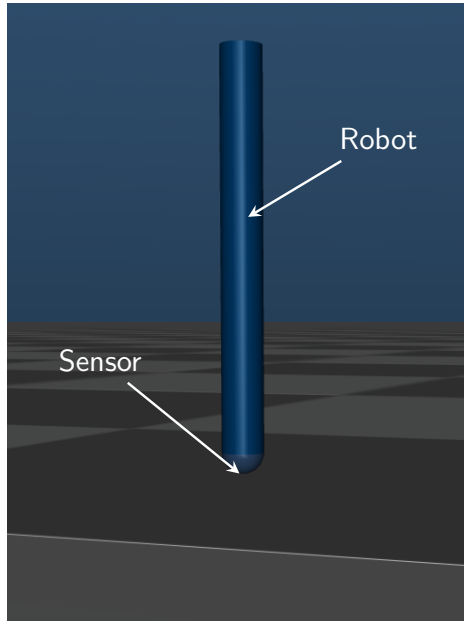
The object identification task is given by applying unsupervised machine-learning methods to generate object classification thresholds or to fit the dedicated model parameters given the collected data measurements.

Regarding the grid-based strategy, clustering algorithms such as K-Means and density-based-spatial-clustering for applications with noise (DBSCAN) are suitable methods as the structure of the inference grid is also bound to a finite number of material types. Thus, the identification process can be used to update the decision boundaries for each inference layer. Furthermore, the belief of the inference grid layers can be corrected using the collected measurement- and state history.

The shape-based strategy is not solely limited to clustering the collected data but further requires to reject and resample new particles to the filter. For this purpose, the estimation error

$$\varepsilon_i = \frac{1}{T_{\max}} \sum_{t=0}^{T_{\max}} \|\mathbf{z}_t - \mathbb{E}[\mathbf{z}_t | \mathfrak{R}_i^s, \mathbf{x}_t, \mathbf{u}_t]\|_2 \quad (7.7)$$

for each particle  $\mathfrak{R}_i^s$  is obtained in order to determine which particles have a great discrepancy between measured values  $\mathbf{z}_t$  and the corresponding expected values. Given the recorded data, the least performant particles are removed from the filter. After deleting the inaccurate particles, new shape parameterizations are sampled. In order to obtain proper samples, it is favorable to partition the provided sensor data. Again unsupervised clustering algorithms are a suitable choice here because no further knowledge over the properties of the underlying data is required and the number of current geometric primitives is finite. This results in a deterministic classification, meaning that each measurement is assigned with a class label. Having obtained these individual components, additional model-fits per cluster result



**Figure 7.4:** Simple 6-degree of freedoms robot with touch sensor attached at end-effector.

in new geometric primitive samples. By finally combining these geometric primitives into a composition object a new particle is added to the filter.

## 7.6 Evaluation

The outlined algorithm is evaluated with an artificial robot explorer using a simulated environment, namely the physics-engine multi-joint dynamics with contact (MuJoCo) (Todorov et al., 2012). The robot is equipped with a force-torque sensor that allows to measure the impact during collision. In order to focus on the exploration process, we directly explore and control in Cartesian space. Thus, the pose of the robot  $\mathbf{x}$  is controlled via a Cartesian impedance controller:

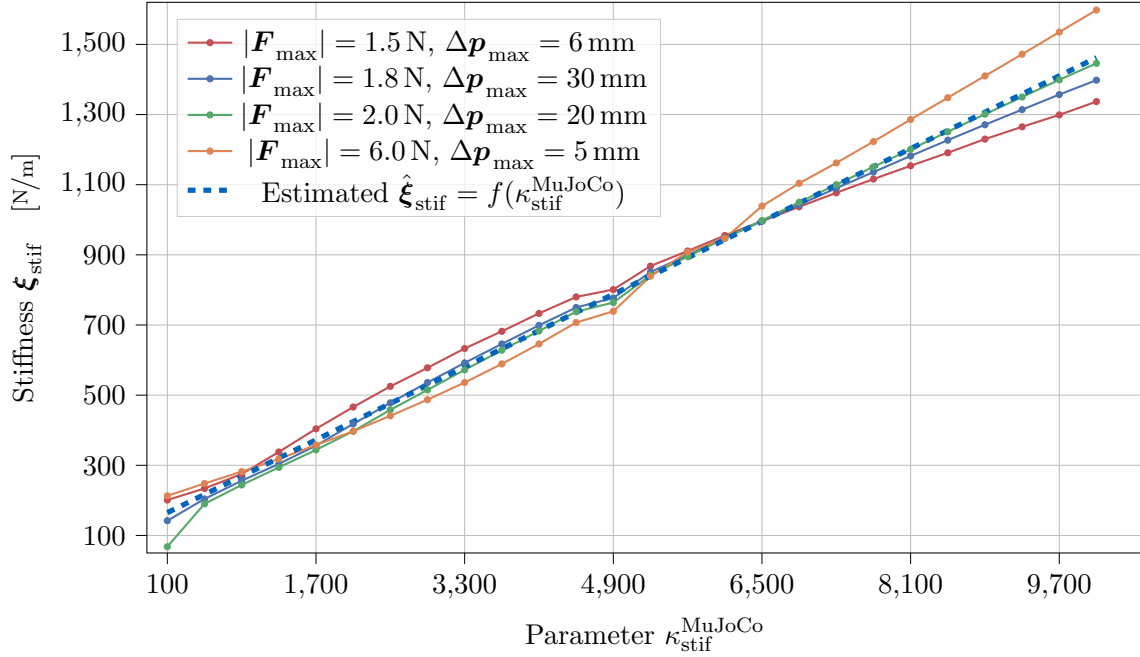
$$\mathbf{u}^F = \mathbf{K}_{\text{stif}}(\mathbf{x}_{\text{des}} - \mathbf{x}) - \mathbf{K}_{\text{dmp}}\dot{\mathbf{x}} + \mathbf{F}_{\text{ext}}, \quad (7.8)$$

where  $\mathbf{u}^F$  is the applied wrench command,  $\mathbf{x}_{\text{des}}$  is the desired pose,  $\dot{\mathbf{x}}$  the velocity,  $\mathbf{K}_{\text{stif}}$  and  $\mathbf{K}_{\text{dmp}}$  describe the stiffness and damping matrices, and  $\mathbf{F}_{\text{ext}}$  describes an additional feed-forward wrench command.

In order to test our methods against the challenges stated in Section 7.2, the robot is faced with a set of unknown objects, which are composed of sub-components of different materials, which differ in their stiffness values. MuJoCo (Todorov et al., 2012) handles all contacts between objects as soft constraints in the dynamic system, which can be seen as a spring-damper system, where one can set the stiffness  $\kappa_{\text{stif}}^{\text{MuJoCo}}$  and damping  $\kappa_{\text{dmp}}^{\text{MuJoCo}}$ . These stiffness-damping values are artificial contact values used for simulation dynamics rather than physically realistic values <sup>2</sup>, such as Young’s modulus, that a robot can regress by obtaining measurements

<sup>2</sup>We refer to Todorov et al. (2012) for detailed information.





**Figure 7.5:** Estimation of relationship  $\xi_{\text{stif}} = f(\kappa_{\text{stif}}^{\text{MuJoCo}})$  between MuJoCo parameter  $\kappa_{\text{stif}}^{\text{MuJoCo}}$  and resulting stiffness  $\xi_{\text{stif}}$  based on multiple measurement series.

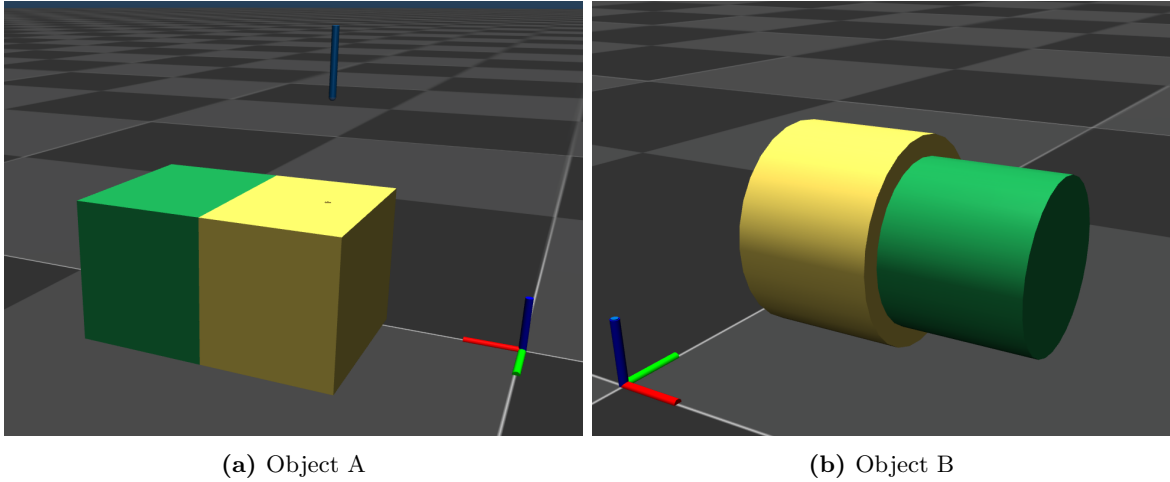
$\mathbf{z}_t = (\mathbf{x}_t, \mathbf{F}_t, \Delta \mathbf{p})$ , i.e., the magnitudes of force and displacement during contact at  $\mathbf{x}_t$ . In order to assess the relationship between stiffness parameter  $\kappa_{\text{stif}}^{\text{MuJoCo}}$  of a MuJoCo object model and the physical stiffness value  $\xi_{\text{stif}} = \frac{\mathbf{F}}{\Delta \mathbf{p}}$ , we fixed the damping values to  $\kappa_{\text{dmp}}^{\text{MuJoCo}} = 1$  for all simulations and performed several experiments with increasing parameter  $\kappa_{\text{stif}}^{\text{MuJoCo}}$  and compared the resulting estimations with the numeric stiffness value  $\xi_{\text{stif}} = \frac{\mathbf{F}_{\text{max}}}{\Delta \mathbf{p}_{\text{max}}}$ . Applying linear regression, the material stiffness can be approximated as

$$\hat{\xi}_{\text{stif}} \leftarrow \kappa_{\text{stif},1}^{\text{MuJoCo}} \kappa_{\text{stif}}^{\text{MuJoCo}} + \kappa_{\text{stif},2} \approx (0.13 \kappa_{\text{stif}}^{\text{MuJoCo}} + 152) \text{ N/m}, \quad (7.9)$$

as visualized in Figure 7.5. Even though the data is just an approximation of the actual material property, it is sufficient to evaluate the capability of our method to differentiate between materials and thus to identify the decomposition of an object.

We evaluate the algorithm on two artificial objects as visualized in Figure 7.6. The first object – denoted as object A in the following – describes a composition of two box-shaped components, where each component has a distinct material type, i.e., constant stiffness value. The soft component – visualized in yellow – has a stiffness of  $\xi_{\text{stif}}^{\text{MuJoCo}} = 100$ , which represents an actual stiffness of  $\xi_{\text{stif}} = 165 \text{ N/m}$ . The second component – visualized in green – has a stiffness value of  $\xi_{\text{stif}}^{\text{MuJoCo}} = 10000$  or  $\xi_{\text{stif}} = 1452 \text{ N/m}$ . In a similar manner, the second object – denoted as object B in the following – consists of two cylinder-shaped objects, where the soft (yellow) component is defined as  $\xi_{\text{stif}}^{\text{MuJoCo}} = 1000$ , i.e.,  $\xi_{\text{stif}} = 282 \text{ N/m}$ , while the stiff component (green) is set to  $\xi_{\text{stif}}^{\text{MuJoCo}} = 8000$ , i.e.,  $\xi_{\text{stif}} = 1192 \text{ N/m}$ .

This results in a fixed number of material types  $N_{\text{mtrl}} = 2$  for both scenarios. Further, the interaction force and contact displacement are limited to  $\mathbf{F}_{\text{max}} = 3 \text{ N}$  and  $\Delta \mathbf{p}_{\text{max}} = 8 \text{ mm}$  for object A and  $\mathbf{F}_{\text{max}} = 2.4 \text{ N}$  and  $\Delta \mathbf{p}_{\text{max}} = 7 \text{ mm}$  for object B.

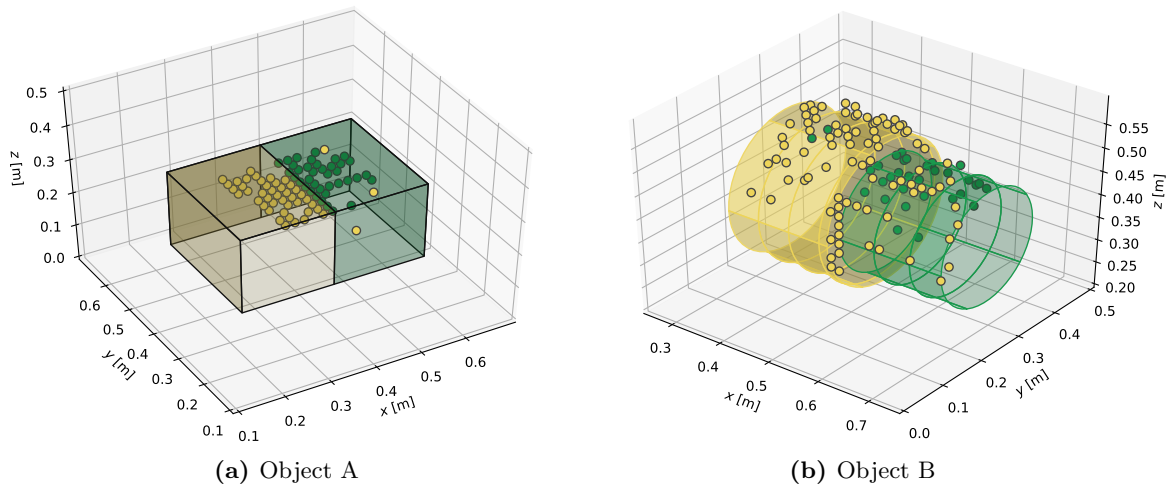


**Figure 7.6:** Evaluation objects within MuJoCo. The box-shaped object on the left hand side is denoted as object A, while the cylinder-shaped object is denoted as object B. In order to highlight the experimental procedure, the explorer from Figure 7.4 is shown in Figure 7.6a. The exploration is run in an episodic manner, where the explorer collects  $N_{\text{step}} = 10$  samples per *episode*. After each episode the object identification according to Figure 7.1 and Section 7.5 is updated.

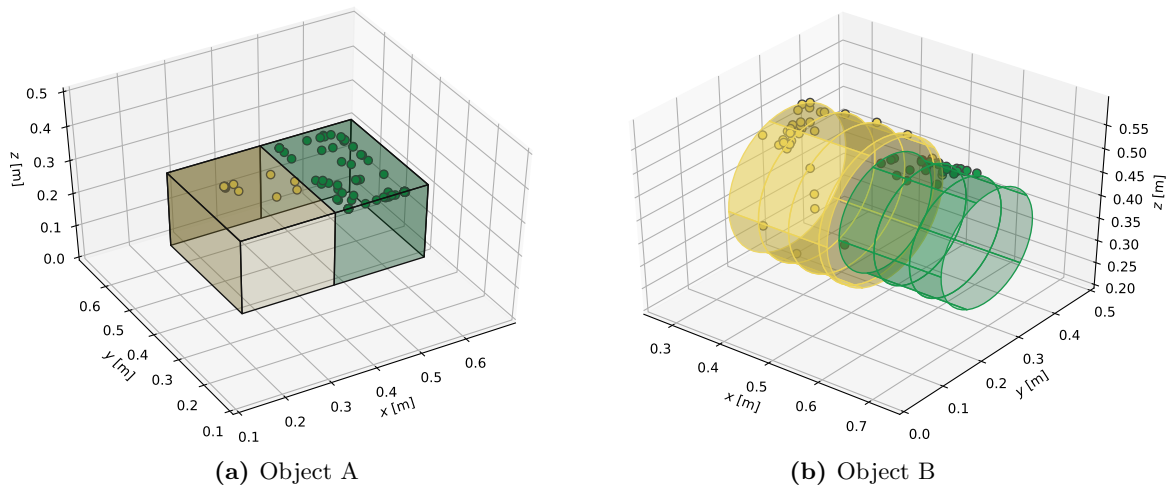
Recalling the iterative procedure from Figure 7.1, the exploration is run  $N_{\text{step}} = 10$  steps for the grid-based approach and  $N_{\text{step}} = 16$  for the shape-based approach. This implies that  $N_{\text{step}}$  measurement samples are obtained before an update for the material type classification is run.

### 7.6.1 Grid-Based Exploration

We employ a grid of 25 cells along  $e_x$ ,  $e_y$  and  $e_z$  using a resolution of 2 cm for each cell. The grid-based algorithm is provided an initial surface estimation, that assigns initial values to  $\mathfrak{S}_0^s$ . However, the initial data only provides a belief for the first layer regarding the occupancy of the grid, so all remaining layers are initialized without any prior knowledge. The clustering results are shown in Figure 7.7 after  $N_{\text{eps}} = 12$  for object A and  $N_{\text{eps}} = 20$  for object B. The different number of episodes emphasize the applied utility-metric from (7.3) that encourages exploration with high-information gain w.r.t. current material-type parameters, i.e., taking samples close to the class borders first. As it can be seen for higher exploration runs of object B the exploration is scattered across the accessible surface of the body compared to the early samples for object A. The resulting material association for the data collected for object A reaches an F1-score of 0.966 for yellow and 0.962 for green and is thus clustered into two clearly distinguishable classes. Object B reaches a F1-score of only 0.834 for yellow and 0.667 for green, thus fails to assign samples to the correct material type. A major reason for these false classifications lies in distorted measurements, which are likely to occur when the contact angle between robot and surface is very small such that the applied force is nearly parallel to the object surface. A major downside of the current grid-based approach is not having access to the normal vector of the underlying geometry, thus approaching an object at an inapt angle is more likely. Especially if the object is specifically curved, such as the cylinders in object B.



**Figure 7.7:** Classification of data measurements for the grid-based approach after  $N_{\text{eps}} = 12$  for object A and  $N_{\text{eps}} = 20$  for object B, with  $N_{\text{step}} = 10$  each.

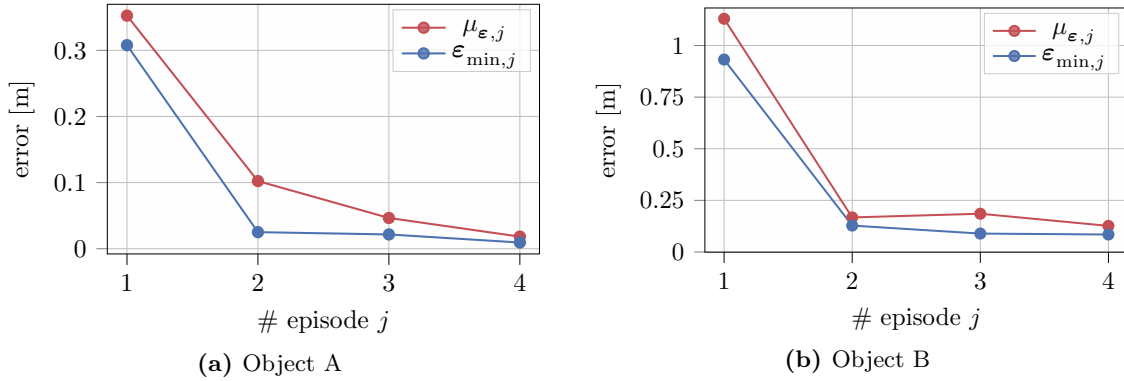


**Figure 7.8:** Classification of data measurements for the shape-based approach after  $N_{\text{eps}} = 4$  episodes with  $N_{\text{step}} = 16$  steps each.

### 7.6.2 Shape-Based Identification

In contrast to the grid-based approach, the object identification requires more exploration steps before a clustering process can be initiated, as a model-fit such as the random sample consensus algorithm requires at least 16 data points to fit the plane model with  $\xi \in \mathbb{R}^{15}$  to provide proper data fittings. An initial particle set is drawn from 10 data points. The results for object A and B are shown in Figure 7.8 after  $N_{\text{eps}} = 4$  episodes. While the F1-score for object A is slightly less for yellow (0.952) compared to the grid-based approach, it is significantly larger for green (0.987) and for object B for both yellow (0.963) and green (0.987).

In contrast to the grid-based approach, which has been shown to be efficient for spatial object refinement in previous work, the aspect of geometric shape refinement process is further evaluated. Thus, we evaluate the evolution of the mean error  $\mu_{e,j}$  over all particles and the



**Figure 7.9:** Evolution of the estimation error of the particle filter over episodes  $j$  and their corresponding values for object A on the left and object B on the right.

minimum error  $\epsilon_{\min,j}$  of all particle errors according to (7.7) as shown in Figure 7.9. The fact that the gap between the average estimation error and the best estimate is reducing over time highlights the effect of rejecting false hypothesis candidates at every iteration. Nonetheless, a slight increase is noticeable from episode 2 to episode 3 for object B, that shows the possibility of drawing false candidates in the resampling process. Overall, the evaluation of the error shows the potential of the shape-based algorithm on iteratively improving the shape estimation.

### 7.6.3 Material Parameter Estimation Accuracy

While both methods successfully classify the objects in three out of four test-cases, we close with evaluating the ability of directly estimating the material parameter, i.e., stiffness  $\xi_{\text{stif}}^*$  in here. The results, obtained as the centers for each cluster, are summarized in Table 7.1 for both objects and approaches. It has to be noted that the data was evaluated as collected without any outlier removal, which has a distinct impact on the stiffness estimation using the grid-based method on object B.

For all components, the stiffness-estimation underestimates the actual values. Again, the small forces upon small contact angles deteriorate these measurements. Besides this, empirical tests have shown that the simulation yields inconsistent measurements if a contact is measured directly on the edge of a geometry. As measurements at the edges are encouraged by the

Object	Class	$\xi_{\text{stif}}^{\text{MuJoCo}}$	$\xi_{\text{stif}}^*$ [N/m]	$\hat{\xi}_{\text{stif}}^{\text{G}^s}$ [N/m]	$\hat{\xi}_{\text{stif}}^{\text{R}^s}$ [N/m]
A	Yellow	100	165	107.81	113.85
A	Green	10000	1452	615.11	717.02
B	Yellow	1000	282	164.75	104.54
B	Green	8000	1192	397.34	947.15

**Table 7.1:** Estimated stiffness for both methods and objects. The last columns show the estimated stiffness values  $\hat{\xi}_{\text{stif}}^{\text{G}^s}$  for the grid-based method and  $\hat{\xi}_{\text{stif}}^{\text{R}^s}$  for the shape-based approach.

utility metrics in (7.3) and (7.6), we leave the investigations of this effect on future work, that will include hardware applications.

## 7.7 Conclusion

This chapter presented two main methods that improve the capability of robots on identifying and understanding unknown objects via haptic data acquisition. The first method extends findings in the field of haptic SLAM by extending the basic method based on occupancy grids to inference grids, that further allow to estimate the material type of the individual components. The second method exploits the concept of particle filter and the assumption that arbitrary objects can be represented as a composition of geometric primitives, by iteratively rejecting and resampling new geometric primitive-decompositions as particles. Both these algorithms are further extended by unsupervised machine-learning methods that allow them to refine decision boundaries for individual class memberships. For the shape-based strategy it is also outlined how explicit model fitting can be used to obtain reasonable particle samples.

The final framework is evaluated in a virtual environment, where unknown objects of different material stiffness have to be explored. Both algorithms are evaluated against their classification accuracy, where the grid-based algorithm is significantly outperformed by the shape-based method. The presented results highlight that the methods outlined in this chapter are a helpful step towards enabling robots in coping with unknown objects and thus increasing their field of applications in the future.

## Future Work

Building upon the examples and results presented in this chapter, future work could explore the usability and performance on real robot data recordings. Nonetheless, the presented approach suffers from high sensitivity to data measurements. Especially, regarding a proper measurement of displacement during contact with an unknown object is subject to sensor-noise, -delays and discretization errors. Thus, future work should focus on improving the accuracy in these measuring metrics first before applying the presented method on a real robot platform. In contrast to that, the basic filter architecture is suitable to be extended by a collection of individual data-driven components such as a *product of experts* or a connection of multiple deep networks. The main motivation here is that such models are directly compatible with current state-of-the-art vision classifiers or estimators. Unlike most commonly applying pure end-to-end learning, the individual classifiers bare potential on approximating a belief over material types or shapes, which can then be embedded in a similar framework to the one presented in this chapter. Eventually it is also of importance to develop suitable controllers for the robot hardware, to ensure contact stability and minimize encountered impacts. This is of utmost importance as undesired penetration of unknown surfaces – especially if said surface is stiff – may endanger to harm the hardware of robotic platforms.



# 8

## A Force-Sensitive Grasping Controller Using Tactile Gripper Fingers and an Industrial Position-Controlled Robot

### Chapter Abstract

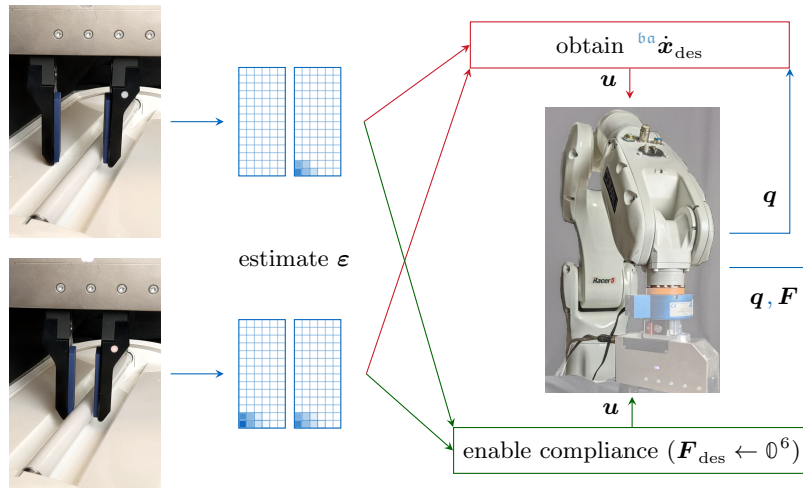
This chapter focuses on the aspect of grasping unknown objects with imperfect grasping pose estimations. Specifically, this chapter presents a novel grasping controller that allows (industrial) robots to compensate for object goal pose uncertainties during grasping by exploiting tactile feedback obtained from digital sensor arrays (DSAs) equipped on the gripper fingers.

First, we outline how the alignment pose error during initial object contact can be estimated from the current raw sensor readings. Given this alignment error estimation, we then provide two grasping control modalities that specifically steer the robotic configuration towards a suitable goal-pose by incrementally decreasing the alignment error. Specifically, the presented grasping control modalities allow to either directly compensate for interaction forces or to solve a constrained model predictive control (MPC)-problem to minimize the estimated alignment error. While the former relies on force-torque-sensor-data, the later assumes that the robot motion dynamics can be locally linearized, such that the MPC can be solved online. Both modalities exploit the structure of a hybrid control-interface given a Cartesian robot controller. The hybrid nature of the controller further allows to combine the presented grasping control modalities along selective axes in a hybrid nature alike.

We evaluate the proposed grasping controller on a parallel two-finger gripper, that is equipped with one DSA per finger, for which we also provide an extended ROS-driver that allows to obtain DSA-data at communication rates above 5 Hz. In our experiments, the presented method distinctly increases the success-rate and final grasping error-pose compared to a compliant controller, that only minimizes interaction wrenches during grasping. Specifically, a hybrid grasping strategy with compliance in translation in combination with an MPC for the attitude achieves the highest success-rate.

Given the empirical evidence in combination with the ability of creating hybrid grasping strategies along selective axes, the presented grasping controller does not only increase the skill-set of industrial robots in the presence of uncertainty but also opens new research paths towards applying stiff robots to handle fragile objects fully autonomously.

*Remark:* A majority of this chapter was previously published in [Gabler et al. \(2022b\)](#).



**Figure 8.1:** Adaptive grasping strategy using tactile sensor data to send hybrid control-strategies along selective axes to an industrial robot via **velocity-based** or **force-based** grasping strategies.

## 8.1 Introduction

In the context of sensitive grasping, a robot is required to react in a compliant manner in order to limit interaction wrenches between the robot gripper and the dedicated object. Thus, a majority of approaches rely on compliant manipulators, fingers or tool-changers that allow to directly adjust for unforeseen impacts, or seek for a sufficiently accurate perception framework. Nonetheless, few examples allow an adaptation of the grasp when neither an accurate perception nor a compliant robot platform is available. Thus, this chapter tackles the issue of sensitive grasping for conventional industrial robot control interfaces – i.e., stiff position-controlled manipulators – without relying on external camera or depth-sensor data.

In fact, we claim that it is beneficial to equip a robot with pressure-sensitive fingers using digital sensor arrays (DSAs). Using the spatial resolution of these DSAs rather than solely the force observed on each finger motor or the force-torque (FT)-sensor directly, we outline how a robot can estimate the alignment error during initial grasp trials.

In order to evaluate this approach empirically, we equip an industrial robot platform, a CO-MAU Racer 5 0.80, with a two-finger parallel gripper – a WSG 50 – that provides a DSA on each finger as shown in Figure 8.1. Besides extending the current robot operating system (ROS) driver with DSA support, the contribution of this chapter proposes an efficient gripper alignment error estimation. Furthermore, we outline how a hybrid force-position control architecture can be used to decrease the estimated alignment error, while keeping interaction forces of the robot and the environment limited. The novel grasping controller is evaluated on an exemplary disassembly scenario, the removal of a light-bulb within an emergency lamp, where the exact pose is noisy and the grasp is executed solely relying on FT and DSA measurements.

Below we sketch the contribution of this chapter in relation to related work. We proceed with a mathematical formulation of the problem in Section 8.2 and outline the presented grasping controller in Section 8.3. We conclude this chapter with an empirical evaluation in Section 8.4, and an overall conclusion in Section 8.5.



### 8.1.1 Related Work

In the context of adaptive grasping, a variety of approaches have focused on developing novel robot grasping grippers (Fan et al., 2018, Fox and III, 2020, Koustoumpardis and Aspragathos, 2004, Krut, 2005, Ma et al., 2013, Mohammadi et al., 2017, Tiziani et al., 2017), such as tendon-driven mechanics (Dollar et al., 2010) or low impedance fingers (Natale and Torres-Jara, 2006). These hardware designs have shown great results in improving robot capabilities, yet these systems are often costly and are not as mechanically robust and broadly available as off-the-shelf industrial grippers. Thus, the majority of related work focused on increasing robotic capabilities using off-the-shelf grippers. In here, several approaches have applied machine learning (ML)-techniques (Guo et al., 2016, 2017, Pinto and Gupta, 2016, Stulp et al., 2011), where the environment is mainly perceived via a camera, but also force-based approaches have been introduced (Dang and Allen, 2014, Merzic et al., 2019, Steffen et al., 2007) that rely on tactile feedback. These approaches show great performance but usually require tremendous amount of data. Similarly, planning-based methods have been outlined for automated grasping. These decision-theoretic concepts range from partially observable Markov decision processes (Garg et al., 2019, Hsiao et al., 2007), heuristic planning strategies (Hsiao et al., 2010, Leeper et al., 2010), visual servoing (Rusu et al., 2009, Saxena et al., 2008), specialized sensors (Hsiao et al., 2009), or online object refinement and adjustment (Dragiev et al., 2013). Regarding the aspect of tactile sensors, previous work has outlined how grasping robustness can be improved by using tactile sensors (Bekiroglu et al., 2011, Su et al., 2015), where the major emphasis is set on ML again or the environment is assumed to be compliant. The application of tactile fingers range from six-axes force-torque sensors (Eberman and Jr., 1994), stress rate sensors and acceleration sensors (Howe et al., 1990), biomimetic tactile sensors with a weakly conductive fluid (Wettels et al., 2008) to optical force measurements (Ward-Cherrier et al., 2018). The concept of DSA was proposed by Romano et al. (2011) that shall mimic mechanoreceptive afferents in glabrous – i.e., non-hairy – human skin (Johansson and Flanagan, 2009). A tactile sensor array unit that does not only measure normal forces but also shear forces, was developed by Kis et al. (2006). As these approaches have focused on detecting events, rather than controlling the motion of a robot based on the DSA feedback, we focus on the concept of versatile manipulation with simple grippers (Mason et al., 2012, 2009). Thus, we seek to improve robot grasping skill-sets by deriving a grasping strategy exploiting the DSA readings.

### 8.1.2 Contribution

In this context the contribution of this chapter is given as:

1. outlining a novel grasping alignment error estimation based on tactile sensor readings without relying on additional FT-data,
2. a publicly available ROS-driver implementation with support for the tactile WSG 50 DSA fingers<sup>1</sup> with a communication rate of up to 5.49 Hz during contact,
3. defining novel grasping strategies that exploit the current alignment error estimation and drive a robot to the desired goal pose while limiting the interaction force between robot and object.

<sup>1</sup>available at <https://gitlab.com/VGab/ros-wsg-50>

## 8.2 Problem Formulation

Given the current robot configuration  $\mathbf{x}$  in SE(3), defined by the current joint configuration  $\mathbf{q} \in \mathbb{R}^n$ , where  $n$  denotes the degrees of freedom (DoF) of the manipulator, there exists a desired pose  ${}^{\text{to}}\mathbf{x}_{\text{des}}$  w.r.t.  $\text{to}$ , that allows a robot to successfully grasp an object. This pose is in general unknown to the robotic system.

In the context of this chapter, an industrial robot is required to align the pose of the end-effector during initial grasping contact, i.e., when the gripper fingers initiate contact with an object. We explicitly focus on situations, where the robot has no access to visual or point-cloud data. Instead, the sensory input is limited to an external FT, i.e.,  $\mathbf{F}_{\text{cur}}$ , as well as tactile sensor-readings in the gripper-fingers of a parallel finger gripper, denoted as  ${}^{\{0,1\}}\mathfrak{D}$ . Thus, the problem of this chapter becomes twofold. First the end-effector pose error

$${}^{\text{to}}\boldsymbol{\varepsilon}_{\text{t}} \leftarrow \mathcal{F}_{\text{estim}}(\mathbf{x}_{\text{t}}, {}^0\mathfrak{D}_{\text{t}}, {}^1\mathfrak{D}_{\text{t}}) \quad (8.1)$$

needs to be estimated from sensor readings at a frequency that allows to obtain suitable control commands online. Given this alignment error, the second problem is then given by alternating the configuration of an industrial, i.e., solely position-controlled, robot

$$\mathbf{x}_{\text{t}+1} \leftarrow \mathcal{F}_{\text{align}}(\mathbf{x}_{\text{t}}, {}^{\text{to}}\boldsymbol{\varepsilon}_{\text{t}}, \mathbf{F}_{\text{curt}}), \quad (8.2)$$

such that eventually  $\mathbf{x} = \mathbf{x}_{\text{des}}$  holds, and the interaction forces with the object remain limited.

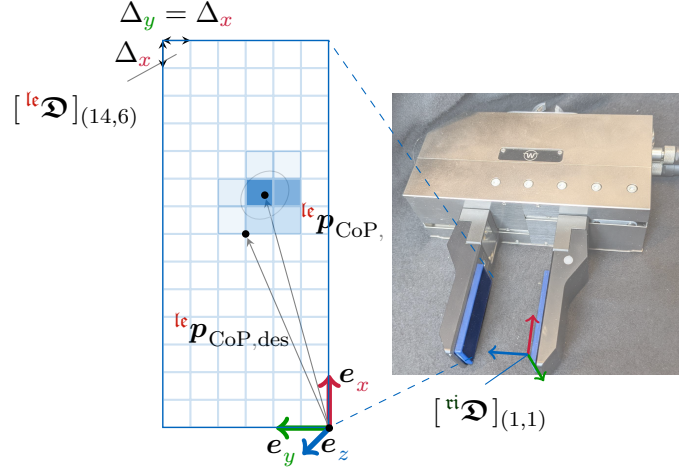
## 8.3 Technical Approach

According to the problem from Section 8.2, our approach is outlined sequentially. First, we show how the end-effector alignment error is estimated from tactile sensor-arrays. Given this, we outline how the industrial robot in use is controlled and how the available control interfaces are used to steer the robot to  $\mathbf{x}_{\text{des}}$ .

### 8.3.1 Alignment Error Estimation Using Tactile Sensor Arrays

Pressure-sensitive DSAs allow a robot to record a discretized sensor-cell matrix reading of an  $N_x \times N_y$  sized array, i.e.,  ${}^{\text{lc}}\mathfrak{D} \in \mathbb{N}^{N_x \times N_y}$ . Introducing frames  ${}^{\text{lc}}\mathbf{r}_i$  for the left and right DSA-fingers, as well as cell-sizes  $\Delta_x$  according to Figure 8.2, the center location  ${}^{\text{lc}}\mathbf{p}_{ij}$  of each cell-element  $[{}^{\text{lc}}\mathfrak{D}]_{(i,j)}$  can be expressed in the respective finger reference frame  ${}^{\text{lc}}\mathbf{p}_{ij} = [i\Delta_x \quad j\Delta_y \quad 0]^\top$ . As a result, the center of pressure (CoP) is obtained as

$$\begin{aligned} {}^{\text{lc}}\mathbf{p}_{\text{CoP,des}} &:= \frac{\iint {}^{\text{lc}}\mathbf{p}_{x,y} {}^{\text{lc}}\mathfrak{D}(x,y) dx dy}{\iint {}^{\text{lc}}\mathfrak{D}(x,y) dx dy} \\ &\approx \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} {}^{\text{lc}}\mathbf{p}_{ij} [{}^{\text{lc}}\mathfrak{D}]_{(i,j)}}{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} [{}^{\text{lc}}\mathfrak{D}]_{(i,j)}} \end{aligned} \quad (8.3)$$



**Figure 8.2:** DSA cell arrays with exemplary pressure data for  ${}^{lc}\mathfrak{D}$ . Colors encode the coordinate-system axes  $e_x$ ,  $e_y$  and  $e_z$ . In here a toy-pressure distribution as a grey ellipsoid is shown, that results in the colored pressure distribution.  ${}^{lc}\mathbf{p}_{CoP,des}$  is the center of the array and assumed to be the desired CoP here, while  ${}^{lc}\mathbf{p}_{CoP}$  is obtained from (8.3).

The center of pressure is visualized in Figure 8.2 for a fictional pressure surface and reading. Assuming that both fingers are in contact with an object, the unit-vector connecting the CoPs of both fingers results in

$${}^{vi}\mathbf{e}_{cur} = \frac{1}{\nu} \left( \underbrace{{}^{vi}\mathbf{T}_{{}^{lc}} \quad {}^{lc}\mathbf{p}_{{}^{lc}CoP} - \quad {}^{vi}\mathbf{p}_{{}^{vi}CoP}}_{{}^{vi}\mathbf{p}_{{}^{vi}CoP,cur}} \right), \quad (8.4)$$

where  $\nu$  is a normalizing constant, the middle of the connecting vector

$${}^{to}\mathbf{p}_{CoP,ctr} = {}^{to}\mathbf{T}_{{}^{vi}} \left( {}^{vi}\mathbf{p}_{{}^{lc}CoP} + \frac{1}{2} {}^{vi}\mathbf{p}_{{}^{vi}CoP,cur} \right) \quad (8.5)$$

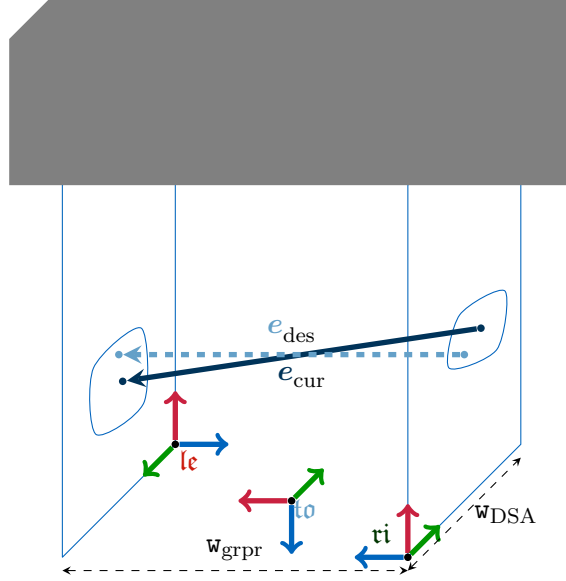
in the tool-frame of the robot. The transformation from  ${}^{vi}$  to  ${}^{lc}$  is given as

$${}^{lc}\mathbf{T}_{{}^{vi}} = \begin{bmatrix} \mathbf{R}_\varphi(\pi) & \begin{bmatrix} 0 \\ \mathbf{w}_{DSA} \\ \mathbf{w}_{grpr} \\ 1 \end{bmatrix} \\ \mathbb{0}^{1 \times 3} & \end{bmatrix}, \quad (8.6)$$

where the current opening width  $\mathbf{w}_{grpr}$  of the gripper is the only time-variant value besides  $\mathbf{w}_{DSA} = N_y \Delta_y$  as the width of the tactile sensor array and the rotation matrix as a  $180^\circ$  flip around  ${}^{vi}\mathbf{e}_x$ . Similarly, the transformation from  ${}^{to}$  to  ${}^{vi}$  is given as

$${}^{vi}\mathbf{T}_{{}^{to}} = \begin{bmatrix} \mathbf{R}_\theta(-\frac{\pi}{2}) & \frac{1}{2} \begin{bmatrix} 0 \\ \mathbf{w}_{DSA} \\ \mathbf{w}_{grpr} \\ 1 \end{bmatrix} \\ \mathbb{0}^{1 \times 3} & \end{bmatrix}, \quad (8.7)$$

with a rotation of  $-90^\circ$  around  $y$ . In order to estimate the current alignment error,  $\{{}^{lc}, {}^{vi}\}\mathbf{p}_{CoP,des}$ , the desired location of each CoP is assumed to be known, such that  ${}^{vi}\mathbf{e}_{cur}$  can be inferred.



**Figure 8.3:** Alignment error estimation visualization given the CoPs in each finger, connected by a blue vector, while the green line denotes the vector connecting the desired CoPs. The centers of both vectors and respective unit-vector are then used to estimate the alignment error according to (8.8).

Assuming that each CoP is located at the center of each finger, the relation for an exemplary pressure distribution is shown in Figure 8.3. In this case, these CoPs are identical to (8.3) if  ${}^{vi}\mathcal{D} = \mathbb{0}^{N_x \times N_y}$  holds. Transforming the unit-vectors into the tool-frame, i.e.,  ${}^{to}\mathbf{e}_p$ , with  $\mathbf{p} = \{\text{des}, \text{cur}\}$ , the alignment error is estimated as

$${}^{to}\boldsymbol{\varepsilon} := \begin{bmatrix} {}^{to}\boldsymbol{\varepsilon}_{\text{trans}} \\ {}^{to}\boldsymbol{\varepsilon}_{\text{rot}} \end{bmatrix}, \quad (8.8)$$

$$\text{where } {}^{to}\boldsymbol{\varepsilon}_{\text{trans}} = {}^{to}\mathbf{p}_{\text{CoP,des}} - {}^{to}\mathbf{p}_{\text{CoP,ctr}}, \quad (8.9)$$

and the rotation error  ${}^{to}\boldsymbol{\varepsilon}_{\text{rot}}$  defines the rotation needed to align  ${}^{to}\mathbf{e}_{\text{cur}}$  to  ${}^{to}\mathbf{e}_{\text{des}}$ . As there are infinite solutions due to the symmetry of the unit-vectors around the  $y$ -axis, the  $y$ -axis can be either obtained from matching the contours on both DSAs, or projected into the  $xy$ -plane of  ${}^{to}$  via  ${}^{to}\tilde{\mathbf{e}}_{y_p} = [[{}^{to}\mathbf{e}_p]_{(y)}, \sqrt{1.0 - [{}^{to}\mathbf{e}_p]_{(y)}^2}, 0.0]^T$ . Using the direction of the unit-vectors as the  $y$ -axis, the dedicated  $z$ -axis is obtained via the cross-product of  ${}^{to}\mathbf{e}_p$  and  ${}^{to}\tilde{\mathbf{e}}_y$ :

$${}^{to}\boldsymbol{\varepsilon}_{\text{rot}} \leftarrow \mathcal{F}_{\text{RPY}} \left( \mathbf{R} \left( {}^{to}\mathbf{e}_{\text{cur}}, {}^{to}\tilde{\mathbf{e}}_{y_{\text{cur}}}, {}^{to}\mathbf{e}_{\text{cur}} \times {}^{to}\tilde{\mathbf{e}}_{y_{\text{cur}}} \right)^T, \mathbf{R} \left( {}^{to}\mathbf{e}_{\text{des}}, {}^{to}\tilde{\mathbf{e}}_{y_{\text{des}}}, {}^{to}\mathbf{e}_{\text{des}} \times {}^{to}\tilde{\mathbf{e}}_{y_{\text{des}}} \right) \right), \quad (8.10)$$

where  $\mathcal{F}_{\text{RPY}}$  denotes the mapping from rotation matrix to  $\varphi, \theta, \psi$  in order  $zyx$ , and the individual rotation matrices are constructed from their individual unit-axes. Finally, a special case is given if only one finger establishes contact with the object. In such cases, the attitude error is set to zero in (8.8), while (8.9) is replaced by an heuristic component

$${}^{to}\boldsymbol{\varepsilon}_{\text{trans}} := \begin{cases} {}^{to}\mathbf{T}_{l_c} \begin{bmatrix} 0 & 0 & \frac{1}{2}w_{\text{grpr}} \end{bmatrix}^T & \text{iff } \|l_c\mathcal{D}\|_2 > 0 \\ {}^{to}\mathbf{T}_{t_i} \begin{bmatrix} 0 & 0 & \frac{1}{2}w_{\text{grpr}} \end{bmatrix}^T & \text{iff } \|t_i\mathcal{D}\|_2 > 0 \\ \mathbb{0}^3 & \text{else} \end{cases}. \quad (8.11)$$

### 8.3.2 Controller Design for Adaptive Grasping with an Industrial Robot

Having obtained the estimated alignment error as a Cartesian pose error in SE(3), the subsequent problem according to Section 8.2 is given by deploying a suitable control strategy to minimize this error. As most common grippers do not allow to control each finger separately, the alignment control needs to be mainly handled by the robot manipulator, which is assumed to be an industrial robot manipulator with a restricted control interface – i.e., controlling the Cartesian pose of the robot end-effector. In order to achieve a sensitive alignment control, we propose two control-strategies, which are schematically outlined in Figure 8.1: a compliant force-based strategy that incorporates additional FT-sensor data readings, as well as a model predictive control (MPC) that generates a constrained Cartesian velocity-profile. Defining the control-command  $\mathbf{u}$  as the Cartesian end-effector-velocity, we outline these strategies below, followed by an insight about the actual implementation on the robot platform.

#### 8.3.2.1 Force-Based Grasping Strategy

This grasping strategy allows a compliant robot behavior towards external Cartesian wrenches, via a PI-control

$$\mathbf{u}^{\text{frc}}_{\text{PI}} := \mathbf{K}_{\text{P}}^{\text{frc}} (\mathbf{F}_{\text{des}} - \mathbf{F}) + \mathbf{K}_{\text{I}}^{\text{frc}} \mathbf{F}_{\text{des}i} - \mathbf{F}_i, \quad (8.12)$$

using positive semi-definite control-gain matrices  $\mathbf{K}_{\text{P}}^{\text{frc}}$  and  $\mathbf{K}_{\text{I}}^{\text{frc}}$ , as well as a sliding window of size  $N_{\text{I}}$  for numerical integration, to steer the interaction wrenches towards the desired value  $\mathbf{F}_{\text{des}}$ . Assuming the object to be grasped is quasi-static, an alignment error during grasping results in a non-zero wrench measurement, while the minimal interaction force is obtained when the robot is aligned correctly w.r.t. the object. Thus, the desired Cartesian wrench is set to  $\mathbf{F}_{\text{des}} = \mathbf{0}^6$ . In order to incorporate the alignment error from Section 8.3.1, the actual command forwarded to the robot is obtained as

$$\mathbf{u}^{\text{frc}}_{\text{p}} := \mathbf{S}^{\text{frc}}_{\mathbf{R}} \mathbf{u}^{\text{frc}}_{\text{PI,p}}, \text{ with} \quad (8.13)$$

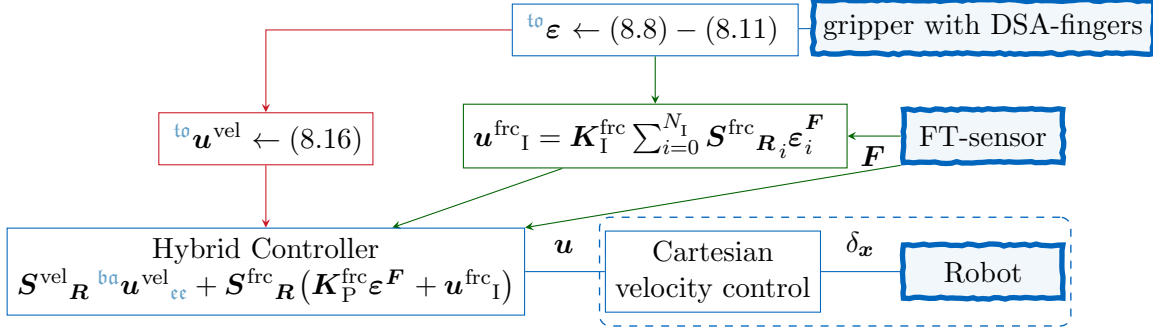
$$\mathbf{S}^{\text{frc}}_{\mathbf{R}} := \begin{bmatrix} \mathbf{R}_{\text{to}}^{\text{ba}} \text{diag}(s_{1:3}) \mathbf{R}_{\text{ba}}^{\text{to}} & \mathbf{0}^{3 \times 3} \\ \mathbf{0}^{3 \times 3} & \mathbf{R}_{\text{to}}^{\text{ba}} \text{diag}(s_{3:6}) \mathbf{R}_{\text{ba}}^{\text{to}} \end{bmatrix}. \quad (8.14)$$

A binary selection signal  $s$  is introduced w.r.t.  $\text{to}$ , that evaluates the alignment error against a non-negative threshold  $\zeta_{\text{align}}$ :

$$s_k := \begin{cases} 1 & \text{if } \|\text{to} \boldsymbol{\varepsilon}_k\|_1 > \zeta_{\text{align}} \\ 0 & \text{else} \end{cases}, \quad (8.15)$$

which is usually set to 0. Nonetheless,  $\zeta_{\text{align}}$  can be adjusted as an additional hyperparameter to diminish the sensitivity of the grasping controller to small alignment errors. Recalling the integral term in (8.12), it has to be noted that this may lead to instability if contact dynamics are changing. However, (8.13)-(8.15) allow adjusting the integral-term during contact-changing events. Therefore, not only the force-errors are stored in a sliding window to calculate the I-term in (8.12), but also the dedicated selection-matrices  $\mathbf{S}^{\text{frc}}_{\mathbf{R}_t}$  from (8.13) at each time step  $\mathbf{t}$ . If eventually the value of  $s_k$  changes, all values of  $k$  are set to zero.

Given that the obtained alignment error stems from a pressure distribution along the sensor cell arrays, minimizing the force-control error along the selected axes via the force-control term in (8.18) eventually minimizes  $\text{to} \boldsymbol{\varepsilon}_{\text{cc}}$  and thus also  $\text{ba} \boldsymbol{\varepsilon}_{\text{cc}}$ .



**Figure 8.4:** Schematic overview of the proposed adaptive grasping controller. Colors encode the **velocity**-based versus the **force**-based grasping strategy, while noisy blue boxes denote hardware components. Desired values have been omitted for brevity.

### 8.3.2.2 Velocity-Based Grasping Strategy

The alternative strategy uses the velocity control interface and the estimated alignment error. In contrast to the force-control, there is no feedback obtained from the environment within the compliance controller, i.e., this strategy is also applicable on robots without FT-sensors and in applications where there is no direct relation between force-data and the alignment error. For this strategy, we exploit the fact that the velocity of the robot needs to be distinctly constrained in order to prohibit high impacts. Using a sufficiently small time step  $\delta_t$ , the Cartesian robot motion-dynamics can be approximated as a linear point-mass  ${}^{to}\boldsymbol{x}_{t+1} \approx {}^{to}\dot{\boldsymbol{x}} + {}^{to}\boldsymbol{u}_t \delta_t$ , such that the alignment error can be decreased by solving the MPC-problem

$$\begin{aligned} \min_{{}^{to}\boldsymbol{u}_{t:T_{\max}}} & \sum_{t=0}^{T_{\max}} {}^{to}\boldsymbol{\varepsilon}_t^\top \boldsymbol{C}_{\text{sys}} {}^{to}\boldsymbol{\varepsilon}_t + \boldsymbol{u}_t^\top \boldsymbol{C}_{\text{inp}} {}^{to}\boldsymbol{u}_t \\ \text{s.t.} & \quad {}^{to}\boldsymbol{\varepsilon}_{t+1} = \boldsymbol{A} {}^{to}\boldsymbol{\varepsilon}_t + \boldsymbol{B} {}^{to}\boldsymbol{u}_t \\ & \quad -\dot{\boldsymbol{x}}_{\max} \leq {}^{to}\boldsymbol{u}_t \leq \dot{\boldsymbol{x}}_{\max} \end{aligned}, \quad (8.16)$$

with the maximum impact velocity  $\dot{\boldsymbol{x}}_{\max}$  as an upper constraint and  $\boldsymbol{A} = \mathbb{1}^{6 \times 6}$ ,  $\boldsymbol{B} = \mathbb{1}^{6 \times 6} \delta_t$  due to the locally linearized model. Solving (8.16) obtains the desired end-effector velocity in tool frame  ${}^{to}$ , i.e.,  ${}^{to}\boldsymbol{u}_t$ , which can then be transformed to  $\boldsymbol{u}_t$  and commanded to the robot.

### 8.3.2.3 Overall Controller Implementation

Given the proposed grasping strategies from above, we now outline how the full grasping controller is implemented. The overall controller is sketched in Figure 8.4, where the upper branch denotes the alignment error-estimation from Section 8.3.1. On the bottom right-hand side, the industrial robot platform – emphasized as a dashed box – is commanded via the Cartesian end-effector-velocity-commands  $\boldsymbol{u}$ , obtained by the grasping strategies from the paragraphs above. While a Cartesian velocity-command can usually be achieved by any industrial robot, this chapter applies a COMAU robot that allows to send a Cartesian deviation command relative to the current end-effector pose, such that the controlled system simplifies to

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t + \delta_x \approx \boldsymbol{x}_t + \boldsymbol{u} \delta_t. \quad (8.17)$$

As the robot runs at a real-time-safe, constant update-rate  $\delta_t$ , it is possible to design a hybrid force-velocity controller, that sends

$$\mathbf{u} := \mathbf{S}^{\text{vel}} \mathbf{R}^{\text{ba}} \dot{\mathbf{x}}_{\text{ccdes}} + \mathbf{S}^{\text{frc}} \mathbf{R} \mathbf{K}_{\text{p}}^{\text{frc}} \boldsymbol{\varepsilon}^{\text{F}} \quad (8.18)$$

to the robot, where the selection-matrices are calculated via (8.14). Thus, the controller either follows a Cartesian velocity-profile  ${}^{\text{ba}}\dot{\mathbf{x}}_{\text{ccdes}}$  or force-profile  $\mathbf{F}_{\text{des}}$  via minimizing the force-error  $\boldsymbol{\varepsilon}^{\text{F}}$  from (8.12). This controller differs from classic hybrid force-position control by the fact that disabling one control-modality does not directly result in switching to the alternative modality. Instead if  $s_k^{\text{x}} = s_k^{\text{F}} = 0$  holds, the robot enables stiff-position control, and keeps the current position according to the internal control loop and (8.17). Nonetheless, for a correct decoupling of the individual control policies, the selection matrices need to hold  $s_k^{\text{x}} s_k^{\text{F}} = 0$ . Thus, by limiting each selection-element (8.15) to one specific grasping strategy, the control outputs of the individual component can directly be inserted in (8.18). Transforming  ${}^{\text{to}}\mathbf{u}_v$  into  ${}^{\text{ba}}\mathbf{u}_{\text{cc},v}$  and replacing the P-control in (8.18) by (8.13), we obtain the final control-architecture from Figure 8.4.

Even though the method sketched above has been outlined explicitly for a parallel gripper, i.e., an alignment error estimation from two DSAs, the presented method is not limited to such setups. As both control-strategies in this chapter solely rely on the estimated alignment error, they are not directly dependent on the number of fingers in use, as long as the number of fingers may decrease the update rate of the alignment error. In general, these control-strategies are expected to profit from an increased number of DSAs, given that the actual alignment error estimation is directly correlated with the number of DSAs in use. The method presented in Section 8.3.1 can be extended to multiple fingers, by evaluating the pressure distribution and / or CoPs of each finger against a desired set-value. In fact, comparing the CoP of each finger against the geometrical center of all CoPs, results in an alignment error per finger. While minimizing the mean of these alignment errors replicates the procedure presented in this chapter, having access to multiple CoP measurements allows to use more advanced methods, such as weighted mean. As this is subject to empirical evaluation, we leave the evaluation of these metrics for future work.

### 8.3.3 Implementation Details

Having outlined the general procedure above, we proceed with the implementation details for the WSG 50 gripper-driver. The WSG 50 records a 12 bit encoded DSA-reading for each finger, where  $N_x = 14$  and  $N_y = 6$ . In order to transfer the sensor-readings from the sensor cells to an external client, the gripper provides the possibility of transferring data via transmission control protocol (TCP) over an ethernet connection. This controller is then capable of running the controllers from Figure 8.4. Within the provided driver, the gripper acts as the server that allows to adjust the feedback sent to the client w.r.t. on-demand flags send by the client. Thus, the DSA reading can be stopped by the client if there is no contact to be expected, which allows to increase the communication speed. Due to the 12 bit encoding, each cell requires two bytes (2B) to be sent. In order to increase communication rate during DSA read outs, two modes are available:

- full DSA read with a message size of  $14 \cdot 6 \cdot 2 \text{ B} = 336 \text{ B}$ .

- CoP with approximated pressure surface from 21 B to 62 B in contact and 10 B without contact.

The compact message, i.e., the CoP-mode, does not transfer the full DSA reading but calculates the CoPs directly on the gripper driver before sending it to the dedicated client via TCP. As a pressure distribution may also be multi-variate, it is insufficient to solely transfer the CoP to the client. Instead, the denominator of (8.3) is sent in combination with contour edge of the pressure surface. The client can thus evaluate the current CoP alongside an equal discrete distribution approximation by flattening the pressure equally over occupied cells. The static body of the dynamic message is given as the total pressure sum as 32 bit integer and the 8 bit integer to denote the number of occupied rows (5 B), where column and row refer to  $e_x$  and  $e_y$  according to Figure 8.2, for both DSA-fingers, resulting in a minimum message size of 10 B. In case a contact is detected, the message is extended by the CoP (8 B), and a list of three byte tuples containing the row-index as well as the dedicated minimum and maximum column cell-index for each DSA-finger. Referring to the example from Figure 8.2, the additional information besides the CoP and pressure sum would be given as  $\{2, 7, 9, 3, 7, 9, 4, 7, 8\}$ . We evaluate the effects on the overall communication rate in the next section empirically.

## 8.4 Experiment

In this section we evaluate the proposed grasping controller and gripper driver w.r.t. communication speed and functionality. We start with empirically validating and highlighting the available communication modalities of the gripper driver.

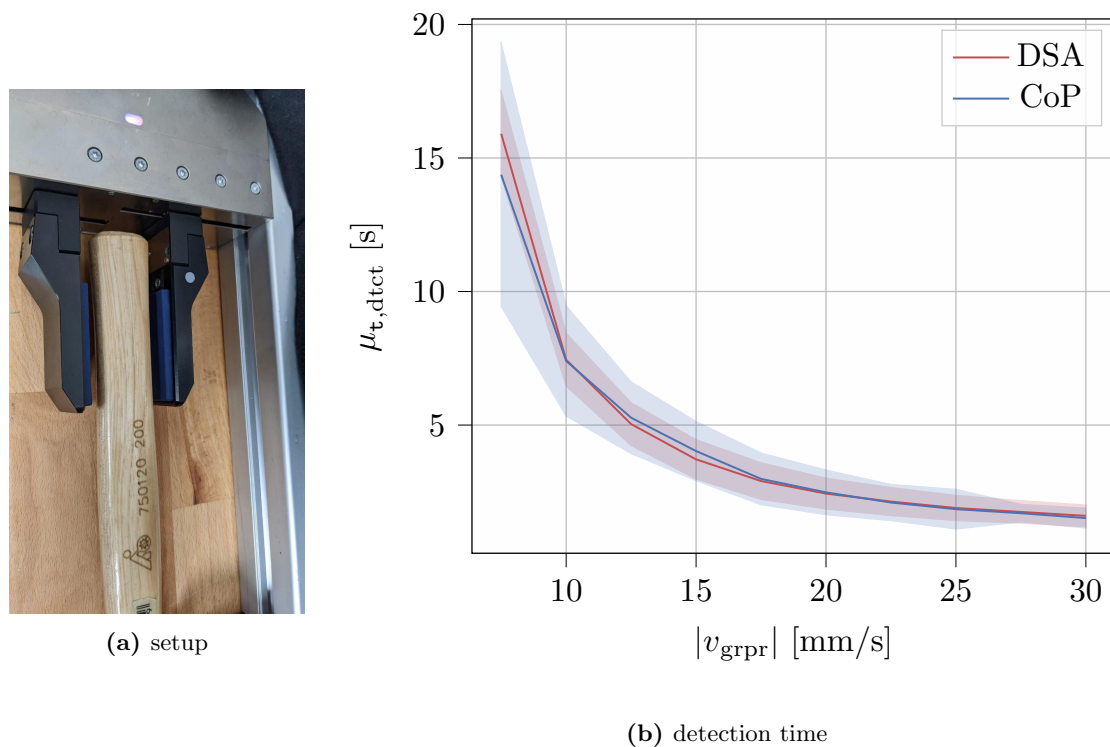
### 8.4.1 Evaluating Communication Speed

In order to compare the different communication modes of the proposed gripper driver, the gripper is tested on a benchmark setup as depicted in Figure 8.5a. Regarding the communication speed, Table 8.1 collects the averaged ROS update rates depending on the control mode over 50 runs each. In order to evaluate the effects on the reaction capabilities of a robot, we evaluated the detection time needed, if the gripper is set to a constant velocity control with a fixed object in the middle of the fingers. Even though the communication rate of the full-DSA reading is reduced according to Table 8.1 compared to the sparse read-out, the detection time is only affected at very slow movements as can be seen by the averaged contact detection-times  $\mu_{t,\text{dct}}$  in Figure 8.5b.

Control-mode	$\mu_{\text{rate}}$	$\sigma_{\text{rate}}$	$\mu_{\text{cont}}$
full DSA read [Hz]	9.08	1.66	4.50
sparse DSA read [Hz]	<b>9.62</b>	<b>1.25</b>	<b>5.49</b>

**Table 8.1:** Average ROS-update rate in Hz of the gripper driver with enabled DSA-reading. Improved results are highlighted in bold.



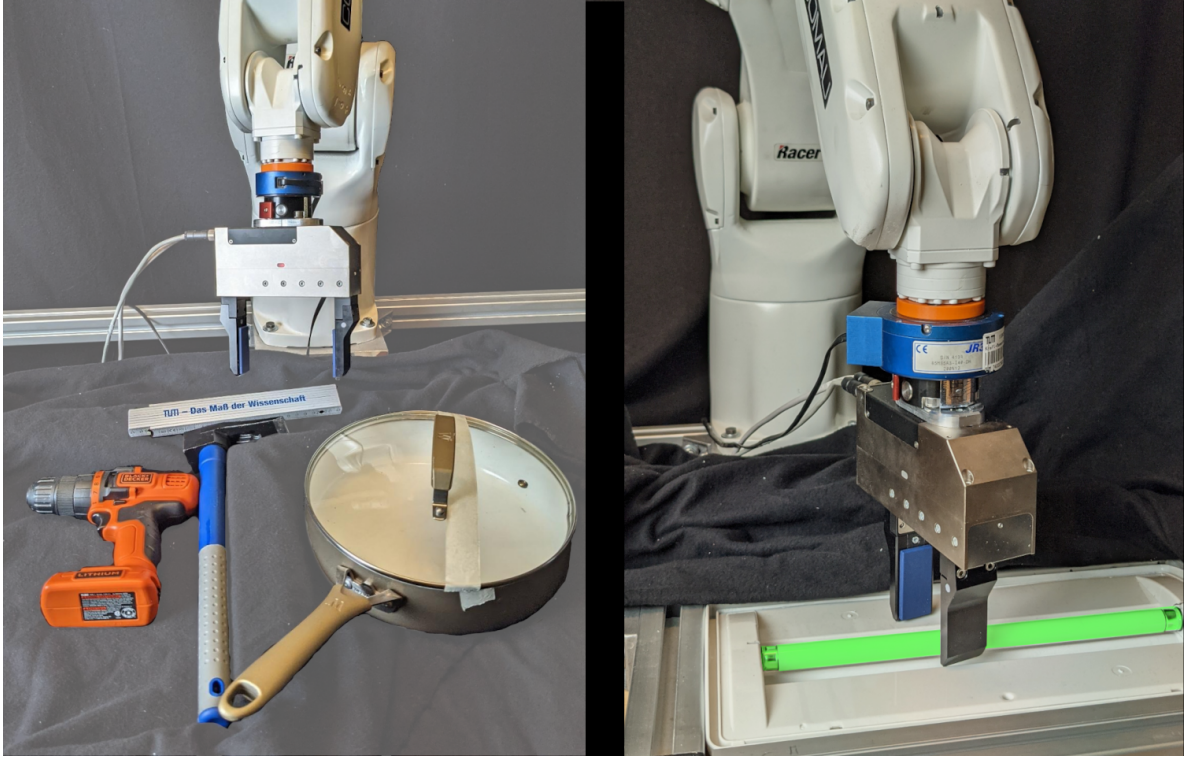


**Figure 8.5:** Empirical comparison of the available control modes in terms of communication speed and contact detection speed. The results are averaged over 50 runs per control mode, while the plot show the mean and a confidence-interval (CI) of 95% for both modes.

### 8.4.2 Evaluation of Proposed Grasping Strategies

In order to highlight the functionality of the proposed controller, we evaluate the grasping strategies on exemplary object grasping tasks. Recalling the main motivation of this chapter, the robot has no access to visual or depth camera sensors and is asked to correctly grasp the objects, given a noisy guess on the object goal poses. Thus, we compare the presented control method against a pure compliant robot control, where the robot follows a compliant force-control policy according to (8.18), with  $\mathbf{K}_I^{\text{frc}} = \mathbb{0}^{6 \times 6}$  and  $s_k^{\mathbf{F}} = 1$ . The robot platform is given as a COMAU Racer 5 0.80 as depicted in Figure 8.6. As our method does not consider collision avoidance or plan for an optimal trajectory, the task is initiated in close distance to the object. For the cross-comparison of the strategies in use, each grasping task is implemented as a sequence of the following action-primitives:

1. approach pre-pose.
2. initiate gripper closure with closing speed  $v_{grpr}$ .
3. align gripper according to Section 8.3.
4. identify translation offset as center of orientation.
5. (optionally) tilt object and remove.



**Figure 8.6:** Experimental evaluation setup. The figure on the left hand side shows the selected objects to be grasped by the robot. For each object, the robot is provided with a noisy grasping goal-pose. Being restricted to sensory inputs from the FT-sensor and DSAs, the robot is then asked to align the gripper correctly w.r.t. the dedicated object. The right hand side shows the task of removing an emergency lightbulb, on which we evaluate the grasping strategies from Section 8.3.2.1 and Section 8.3.2.2.

The closing speed of the gripper  $v_{\text{grpr}}$  is set to 10 mm/s, which also serves as the control constraint in (8.16). The proportional gains for the force-controller and the force-based strategy (frc-based) grasping strategy are  $[\mathbf{K}_P^{\text{frc}}]_{\{x,y,z\}} = 8 \cdot 10^{-4}$  for translation and  $[\mathbf{K}_P^{\text{frc}}]_{\{\varphi,\theta,\psi\}} = 2 \cdot 10^{-3}$  for rotation. In addition, we set  $N_I = 50$  and  $\mathbf{K}_I^{\text{frc}} = 1 \cdot 10^{-6} \mathbb{1}^{6 \times 6}$  for the integral part from Figure 8.4. For the velocity-based strategy we applied  $\mathbf{C}_{\text{sys}} = \mathbb{1}^{6 \times 6}$  and  $\mathbf{C}_{\text{inp}} = 1 \cdot 10^{-4} \mathbb{1}^{6 \times 6}$  and used [Andersson et al. \(2012\)](#), [Wächter and Biegler \(2006\)](#) and [Lucia et al. \(2017\)](#) to solve (8.16). In order to not only neglect small estimation errors but also allow the robot to detect a successful alignment procedure, the threshold in (8.15) is set to  $\zeta_{\text{align}} = 1 \cdot 10^{-2}$  rad and  $\zeta_{\text{align}} = 0.5$  mm for the velocity- and frc-based grasping strategies from Section 8.3.2, respectively.

As shown in Table 8.2, we empirically evaluated four variants of our proposed grasping controller against the baseline method while grasping the objects from Figure 8.5a. The grasping modes frc-based and velocity-based strategy (vel-based) solely forward the commands obtained from Section 8.3.2.1 and Section 8.3.2.2, while hybrid-force-velocity strategy (hybrid-f-v) and hybrid-velocity-force strategy (hybrid-v-f) apply a hybrid strategy along complementary Cartesian directions. In detail, hybrid-f-v follows the frc-based strategy in translation and the velocity-based strategy for the rotation, while hybrid-v-f reverts this scheme. As the presented methods rely on encountered interaction wrenches and/or sensor readings from

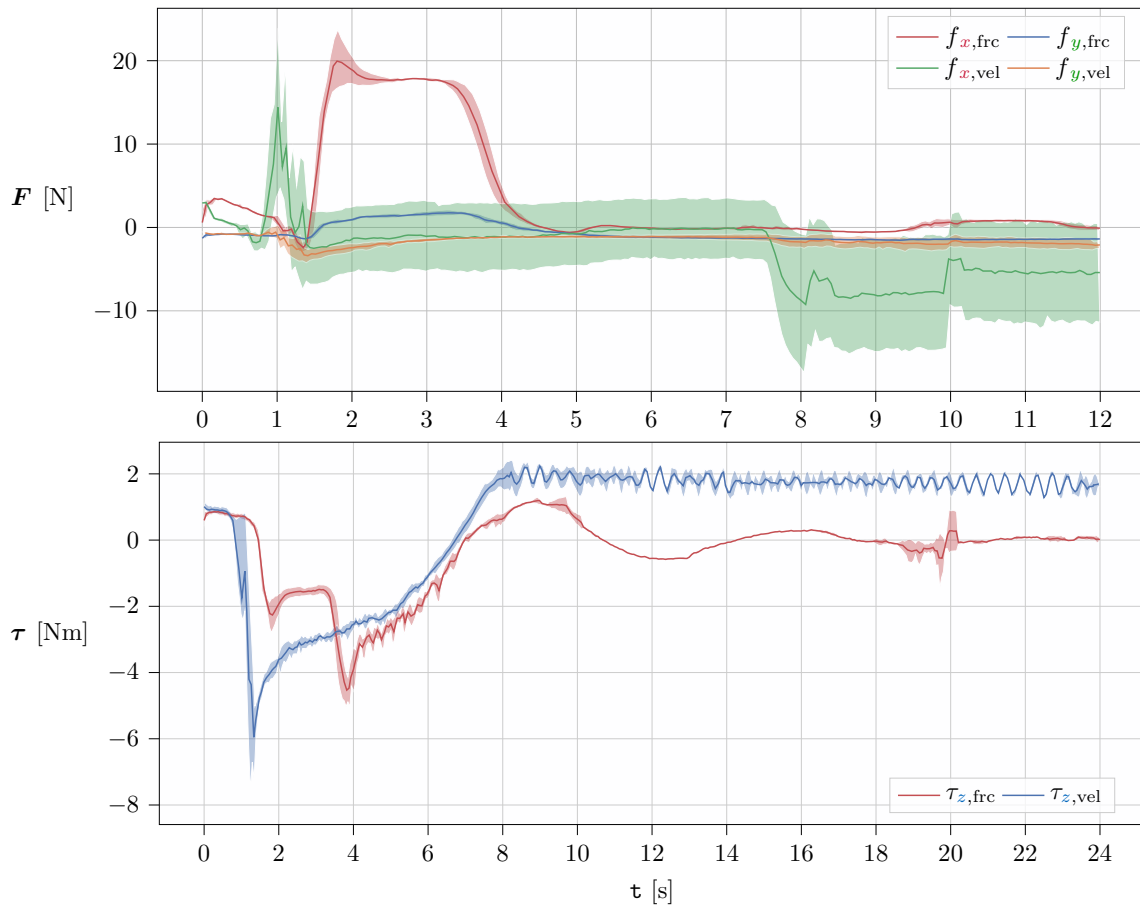
the DSAs, which result from said interaction forces, the objects from Figure 8.6 were fixed to the table in front of the robot. For each of these objects 10 starting poses were sampled and evaluated against all methods given a fixed length of 2000 steps – using an update rate of 100 Hz – for the actual grasping phase to allow for a fair cross-comparison. Recalling the motivation from Section 8.2, Table 8.2 lists the averaged norm-values for the alignment error of the grasping phase as well as the accumulated wrench per step. The last column denotes the success-ratio over all runs, where a task is defined as successful if the final error is below 0.5. As our strategies are non-compliant if there is no contact or feedback obtained from the gripper, the baseline method outperforms our approaches in terms of overall interaction wrenches. Nonetheless, the baseline solution fails on solving the task and remains stuck at unacceptable final error residuals. Even though the velocity-based strategy performs best w.r.t. the overall alignment error, this strategy still suffers from large interaction wrenches. Given that the hybrid strategy performs best in terms of success-ratio, we propose that our method is best to be applied by its’ hybrid nature. Adding feasible upper wrench constraints can thus be used to apply the velocity-based control strategy if possible and switch to a compliant control otherwise. We outline the evaluation of the interaction wrenches for both strategies in more detail below.

#### 8.4.2.1 Cross-Comparison of the Presented Grasping Strategies

In this section we explicitly compare the presented strategies from Section 8.3.2.1 and Section 8.3.2.2 based on the removal of an emergency lightbulb as presented in the accompanied media attachment. Except using another object for the grasping task, the experimental procedure is identical to Section 8.4.2. The resulting force-profiles in  $xy$ -plane of the robot as well as the Cartesian torque in  $\psi$  are sketched in Figure 8.7. Due to the translational offset, the robot encounters contact with the object on one finger, which results in an increased force-peak for both strategies. While the MPC controller is tuned well, and adjusts the velocity to the current closing speed of the gripper, the force remains limited in this stage. In contrast, the compliant force-controller suffers from the induced temporal delay due to the DSA-readings, which allows to compensate for the current force-error but results in an error-residual while being dragged across the workspace by the lightbulb. In contrast, the force-controller successfully steers the Cartesian wrench to zero and thus also compensates the attitude error in  $\psi$ , while the MPC-controller focuses on the positional error. Even though

Object Strategy	Pocket Ruler		Hammer		Electrical Drill		Cooking Pan		$\mathbb{P}_{\text{suc}}$
	$\mu_{\epsilon}$	$\mu_{\mathbf{F}}$	$\mu_{\epsilon}$	$\mu_{\mathbf{F}}$	$\mu_{\epsilon}$	$\mu_{\mathbf{F}}$	$\mu_{\epsilon}$	$\mu_{\mathbf{F}}$	
frc-based	1.576	12.34	2.118	18.93	0.526	17.92	0.797	10.36	41.17%
vel-based	<b>1.216</b>	54.98	<b>1.484</b>	55.62	<b>0.376</b>	60.25	<b>0.615</b>	28.79	77.58%
hybrid-v-f	1.567	64.02	2.039	20.68	0.576	25.73	0.832	10.21	22.54%
hybrid-f-v	1.597	16.56	1.936	45.38	0.323	48.77	0.627	22.05	<b>84.17%</b>
baseline	1.606	<b>8.98</b>	2.249	<b>8.83</b>	0.647	<b>4.30</b>	0.855	<b>5.591</b>	9.55%

**Table 8.2:** Cross-comparison of the proposed grasping controller against a force-control baseline on four exemplary grasping tasks. The entries contain the averaged norm-values for the alignment error and interaction wrench per step over 10 runs for each object and method, as well as the total success-ratio. The best performing values are highlighted in bold.



**Figure 8.7:** Cartesian wrench profiles for the lightbulb removal task averaged over 10 trials for the vel-based and the frc-based strategies.

this error is minimized around this point, there remains a static error-residual in Cartesian torque.

## 8.5 Conclusion

In this chapter we proposed a novel grasping control strategy tailored to industrial, i.e., stiff position-controlled robot manipulators that exploits tactile feedback during contact with the object. In contrast to most related work, we did not focus on the optimal grasping pose detection from visual sensor-data, but rather the alignment correction that a robot encounters during initial contact with the object. For such cases, we claim that a robot is able to exploit the spatial distribution on pressure-sensitive sensor-readings on the gripper fingers. Given such a pressure distribution, we propose to apply geometric reasoning by numerically approximating the center of pressure (CoP) on each finger, such that the robot-pose can be evaluated against the actually desired CoP. Given this estimated alignment error, this chapter further outlined a novel grasping strategy that sends a Cartesian deviation command to the robot and thus either compensates for interaction wrenches along selected axes, or follows a constrained velocity-profile obtained from a model predictive control problem that uses

a locally linearized alignment error model. In order to evaluate this approach, this chapter provides an extended robot operating system (ROS)-driver for the WSG 50 finger gripper with optional digital sensor array (DSA) reading. The driver was empirically validated and could achieve a reliable communication rate of up to 5.49 Hz. In addition, the presented grasping strategies have been evaluated on exemplary grasping tasks with unknown pose uncertainty. Both approaches have shown promising results towards improving the skill-set of industrial robots in the context of versatile grasping, even though the predicted goal poses were distinctly incorrect.

## Future Work

Given the proposed robot control and gripper-interface, there is also room for further applications and extensions. A suggestion for future research lies in the possibility of improving the performance and success rate of the proposed grasping strategies. On the one hand side, the controller parameterization may profit from being updated online. While the parameters have been tuned manually in the scope of this chapter, updating these values online, e.g., by means of adaptive control, is a promising path for future research to easily adopt the framework to new and unforeseen scenarios and / or applications. Eventually, the presented approach could profit from incorporating recent data-driven approaches for robotic grasping or manipulation. Given the fact that the DSA-readings are closely related to the information obtained from a camera image, fusing visual and haptic data bears great potential on improving the overall success-rate and combine the advantages of these sensory inputs. A major benefit of such an approach lies in the independence of the robot hardware in use, making it applicable to arbitrary robot systems, without the need of installing force-sensitive compliant and often costly manipulators.



# 9

## Bayesian Optimization with Unknown Constraints in Graphical Skill-Models for Compliant Manipulation Tasks Using an Industrial Robot

### Chapter Abstract

This chapter focuses on learning manipulation skills from episodic reinforcement learning (RL) in unknown environments using industrial robot platforms. These platforms usually do not provide the required compliant control modalities to cope with unknown environments, e.g., force-sensitive contact-tooling. This requires to design a suitable controller, while also providing the ability of adapting the controller parameters from collected evidence online.

Thus, this chapter extends existing work on meta learning for graphical skill-formalisms. First, we outline how a hybrid force-velocity-controller can be applied to an industrial robot in order to design a graphical skill-formalism. This skill-formalism incorporates available task knowledge and thus allows for online episodic RL.

In contrast to existing work, we further propose to extend this skill-formalism by estimating the success-probability of the task to be learned by means of factor graphs. This method allows to assign samples to the individual factors, i.e., Gaussian processes (GPs) more efficiently and thus allows to improve the learning performance especially at early stages, where successful samples are usually only drawn in a sparse manner. Finally, we propose suitable constraint GP-models and acquisition functions to obtain new samples in order to optimize the information gain, while also accounting for the success-probability of the task.

We outline a specific application example on the task of inserting the tip of a screwdriver into a screwhead with an industrial robot, and evaluate our proposed extension against the state-of-the-art. The collected data outlines that our method allows artificial agents to obtain feasible samples faster than existing approaches while achieving a smaller regret value. This highlights the potential of our proposed work for future robotic applications.

*Remark:* A majority of this chapter has been published in [Gabler et al. \(2022a\)](#) and [Gabler and Wollherr \(2022\)](#).

## 9.1 Introduction

Robotic manipulators have been established as a key-component within industrial assembly lines for many years. However, applications of robotic systems beyond such well-defined and usually caged environments remain challenging. Simply reversing the process, i.e., asking a robot to disassemble a product that has been assembled by a robot manipulator in the past, uncovers the shortcomings of currently available (industrial) robot manipulators: the impacts of damage, temporal wear-offs or dirt most often diminish available model knowledge and thus do not allow an accurate perception of the environment. Rather than relying on the well-defined environment model, robot manipulators are required to account for this uncertainty and thus find a suitable control strategy to interact with the object in a compliant manner. While RL has found remarkable success in dealing with unknown environments, most of these approaches rely on a tremendous amount of data, which is usually costly to obtain, cf. [Levine et al. \(2016, 2015\)](#). In contrast, GPs allow acquiring data efficiently, but suffer from poor scaling w.r.t. state-size and dimension. Previous work has proposed to exploit existing model- and task-knowledge in order to reduce the parameter space from which a robot has to extract a suitable control policy.

Nonetheless, these approaches have usually been applied on (partially) compliant robots, where constraint violations, e.g., unforeseen contact impulses, can easily be compensated and are thus neglected. In the context of this article instead, a non-compliant – i.e., position-controlled – industrial robot is intended to solve manipulation tasks that require compliant robot behavior, such as screwdriver insertion given a noisy goal location. Therefore, this article outlines an episodic RL-scheme that uses Bayesian optimization with unknown constraints (BOC) to account for unsafe exploration samples during learning. In order to apply the proposed scheme on an industrial robot platform that does not provide the default interfaces for compliant controllers, such as a hybrid Cartesian force-velocity, we outline a slightly modified version of existing controllers. The resulting controller allows enabling force-/velocity profiles along selective axes, while using a high frequent internal position-controller as an alternative fallback. The hybrid nature of this controller allows a direct application of a graphical skill-formalism for meta learning in robotic manipulation from previous work. Thus, the state-complexity can be reduced to a level where the advantages of GPs outweigh their scaling deficiency. The core contribution of this article lies in the extension and adjustment of BOC to the outlined graphical skill-formalism such that safety constraints can not only be incorporated, but also directly added to the graphical skill-formalism. Specifically, we outline how the underlying graph structure can be extended to directly account for safety constraints and thus improve exploration behavior during early exploration stages, where a successful episode is unlikely.

Before sketching our contribution against related work below, we briefly outline of the terminology used in this chapter. Given the mathematical problem in Section 9.2, we shortly sketch the methodical background of our work in Section 9.3 and outline the technical insights of our approach in Section 9.4. Eventually, we outline a specific application example in Section 9.5 and present our experimental results collected with an industrial robot manipulator in Section 9.6 before concluding this chapter in Section 9.7.



### 9.1.1 Terminology

This section summarizes the terminology of this chapter. For brevity, we only highlight technical terms, which distinctly differ in their meaning across research fields.

- A (manipulation) *task* describes the challenge for a robot to reach a predefined goal-state, closely related to the definitions from automated planning (Nau et al., 2004). As this chapter focuses on episodic RL, the result of an episode is equal to the outcome of a task.
- A manipulation primitive (MP) defines a sub-step of a *task*. In contrast to automated planning, this chapter does not intend to plan a sequence of (feasible) MPs, but instead focuses on the parameterization of a predefined sequence of MPs. In contrast to hierarchical planning, we omit further hierarchical decompositions – e.g., methods (Nau et al., 2004) – such that a task can only be realized as a sequence of primitives.
- Using such MPs in order to solve a manipulation task directly leads to the introduction of the term of a *skill*. While a (robotic or manipulation) *skill* denotes the ability of a robot to achieve a task, we explicitly use the term (graphical) *skill-formalism* to denote a specific realization of sequential MPs to solve a manipulation task.
- Our approach seeks to increase the learning speed for episodic RL by limiting learning to a reduced parameter-space, which we denote as *meta learning* as used in existing work (Johannsmeier et al., 2019). It still has to be noted that this terminology is different to common terminologies such as meta-RL (Frans et al., 2018).
- Within episodic RL, a robot is usually asked to find an optimal parameter sample or a policy w.r.t. a numeric performance metric that is obtained at the end of a multi-step episode. In the scope of this chapter, we specifically focus on the former, i.e., a robot is asked to *sample* parameter values during the learning phase. Similarly to literature in Bayesian optimization, we often denote this sampling-process as the *acquisition of samples* (Rasmussen and Williams, 2006). Eventually, the performance metric is obtained at the very end of a successful trial episode.

### 9.1.2 Related Work

In the context of learning force-sensitive manipulation skills, a broad variety of research work has been presented in the last decade. Profiting from compliant controllers that were designed to mimic human motor skills (Vanderborght et al., 2013), the concept of adaptive robot skills has found interest way beyond adaptive control design. Thus, this section outlines the state-of-the-art across multiple research fields before setting the contribution of this chapter in relation to these works.

#### 9.1.2.1 Force-Adaptive Control for Unknown Surfaces or Objects

As covering all aspects of interacting with unknown surfaces, e.g., tactile sensing (Li et al., 2018b), is beyond the scope of this chapter, we refer to existing surveys (Li et al., 2020) and specifically summarize findings on learning force-adaptive manipulation skills.

In this context, the peg-in-hole problem is one of the most covered research challenges. Early work, such as [Gullapalli et al. \(1994, 1992\)](#) proposed to apply machine learning (ML), e.g., real-valued RL to learn a stochastic policy for the peg-in-hole task. The neural networks for the force controllers were trained by conducting a search guided by evaluative performance feedback.

Besides ML, many approaches have applied learning from demonstration to obtain suitable Cartesian space trajectories, cf. [Nemec et al. \(2013\)](#) or [Kramberger et al. \(2016\)](#) who adjust dynamic movement primitives conditioned on environmental characteristics using online inference. While first attempts have focused on adjusting the position of the robot end-effector directly, recent approaches have also investigated the possibility of replicating demonstrated motor-skills that also involve interaction wrenches ([Cho et al., 2020](#)) or compliant behavior ([Deniša et al., 2016](#), [Petric et al., 2018](#)).

Alternative work proposes adaptive controllers that adjust the gains of a Cartesian impedance controller as well as the current desired trajectory based on the collected interaction dynamics. [Li et al. \(2018c\)](#) for example evaluate observed error-dynamics, current pose, velocity and excited wrenches.

Even though, these works have achieved great results for modern and industrial robot manipulators in their application fields, they do not allow robots to autonomously explore and refine a task. While learning from demonstration always requires a demonstration to be given, adaptive controllers assume to have access to a desired state or trajectory. In addition, the majority of proposed controllers usually require high-frequent update rate on the robot joints, cf. [Scherzinger et al. \(2019b\)](#), [Stolt et al. \(2015, 2012\)](#) which usually is only accessible for the robot manufacturer. In contrast to this, we seek for a setup that can be deployed on off-the-shelf industrial robot manipulators.

A few years ago, the idea of end-to-end learning via means of deep RL-techniques has been studied thoroughly to combine the efforts of the former and the latter in a confined black-box system. In these studies, the concept of controlling the gains is omitted and instead replaced by a feed-forward torque policy that generates joint-torques from observed image data using a deep neural network (NN). [Levine et al. \(2016, 2015\)](#) use guided policy search that leverages the need for well-known models or demonstrations. Instead, the system learns contact-rich manipulation skills and trajectories through time-varying linear models which are unified into a single control policy. [Devin et al. \(2017\)](#) have tackled the issue of slow converging rates due to the enormous amount of required data by introducing distributed learning, where evidence is shared across robots, and the network structure allows to distinguish between task-specific and robot-specific modules. These models are then trained by means of mix-and-match modules, which can eventually solve new visual and non-visual tasks which were not included in the training data. The issue of low precision has been improved by [Inoue et al. \(2017\)](#), who evaluated the peg-in-hole task with a tight clearance.

Recently, the application of deep-RL has stepped back to use existing controllers and improve their performance by applying deep-NNs in addition, e.g., [Luo et al. \(2019\)](#) proposed to learn the interaction forces as Pfaffian constraints via a NN. [Beltran-Hernandez et al. \(2020\)](#) apply an admittance controller for a stiff position-controlled robot in joint space and apply RL via soft actor-critic ([Haarnoja et al., 2018](#)) to achieve a compliant robot behavior that successfully learns a peg-in-hole task by adjusting the gains of the admittance controller. Similarly, the feed-forward wrench for an insertion task is learned from human demonstrations ([Scherzinger](#)

et al., 2019a) using NNs and a Cartesian admittance controller tailored to industrial platforms (Scherzinger et al., 2017).

Aside of the aspect of meta-RL (Frans et al., 2018, Gupta et al., 2018), which investigates the idea of bridging data generated in simulations to physical platforms, the performance benefits and ability to learn almost arbitrarily complex tasks, existing methods for deep-RL still require tremendous amount of experimental data to be collected to achieve reliable performance.

### 9.1.2.2 Robot Skill Learning On Reduced Parameter-Spaces

The size of required data is directly subject to the size of the parameter-space that needs to be regressed. Thus, another promising line of research is given by decreasing the search space and problem complexity.

Recent research has proposed to use available expert knowledge rather than learning a skill from scratch. LaGrassa et al. (2020) propose to categorize the working space into regions where model knowledge is sufficient and into unknown regions, where a policy is obtained via deep RL. Johannsmeier et al. (2019) propose to incorporate expert knowledge in order to reduce the search space for adaptive manipulation skills by introducing MPs. Based on this, they showcase a peg-in-hole task where a robot adjusts the stiffness, and feed-forward interaction wrenches of a Cartesian impedance controller by means of Bayesian optimization (BO) and black-box optimization.

The application of such MPs also encouraged the application of deep-RL approaches. Zhang et al. (2021b) propose two RL approaches based on the principle of MPs, where the policy is represented by the feed-forward Cartesian wrench and the gains of a Cartesian impedance controller. Martín-Martín et al. (2019) similarly propose to learn the controller selection and parameterization during a peg-in-hole task. Hamaya et al. (2020) apply model-based RL via GP on a peg-in-hole task for an industrial position-controlled robot by attaching a compliant wrist to the robot end-effector, that compensates for perception inaccuracy. Mitsioni et al. (2021) instead propose to learn the environment dynamics from a NN in order to apply model predictive control, if the current state is classified as safe via a GP-classifier. Alt et al. (2021) also apply NNs via *differentiable shadow-programs* that employ the parameterization of robotic skills in the form of Cartesian poses and wrenches in order to achieve force-sensitive manipulation skills, even on industrial robots. They include the success-probability in the output of the NNs, in order to minimize the failure rate.

While these approaches have shown promising results by solely collecting experimental data within reasonable time, neither of those approaches include interaction constraints – e.g., maximum contact wrenches – during the acquisition or evaluation of new data-samples, nor allow the application of the presented results on an industrial platform without an additional compensation unit. As for the former, the majority of research projects have applied BOC to account for safety critical or unknown system constraints during learning, we continue with a dedicated overview of research in this field.

### 9.1.2.3 Bayesian Optimization with Unknown Constraints for Robotics

Within robotic applications BO has been promising to achieve online RL due to effective acquisition of new samples (Calandra et al., 2016, Deisenroth et al., 2015), that is still used within robotic research applications (Demir et al., 2021).

In the context of BOC safe RL methods have been proposed that estimate safe or feasible regions of the parameter-space into account to allow for safe exploration, cf. Berkenkamp et al. (2016a,b), Sui et al. (2015) or Baumann et al. (2021).

Similarly, Englert and Toussaint (2016) proposed the probability of improvement with a boundary uncertainty criterion (PIBU) acquisition function that encourages exploration in the boundaries of safe states. Their approach was further evaluated on generalizing small demonstration data autonomously in Englert and Toussaint (2018) as well as on force-adaptive manipulation tasks by Drieß et al. (2017). A similar acquisition function has been proposed by Rakicevic and Kormushev (2019) even though they do not approximate the success as a GP.

Approaches like Wang et al. (2021), who use GPs to regress the success of an atomic planning skill from data, have further shown that BOC is well suited to regress high-level, i.e., task-planning constraints from data. While they approximated this success-probability as a constraint with a predefined lower bound 0, Marco et al. (2021) outlined a constraint-aware robot learning method based on BOC that allows to improve sampling even if no successful sample is available yet. Recent practical application examples of BOC are found in König et al. (2020), Stenger et al. (2022), Yang et al. (2022).

While these approaches have achieved promising results within small-scale (robot) learning problems, they suffer from poor scaling properties as GPs require to use the covariance matrix for prediction and acquisition of new data-samples, and which grows exponentially in the state-space of the underlying problem. While various work has focused on finding proper approximation methods to leverage this problem, we propose that within a robotic context, it is preferable to explicitly incorporate structural knowledge whenever possible. To conclude this overview of the state-of-the-art, we shortly summarize the contribution of this chapter in relation to the work stated above.

### 9.1.3 Contribution

This chapter introduces a novel episodic RL-scheme for compliant manipulation tasks tailored to industrial robots. In order to allow for compliant manipulation tasks, the control interfaces of an industrial robot are adjusted to follow a Cartesian hybrid force-velocity controller (Craig and Raibert, 1979, Khatib and Burdick, 1986). By exploiting the hybrid nature of this controller and available expert knowledge, a complex manipulation task can be reformulated into graphical skill-formalisms – i.e., a sequence of simplified MPs — from existing work. Eventually, we outline an extension of these graphical skill-formalisms by taking parameter constraints and success-probabilities at each sub-step into account. This improves learning especially at early stages and allows to refine the individual sub-steps of a robotic manipulation task even when no successful episode could have been observed yet. Furthermore, we define suitable BOC-models to estimate the success-probability of each MP as well as the

overall task, as well as the outline of suitable acquisition functions that allow collecting data efficiently during learning.

## 9.2 Problem Formulation

The mathematical problem tackled in this chapter is the optimization of an unknown objective function  $\mathcal{J}(\boldsymbol{\xi})$  w.r.t. meta parameter vector  $\boldsymbol{\xi}$  subject to unknown constraints  $\mathbf{g}$

$$\begin{aligned} \min \mathcal{J}(\boldsymbol{\xi}) \quad & \boldsymbol{\xi} \in \mathbb{R}^m \\ \text{s.t.} \quad & g_i(\boldsymbol{\xi}) \leq c_i, \forall i \in [1, |\mathbf{c}|] \end{aligned} \quad (9.1)$$

specifically tailored to robotic applications. In here, the objective  $\mathcal{J}(\boldsymbol{\xi})$  describes the performance metric of a task, while a finite set of constraints  $\mathbf{g}(\boldsymbol{\xi}) \leq \mathbf{c}$  defines a safe subset of the meta parameter-space  $\boldsymbol{\xi}$ . In the context of this chapter, this function mapping  $\mathcal{J}(\boldsymbol{\xi})$ , as well as the constraints – i.e.,  $\mathbf{g}$  and  $\mathbf{c}$  – are regressed from data by means of episodic RL. In contrast to most RL-approaches, where the environment is assumed to be Markovian, episodic RL needs to execute a multi-step exploration before obtaining a feedback, that can be used to update the current model(s). In the scope of this chapter, an episode is given as a manipulation task. which can be either evaluated in simulation or directly on a robot platform. In the remainder of this chapter, we mainly focus on the direct application on the latter. Similar to related work in this area (Marco et al., 2021), we assume that the feedback of an episode is expected to be given in the form of

$$\mathcal{J}_{\text{spl}}, \mathbf{g}_{\text{spl}}, \mathbf{s}_{\text{spl}} \leftarrow \begin{cases} \mathcal{J}(\boldsymbol{\xi}), \mathbf{g}(\boldsymbol{\xi}), \top & \text{iff } g_i(\boldsymbol{\xi}) \leq c_i, \forall i \in [1, |\mathbf{c}|] \\ \infty, \quad \infty, \quad \perp & \text{else} \end{cases}, \quad (9.2)$$

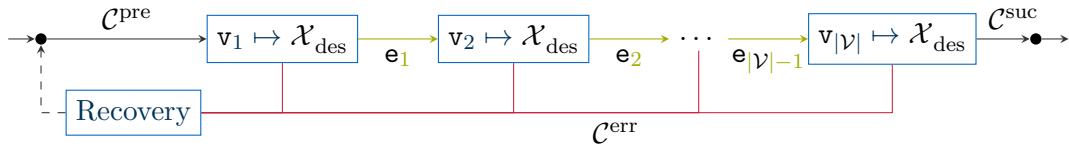
as the current performance sample  $\mathcal{J}_{\text{spl}}$ , and the constraint and success-return vectors  $\mathbf{g}_{\text{spl}}, \mathbf{s}_{\text{spl}} \in \mathbb{R}^{|\mathbf{c}|}$ . Therefore, a major challenge lies in handling episodes where *infeasible* / *unsafe* parameters have been selected, and neither information about  $\mathcal{J}$  nor the constraint metric is gained. It is often costly to select and evaluate new samples within robotic applications. GP-regression has shown great potential in ML and robotics, if only a handful of samples should be evaluated. Thus,

$$\begin{aligned} \mathcal{J}(\boldsymbol{\xi}) & \leftarrow \check{\mathcal{F}}_{\mathcal{J}}(\boldsymbol{\xi} | \mathcal{D}_{\mathcal{J}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathcal{J}}, \boldsymbol{\Sigma}_{\mathcal{J}}) \\ g_i(\boldsymbol{\xi}) \leq c_i & \leftarrow \check{\mathcal{F}}_{g_i}(\boldsymbol{\xi} | \mathcal{D}_{g_i}) \sim \mathcal{N}(\boldsymbol{\mu}_{g_i}, \boldsymbol{\Sigma}_{g_i}) \quad \forall i \in [1, |\mathbf{c}|] \end{aligned} \quad (9.3)$$

approximate the objective  $\mathcal{J}$  and constraints  $\mathbf{g}_i$  via GPs using collected empirical data  $\mathcal{D}_{\mathcal{J}} = \{\boldsymbol{\xi}, \mathcal{J}(\boldsymbol{\xi})\}$  and  $\mathcal{D}_{g_i} = \{\boldsymbol{\xi}, (g_i(\boldsymbol{\xi}), \{\top, \perp\})\}$ . Finally, the optimal guess for (9.1) can be obtained by minimizing the posterior of  $\check{\mathcal{F}}_{\mathcal{J}}(\boldsymbol{\xi})$ :

$$\boldsymbol{\xi}^* \leftarrow \arg \min_{\boldsymbol{\xi}} \mathbb{E}_{\check{\mathcal{F}}_{\mathcal{J}}} \left[ \check{\mathcal{F}}_{\mathcal{J}}(\boldsymbol{\xi}) \prod_{i=1}^{|\mathbf{c}|} \mathbb{P} \left[ \check{\mathcal{F}}_{g_i}(\boldsymbol{\xi}) \right] \right], \quad (9.4)$$

weighted by the success-probability of  $\boldsymbol{\xi}$  given as the joint-probability over all constraints. Thus, (9.4) does not only optimize the main task-objective, but also accounts for the probability of violating imposed constraints. This directly allows to optimize the performance of an unknown manipulation task for robotic systems, while accounting for constraints, such as limited interaction wrenches during contact-tooling.



**Figure 9.1:** Schematic skill-formalism for manipulation tasks as presented in [Johannsmeier et al. \(2019\)](#). Each MP – i.e., node  $v_i$  – defines the current set-values for the underlying controller, e.g., desired wrench or velocity, as well the current meta parameters that define the performance of the skill, e.g., controller parameterization. Eventually, the *Recovery* node intends to steer the robot to the initial state whenever an error occurs.

## 9.3 Preliminaries and Background

Before outlining our approach in detail, we give a brief introduction into the graphical skill-formalisms from [Johannsmeier et al. \(2019\)](#) and the BOC approach from [Marco et al. \(2021\)](#) and [Englert and Toussaint \(2016\)](#), which we use as a baseline comparison in our experimental evaluation.

### 9.3.1 Meta Learning for Robotic Systems Using Graphical Skill-Formalisms

Within robotic tasks, the hyper-parameter space is usually large due to the degrees of freedom in  $SE(3)$  or the configuration space of the robot. Therefore, [Johannsmeier et al. \(2019\)](#) proposed to model tasks in fine-grained Moore finite-state automaton (FSA), according to the schematic shown in Figure 9.1. The vertices  $\mathcal{V}$  of the FSA-graph  $\mathbf{G}$  define MPs as atomic primitive tasks. In these FSAs, the output alphabet is defined by the meta parameters  $\xi$  and desired set-values, e.g.,  $\mathbf{x}_{des}$ , that are sent to the robot at each MP, denoted as the dedicated space  $\mathcal{X}_{des}$  in Figure 9.1. Therefore, the more task-knowledge can be exploited for each MP, the smaller the space of the resulting meta parameter per node.

Eventually, the manipulation skill is further defined by a set of constraints that define the start- and end-constraints, as well as any time constraints that the robot shall never violate. This brings in the benefit of exploiting available object knowledge, while also providing a skill-formalism that is closely related to those of automated task planning ([Nau et al., 2004](#)). In fact, these constraints are closely related to autonomous planning and first-order-logic, where planning primitives are often described by a set of *pre-conditions* and *effects*. In the context of concurrent planning, this is also extended to any time constraints, that must not be violated while the task primitive is executed. This results in a skill-representation as shown in Figure 9.1, where the task-constraints are defined as deterministic mapping functions  $\mathcal{C} := \mathcal{X} \mapsto \{\perp, \top\}$ , which map the state-space of the robot to a Boolean return value. In particular, individual manipulation skills are defined by:

- Initialization-constraints  $\mathcal{C}^{pre}$ , or pre-conditions. They define the initialization of the task. In general  $\mathcal{C}^{pre}$  is given as a set of constraints, that only evaluates to  $\top$ , if **all** conditions evaluate to  $\top$ , i.e., if  $\mathbf{s}_0$  denotes the initial state of the robot, then  $c(\mathbf{s}_0) \mapsto \top, \forall c \in \mathcal{C}^{pre}$  has to hold.

- Success-constraints  $\mathcal{C}^{\text{suc}}$ , or termination-conditions. They evaluate if the manipulation skill has been executed successfully. This terminates the overall FSA shown in Figure 9.1 and requires as well **all** conditions to evaluate to  $\top$  i.e., if  $\mathbf{s}_{T_{\text{max}}}$  denotes the final state of the robot in the manipulation skill, then  $c(\mathbf{s}_{T_{\text{max}}}) \mapsto \top, \forall c \in \mathcal{C}^{\text{suc}}$  has to hold.
- Safety and performance constraints  $\mathcal{C}^{\text{err}}$ , or error conditions. They evaluate if the current MP has violated any constraints, e.g., timeouts, accuracy violations that may exceed information provided by a task planner. In contrast to  $\mathcal{C}^{\text{pre}}$  and  $\mathcal{C}^{\text{suc}}$ , the error constraint-set  $\mathcal{C}^{\text{err}}$  evaluates to  $\top$  if **any** condition is violated at any time, i.e., if  $\mathbf{s}_t$  denotes the state of the robot at any time during the manipulation skill, then  $\exists c \in \mathcal{C}^{\text{err}} : c(\mathbf{s}_t) \mapsto \top$  has to be fulfilled. Furthermore, the robot enters a recovery node, in which the robot tries to reach the initial state to initiate a new trial-episode – as emphasized by the dashed line in Figure 9.1.

In the context of the graphical skill-formalism from [Johannsmeier et al. \(2019\)](#),  $\mathcal{C}^{\text{pre}}$  are defined by the adjacency matrix of the graph and the success-constraint from the predecessor-node, i.e., if a node raises the success-constraint, there is a unique successor-node, whose precondition holds by design.

### 9.3.2 Bayesian Optimization with Unknown Constraints

Within BO an unknown function or system is regressed from data as a stochastic process. A common model is a GP, which is defined as a collection of random variables, namely joint normally distributed functions over any subset of these variables. They are fully described by their second-order statistics, i.e., a prior mean and a covariance-kernel-function  $\mathfrak{k}(\boldsymbol{\xi}, \boldsymbol{\xi}')$ , that encodes prior function properties or assumptions.<sup>1</sup> A key-benefit of stochastic processes is their ability to draw samples efficiently. This strongly depends on the choice of the acquisition function  $\mathcal{F}_{\text{aqu}}$ , which usually intends to maximize the information gain for the estimated posterior  $y_{\text{spl}}$ . Famous examples are the expected improvement (EI) and expected improvement with constraints (EIC)

$$\mathcal{F}_{\text{aquEI}}(\boldsymbol{\xi}, \mathcal{D}) = \mathbb{E}_{y_{\text{spl}} \sim \mathcal{N}_{\mathcal{F}}(\mu, \sigma | \boldsymbol{\xi})} [\max(y_{\text{spl}} - \mathcal{F}^{\circledast}, 0)], \quad (9.5)$$

$$\mathcal{F}_{\text{aquEIC}}(\boldsymbol{\xi}, \mathcal{D}) = \mathbb{E}_{y_{\text{spl}} \sim \mathcal{N}_{\mathcal{F}}(\mu, \sigma | \boldsymbol{\xi})} \left[ \max(y_{\text{spl}} - \mathcal{F}^{\circledast}, 0) \prod_{j=0}^G \mathbb{P}[\mathbf{g}_j(\boldsymbol{\xi}) \leq \mathbf{c}_j] \right], \quad (9.6)$$

where the probability of improvement (PI) is maximized

$$\text{PI}_{\text{GP}(\mathcal{F})}(\boldsymbol{\xi}) = \Phi \left( \frac{\mu_{\text{GP}(\mathcal{F})}^{\boldsymbol{\xi}} - \mathcal{F}_{\mathcal{D}}^{\circledast}}{\sigma_{\text{GP}(\mathcal{F})}^{\boldsymbol{\xi}}} \right). \quad (9.7)$$

In here  $\Phi$  denotes the normal cumulative distribution function (CDF), while  $\mathcal{F}_{\mathcal{D}}^{\circledast}$  represents the best output sample in the data set  $\mathcal{D}$ , which serves as the lower bound for the improvement. The mean  $\mu_{\text{GP}(\mathcal{F})}^{\boldsymbol{\xi}}$  and variance  $\sigma_{\text{GP}(\mathcal{F})}^{\boldsymbol{\xi}}$  are obtained as the posterior of the GP at new sample candidates  $\boldsymbol{\xi}$ . While modeling the task-performance via a GP is commonly applied in BOC, regressing a discriminative success function is non-trivial. In [Drieß et al. \(2017\)](#), [Englert and](#)

<sup>1</sup>For more information about GPs and GP-classification, we refer to [Rasmussen and Williams \(2006\)](#).

Toussaint (2016, 2018) GP-classification with a sigmoid-function to classify the output of a latent GP is proposed. Given this, the authors propose a constrained sensitive acquisition function, which they denote as PIBU

$$\mathcal{F}_{\text{aquPIBU}}(\boldsymbol{\xi}, \mathcal{D}) = \begin{cases} \text{PI}_{\text{GP}(\mathcal{F})}(\boldsymbol{\xi}) & \check{\mathcal{F}}_g(\boldsymbol{\xi}) > 0 \\ \sigma_{\text{GP}(\check{\mathcal{F}}_g)}(\boldsymbol{\xi}) & \check{\mathcal{F}}_g(\boldsymbol{\xi}) \mapsto 0 \end{cases}, \quad (9.8)$$

that uses the PI in admissible regions of the parameter-space, and the variance  $\sigma$  of the latent GP in the boundary regions to encourage a safe exploration. They further use a constant negative mean prior for the latent GP to limit sampling to the boundary regions of the safe parameter-space. In contrast to this, Marco et al. (2021) propose to use a constraint-aware GP-model that allows to use EIC, which they denote as Gaussian process for classified regression (GPCR). GPCR allows updates even if no successful constraint sample has been drawn yet, based on the environmental feedback in (9.2). Further, Marco et al. (2021) propose to regress the constraint thresholds  $\mathbf{c}_j$  directly from data. Thus having  $N_{\text{spl}} \leq |\mathcal{D}|$  successful samples, the likelihood is defined as

$$\mathbb{P}[\mathcal{D} | \mathbf{g}_j] = \prod_{j=0}^{N_{\text{spl}}} \mathcal{F}_H(\mathbf{c}_j - \mathbf{g}_j) \mathcal{N}(\mathbf{g}_j, \sigma_{\text{noise}}^2) \prod_{j=N_{\text{spl}}+1}^{|\mathcal{D}|} \mathcal{F}_H(\mathbf{g}_j - \mathbf{c}), \quad (9.9)$$

where  $\mathcal{F}_H$  denotes the Heaviside function. Using a zero-mean Gaussian prior, the posterior is given as

$$\mathbb{P}[\mathbf{g} | \mathcal{D}] = \mathcal{N}(\mathbf{g} | \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) \prod_{i=0}^{N_{\text{spl}}} \mathcal{F}_H(\mathbf{c}_j - \mathbf{g}_j) \prod_{j=N_{\text{spl}}+1}^{|\mathcal{D}|} \mathcal{F}_H(\mathbf{g}_j - \mathbf{c}_j) \approx \mathcal{N}(\mathbf{g} | \boldsymbol{\mu}_{\text{EP}}, \boldsymbol{\Sigma}_{\text{EP}}), \quad (9.10)$$

where the Gaussian distribution  $\mathcal{N}(\mathbf{g} | \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$  is obtained by the multivariate Gaussian from the observation noise and the observation samples. As the Heaviside functions in (9.10) do not allow obtaining an analytic solution for (9.10), the authors propose to use a variational approximation, namely expectation propagation (EP), such that the predictive distribution at unobserved samples  $\boldsymbol{\xi}'$  is obtained via a Gaussian distribution defined by mean and variance

$$\begin{aligned} \mu_{g_j}(\boldsymbol{\xi}') &= \mathbf{k}_{\mathfrak{X}}(\boldsymbol{\xi}')^\top \mathbf{R}^{-1} \boldsymbol{\mu}_{\text{EP}} \\ \sigma_{g_j}(\boldsymbol{\xi}') &= \mathbf{k}(\boldsymbol{\xi}', \boldsymbol{\xi}') - \mathbf{k}_{\mathfrak{X}}(\boldsymbol{\xi}')^\top \mathbf{R}^{-1} \left( \mathbb{1}^{|\mathcal{D}| \times |\mathcal{D}|} - \boldsymbol{\Sigma}_{\text{EP}} \mathbf{R}^{-1} \right) \mathbf{k}_{\mathfrak{X}}(\boldsymbol{\xi}'), \end{aligned} \quad (9.11)$$

where  $\mathfrak{X}$  denotes observed parameter-samples in  $\mathcal{D}$ . The success-probability is then given as

$$\mathbb{P}[\mathbf{g}(\boldsymbol{\xi}) \leq \mathbf{c}(\boldsymbol{\xi}')] = \prod_{i=0}^{|\mathcal{C}|} \Phi \left( \frac{\mathbf{c}_j - \mu_{g_j}(\boldsymbol{\xi}')}{\sigma_{g_j}(\boldsymbol{\xi}')} \right). \quad (9.12)$$

## 9.4 Technical Approach

In order to allow online RL to be applied from a handful of exploration samples, it is favorable to exploit available knowledge and thus decrease the overall meta parameter-space of the observed system. As mentioned before, we thus extend the concept of modeling robotic tasks



as *skill-graphs* from [Johannsmeier et al. \(2019\)](#) to allow compliant manipulation tasks to be tuned online. In contrast to preliminary work, we outline how a stiff position-controlled industrial robot platform can be controlled in order to allow for compliant robot behavior. Building upon this, we emphasize how a graphical skill-formalism can exploit the structure of the presented controller, such that the controller parameters can be adjusted online. As crash constraints are critical, if a stiff robot is asked to interact with unknown objects, we conclude our technical contributions by not only outlining how the structure of the skill-graph can be further exploited to simplify the BOC-RL algorithm, but also proposing suitable BOC-models and acquisition functions in order to improve the overall learning performance.

### 9.4.1 Compliant Controller Design for an Industrial Robot

In the context of this chapter, we use a COMAU robot <sup>2</sup>. While this robot prohibits the control of the motor torques or impedance-based controller interfaces, it allows to control the position of the end-effector  $\mathbf{x}$  of the robot via an external client in the form of a Cartesian deviation relative to the current end-effector pose, such that the controlled system simplifies to

$$\mathbf{x}_{t+1} := \mathbf{x}_t + \delta_{\mathbf{x}} \approx \mathbf{x}_t + \mathbf{u}_{\dot{\mathbf{x}},\text{des}} \delta_t. \quad (9.13)$$

where  $\delta_{\mathbf{x}}$  forms the control command being sent to the robot. As the robot runs at a real-time safe, constant update-rate  $\delta_t$ , the Cartesian deviation command  $\mathbf{u}_{\dot{\mathbf{x}},\text{des}}$  can also be used to command a feed-forward Cartesian velocity command to the robot. In order to achieve a hybrid force-velocity control policy for the robot system, this feed-forward end-effector velocity follows to a hybrid Cartesian force-velocity controller ([Khatib and Burdick, 1986](#))

$$\begin{aligned} \mathbf{u}_{\dot{\mathbf{x}},\text{des}} &:= \mathbf{S}^{\text{vel}} \mathbf{R} \dot{\mathbf{x}}_{\text{ccdes}} + \mathbf{S}^{\text{frc}} \mathbf{R} \mathbf{K}_{\text{P}}^{\text{frc}} (\mathbf{F}_{\text{des}} - \mathbf{F}) \\ &= \mathbf{S}^{\text{vel}} \mathbf{R} \dot{\mathbf{x}}_{\text{ccmax}} + \mathbf{S}^{\text{frc}} \mathbf{R} \mathbf{K}_{\text{P}}^{\text{frc}} (\mathbf{F}_{\text{des}} - \mathbf{F}), \end{aligned} \quad (9.14)$$

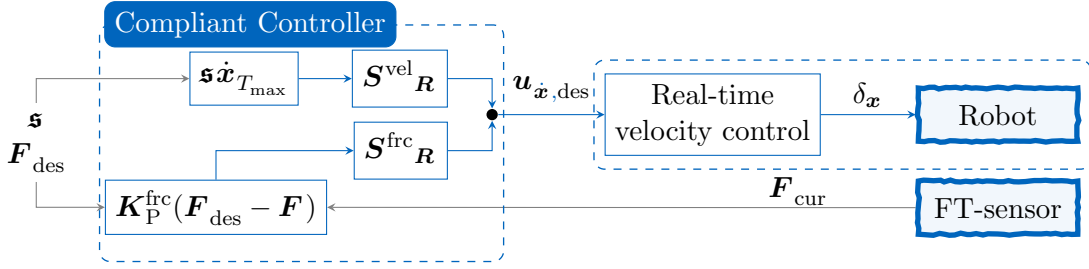
where  $\mathbf{s} \mapsto [0, 1]^6$  is a scaling vector given the maximum end-effector velocity  $\dot{\mathbf{x}}_{\text{max}}$  and  $\mathbf{K}_{\text{P}}^{\text{frc}}$  is a positive definite proportional control gain matrix. The selection matrices  $\mathbf{S}^{\text{frc}} \mathbf{R}$  and  $\mathbf{S}^{\text{vel}} \mathbf{R}$  in (9.14) are given as

$$\begin{aligned} \mathbf{S}^{\text{frc}} \mathbf{R} &:= \begin{bmatrix} \mathbf{R}_{\text{ct}}^{\text{ba}} \text{diag}(s_{1:3}^{\text{frc}}) \mathbf{R}_{\text{ba}}^{\text{ct}} & \mathbb{0}^{3 \times 3} \\ \mathbb{0}^{3 \times 3} & \mathbf{R}_{\text{ct}}^{\text{ba}} \text{diag}(s_{3:6}^{\text{frc}}) \mathbf{R}_{\text{ba}}^{\text{ct}} \end{bmatrix}, \\ \mathbf{S}^{\text{vel}} \mathbf{R} &:= \begin{bmatrix} \mathbf{R}_{\text{ct}}^{\text{ba}} \text{diag}(s_{1:3}^{\text{vel}}) \mathbf{R}_{\text{ba}}^{\text{ct}} & \mathbb{0}^{3 \times 3} \\ \mathbb{0}^{3 \times 3} & \mathbf{R}_{\text{ct}}^{\text{ba}} \text{diag}(s_{3:6}^{\text{vel}}) \mathbf{R}_{\text{ba}}^{\text{ct}} \end{bmatrix}, \end{aligned} \quad (9.15)$$

for position and force control.

Thus, a Cartesian velocity as well as the force-profile  $\mathbf{F}$  can be followed along selective axes. The presented controller differs from classic hybrid force-position control by the fact that disabling the force-control along an axis does not directly result in position control. If  $s_i^{\text{vel}} = s_i^{\text{frc}} = 0$ , the robot automatically holds the current position according to the internal control loop and (9.13). Nonetheless, for a correct decoupling of the individual control policies, the selection matrices need to hold  $s_i^{\text{vel}} s_i^{\text{frc}} = 0$ . The final control architecture, as visualized in Figure 9.2, is well suited for a graphical skill-formalism from [Johannsmeier et al. \(2019\)](#), as it can directly exploit hybrid policies along selective axes.

<sup>2</sup>The extension to arbitrary robots is subject to the internal robot control and dynamics. As the methods in this chapter are dynamically independent, this extension is left for future work.



**Figure 9.2:** Schematic overview of a hybrid force-velocity controller (Craig and Raibert, 1979, Khatib and Burdick, 1986) using the Cartesian deviation control interface of an industrial COMAU robot. In here, the selection matrices  $S^{\text{vel}}_R$  and  $S^{\text{frc}}_R$  activate velocity- and force-control modalities along selective axes using a scaled feed-forward velocity profile  $s \dot{x}_{T_{\max}}$  and a proportional force-controller with gain-matrix  $K_P^{\text{frc}}$ , Cartesian FT-readings  $F$  and the desired wrench  $F_{\text{des}}$ . Eventually, the Cartesian velocity is emulated on the COMAU robot by using the Cartesian deviation command interface  $\delta_x$ .

### 9.4.2 Applying Bayesian Optimization with Unknown Constraints on Graphical Skill-Representations

Even though a skill-graph can decrease the search space complexity, the resulting space may still suffer from the curse of dimensionality. Furthermore, collecting data from actual experiments is at risk of gathering various incomplete and thus useless data-samples. In the context of episodic RL, one (successful) graph iteration represents a single episode. This requires all steps to succeed for a useful return value. Thus, we outline how the BOC-problem from Section 9.2 can be reformulated to exploit available model knowledge in this skill-graph to improve sampling and learning. We assume that the feedback from (9.2) can be obtained at each node of the skill-graph and that each parameter in  $\xi$  is bounded. Given a graphical skill-representation as in Figure 9.1, represented by MP-nodes  $\mathcal{V}$  and transitions  $\mathcal{E}$ , the objective  $\mathcal{J}$  can be decomposed into the sum of all nodes, while the dedicated constraints need to be fulfilled at each step

$$\begin{aligned} \mathcal{J}(\xi) &:= \sum_{v \in \mathcal{V}} \check{\mathcal{J}}_{\mathcal{J}^v}(\xi_v) \quad \xi_v \in \mathbb{R}^n, n \leq m \\ \text{s.t. } &g_v(\xi_v) \leq c_v \quad \forall v \in \mathcal{V}. \end{aligned} \quad (9.16)$$

Thus, (9.4) results in

$$\xi^* \leftarrow \arg \min_{\xi} \Lambda_{\xi}^{\mathcal{V}} \sum_{v \in \mathcal{V}} \mathbb{E}_{\check{\mathcal{J}}_{\mathcal{J}^v}} \left[ \check{\mathcal{J}}_{\mathcal{J}^v}(\xi^v) \right], \quad (9.17)$$

where  $\Lambda_{\xi}^{\mathcal{V}}$  denotes the joint success-probability over all MP-nodes. While preliminary work (Johannsmeier et al., 2019) has shown that the application of the summation in (9.17) improves learning speed and quality, we claim that it is furthermore beneficial to exploit the structure of the MP-graph in order to regress  $\Lambda_{\xi}^{\mathcal{V}}$  and thus to design suitable acquisition functions from it. Due to the structure of the graph and the underlying BOC-problem, the objective and success-probability are conditionally independent. This allows to outline specific graph-based representations for the success-probability of  $\Lambda_{\xi}^{\mathcal{V}}$ , which we outline below.

### 9.4.2.1 Naive-Bayes Approach

In order to approximate  $\Lambda_{\xi}^{\mathcal{V}}$  from the underlying MP-graph, a commonly applicable solution is given as a Naive-Bayes approach, i.e., assuming conditional independence for all nodes. This is usually valid due to the condition-checking within the MP-graph from Section 9.3.1. Recalling the constraint in (9.16), the success-probability of each MP-node is subject to

$$\Gamma_{\xi}^{\mathbf{v}} = \prod_{j=1}^{|\mathbf{g}|} \begin{cases} \mathbb{P} \left[ \check{\mathcal{F}}_{\mathbf{g}, j}^{\mathbf{v}}(\xi^{\mathbf{v}}) \right] & \text{if } \mathbf{active}(\mathbf{g}, j, \mathbf{v}) \\ 1 & \text{else} \end{cases}, \quad (9.18)$$

where  $\mathbf{active}(\mathbf{g}, j, \mathbf{v}) \mapsto \{\top, \perp\}$  encodes if the constraint is active in the current node or not. This allows to directly encode the structure of the graph – i.e., available task-knowledge – in the success-probability of each node. Given the sequential structure of an MP-graph, the overall success-probability results in

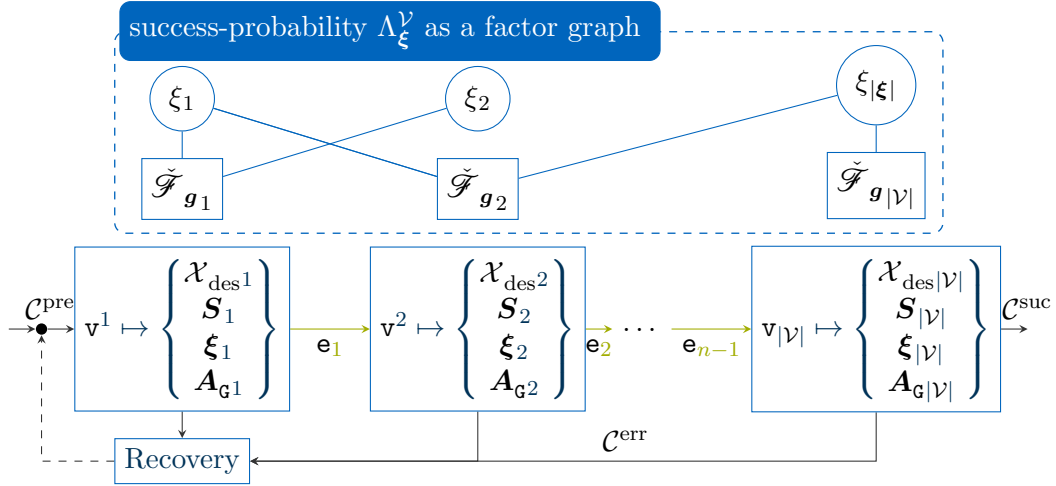
$$\Lambda_{\xi}^{\mathcal{V}} = \prod_{\mathbf{v} \in \mathcal{V}} \Gamma_{\xi}^{\mathbf{v}}, \quad (9.19)$$

while the success-probability of each intermediate node is obtained as the product of individual terms  $\Gamma_{\xi}^{\mathbf{v}}$  from the initial to the current node. In order to estimate  $\Lambda_{\xi}^{\mathcal{V}}$  from data, we thus regress each active success-constraint per node as an individual GP. These GPs are independent and use the success or failure as well as the constraint-metric of the current subset of the meta parameter at each MP node. This results in at most  $|\mathcal{V}||\mathbf{g}|$  GPs for the overall task. Nonetheless, the Naive-Bayes approach suffers from two disadvantages. First, the success-function for the current node may depend on the full vector  $\xi$  instead of  $\xi^{\mathbf{v}}$  in (9.18). This contradicts the assumption of conditional independence and limits the applicability of the Naive-Bayes approach to tasks, where not only the task but also the constraints can be modeled individually for each MP. Given the structure of the MP-graph, namely the existence of error-constraints at each transition, the Naive-Bayes approach is still applicable to a broad variety of tasks, but may not allow adding constraints that affect the choice of parameters across multiple MP-nodes. Second, in case an episode fails at a dedicated node, no labels can be added to the subsequent nodes as each node is handled fully independently. While this still allows to collect samples earlier during the learning stage, the number of samples needed is expected to increase until successful samples can be obtained. In order to diminish these effects, we propose to model the success-probability by a specialized factor graph in the next section.

### 9.4.2.2 Modeling the Success-Function as a Factor Graph

Besides the Naive-Bayes approach it is also possible to directly impose the structural task-knowledge that results from the graph-structure. Namely, we propose to model the overall success-probability as a factor graph representation [Kschischang et al. \(2001\)](#) for the task-constraints, where the scalar elements of  $\xi$  form the variables, and the constraints from (9.1) form the factors, cf. Figure 9.3.

Having obtained the general factor graph for a manipulation skill, this graph is fully described by an adjacency matrix  $\mathbf{A}_{\mathbf{G}}$ , where element  $[\mathbf{A}_{\mathbf{G}}]_{(i,j)} \mapsto 1$  denotes an existing edge from  $i$  to  $j$ . Within factor graphs an edge is only connecting a variable with a factor-node, such that



**Figure 9.3:** Graphical skill-formalism with an additional factor graph representation for the task success-probability. The individual node-parameters  $\xi^v$  denote the meta parameters for each node, while  $\mathcal{X}_{\text{des}^v}$  denote the set-values and  $\mathbf{S}_v$  denotes the selection-matrices  $\mathbf{S}_R^{\text{fic}}$  and  $\mathbf{S}_R^{\text{vel}}$  for the controller from Figure 9.2. The factor graph denotes the overall task completion probability, while the adjacency matrix  $\mathbf{A}_{G^i}$  for each vertex defines the *active* sub-graph for each vertex, which define the success-probability for the current node  $\Gamma_{\xi}^v$ . Again, the *Recovery* node intends to steer the robot to the initial robot state whenever an error occurs in order to initiate a new episode.

it is sufficient to denote the adjacency matrix as  $\mathbf{A}_G \in \mathbb{R}^{|\xi| \times |c|}$ . Therefore, columns denote individual constraints, and the rows define the sub-set of  $\xi$  for each individual constraint. Consequently, the success-probability results in

$$\Lambda_{\xi}^V = \prod_{j=0}^{|c|} \min \left( \sum_{i=0}^{|\xi|} [\mathbf{A}_G]_{(i,j)}, 1 \right) \mathbb{P} \left[ \check{\mathcal{F}}_g^c(\xi_{c_j}) \right] \quad \text{where } [\mathbf{A}_G]_{(i,j)} \mapsto 1 \quad \forall \xi_i \in \xi_{c_j}. \quad (9.20)$$

If for example, only the active success-constraint per MP-node is introduced and each constraint has the same input dimension, the Naive-Bayes approach is reconstructed. In contrast to the Naive-Bayes approach each MP-constraint can depend on arbitrary subsets of  $\xi$ . In order to fully exploit the structure of the MP-graph, we propose to embed the underlying success-probability for each vertex in the skill-graph. This can be directly achieved by extending the current set-values commanded to the robot system by an MP-specific adjacency matrix  $\mathbf{A}_{G^i}$ . The success-constraint at each MP  $\Gamma_{\xi}^v$  can then be obtained by replacing  $\mathbf{A}_G$  with  $\mathbf{A}_{G^i}$  in (9.20). As a result, samples can be added to each constraint metric dependent on the current progress within the MP-graph. Thus, if a skill fails at a specific node, the samples obtained until said notes can be added to the data set as successful, while the samples for the failed node can be assigned to the current and subsequent MPs success-estimators.

### 9.4.3 BOC-Model and Acquisition Function

Given the extended MP-graph, the objective of acquiring samples efficiently is again subject to the choice of the acquisition function and underlying GP-model. Recalling Section 9.3.2, a key-benefit of the method from Marco et al. (2021) is the ability to push the probability mass above the current threshold estimate, which allows to gain more knowledge from failed

samples. Nonetheless, this model relies on approximating the posterior due to nonlinear components in (9.10). Instead, we propose to induce artificial data-points and fit GPs on this artificial data set instead. The algorithmic skeleton is sketched in Algorithm 9.1, where we again assume to have safe and failed data-samples in the data-buffer  $\mathcal{D}$  for each constraint.

---

**Algorithm 9.1:** Induce artificial data-points to fit GP on data sets with failed samples

---

**input :**  $\mathbf{p}_{\text{safe}}, \mathbf{p}_{\text{fail}}, \mathcal{D}, \nu, \kappa_{\text{spl}}, \zeta_{\text{spl}}$   
**output:**  $\tilde{\mathcal{F}}_{\mathbf{g}_i}$

```

1 ConstraintFit:
2    $\tilde{\mathcal{F}}_{\mathbf{g}}^{\text{safe}} \leftarrow \text{ParameterFit}(\xi_{\text{safe}}, \mathbf{g}_{\text{safe}})$             $\triangleright$  fit valid constraint samples
   /* estimate threshold for safe GP                                     */
3    $\xi_{\text{spl}} \sim \xi_{\text{safe}} + \kappa_{\text{spl}} \mathcal{N}(0|\xi|, \mathbb{1}|\xi| \times |\xi|)$ 
4    $\hat{c}_i \leftarrow \Phi^{-1}(\mathbf{p}_{\text{safe}}) \sigma_{\mathbf{g}_i}^{\text{safe}}(\xi_{\text{spl}}) + \max(\mathbf{g}_{\text{safe}}, \mu_{\mathbf{g}_i}^{\text{safe}}(\xi_{\text{spl}}))$ 
5    $\hat{\mathbf{g}}_{j,\text{fail}}^{\text{safe}} \leftarrow \max(\Phi^{-1}(\mathbf{p}_{\text{fail}}), \zeta_{\text{spl}}) \sigma_{\mathbf{g}_i}^{\text{safe}}(\xi_{\text{fail}})$     $\triangleright$  approximate failed data (safe GP)
6    $\mathcal{D}_{\text{art}} \leftarrow \{\xi_{\text{safe}}, \mathbf{g}_{\text{safe}} - \hat{c}_i\} \cup \{\xi_{j,\text{fail}}, \hat{\mathbf{g}}_{j,\text{fail}}^{\text{safe}}\}$     $\triangleright$  generate artificial data set
7    $\tilde{\mathcal{F}}_{\mathbf{g}}^{\text{fail}} \leftarrow \text{ParameterFit}(\mathcal{D}_{\text{art}})$             $\triangleright$  fit artificial data set
8    $\hat{\mathbf{g}}_{i,\text{fail}}^{\text{fail}} \leftarrow \max(\Phi^{-1}(\mathbf{p}_{\text{fail}}), \zeta_{\text{spl}}) \sigma_{\mathbf{g}_i}^{\text{fail}}(\xi_{\text{fail}})$     $\triangleright$  approximate failed data (virtual GP)
9    $\hat{\mathbf{g}}_i \leftarrow (1 - \nu) \hat{\mathbf{g}}_{j,\text{fail}}^{\text{safe}} + \nu \hat{\mathbf{g}}_{j,\text{fail}}^{\text{fail}}$             $\triangleright$  Polyak average approximated data
10   $\mathcal{D}_{\text{art}} \leftarrow \{\xi_{\text{safe}}, \mathbf{g}_{\text{safe}} - \hat{c}_i\} \cup \{\xi_{i,\text{fail}}, \hat{\mathbf{g}}_i\}$     $\triangleright$  generate artificial data set
11   $\tilde{\mathcal{F}}_{\mathbf{g}_i} \leftarrow \text{ParameterFit}(\mathcal{D}_{\text{art}})$             $\triangleright$  fit GP to artificial data set
```

---

We propose fitting a GP into the safe data set first. Given this safe distribution, we propose to estimate the constraint-value  $\hat{c}_i$ . This can be achieved by evaluating the posterior at the safe input samples and applying the inverse CDF and a predefined probability threshold  $\mathbf{p}_{\text{safe}}$  that should be held for legal samples. As the variance is usually small in the near distance of collected evidence, mean-free Gaussian noise is added on the existing samples. Taking the maximum of the predictive mean and the collected samples from the safe data set, the predictive variance can be used to calculate the value of the constraint from the inverse CDF. Using the estimated constraint-value  $\hat{c}_i$ , the predictive variance at the infeasible data-samples can be used to estimate the predictive mean-value that would result in a posterior infeasibility probability threshold  $\mathbf{p}_{\text{fail}}$ . In this estimation, we treat zero as the decision threshold for the current constraint-GP and limit the inverse CDF to a lower bound  $\zeta_{\text{spl}} \in \mathbb{R}^+$ . As the safe GP does not contain any data-sample within the unsafe parameter-space, the variance of the posterior is expected to be large. Thus, we propose to apply a model-fit with the artificial data set and repeat the process from above to obtain new virtual output values. As the decision threshold is set to zero, the safe data-samples are shifted by the current constraint estimate. Within our implementation, we also normalize the collected safe samples, but omitted this in Algorithm 9.1 for brevity. The predictive posterior distribution of this artificial data set will usually impose a conservative variance given the added data-sample support. Thus, the final artificial value is obtained by a Polyak average of the two estimated posterior values. Given this, another parameter fit returns the final constraint GP. In order to embed ambiguity over unobserved parameters, the GPs use a zero-mean prior, which is equal to the constraint threshold of the virtual GP. In case there is no feasible data set found, Algorithm 9.1 shortens. Instead of fitting existing data, the constraint is explicitly set to zero and the artificial data

is set to  $\min(\mathbf{p}_{\text{safe}}, \mathbf{p}_{\text{fail}})$ . Eventually, a parameter fit is obtained to get an estimate of the constraint GP. For the Naive-Bayes approach, each MP and for the factor graph approach each factor are finally realized by a constrained GP according to Algorithm 9.1. Given the structure of the factor graph and the dedicated skill, we propose to use a sequential form of (9.6). For the Naive-Bayes approach this results in applying (9.6) at all nodes

$$\mathcal{F}_{\text{aquEIC,G}}(\boldsymbol{\xi}, \mathcal{D}_{\text{G}}) = \Lambda_{\boldsymbol{\xi}}^{\mathcal{V}} \sum_{\mathbf{v} \in \mathcal{V}} \mathbb{E}_{y_{\text{spl}} \sim \mathcal{N}_{\mathcal{F}_{\mathbf{v}}}(\mu, \sigma | \boldsymbol{\xi}^{\mathbf{v}})} \left[ \max(y_{\text{spl}} - \mathcal{F}_{\mathbf{v}\mathcal{D}}^{\otimes}, 0) \Gamma_{\boldsymbol{\xi}}^{\mathbf{v}} \right], \quad (9.21)$$

and weigh the sum of acquisition functions by the overall success-probability to encourage acquisition of samples that are expected to succeed in the overall task. Due to the linear structure and the conditional independence of each node,  $\Gamma_{\boldsymbol{\xi}}^{\mathbf{v}}$  is directly obtained by  $\check{\mathcal{F}}_{\mathbf{g}_i}$  according to Algorithm 9.1 given that each node can be represented by a dedicated constraint-metric. For the factor graph version, we do not assume conditional independence for the success-probabilities. Instead, the adjacency-graph of each node is used to calculate  $\Gamma_{\boldsymbol{\xi}}^{\mathbf{v}}$  in (9.21)

$$\Gamma_{\boldsymbol{\xi}}^{\mathbf{v}} = \prod_{i=1}^{|\mathcal{c}|} \begin{cases} \check{\mathcal{F}}_{\mathbf{g}_i} & \text{if } \exists j \in [1, |\boldsymbol{\xi}|] : [\mathbf{A}_{\mathbf{G}_{\mathbf{v}}}]_{(\cdot, j)} \mapsto 1 \\ 1 & \text{else} \end{cases}, \quad (9.22)$$

using  $\check{\mathcal{F}}_{\mathbf{g}_i}$  according to Algorithm 9.1. Eventually, the best sample estimate is given by optimizing over the best-guess of each MP-objective estimate at each MP and setting all samples with a success-probability below  $\mathbf{p}_{\text{safe}}$  to  $\mathcal{F}_{\mathcal{D}}^{\ominus}$ . Thus, the EI in (9.21) is replaced by the objective of each node, and  $\Gamma_{\boldsymbol{\xi}}^{\mathbf{v}}$  is set to 1 for feasible estimates. For infeasible samples we assume the worst observed objective for the current objective estimate, such that the optimal parameter estimate is obtained as:

$$\boldsymbol{\xi}^* \leftarrow \arg \min_{\boldsymbol{\xi}} \sum_{\mathbf{v} \in \mathcal{V}} \begin{cases} \mathcal{F}_{\mathbf{v}}(\boldsymbol{\xi}_{\mathbf{v}}) & \text{if } \Gamma_{\boldsymbol{\xi}}^{\mathbf{v}} \geq \mathbf{p}_{\text{safe}} \\ \mathcal{F}_{\mathcal{D}}^{\ominus} & \text{else} \end{cases}. \quad (9.23)$$

#### 9.4.4 Exploit Conditional Dependencies for Collected Samples

The final BOC-algorithm for the proposed online RL approach is sketched in Algorithm 9.2. In contrast to learning the full parameterization of the task, the sequential skill-graph receives the additional feedback, which node was explored last in Line 4. This information is crucial to assign samples correctly for the success- and constraint data-buffers in Line 7 and Line 6. While the assignment for the MP-node objectives is straightforward, i.e., only valid samples for explored nodes are assigned to the data sets, invalid samples may also be assigned even though the related MP or constraint-factor has not yet been evaluated. The necessary condition for a sample to be added to the dedicated data set is that at least one scalar component has to be explored or visited. Due to the sequential procedure of the skill-graph, the mapping of the last explored node  $\mathbf{v}_i$  to the dedicated data sets is deterministic and known beforehand. For the factor graph representation, it is further possible to add artificial samples to the data set if a conditional dependence exists. If a subsequent node contains scalar components that have not yet been explored or visited, while other scalar components have been explored before a failure is detected, artificial data can be added to the data set of said constraint GP. Thus, the unexplored sample can be exchanged by drawing samples from a Sobol-sequence (Sobol', 1967) or linearly distributed data. Using the adjacencies-matrices of the factor graph, the

---

**Algorithm 9.2:** Overall BOC-algorithm

---

**input :**  $P_{\text{safe}}, P_{\text{fail}}, \mathcal{D}, \nu, \kappa_{\text{spl}}, \zeta_{\text{spl}}, N_{\text{eps}}$

**output:**  $\hat{\xi}^*$

```

1  $\mathcal{D}_{\check{\mathcal{F}}_{\mathcal{F}}}^{\nu} \leftarrow \emptyset, \mathcal{D}_{\check{\mathcal{F}}_{\mathcal{G}}}^{\nu} \leftarrow \emptyset \forall \nu \in \mathcal{V}$  ▷ init data sets
2 for  $k = 1$  to  $N_{\text{eps}}$  do
3    $\xi_{\text{spl}} \leftarrow \mathcal{F}_{\text{aquEIC,G}}(\xi, \mathcal{D}_{\mathcal{G}})$  ▷ cf. Section 9.4.3
4    $\mathcal{F}_{\text{spl}\mathcal{V}}, \mathbf{g}_{\text{spl}}, \mathbf{s}_{\text{spl}}, \mathbf{v}_i \leftarrow \text{evaluate}(\xi_{\text{spl}})$  ▷ evaluate sample
   /* assign environment feedback to data sets */
5   for  $\nu \in \mathcal{V}$  do
6      $\mathcal{D}_{\check{\mathcal{F}}_{\mathcal{G}}}^{\nu} \leftarrow \text{AddConstraint}(\mathcal{D}_{\check{\mathcal{F}}_{\mathcal{G}}}^{\nu}, \mathbf{g}_{\text{spl}}, \mathbf{s}_{\text{spl}}, \mathbf{v}_i)$ 
7      $\mathcal{D}_{\check{\mathcal{F}}_{\mathcal{F}}}^{\nu} \leftarrow \text{AddObjective}(\mathcal{D}_{\check{\mathcal{F}}_{\mathcal{F}}}^{\nu}, \mathcal{F}_{\text{spl}\mathcal{V}}, \mathbf{s}_{\text{spl}}, \mathbf{v}_i)$ 
8      $\check{\mathcal{F}}_{\mathcal{F}\nu} \leftarrow \text{ParameterFit}(\mathcal{D}_{\check{\mathcal{F}}_{\mathcal{F}}}^{\nu})$  ▷ fit objective-GP
   /* apply Algorithm 9.1 for all constraints */
9   for  $k = 1$  to  $|c|$  do
10     $\check{\mathcal{F}}_{\mathcal{G}i} \leftarrow \text{ConstraintFit}(P_{\text{safe}}, P_{\text{fail}}, \mathcal{D}, \nu, \kappa_{\text{spl}}, \zeta_{\text{spl}})$ 
11  $\hat{\xi}^* \leftarrow \arg \min_{\xi} \sum_{\nu \in \mathcal{V}} \check{\mathcal{F}}_{\mathcal{F}}^{\nu}(\xi^{\nu}) \mathcal{F}_{\text{H}}(\Gamma_{\xi}^{\nu}(\xi^{\nu}) \geq P_{\text{safe}})$ 
   ▷ get optimal parameter-guess

```

---

visited parameters can be obtained by  $\text{diag}(\mathbf{A}_{\mathbf{G}\nu_i} \mathbf{A}_{\mathbf{G}\nu_i}^{\top}) \geq 1$  for each MP and thus, for the partially explored MP-graph as  $\sum_{i=1}^{\nu_i} \text{diag}(\mathbf{A}_{\mathbf{G}\nu_i} \mathbf{A}_{\mathbf{G}\nu_i}^{\top}) \geq 1$ . Similarly, the samples that can be replaced by artificial samples are obtained as

$$\left( \text{diag}(\mathbf{A}_{\mathbf{G}\nu_i} \mathbf{A}_{\mathbf{G}\nu_i}^{\top}) - \sum_{j=1}^{\nu_i} \text{diag}(\mathbf{A}_{\mathbf{G}\nu_j} \mathbf{A}_{\mathbf{G}\nu_j}^{\top}) \right) \geq 1. \quad (9.24)$$

Before outlining an application example, we shortly outline the theoretical improvements of our approach, i.e., the scaling w.r.t. size of the meta parameter space.

### 9.4.5 Complexity Analysis

In this section, we analyze the proposed method in terms of scaling w.r.t. size of the meta parameter space. It has to be noted that we do not emphasize improving GP-scaling against big-data, for which there is existing work (e.g., [Ambikasaran et al. \(2016\)](#)) available. For brevity, we denote the dimension of the original learning problem as  $n_{\xi}$ , i.e.,  $\xi \in \mathbb{R}^{n_{\xi}}$  and denote the largest dimension of all nodes within a graphical skill-formalism as  $m_{\xi, \mathcal{G}}$ , i.e.,  $\xi_i \in \mathbb{R}^{m_{\xi, \mathcal{G}}}$ .

**Definition 9.1: Valid MP-Graph**

An MP-graph is a valid representation for (9.4), if the following constraints are given

- the graph has no absorbing nodes.
- there exists a finite path from the start- to the end-node
- the underlying objective can be represented by a convex composition of sub-objectives

**Definition 9.2: Feasible MP-Graph**

An MP-graph is a feasible representation for (9.4), if the following constraints are given

- the meta parameter space for each node of the MP-graph is bounded by  $m_{\xi, \mathbf{g}} < n_{\xi}$
- the meta parameter space for all constraints is bounded by  $m_{\xi, \mathbf{g}} < n_{\xi}$
- the number of active constraints per node is bounded by  $|\mathcal{C}|$

**Claim 9.1**

Regressing a general robot task (9.1) as a stochastic representation (9.4) via GPs according to Definition 9.2, the resulting complexity can be reduced from  $\mathcal{O}(n_{\xi}^3)$  to  $\mathcal{O}(\max(|\mathcal{C}|, 1)|\mathcal{V}|m_{\xi, \mathbf{g}}^3)$  by modeling the task as an MP-graph, using the Naive-Bayes approach.

*Proof.* Recalling (9.16), the objective function of the algorithm scales linearly with the numbers of nodes within the graph. According to Definition 9.2, the meta parameter space of each MP-node is bounded, thus

$$\mathcal{O}(\mathcal{J}) = \mathcal{O}\left(|\mathcal{V}|m_{\xi, \mathbf{g}}^3\right), \quad (9.25)$$

This proves claim 9.1 if there are no success-constraints active, i.e.,  $|\mathcal{C}| = 0$ . In case there is a success-constraint active, the upper bound of the complexity is defined by the complexity of the success-probability as it may contain feasible and infeasible data-samples. For the Naive-Bayes approach,  $|\mathcal{C}|$  constraints have to be evaluated at  $|\mathcal{V}|$  nodes via GPs, for which the meta parameter space is bounded by  $m_{\xi, \mathbf{g}}$ , thus resulting in an overall complexity of  $\mathcal{O}\left(|\mathcal{V}||\mathcal{C}|m_{\xi, \mathbf{g}}^3\right)$ .  $\square$

**Claim 9.2**

Using the factor graph method and a task-representation as outlined in claim 9.1, the complexity from  $\mathcal{O}(n_{\xi}^3)$  can be reduced to  $\mathcal{O}\left(|\mathcal{C}|m_{\xi, \mathbf{g}}^3\right)$  if there are active constraints, i.e.,  $|\mathcal{C}| \geq 1$ .



*Proof.* In contrast to the Naive-Bayes approach, the system complexity grows linearly w.r.t. the number of constraints  $|\mathcal{C}|$ . While the complexity follows (9.25), the success-probability for  $|\mathcal{C}| \leq 1$  and thus the overall system complexity results in  $\mathcal{O}\left(|\mathcal{C}|m_{\xi, \mathbf{g}}^3\right)$ .  $\square$

Eventually, it has to be noted that adding artificial data adds data to the data sets of MP-nodes or factors which decreases scaling behavior. Nonetheless, it has to be noted that adding artificial data is not mandatory and intends to add support during early exploration when data sets are usually small. Therefore, we omitted the possibility of adding artificial data in the complexity analysis above.

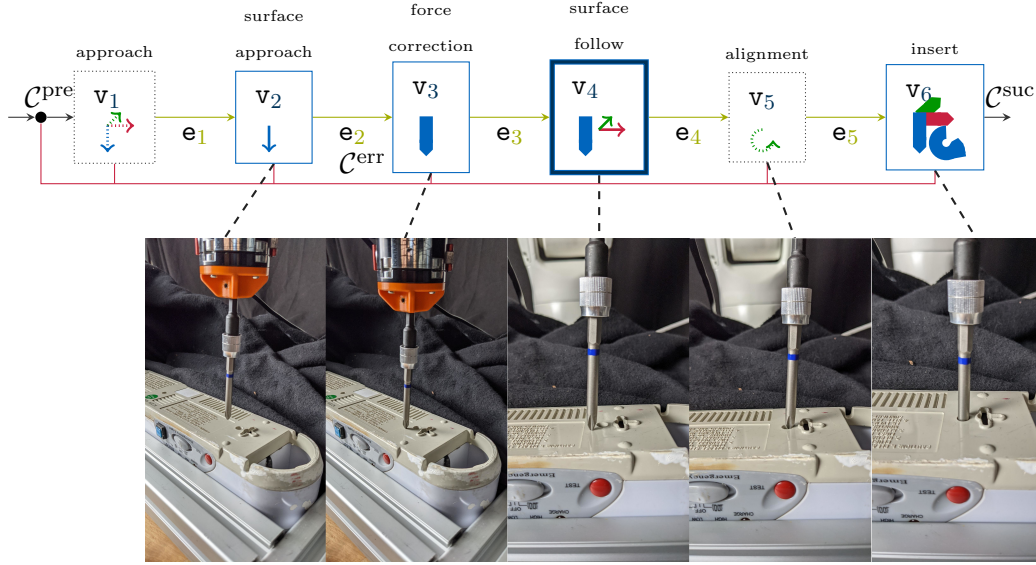
## 9.5 Application Example - Screw Insertion

In this section, we outline an application example for the proposed manipulation learning framework, that uses the proposed controller from Section 9.4.1: the insertion of a screwdriver into a screw-head. Even though the environment suffers from high uncertainty, there exists available pre-knowledge that can be incorporated to reduce the problem size and thus use a skill-graph according to Section 9.4.2. While the previous sections have outlined the generic modalities of our method, this section intends to present an application example, that is eventually used to evaluate our approach. The main motivation of constructing a graphical skill-formalism is the reduction of the actual search-space for the episodic RL task, i.e., the dimension of the parameter-vector  $\xi$ . Therefore, we assume the following constraints to be given:

- the screw is accessible by the robot end-effector, i.e., there exists a robot configuration that does not result in a (self-)collision of the robot with any surrounding object when the screwdriver is inserted. Furthermore, the robot configuration is singularity-free as this would not allow using the underlying Cartesian robot controller reliably.
- in case the position of the screw-head is subject to uncertainty, the condition above needs to be guaranteed for the full range of the uncertain region.<sup>3</sup>
- the robot is equipped with a screwdriver and the transformation from the screwdriver-pin, i.e., control frame  $\mathbf{ct}$ , to the robot end-effector, i.e.,  ${}^{\text{ee}}\mathbf{T}_{\mathbf{ct}}$  is known.
- the type of screw matches the pin of the screwdriver of the robot.

Given these assumptions, motion planning or pose optimization against infeasible states or collisions can be omitted in the following. Instead, the framework focuses on finding a correct parameterization of the controller presented in Section 9.4.1. In approaches such as end-to-end-learning the problem could be represented as an RL-problem, with sparse rewards that penalize any constraint violations and add positive feedback for a successful task. While this allows to learn such a skill from visual data on arbitrary robot platforms, first a supervised learning method is required to classify task success or constraint violation, and infeasible amount of data needs to be collected from experimental trial and error a supervised learning method is required, which will violate feasible time-budgets. In contrast, directly applying

<sup>3</sup>This condition also includes, that a Cartesian path applied within this region will not result in a collision of the robot or a singularity, since the actual control input is commanded directly in task- and not in joint-space.



**Figure 9.4:** Schematic screw insertion skill as an MP-graph. Each vertex shows the dedicated control-direction, and thus the selection matrix from (9.14), where bold arrows represent force-control and thin lines velocity control. Straight arrows in each MP denote translation w.r.t. axes  ${}^{ct}e_x$ ,  ${}^{ct}e_y$  and  ${}^{ct}e_z$ . Circular arrows denote a rotation along the dedicated axis. Parametric nodes are highlighted by a solid edge, while a bold edge denotes a hybrid control policy.

a GP-policy would result in extremely large data sets, which will in return affect the evaluation or acquisition calculation. Thus, we propose to exploit the available expert knowledge and construct a skill-graph formalism similar to [Johannsmeier et al. \(2019\)](#). First, the normal vector  $\mathbf{n}$  of the surface and the screw<sup>4</sup> are approximately known from vision. Furthermore, we assume that an expert has set the desired contact wrench-magnitudes beforehand. Similarly, a designer has chosen a tilting angle for the robot end-effector to ease the contact tooling.

Given this, we outline the resulting skill-graph as visualized in Figure 9.4 from left to right. In this skill-graph, we explicitly denote the output alphabet, i.e., the desired set-values per node as well as the MP-parameters  $\xi_i$ . For the sake of brevity, only non-zero values are explicitly mentioned, e.g., if not explicitly noted, all values of  $\mathbf{S}^{\text{vel}}_{\mathbf{R}}$  and  $\mathbf{S}^{\text{frc}}_{\mathbf{R}}$  are set to 0. The first node  $v_1$  is non-parametric and describes the *approach*-MP, where the robot is asked to steer the tip of the tool and hover above the surface. As obtaining a suitable trajectory is beyond the scope of the presented method, we refer e.g., to [Bari et al. \(2021\)](#) for further insights. The success of this node, thus advancing the graph to  $v_2$  is evaluated via

$$\mathcal{C}_{v_1}^{\text{suc}} := \|\mathbf{x}_{\text{des}} - \mathbf{x}_{\text{cur}}\|_2 \leq \zeta_{\text{pos}}. \quad (9.26)$$

The second node  $v_2$  – *approach-surface* – contains the first parametric node and describes the motion of the robot towards the surface until contact with the environment is established. Thus, a constant velocity along the negative surface-normal is applied, such that the set-values

<sup>4</sup>We set these normal-vectors as constant within this evaluation, but it is possible to update the normal-vectors online if needed.

for the robot-controller for this node are given as

$$\begin{aligned}\mathcal{X}_{\text{des}} &= \left\{ \dot{\mathbf{x}}_{\text{des}} \leftarrow -\mathfrak{s}_{\text{pos}} \mathbf{v}_{\text{max}} \mathbf{n} \right\} \\ \boldsymbol{\xi}_2 &= \mathfrak{s}_{\text{pos}} \\ \mathbf{S}_{\mathbf{R}}^{\text{frc}} &= \text{diag}(0, 0, 1, 0, 0, 0)\end{aligned}\quad . \quad (9.27)$$

The success of this MP is given as an established contact with the environment, which is defined as

$$\mathcal{C}_{\mathbf{v}_2}^{\text{suc}} := \mu_{\mathbf{F}} > \sigma_{\mathbf{F}}, \quad (9.28)$$

where the variance  $\sigma_{\mathbf{F}}$  denotes the approximated sensor-noise and  $\mu_{\mathbf{F}}$  the filtered force-torque (FT)-sensor readings over a sliding window of fixed size  $N_{\text{FT}}$ . This node further checks against the maximum allowed contact force  $\mathbf{F}_{\text{max}}$

$$\mathcal{C}_{\mathbf{v}_2}^{\text{err}} := \left| \mu_{\mathbf{F}} - \sigma_{\mathbf{F}} \right| \geq |\mathbf{F}_{\text{max}}|, \quad (9.29)$$

to raise a failure of the skill. The subsequent node  $\mathbf{v}_3$  – *force correction* – corrects the encountered force-impulse stemming from the contact at the end of the previous MP. Thus, the controller switches from the feed-forward velocity command to force-control along the normal-vector of the surface:

$$\begin{aligned}\mathcal{X}_{\text{des}} &= \left\{ \mathbf{F}_{\text{des}} \leftarrow -\mathbf{F}_{\text{des}} \mathbf{n}, [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(z)} \right\} \\ \boldsymbol{\xi}_3 &= [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(z)} \\ \mathbf{S}_{\mathbf{F}_{\text{des}}}^{\mathbf{R}} &= \text{diag}(0, 0, 1, 0, 0, 0)\end{aligned}\quad . \quad (9.30)$$

The success of this MP is evaluated by the accumulated force-error for a fixed window-size  $N_{\text{cont}}$ :

$$\mathcal{C}_{\mathbf{v}_3}^{\text{suc}} := \sum_{i=1}^{N_{\text{cont}}} \left( \mathbf{F}_{\text{cur}, t-1}^f - \mathbf{F}_{\text{des}} \right) \mathbf{n} \leq \zeta_{\mathbf{F}}, \quad (9.31)$$

using only the force-measurement  $\mathbf{F}_{\text{cur}}^f$  of the wrench  $\mathbf{F}$ . The error-constraint also checks against the force-threshold in (9.29), but also evaluates

$$\mathcal{C}_{\mathbf{v}_3}^{\text{err}} := \mathcal{C}_{\mathbf{v}_2}^{\text{err}} \wedge \left( \mu_{\mathbf{F}} - \sigma_{\mathbf{F}} \right)^{\top} [\mathbf{n}, \mathbf{0}^3] \geq 0.0, \quad (9.32)$$

to detect contact-loss with the environment as an error-constraint. During the next node  $\mathbf{v}_4$  – *surface search* – the robot steers along the surface of the object in order to detect the screw. This implies a hybrid force-velocity profile, where the robot seeks to regulate the normal force with the surface while following a velocity profile along the surface. Using a parameterized velocity profile  $\dot{\mathbf{x}}_{\text{des}, \kappa_{xy}}$ , the output of  $\mathbf{v}_4$  is given as

$$\begin{aligned}\mathcal{X}_{\text{des}} &= \left\{ \dot{\mathbf{x}}_{\text{des}} \leftarrow \dot{\mathbf{x}}_{\text{des}, \kappa_{xy}}, \mathbf{F}_{\text{des}} \leftarrow -\mathbf{F}_{\text{des}} \mathbf{n}, [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(z)} \right\} \\ \boldsymbol{\xi}_4 &= \left\{ \kappa_{xy}, [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(z)} \right\} \\ \mathbf{S}_{\mathbf{x}_{\text{des}}}^{\mathbf{R}} &= \text{diag}(1, 1, 0, 0, 0, 0) \\ \mathbf{S}_{\mathbf{F}_{\text{des}}}^{\mathbf{R}} &= \text{diag}(0, 0, 1, 0, 0, 0)\end{aligned}\quad . \quad (9.33)$$

The success of this MP is evaluated via the force impulse encountered in the current motion direction, i.e.,  $\frac{\dot{\mathbf{x}}_{\text{cur}}}{\|\dot{\mathbf{x}}_{\text{cur}}\|_2}$  and the perpendicular torque.

$$\mathcal{C}_{v_4}^{\text{suc}} := \left| \mathbf{F}_{\text{cur}}^f \frac{\dot{\mathbf{x}}_{\text{cur}}}{\|\dot{\mathbf{x}}_{\text{cur}}\|_2} \right| + \left| \mathbf{F}_{\text{cur}}^\tau \left( \mathbf{n} \times \frac{\dot{\mathbf{x}}_{\text{cur}}}{\|\dot{\mathbf{x}}_{\text{cur}}\|_2} \right) \right| \geq \zeta_{\text{impls}}. \quad (9.34)$$

For the error-constraint, this node applies (9.29), (9.32) and also checks against the robot position

$$\mathcal{C}_{v_4}^{\text{err}} := \mathcal{C}_{v_2}^{\text{err}} \wedge \mathcal{C}_{v_3}^{\text{err}} \wedge \|\mathbf{p}_{\text{cur}} - \mathbf{p}_{v_3}\|_2 \geq \zeta_{\text{dspl}}, \quad (9.35)$$

where  $\mathbf{p}_{\text{cur}}$  denotes the translational component of the tool-tip of the robot, while  $\mathbf{p}_{v_3}$  represents the tool-tip position at the end of node  $v_3$ . The node  $v_5$  – *alignment* – is non-parametric and optional. It denotes the alignment of the tool-tip to be perpendicular to the surface. Thus, if the initial tilting angle is set to zero, this step is omitted. The success-constraint is identical to  $v_1$ , but the translation component is ignored. The final node  $v_6$  – *insert* – describes the insertion MP, that applies a Cartesian wrench control. Thus, the MP is defined as

$$\begin{aligned} \mathcal{X}_{\text{des}} &= \left\{ \mathbf{F}_{\text{des}} \leftarrow -\mathbf{F}_{\text{des}} \mathbf{n}, \text{diag} \left( [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(xy)}, [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(xy)}, [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(z)}, 0, 0, [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(\psi)}, \right) \right\} \\ \boldsymbol{\xi}_6 &= \left\{ [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(xy)}, [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(z)}, [\mathbf{K}_{\text{P}}^{\text{frc}}]_{(\psi)} \right\} \\ \mathbf{S}_{\mathbf{F}_{\text{des}}}^R &= \text{diag}(1, 1, 1, 0, 0, 1) \end{aligned} \quad (9.36)$$

While the error-constraint is identical to  $v_3$  – i.e.,  $\mathcal{C}_{v_6}^{\text{err}} := \mathcal{C}_{v_3}^{\text{err}}$  – the success-constraint is checked via comparing the displacement along the normal-vector

$$\begin{aligned} \mathcal{C}_{v_6}^{\text{suc}} &:= \left\| (\mathbf{p}_{\text{cur}} - \mathbf{p}_{v_5}) (\mathbb{1}^3 - \mathbf{n}) \right\|_2 \geq \zeta_{\text{dspl}} \wedge \\ &\quad ((\mathbf{p}_{\text{cur}} - \mathbf{p}_{v_5}) \mathbf{n})^\top ((\mathbf{p}_{\text{cur}} - \mathbf{p}_{v_5}) \mathbf{n}) \geq \zeta_{\text{insrt, min}} \wedge, \\ &\quad ((\mathbf{p}_{\text{cur}} - \mathbf{p}_{v_5}) \mathbf{n})^\top ((\mathbf{p}_{\text{cur}} - \mathbf{p}_{v_5}) \mathbf{n}) \leq \zeta_{\text{insrt, max}} \end{aligned} \quad (9.37)$$

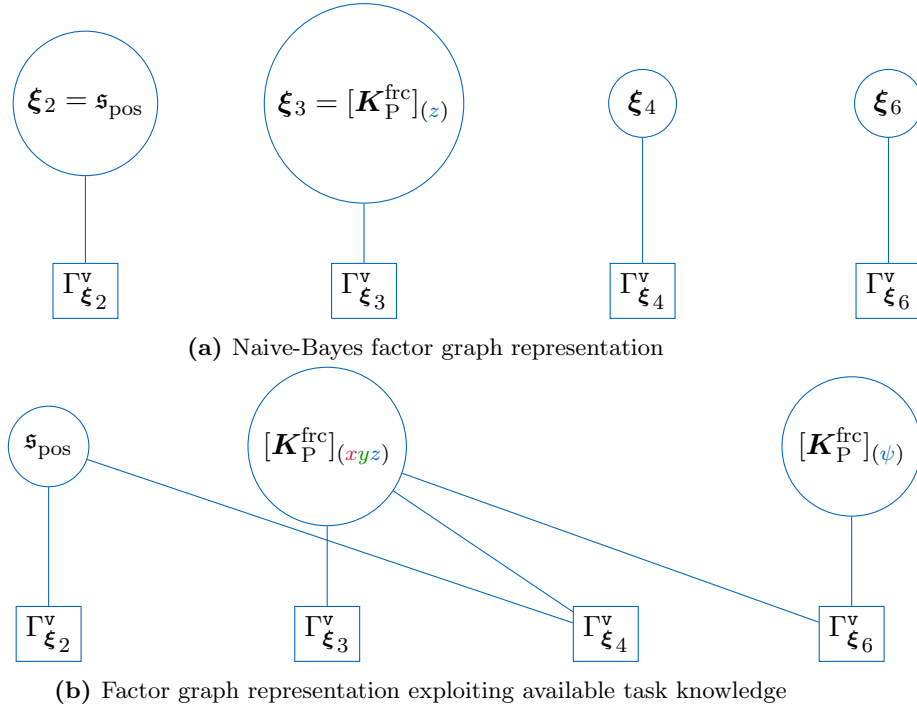
where  $\mathbf{p}_{v_5}$  again denotes the tool-tip position when the current node is initiated.

Having introduced the general MP-graph, we now outline how the success-constraint of the overall skill can be derived as a factor graph for the outlined skill-graph. First, the Naive-Bayes approach retrieves the success-constraint as the joint probability of

$$\Lambda_{\boldsymbol{\xi}}^{\mathcal{V}} = \prod_{i=\{2,3,4,6\}} \Gamma_{\boldsymbol{\xi}}^{\mathcal{V}}(\boldsymbol{\xi}_i). \quad (9.38)$$

This results in the factor graph from Figure 9.5a. For the factor graph representation, the actual parameter-vector needs to be decomposed into the scalar components to obtain the underlying factors. Thus, this strongly depends on the actual parameterization of  $v_4$  and  $v_6$ . As both nodes  $v_2$  and  $v_3$  are scalar parameters, we evaluate the presented approach by introducing two further simplifications:

- the search pattern on the surface of the object is restricted to a constant velocity, where the direction is set by an expert, while only the velocity needs to be adjusted to prevent the robot to miss the screw. Thus, we replace  $\kappa_{xy} \leftarrow \mathfrak{s}_{\text{pos}}$ .
- for the force-controller the proportional gain is set equally for all translational components  $x, y, z$ .



**Figure 9.5:** Representation of the success probability of the unscrewing skill as factor graphs. In here, the Naive-Bayes approach is also highlighted as a factor graph, while the actual factor graph exploits available task knowledge to introduced conditional dependence and independence in the regression problem that allows to add samples efficiently during learning.

As a result, the overall success-probability results in the factor graph from Figure 9.5b. The according adjacency-matrices are then given as

$$\begin{aligned}
 \mathbf{A}_{\mathbf{G}v_2} &:= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{A}_{\mathbf{G}v_3} &:= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \mathbf{A}_{\mathbf{G}v_4} &:= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{A}_{\mathbf{G}v_6} &:= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{9.39}$$

Recalling Section 9.4.4, the graph-structure needs to be respected when assigning samples. Failed trials at  $v_2$  can be added to the failure of  $v_2$  and  $v_4$ , while failures at  $v_3$  and forward can be added to all nodes. In addition, the factor graph from Figure 9.5b. allows to create artificial data-samples for  $[\mathbf{K}_P^{\text{frc}}]_{(xyz)}$  in  $\Gamma_{\xi_4}^v$  if a failure at  $v_2$  is detected and similarly to generate samples for  $[\mathbf{K}_P^{\text{frc}}]_{(\psi)}$  in  $\Gamma_{\xi_6}^v$  if a failure for  $v_3$  or  $v_4$  is encountered. Eventually, the RL-problem for the unscrewing task results in regressing the parameter-vector  $\xi \in \mathbb{R}^3$ , as well as  $\xi_4 \in \mathbb{R}^2$  and  $\xi_6 \in \mathbb{R}^2$  for the Naive-Bayes approach. Using the bounds from Table 9.1 a normalized parameter-vector  $\xi \mapsto [0, 1]^3$  can be incrementally evaluated using the acquisition functions from Section 9.4.3 and existing work. We continue with comparing the improvements of our method against existing work in the next section.

## 9.6 Experimental Results

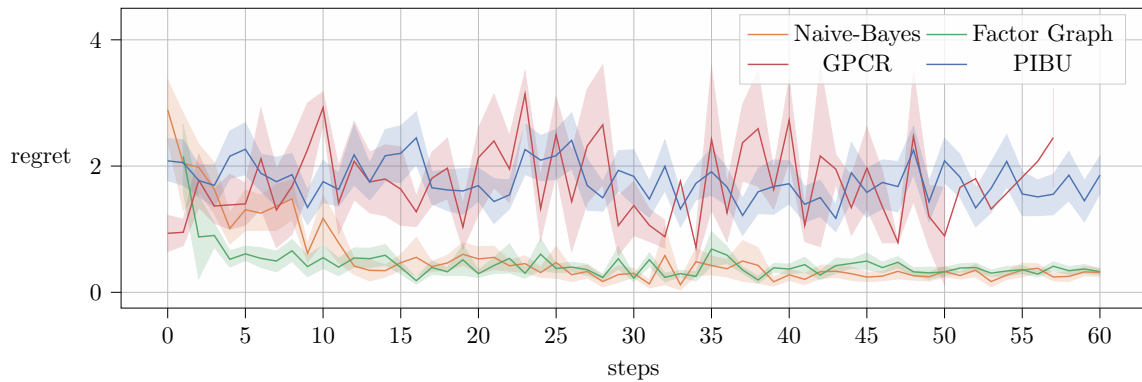
Given the exemplary MP-graph for the unscrewing task from Figure 9.4, a suitable controller parameterization is regressed from data by setting the objective  $\mathcal{J}$  as the negative overall runtime. A parameterization is set as successful, if the full graph has been executed without raising an error-flag. In addition, each node can be repeated up to five times in case a timeout is encountered. The set-values chosen by a designer in our experimental recordings are listed in Table 9.2, where the insertion force is set higher than the environment contact force to enforce an insertion into the screw-head. In order to arrange for a fair comparison over the presented algorithms, the start pose has been chosen identically for all algorithms and the search-direction is set to the static straight line on the object surface as shown in Figure 9.4. Similarly, the tilting angle is chosen to  $2^\circ$  for all approaches and is tilted perpendicular to the motion direction along the object surface. Further, the constraint-thresholds are set to  $\zeta_{\text{pos}} = 0.1$  mm in translation and  $\zeta_{\text{rot}} = 0.1$  rad in rotation. The variance of the FT-sensor has been obtained before running the experiment from collected sensor-data and evaluated to 0.3 N for the force-measurements and  $0.2 \text{ N/m}$  for the Cartesian torque-measurements. The window-size  $N_{\text{FT}}$  to evaluate the sensor-readings has been chosen to 50 using a reading-rate of 170 Hz. Unfortunately, the presented force-controller from Section 9.4.1 suffers from noisy sensor-data and thus misses a proper damping term that could stabilize an aggressive proportional gain controller. To diminish the sensitivity to unstable controller behavior, an explored sample is set to failed if the standard-deviation of the observed force signal during contact is above 2.5 N using a sliding window of 1 s, with a sampling rate of 50 Hz, i.e.,  $N_{\text{cont}} = 50$  and  $\zeta_{\mathbf{F}} = 0.25$  N. In order to detect a contact-impulse during planar search, we set  $\zeta_{\text{impls}} = 5.0$  N and allowed a maximum search range of  $\zeta_{\text{dspl}} = 25.0$  mm. For the GP-models, we assumed a zero-mean prior and used a Matern-Kernel  $\frac{5}{2}$  assuming a prior Gamma-distribution with concentration of 3 and a rate of 6 for the length-scale and a concentration of 2 and a rate of 0.15 for the variance of the kernel. For the related work, we initialized their models according to their manuscripts (Englert and Toussaint, 2016, Marco et al., 2021). In order to allow for a fair comparison of the proposed algorithms and existing work, a grid-search was recorded to collect empirical evidence data-base and mapped to a normalized hyper-cube of  $\xi$  given the parameter-bounds from Table 9.1.

Given this, each algorithm was run 25 times using  $N_{\text{eps}} = 60$  iteration-steps for each run. In each run new samples were added to the dedicated data sets and the current optimum guess is stored at each step. Using the collected empirical evidence as ground-truth, the best empirical sample  $\xi^* = [0.421 \quad 0.316 \quad 0.495]^\top$  is used to calculate the regret  $\text{regret} = \mathcal{J}(\hat{\xi}^*) - \mathcal{J}^*$ .

The averaged regrets over 25 trials per method are plotted in Figure 9.6, where the shaded area highlights the CI of 70%. The presented data underlines that our graphical representations

Parameter	lower bound	upper bound
$\ \dot{\mathbf{x}}\ _2(\mathbf{s}_{\text{pos}})$	1 mm/s	20 mm/s
$[\mathbf{K}_{\text{P}}^{\text{frc}}]_{(\{xyz\})}$	1e-5	1e-3
$[\mathbf{K}_{\text{P}}^{\text{frc}}](\psi)$	1e-5	1e-3

**Table 9.1:** This table summarizes the unknown controller-parameters for the unscrewing skill given the presented controller from Section 9.4.1.



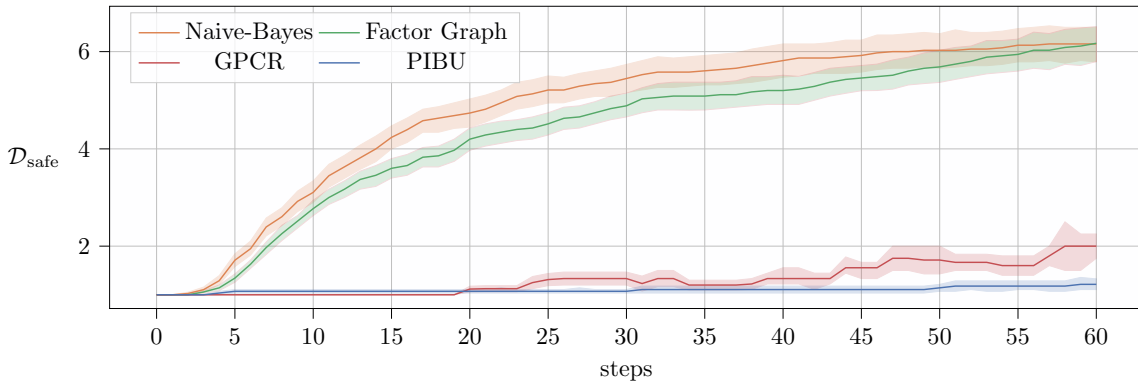
**Figure 9.6:** Regret evolution of the experimental screw-insertion task over the number of trials. The data is averaged over 25 runs per algorithm, with the shaded areas denoting a confidence-interval (CI) of 70 %.

allow acquiring feasible data distinctly faster compared to GPCR and PIBU. This improved learning performance mainly stems from the decreased meta parameter space and the ability to collect evidence of the individual factors rather than learning the full task.

This is further underlined by the evolution of successful samples that are collected by the algorithms as visualized in Figure 9.7. Again, a CI of 70 % is added over the averaged temporal evolution of the successful samples. It also has to be noted that this number is only increased if all nodes of the proposed graphical structures receive a successful sample, i.e., the overall exploration sample returns a successful sample. In this experiment, the Naive-Bayes approach is able to collect new successful samples earlier than the factor graph version. Nonetheless, the difference diminishes by the end of the 60 trials and the evolution of successful samples equals out for both graphical approaches. Within our experimental evaluations, the GPCR-method suffered from numerical instability after latest 60 iterations, while our approaches were able to evaluate further trials. As samples above 60 do not allow for a fair comparison, we omit the continuation of the plots. Still, we ran extended simulations for the proposed graphical methods with 80 steps, and the evolution of the successful samples converged to similar values for the final trial-episodes. While Figure 9.6, denotes the performance of the evaluated methods, Figure 9.7. denotes how many safe samples are explored. Nonetheless, Figure 9.6, only contains valid evaluations of the MPs or the tasks, as even if only a single MP fails, the regret would return an infinite value. In order to compare our algorithms in terms of safety awareness, the rates of estimating a valid optimal sample are listed for each algorithm in Table 9.3. As it can be seen, the pure GP-classification within PIBU outperforms the remaining methods distinctly. This effect mainly stems from the structure of the task, where the approaching speed scaling is linearly increasing the objective, while the constraint is given

Parameter	$\ \mathbf{F}_{\text{des,cont}}\ _2$	$\ \mathbf{F}_{\text{des,insrt}}\ _2$	$\mathbf{n}$	$\frac{\dot{\mathbf{x}}_{\text{des}}}{\ \dot{\mathbf{x}}_{\text{des}}\ _2}$
Value	10.0 N	30.0 N	$[0 \ 0 \ 1]^\top$	$[-0.71 \ 0.71 \ 0]^\top$

**Table 9.2:** Predefined parameters for the unscrewing skill. The value for  $\frac{\dot{\mathbf{x}}_{\text{des}}}{\|\dot{\mathbf{x}}_{\text{des}}\|_2}$  denotes the motion direction that is to be followed during the search on the surface of this object.



**Figure 9.7:** Number of safe samples for the experimental screw-insertion task over the number of trials. The data is averaged over 25 runs per algorithm, with the shaded areas denoting a CI of 70%.

as a strict upper threshold, that also represents the optimal value. With only a handful samples, estimating the constraint rather than the classification labels remains numerically challenging.

In contrast, the application of a pure classification GP may also be overly conservative and being only provided with a small number of successful samples, the classification may not be capable of returning a useful solution for the task to be learned. Besides receiving a distinctly smaller regret, our approaches also converge closer to the actual optimum parameter samples. This is visualized by the temporal evolution of the estimated optimal parameter samples in Figure 9.8, where the shaded areas again denote a CI of 70%. In contrast to related work, our approaches quickly converge to a solution for  $\xi_1$  and  $\xi_2$ , while  $\xi_3$  is only slowly converging towards the optimal value. This delay stems from  $\xi_3$  being conditionally dependent on the performance of the remaining data-samples. Even though the estimation of  $\xi_3$  also suffers from higher variance compared to related work, our approaches distinctly outperform the related work in this aspect. This underlines that our approaches do not result in suitable parameter-estimation by chance but due to efficient data-acquisition.

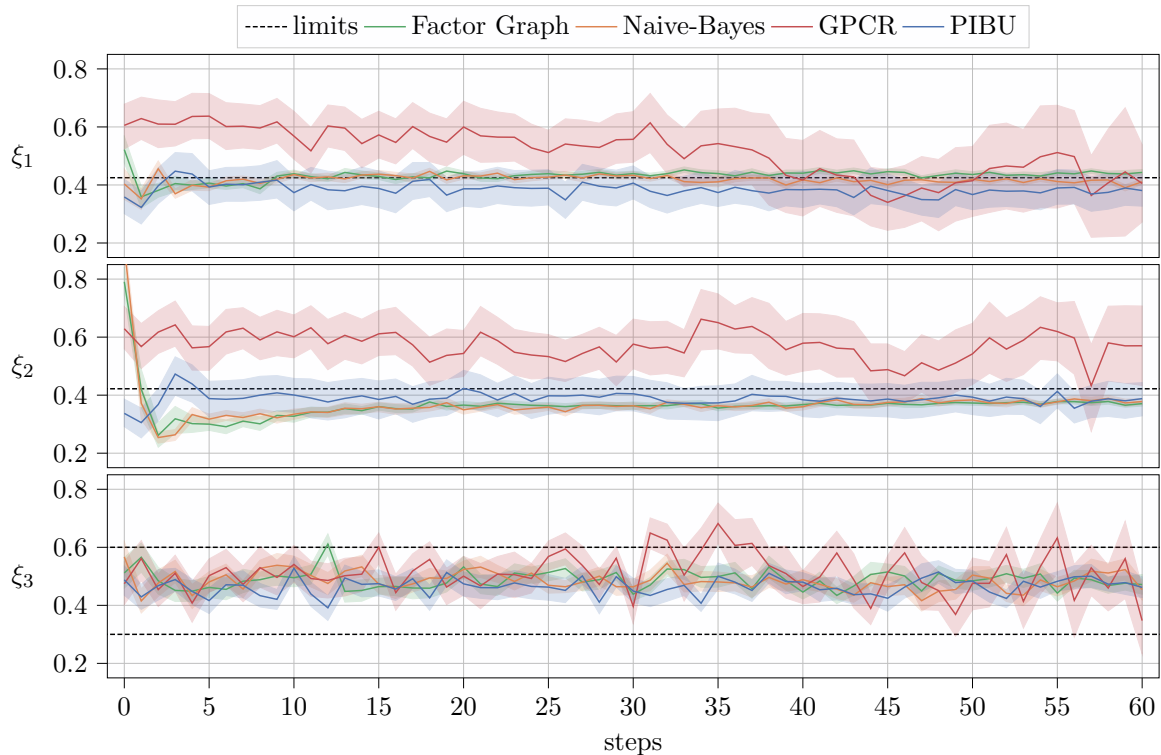
## Discussion

Having collected the experimental data, our approaches outperform existing work in terms of data-efficiency and allows to obtain suitable results from only a handful of samples. Furthermore, our approaches apply standard GPs on smaller meta parameter spaces. Even though our regression method requires multiple parameter fit for multiple nodes, each GP is conditionally independent by definition within a factor graph. This allows for full parallelization, even though we evaluated our method in a purely sequential manner.

	GPCR	PIBU	Naive-Bayes	Factor Graph
success-probabilities (in %)	29.1	<b>53.7</b>	29.7	23.6

**Table 9.3:** Rate of estimating a correct optimal sample. The best performing, i.e., highest success-percentage is highlighted in bold.





**Figure 9.8:** Temporal evolution of the estimated optimum parameter value, where the dashed lines denote the upper bound for the parameter, while the lower dashed line in the bottom plot denotes the dedicated lower bound. Shaded areas denote the CI of 70%. The optimal values to be regressed from data are  $\xi^* = [0.421 \quad 0.316 \quad 0.495]^T$ .

Nonetheless, the presented results also highlighted a particular downside of our method, which is exposed by the small chance of drawing successful samples. While our method outperformed existing work in drawing successful samples during exploration, this effect can be neglected during exploitation. If the current estimate is to be applied on safety-critical applications, the provided success-rate needs to be improved. While it has to be noted that neither GPCR nor a classification GP can provide a safety-guarantee when drawing success-estimate, the combination of our method with one of the formers allows to alleviate this issue. Thus, the overall graph-success probability can be replaced by a product of experts, where the experts are given as the individual success-models. Another possible solution is given by using a negative prior mean similar to PIBU in the constraint-GP and evaluating the constraint-metric by shifting the probability of the posterior. Using this, the search of the optimal value is constrained to a tightened set of the parameter-space, which automatically results in an increased probability to draw a correct sample.

Eventually, it has to be mentioned that the presented problem on regressing  $\xi_1$  is a special case, while in general cases, where the optimum value is not in the near distance of the success-constraint, our method reliably converges to the correct parameter-guess. Given the overall improvements of our method that is evident in the collected experimental data and the overall framework, it can be summarized, that our method improves existing methods on regressing task-parameters for autonomous robots in a constraint-aware manner. Referring to the ability of converging to correct values within a reasonable time and amount of data, make

our application a reasonable method to be applied on future robot platforms and manipulation tasks.

Finally, the question of whether either our factor graph or the Naive-Bayes approach is favorable needs discussion. Referring to the overall results, the performance of both methods is comparably similar. This mainly stems from the fact that the first parameter and thus the first sample is the most critical evaluation parameter of the task to be learned. As this node is conditionally independent of the last parameter, the benefit of generating artificial data-samples can only be applied rarely. Nonetheless, the preferable major advantage of the factor graph is given by the ability to apply it to arbitrary tasks, and allows to regress constraints that have a different input-space than the current objective node. Given that both approaches obtained almost identical performance results, the factor graph method forms the generic representation and preferable method, whereas the Naive-Bayes version stands out by its simplicity and simple adjustment to alternative models.

## 9.7 Conclusion

In this chapter we proposed an episodic RL-scheme that uses BOC to account for unsafe exploration samples during learning. In order to apply the proposed scheme online, we further outlined a suitable control architecture for an industrial robot platform that uses a Cartesian displacement control interface at a comparably low update rate. The hybrid controller interface is well suited to apply selective control-strategies along individual axes, which can then be embedded into a graphical skill-formalism from previous work to reduce the required parameter-space for the task to be learned.

In contrast to existing work, we further claimed that it is beneficial to not only exploit available task-knowledge to decrease the parameter- or search-space for the current task, but also to incorporate task-knowledge on regressing the failure constraints. For this reason, we proposed a graphical skill-formalism for the overall success-probability as factor graphs. Here, we proposed a pure Naive-Bayes method that regresses the failure of the overall task as the joint probability of each node failing for a given sample. While this method improves the overall sampling, it may hinder assigning failed samples to subsequent nodes, even though conditional dependencies are well known beforehand. Thus, we further proposed to incorporate these relations into a graphical skill-formalism for the success-probability, and thus improve scaling behavior to eventually regress feasible samples. In addition, we proposed suitable acquisition functions for the individual representations and proposed a novel conservative acquisition method.

Finally, we outlined an application example for the proposed method as the screw insertion task for an industrial robot, where the exact goal-pose is unknown and the controller parameterization of our proposed controller needs to be regressed from data.

Given the outlined screw insertion task, we compared our approaches against existing state-of-the-art methods for BOC-based RL using an industrial robot manipulator in a lab-environment. Given the collected experimental data, our method distinctly outperformed the state-of-the-art in performance, which we have evaluated by the collected objective regret. Furthermore, our method required distinctly smaller amount of data-samples and thus learning time and steps compared to existing work. These results underline that it is preferable to not only

incorporate available task-knowledge for the objective but also the constraints of robotic manipulation tasks during learning whenever possible in order to decrease the number of samples needed.

### Future Work

Building upon the data collected and the presented method, a promising path for future research projects lies in combining our method with visual feedback. This may further allow defining robust success- and error-constraints, as for example missing the screw-head or hole remains unreliable solely from FT-data; especially if a constant velocity vector results in a robot missing the screw-head completely. If such feedback is obtained, the presented method would strongly benefit in learning advanced motion policies, i.e., comparing different search patterns, e.g., spirals or straight-line patterns. Nonetheless, regressing the optimal search pattern usually is preferably solved by visual servoing. In these scenarios, the interaction does not rely on accurate FT-data and feedback control. Thus, this allows to collect data within simulated environments and to apply recent results from machine learning, especially meta-RL.

Eventually, future research should evaluate the possibility of self-evaluating models, i.e., artificial agents should be aware that some of the imposed model-knowledge may be subject to false design. Thus, another line of research is given by designing new methods that allow not only to exploit available task knowledge but also to evaluate the accuracy and discrepancy of the assumed model against the empirical evidence.



# 10

## Conclusion

This thesis has aimed to investigate the challenge for autonomous robots to cope with the stochasticity of the environment for real-world applications. Handling uncertainties and decreasing knowledge gaps is a key-challenge to bridge findings from well-defined lab-environments towards practical applications, where the perception of the environment often is subject to faulty sensors or missing model knowledge. As the sources of uncertainty for complex systems such as robotic applications are beyond the scope of a single thesis and may remain an open research field for many decades, this thesis has specifically focused on selected, yet utmost important, sub-areas of these challenges. In particular, a major emphasis was set on interacting with humans in the close – yet well-defined – distance in the context of industrial assembly, the interaction with autonomous agents and eventually manipulation tasks within unknown or only partially defined environments.

### 10.1 Interactive Action-Selection within Human-Robot Collaboration

In the first part of this thesis, we have evaluated the aspect of human behavior models and interactive decision making in human-robot collaboration (HRC), where a team of one robot and one human are required to perform a joint task. In here, we have evaluated the question *How can robots estimate and track human suboptimal behavior within HRC?* by proposing a mixed observability Markov-model that allows to depict the HRC-scenario as a collection of fully observable representations by means of Markov decision processes (MDPs) given a discretized realization of the partially observable states of the environment. Specifically, the not observable state depicts the human belief of the joint task, as an exemplary use-case for the suboptimal human behavior. Nonetheless, the presented method is not limited to this specific use-case, as alternative human states such as trust or fatigue are directly applicable in the presented framework. Similarly, we proposed to directly track suboptimal human behavior into a human-robot decision making framework in our conceptual HRC task and motion planning-framework. Given the collected experimental data, our methods allow robots to account for suboptimal human behavior and thus to improve the decision-making online.

Besides evaluating the concept of human suboptimal behavior, we further evaluated the research question *How can mutually interactive game-theory be applied for autonomous decision-making on robotic systems within HRC?*. Specifically, we proposed to use normal form games to model the action assignments in joint HRC-tasks. We have evaluated the efficiency and increased performance of the presented method on exemplary human-robot pick-and-place tasks within a lab environment. While this method can be straight-forwardly extended to

arbitrary tasks, a major downside is given by the poor scaling of normal-form games to long-term decision making, i.e., planning problems. In general tasks, the decision horizon requires to evaluate more steps than solely a single step. Thus, we further outlined how our method can be extended to a generic Markov game (MG), which allows for long-term decisions under uncertainty and extending the concept of MDPs to the multi-agent domain.

As MGs suffer from poor scaling and do not allow to control the human policy, we outlined how an approximated solution of the interactive HRC-MG can be obtained. We propose to solve this by directly answering the third research question of this part, i.e., *How can robots account for incorrect models during the interaction with humans?* Consequently, our concept incorporates the idea of a mixed observable state-representation to model the interactive MG as a collection of local MDPs that tracks human rationality as a metric on the model accuracy of the human counterpart. Thus, the direct incorporation and discretization of the human rationality represents a different evolution, i.e., state-transition for the MG, which can be approximated by a local MDP instead of solving the full MG.

Summarizing the individual findings of this part of the thesis, our methods improve human behavior models as well as the interactive decision-making of robots within HRC, and allow for future applications of robots in a close distance of humans. Directly incorporating safety aspects such as collision avoidance in the decision-making allows robots to improve the interaction safety. This is of utter importance to allow robot applications without the necessity of safety fences such as cages or light-barriers.

## 10.2 Learning Behavior-Policies in Groups of Artificial Agents

In the second part of this thesis, we have focused on the research field of multi-agent systems, specifically the area of multi-agent reinforcement learning (MARL). Specifically, we investigated the research question: *How can reinforcement learning (RL) of multi-agent systems be embedded into a hierarchical framework without relying on overly restrictive assumptions such as fully synchronous decisions and centralized learning?* Thus, we proposed a novel decentralized MARL-framework that embeds methods from game-theory and multi-robot systems. This model allows to decentralize the interaction of the agents within MARL, while still accounting for the behavior or other agents by modeling each agent as a best-response (br)-policy, which eventually converges to a Stackelberg-equilibrium. Furthermore, our approach explicitly differentiates between joint task rewards and native – or agent-specific – cost-terms. Finally, we proposed a novel hierarchical MARL-approach, that builds upon the presented method in order to provide a valid answer to our first research question.

Recalling our second research question: *How can the effect of hierarchical performance be evaluated against joint task performance?*, the hierarchical concept directly exploits the ability of our br-policy-based approach on differentiating between native costs and interactive task-rewards. Namely within a hierarchical context, each agent is able to explicitly differentiate between hierarchical – or artificial – sub-goals and the current interactive task reward.

Eventually, our hierarchical MARL-concept also introduces factored observations for each agent. The main idea is to directly distinguish between internal agent-states and external observations. This allows to directly distinguish between internal – most often known – agent dynamics and external – mostly unknown – environment dynamics. This part of the thesis

only closes with a theoretical concept on how to answer our last research question: *How can basic model-knowledge about the individual agent-dynamics be embedded into model-free MARL without adding overly restrictive model and system assumptions?*, Nonetheless, we present a collection of possible extensions of our proposed methods, that bare great potential to answer the stated research question within future work.

Given the empirical comparison of our br-based MARL approach against recent state-of-the-art MARL-algorithms, it can be summarized that this thesis has presented valid answers to the first and second research question of this part. Furthermore, the provided MARL-concept serves as a valuable basis to tackle the remaining open research question(s). While a full summary of future research will close this chapter and thesis, we proceed with a summary of the research questions in Part III.

### 10.3 Advanced Manipulation Tasks with Unknown Objects

In the last part of this thesis, we have evaluated three individual research questions, that have been motivated from actual robotic manipulation skills, that have not yet been achievable for a robotic system by the beginning of this thesis. Given the collected results and methods, our work extends the skill-set and performance of robots in unknown environments and allows for future applications. In contrast to most common approaches, our methods are applicable to industrial robots, which are already broadly established in production plants, thus opening the door for advanced factory plants, where new tasks can be established faster while also allowing for uncertainties in the perception of the robot surroundings.

The first research question analyzed in this part of the thesis was framed as: *How can robots refine their model knowledge about the characteristics of unknown objects if there is no vision data or insufficient vision data available?* In order to accomplish this research question, we proposed a novel state-estimation framework, that applies concepts from Bayesian filter-theory and thus allows a robot to regress not only the structure but also the material characteristics of the unknown object from haptic sensor data. Specifically, our framework uses an incremental Bayesian-filter that uses a prediction and correction update step to update the current estimation of the geometric shape of the object, while also using unsupervised machine-learning methods to cluster the obtained data measurements into dedicated groups of material types. Given these clusters, a robot eventually is able to estimate the material characteristics of each cluster to identify the underlying physical properties. This novel concept has been evaluated in a simulation environment, where the material stiffness has been chosen as the distinct material parameter. The collected data outlined the capability of our presented method on regressing the shape of an object while also differentiating between different material types. This allows robots to refine the received object knowledge when visual data may suffer from imprecision or may not be accessible at all, which provides a usable answer to the stated research question and thus opens the door for future robotic applications in handling unknown devices.

Furthermore, we have evaluated the research question: *How can industrial robots execute sensitive grasping skills if neither the robot hardware provides compliant control interfaces nor a force-torque (FT)-sensor is available to account for undesired interaction wrenches?* For this, we proposed a novel grasping controller, that allows for an application on industrial robot platforms while still providing the ability to compliantly grasp objects. Specifically

we propose a novel alignment error estimator, that uses the haptic sensor feedback of digital sensor arrays (DSAs), that are equipped on the gripper fingers. Using this alignment error, a hybrid controller is outlined that allows to command hybrid grasping strategies by either following a model predictive control (MPC)-based velocity signal or to apply a Cartesian wrench-controller to diminish the interaction wrenches. We empirically evaluated the presented grasping strategies in their pure forms as well as hybrid combinations of the former and the latter against a pure compliant robot control. The presented controller allows to attenuate the individual control policies along selective axes depending on the current estimation error, thus distinctly improving the final pose error and therefore the overall success-rate compared to the baseline method. In summary, our presented grasping controllers provide a realistic and suitable answer to the stated research question and thus provide the basis for future compliant grasping tasks for (industrial) robots.

Eventually, this part of the thesis evaluated a third research question: *How can industrial robots efficiently learn compliant manipulation tasks within a reasonable time, i.e., only from a handful of experimental trials?* This question imposes the challenge of designing a suitable controller for an industrial robot, imposing an additional constraint on not allowing to command the joint-position or -velocities at suitable update rates. Thus, this thesis proposed a hybrid Cartesian velocity-force controller that is applicable to compliant manipulation tasks. Further, we outlined how the hybrid nature of the presented controller is well suited to directly impose a graphical representation of the manipulation skill from existing work, that explicitly incorporates available task knowledge. In contrast to existing work, where the emphasis was laid upon decreasing the parameter-space of the objective to be learned, our method proposes to extend this graphical formalism on also regressing the task-constraints, i.e., the feasible parameter-space. Besides proposing two novel graphical skill-formalism for the success estimation of the manipulation task, we also proposed suitable Bayesian optimization with unknown constraints (BOC)-models and acquisition functions, that allow to regress the optimal task parameters within reasonable time given only a small number of evaluation samples. We eventually introduced an application example that we also used to evaluate the advances of our proposed method against existing work: the insertion of a screwdriver in the screwhead. In here, we emphasized on regressing the optimal controller parameterization while considering interaction wrench constraints. Our presented methods distinctly outperformed existing work in terms of the number of samples that are required to obtain feasible results, while also obtaining a distinctly improved regret and number of safe samples during exploration. Given the collected data, the presented method is well suited to allow the application of (industrial) robots on unknown manipulation tasks that may require the robot to act compliantly. Overall, the presented method forms a useful and promising answer to the stated research question and thus allows to further extend the application of (industrial) robots for unknown manipulation tasks.

## 10.4 Recommendations for Future Research

Even though the presented work contains a collection of improvements and allow for novel applications of robot systems, there are still limitations and open questions that require future research.



In the aspect of HRC, the proposed concept from Chapter 5 requires proper empirical validation as well as the development of a suitable motion-planner that allows for the proposed interaction-framework. In here, the concept proposed by Osa (2020) contains a promising concept as it allows to extract multiple solutions for a single optimization task. Their method would also allow to regress multiple policies if the goal-pose contains free-parameters. Nonetheless, their current motion planner suffers from slow convergence, such that incorporating their concepts in faster motion planners, e.g., (Bari et al., 2021, Mukadam et al., 2018). Alternatively, sampling based methods usually allow to reuse the graph-structure to recover alternative solutions quickly, such that methods as in Englert et al. (2021) may also serve as a good start.

While game-theory has found already broad application in autonomous driving, the most prominent line of research lies in machine learning, where concepts such as (Bai et al., 2019, Geiger and Straehle, 2021, Ling et al., 2018) already proposed ideas on how artificial agents can autonomously learn not only a policy but equilibria and thus directly encode game-dynamics in the learning framework. In order to diminish the reliance on autonomous planning, and thus rely on poor scaling for certain applications, the incorporation of these recent findings within applications such as autonomous driving or interactive HRC might improve the current state-of-the-art in said application areas.

In the context of MARL, research progress has reached an incredible production speed, such that the proposed extensions below may already be outdated. Nonetheless, we briefly sketch further lines of research in the context of MARL that are worth further investigation:

- **meta-RL.** In the context of meta-learning (Frans et al., 2018), advances from simulation allows to be applied on physical platforms by regressing suitable representation-similarities between the simulated and real environment observations. This area is a promising line of research and by its nature closely related to the hierarchical framework presented in this thesis. In here, a promising field of research lies in incorporating constraints and safety metrics, which is a crucial aspect for real-world robotic applications.
- **inverse cost and reward learning.** In the context of MARL and RL, an alternative to learning a policy directly is the idea of learning a convex objective that not only allows to represent the task to be learned but also to apply differentiable controllers, such as MPC or optimal control for the online execution. Thus a promising line of research is to extend concepts such as Englert and Toussaint (2016) on the MARL-domain.
- **combining data-driven controllers with deep-RL.** Recently, data-driven control methods (Bevanda et al., 2021, Kerz et al., 2021) have resulted in impressive results. As their methods are well suited to cope for the unknown system-dynamics of MARL, the extension of these methods to multi-agent-systems for either distributed control or even as a combination of model-free and model-based RL is an interesting and promising research path, that may eventually allow for improved scalability, while also guaranteeing safe execution.
- **impose robot controllers into hierarchical-MARL.** The most promising line of research of the presented method from Chapter 6 is closely related to the previous bullet-point. Namely, the direct incorporation of model-knowledge into hierarchical reinforcement learning, especially within multi-agent systems. In contrast to most approaches from control-theory, future research needs to investigate conceptual approaches that

only impose minor system-assumptions and constraints, e.g., only assume knowledge about the internal agent dynamics. Still, results from single-agent RL has outlined how advanced robot behavior can be extracted by composing policies from multiple low-level controllers (Saxena et al., 2021, Sharma et al., 2020). On the other hand, a controller may also provide suitable priors to achieve suitable results faster (Rana et al., 2021).

- **limit RL policies to regions of need.** Eventually, findings from safe-learning (Zhou et al., 2021) bear great potential to improve (hierarchical) MARL. In contrast to only identifying safe regions, it is also of interest to directly classify the regions of the state-space, in which the existing model knowledge is insufficient, i.e., where RL is actually needed.
- **evaluate the applicability of end-to-end differentiable models.** In order to profit from powerful tools such as MPC and optimal control, the investigation of applying end-to-end differentiable MPC (Amos et al., 2018) bears great potential for future research within the MARL-domain and beyond. Especially within multi-agent systems, the (response-)behavior of other agents is best to be represented by deep neural network. In order to exploit the full capabilities of optimization, future research needs to investigate suitable solutions to apply e.g., MPC on such complex models in real time.
- **incorporate stabilizing feedback-control in MARL:** similarly to the above, investigating the possibility of embedding stabilizing feedback-controllers into data-driven MARL-problem forms an interesting line of research, where a hierarchical framework may profit from finding stable control-funnels during learning (Ames and Konidaris, 2019, Reist et al., 2016, Tedrake et al., 2010).

Given our presented results on haptic object identification, the presented methods need to be applied and further elaborated on a physical platform to account for noisy sensor-output. In here, it is important to apply a suitable controller, that allows to explicitly compensate for interaction wrenches. A promising line of extensive research thus might be given by combining adaptive controllers such as Li et al. (2018c) with a prior on the surface materials, to directly account for the model-mismatch, which can be embedded in an extended Bayes-Filter. Furthermore, incorporating recent findings of haptic-rendering (Mercado et al., 2021) into such controllers is a promising field of research, that is worth to investigate. Eventually, the usage of additional sensors and evaluation of multiple features during the classification is of utter importance in order to obtain robust material classification, cf. Strese (2021).

In the aspect of adaptive grasping, the currently presented method only has access to the DSA or FT-sensor-readings. This achieves useful results but may eventually profit from directly extending the findings by data-driven grasping methods, that directly use vision data or depth information. Future research should thus investigate the possibility of adding the obtained sensor-readings into existing grasping pose detector methods to account for repositioning online. In here, a major challenge is given by designing a suitable and precise physics engine that simulates the behavior of DSAs-readings, to train the combined models by concepts such as meta-RL (Frans et al., 2018). Alternatively, the behavior of the DSAs-sensors may be regressed from data-recordings first, starting from existing physical interaction learning work (de Avila Belbute-Peres et al., 2018). Eventually, the selection of control-strategies may be extended by alternative control strategies, especially if the robot platform allows for advanced controllers. In here, the concept of selecting control-strategies from data-exploration is an interesting line of research, as for example proposed by Sharma et al. (2020).

Regarding the aspect of learning manipulation tasks from collected empirical data, the presented method is error-prone in the actual detection and classification of successful or failed trials. Thus, future research should investigate the possibility of improving the system against false-positive or false-negatives. Partially related to this, the concept of handling concurrent data collection needs further investigation. This requires to allow robots to distinguish between useful and less useful data-evidence. Starting from work such as [Umlauf et al. \(2020\)](#) may thus improve the proposed methods.

Having collected the various line of research projects that may follow this thesis, we want to point to available open-software modules – further modules are listed in Appendix A and Appendix B – that may ease the progress of future research projects<sup>1</sup>. Even though the list may lack in completion, there exists useful simulation benchmark-environments [Leurent \(2018\)](#)<sup>2</sup>, [Panerati et al. \(2021\)](#)<sup>3</sup> [Leibo et al. \(2021\)](#)<sup>4</sup> or [Lenton et al. \(2021\)](#)<sup>5</sup>, control-toolboxes [Sekiguchi \(2018-2022\)](#)<sup>6</sup>, [Amos et al. \(2018\)](#)<sup>7</sup>, or [Lucia et al. \(2017\)](#)<sup>8</sup>, machine-learning baselines [Dhariwal et al. \(2017\)](#)<sup>9</sup>, [Achiam \(2018\)](#)<sup>10</sup>, [Delhaisse et al. \(2019\)](#)<sup>11</sup> or [Moritz et al. \(2018\)](#)<sup>12</sup>, as eventually online-blogs ([DeWolf, 2012–2022](#))<sup>13</sup>.

<sup>1</sup>Please not that these projects are subject to external maintenance and may thus be outdated at some point.

<sup>2</sup><https://github.com/eleurent/highway-env> for autonomous driving.

<sup>3</sup><https://github.com/utiasDSL/gym-pybullet-drones> for MARL using quadrotors.

<sup>4</sup><https://github.com/deepmind/meltingpot> a benchmark-suite for MARL.

<sup>5</sup><https://github.com/unifyai/gym> for optimization of fully differentiable systems.

<sup>6</sup><https://github.com/Shunichi09/PythonLinearNonlinearControl> containing implementations for (non)linear control.

<sup>7</sup><https://locuslab.github.io/mpc.pytorch/> a fast and differentiable MPC-solver for PyTorch.

<sup>8</sup><https://github.com/do-mpc/do-mpc> comprehensive toolbox for robust MPC.

<sup>9</sup><https://github.com/openai/baselines>.

<sup>10</sup><https://spinningup.openai.com/en/latest/>.

<sup>11</sup><https://github.com/robotlearn/pyrobolearn>.

<sup>12</sup><https://docs.ray.io/en/master/rllib/index.html>.

<sup>13</sup><https://studywolf.wordpress.com/>.





## Summary of Software Modules

The software modules produced within this thesis<sup>1</sup> are available online, partially in public and partially only upon request. The software is provided as is, without further warranty or support. For further details, each module contains a dedicated license. For a brief and concise overview, the list below sketches the individual software-modules and how they are connected to the individual chapters of this thesis. First, this thesis resulted in the following open-source modules<sup>2</sup>:

- <https://gitlab.com/vgab/ros-wsg-50>: contains the robot gripper-driver of the WSG 50 presented in Chapter 8, as applied in Part I and Part III.
- [https://gitlab.com/vg\\_tum/relational-engine](https://gitlab.com/vg_tum/relational-engine): contains a relational planning module used in Chapter 4 and Chapter 5.
- [https://gitlab.com/vg\\_tum/multi-agent-gym](https://gitlab.com/vg_tum/multi-agent-gym): contains a multi-agent simulation environment for multi-agent reinforcement learning (MARL)-methods, that has been used to evaluate the approaches from Chapter 6.
- [https://gitlab.com/vg\\_tum/mahac\\_rl](https://gitlab.com/vg_tum/mahac_rl): contains an implementation of the baseline methods and the proposed method from Chapter 6.
- [https://gitlab.com/vg\\_tum/sim-robots](https://gitlab.com/vg_tum/sim-robots): contains a collection of robot kinematic handlers / helpers that were used within Chapter 8 and Chapter 9.
- [https://gitlab.com/vg\\_tum/graph-boc](https://gitlab.com/vg_tum/graph-boc): contains an implementation of the baseline methods and the proposed method from Chapter 9.

Furthermore,  $\text{\LaTeX}$  content to reproduce selective latest content of this thesis is available as

- [https://gitlab.com/vg\\_tum/latex\\_styles](https://gitlab.com/vg_tum/latex_styles): contains a collection of  $\text{\LaTeX}$ -utilities used within this thesis.
- [https://gitlab.com/vg\\_tum/dissaster](https://gitlab.com/vg_tum/dissaster): contains the  $\text{\LaTeX}$  source-code to generate this report and presentation slides.
- [https://gitlab.com/vg\\_tum/graph-boc-article](https://gitlab.com/vg_tum/graph-boc-article): contains the  $\text{\LaTeX}$  source-code for the manuscript from Gabler and Wollherr (2022).
- [https://gitlab.com/vg\\_tum/compliant-grasper](https://gitlab.com/vg_tum/compliant-grasper): contains videos, presentations and  $\text{\LaTeX}$ -files for the content from Gabler et al. (2022b).

---

<sup>1</sup>Please refer to the dedicated repositories for a full list of contributors.

<sup>2</sup>Some of the modules listed below are subject to articles, which were still under review by the time of submission. Therefore, the availability of the dedicated data directly depends on the publication status.

Beyond this, this thesis has distinctly contributed to the development of the following robot operating system (ROS)-projects, which are available as repositories of the Technical University of Munich and Chair of Automatic Control Engineering. As these repositories are subject to internal and thus ongoing research work, access can only be provided upon request:

- [https://gitlab.lrz.de/lsr-itr-ros//kuka\\_lwr](https://gitlab.lrz.de/lsr-itr-ros//kuka_lwr): contains an extension of the Kuka-LWR 4+ ROS-driver from [Bioengineering and Robotics Research Center "E. Piaggio" University of Pisa \(2014\)](#), applied in Part I.
- [https://gitlab.lrz.de/lsr-itr-ros/robot\\_parts](https://gitlab.lrz.de/lsr-itr-ros/robot_parts): contains a collection of simulation models and/or unified robot description formats of robot-components, applied in in Part I and Part III.
- [https://gitlab.lrz.de/lsr-itr-ros/ros\\_jr3](https://gitlab.lrz.de/lsr-itr-ros/ros_jr3): contains a ROS-driver of the *JR3*-force-torque (FT)-sensor, applied in Part I and Part III.
- <https://gitlab.lrz.de/lsr-itr-ros/comau-data>: robot programs and driver interfaces, necessary to run a COMAU Racer 5 robot setup, applied in Part III.
- <https://gitlab.lrz.de/lsr-itr-ros/comau-experimental>: modified and extended ROS-driver, as developed within the HR-Recycler project, applied in Part III.
- [https://gitlab.lrz.de/hr\\_recycler/sim\\_robots](https://gitlab.lrz.de/hr_recycler/sim_robots): contains an extended version of the public-version from <https://gitlab.com/vgab/sim-robots> that is mainly tailored to the COMAU Racer 5 setup of the HR-Recycler project, applied in Part III.
- [https://gitlab.lrz.de/hr\\_recycler/hrr\\_cobot](https://gitlab.lrz.de/hr_recycler/hrr_cobot): high-level python control interface to control the COMAU Racer 5 setup of the HR-Recycler project, applied in Part III.

Eventually, the a collection of projects have been developed within Part I, which have been archived by the end of this thesis. Nonetheless, they can be made available upon request:

- `vg_hgmdp`: contains the software to reproduce the experiments from Chapter 3.
- `vg_game`: contains the software to reproduce the experiments from Chapter 4.
- `user_interfaces`: contains a collection of ROS packages to create the user-interfaces as e.g., needed for the experiment in Chapter 3.
- `hrc-main`<sup>A</sup>: contains a collection of high-level configuration, utilities and visualization helper, used within human-robot collaboration (HRC)-experiments in Part I.
- `env-perception`<sup>A</sup>: contain the required ROS-packages for system calibration and
- `hrc-common`<sup>AB</sup>: contains a collection of MATLAB<sup>®</sup>-Simulink handles to control a Kuka LWR 4+ robot
- `hrc-state-machine`<sup>AB</sup>: contains a state-machine in MATLAB<sup>®</sup>-Simulink to command high-level manipulation skills to a Kuka LWR 4+ robot.

---





















<sup>A</sup> Additional credits belong to: [Dr.-Ing. Gerold Huber](#)

<sup>B</sup> Additional credits belong to: [M.Sc. Khoi Hoang Dinh](#)

# B

## Hyper-Parameters and Implementation

The software modules that have been developed or used within this thesis build upon a variety of open-source projects, which need to be references appropriately. For brevity, we summarize the dedicated usage in a per-part bases in Table B.1.

Module(s)	Part I	Part II	Part III
 Van Rossum and Drake (2009)	✓	✓	✓
 Kluyver et al. (2016)	✓	✓	✓
 Harris et al. (2020)	✓	✓	✓
 Hunter (2007), Waskom et al. (2017)	✓	✓	✓
 McKinney et al. (2010)	✓	✓	✓
 Virtanen et al. (2020)	✓	✓	✓
 Meurer et al. (2017)	✓	✓	✓
C++ Stroustrup (2000)	✓		✓
C++ Guennebaud et al. (2010)	✓		✓
C++ Coleman et al. (2014)	✓		✓
 Quigley et al. (2009)	✓		✓
 Chitta et al. (2017)	✓		✓
 Koenig and Howard (2004)	✓		✓
 Todorov et al. (2012)		✓	✓
 Coumans and Bai (2016–2020)		✓	✓
 Van Rossum and Drake Jr (1995)	✓		
 E. Rohmer S. P. N. Singh (2013)	✓		
 Paszke et al. (2019)		✓	
 Brockman et al. (2016)		✓	
 Klaus Greff et al. (2017), Yadan (2019)		✓	
 Corke and Haviland (2021)			✓
 Lucia et al. (2017)			✓
 Balandat et al. (2019), Gardner et al. (2018)			✓
C++ Rusu and Cousins (2011)			✓

**Table B.1:** Usage of available open-source modules. Grouped models have only been used in shown combination.

In addition, this thesis and the created projects distinctly profited from **git** (Torvald et al., 2005), **L<sup>A</sup>T<sub>E</sub>X** (Lamport, 1986), Linux (Torvalds et al. (1991)), docker (Merkel (2014)) and zsh-projects (Perepelitsa, 2019, Russell, 2009).

## Implementation Details of Chapter 6

In order to evaluate our MARL-algorithms, we evaluated our implementation on our implementation of the multi-agent particle environment (MPE)-environment. The hyper-parameters for the learning procedure that is applied for all algorithms identically is listed in Table B.2.

Parameter	Value
batch-size	1024
polyak value for the target-net update	0.95
decay-parameter $\gamma$	0.95
exploration episode-length	200
evaluation episode-length	100
number of episodes	5000
number of update-steps	5
update rate	every fifth episode
episode-buffer size	1000
number of agents $N_{\mathfrak{A}}$	3
polynomial filter-order (Savitzky and Golay, 1964)	3
filter window-size (Savitzky and Golay, 1964)	31

**Table B.2:** Hyper-parameters for the experimental evaluation and all evaluated algorithms.

Similarly, the (physical) parameters for the MPE-environment are listed in Table B.3.

Parameter	Value
$dt$	0.02 s
$N_{\mathfrak{A}}$	3
goal-threshold $\zeta_{g,MPE}$	0.02 m
collision-cost	1
$\mathbf{v}$	0.9
$\mathbf{m}$	1 kg
$\mathbf{v}_{\max}$	1.0 <sup>m</sup> /s

**Table B.3:** Environment parameters for the MPE and *cooperative collection* task.

Eventually, the hyper-parameters for the individual algorithms are listed in Table B.4. Our best-response (br)-based policies used identical parameters for the dyadic and game against nature scheme. Therefore, only one column per algorithm is provided in the table below. We used Adam Kingma and Ba (2015) for the stochastic gradient descent (SGD)-optimization for all algorithms.

The experimental evidence was collected on two distributed computers with the following hardware components

- **OS-Kernel:**
  - (Ubuntu) Linux-5.13.0-44
  - (Ubuntu) Linux-4.15.0-187



---

Parameter	MADDPG	MASAC	br-TD3	br-SAC
size of (hidden) critic-layers	(64, 64)	(64, 64)	(64, 64)	(64, 64)
size of (hidden) policy-layers	(64, 64)	(64, 64)	(64, 64)	(64, 64)
learning-rate critic $\iota_q$	0.01	0.01	0.01	0.01
learning-rate policy $\iota_\pi$	0.01	0.01	0.01	0.01
polyak-value $\nu$	0.01	0.01	0.01	0.01
$\alpha_0$		0.2		0.2

**Table B.4:** Hyper-parameters for each algorithm. In case an entry is left blank, the algorithm does not have this hyper-parameter. The critic-parameters for our br-based approaches have been chosen identically for the interaction-critic and native critic. Similarly, the parameters for the br-policies are identical as the actor-policy.

- **Processor:**
  - Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz
  - AMD Ryzen Threadripper 2990WX 32-Core
- **Python-version:** 3.9.7
- **GPU-acceleration:** disabled



# Glossary

## Acronyms

<b>AC</b>	actor-critic.
<b>BO</b>	Bayesian optimization.
<b>BOC</b>	Bayesian optimization with unknown constraints.
<b>br</b>	best-response.
<b>CDF</b>	cumulative distribution function.
<b>CHOMP</b>	covariant Hamiltonian optimization for motion planning.
<b>CI</b>	confidence-interval.
<b>COMA</b>	counterfactual multi-agent.
<b>CoP</b>	center of pressure.
<b>CT</b>	control-theory.
<b>DBN</b>	dynamic Bayesian network.
<b>DBSCAN</b>	density-based-spatial-clustering for applications with noise.
<b>DDPG</b>	deep deterministic policy gradient.
<b>DEC-POMDP</b>	decentralized partially observable Markov decision process.
<b>DMP</b>	dynamic movement primitive.
<b>DoF</b>	degree of freedom.
<b>DPG</b>	deterministic policy gradient.
<b>DQN</b>	deep Q-network.
<b>DSA</b>	digital sensor array.
<b>E</b>	efficient policy in Section 3.4.
<b>ebr</b>	$\varepsilon$ -best-response.
<b>EI</b>	expected improvement.
<b>EIC</b>	expected improvement with constraints.
<b>EP</b>	expectation propagation.

<i>Fixed</i>	fixed behavior policy in Section 4.5.
<b>FOL</b>	first-order-logic.
<b>frc-based</b>	force-based strategy in Section 8.4.
<b>FSA</b>	finite-state automaton.
<b>FT</b>	force-torque.
<b>GAE</b>	generalized advantage estimator.
<b>GMM</b>	Gaussian mixture model.
<b>GP</b>	Gaussian process.
<b>GPCR</b>	Gaussian process for classified regression.
<b>GPMP</b>	Gaussian process motion planner.
<b>GPU</b>	graphics processing unit.
<b>GUI</b>	graphical user-interface.
<b>HAC</b>	hierarchical actor-critic.
<b>HATP</b>	hierarchical agent-based task planner.
<b>HER</b>	hindsight experience replay.
<b>HGMDP</b>	hidden goal Markov decision process.
<b>HRC</b>	human-robot collaboration.
<b>HRI</b>	human-robot interaction.
<b>HRL</b>	hierarchical reinforcement learning.
<b>HRT</b>	human-robot team.
<b>hybrid-f-v</b>	hybrid-force-velocity strategy in Section 8.4.
<b>hybrid-v-f</b>	hybrid-velocity-force strategy in Section 8.4.
<b>Hyp</b>	hypothesis.
<b>iLQR</b>	iterative linear-quadratic regulator.
<b>KB</b>	knowledge base.
<b>KL</b>	Kullback–Leibler.
<b>L</b>	hidden goal Markov decision process (HGMDP)-policy in Section 3.4.
<b>LF</b>	partially HGMDP-policy with feedback in Section 3.4.
<b>LfD</b>	learning from demonstration.
<i>Line</i>	spline based human-prediction game-policy in Section 4.5.

<b>LQR</b>	linear-quadratic regulator.
<b>MAAC</b>	multi-actor-attention-critic.
<b>MADDPG</b>	multi-agent deep deterministic policy gradient.
<b>MARL</b>	multi-agent reinforcement learning.
<b>MASAC</b>	multi-agent soft actor-critic.
<b>MCTS</b>	Monte-Carlo tree search.
<b>MDP</b>	Markov decision process.
<b>MG</b>	Markov game.
<b>ML</b>	machine learning.
<b>MOMDP</b>	mixed observable Markov decision process.
<b>MP</b>	manipulation primitive.
<b>MPC</b>	model predictive control.
<b>MPE</b>	multi-agent particle environment.
<b>MPO</b>	maximum a-posteriori policy-optimization.
<b>MRK</b>	Mensch-Roboter-Kollaboration.
<b>MuJoCo</b>	multi-joint dynamics with contact.
<b>NE</b>	Nash-equilibrium.
<b>eNE</b>	$\varepsilon$ -Nash-equilibrium.
<b>NN</b>	neural network.
<b>OC</b>	optimal control.
<b>PDF</b>	probability density function.
<b>PG</b>	policy gradient.
<b>pHRI</b>	physical human-robot interaction.
<b>PI</b>	probability of improvement.
<b>PIBU</b>	probability of improvement with a boundary uncertainty criterion.
<b>POMDP</b>	partially observable Markov decision process.
<b>POMG</b>	partially observable Markov game.
<b>PPO</b>	proximal policy optimization.
<b>Q</b>	question.

<b>RANSAC</b>	random sample consensus.
<b>RAP</b>	relational activity process.
<b>RL</b>	reinforcement learning.
<b>ROS</b>	robot operating system.
<b>SAC</b>	soft actor-critic.
<b>SE</b>	Stackelberg-equilibrium.
<b>SGD</b>	stochastic gradient descent.
<b>SLAM</b>	simultaneous localization and mapping.
<b>SMDP</b>	semi-Markov decision process.
<b>SE(3)</b>	special euclidean group, i.e., the Lie-group to express Cartesian poses.
<b>SO(3)</b>	special orthogonal group, i.e., the Lie-group to express Cartesian attitude.
<i>Spline</i>	spline based human-prediction game-policy in Section 4.5.
<b>STOMP</b>	stochastic trajectory optimization for motion planning.
<b>TAMP</b>	task and motion planning.
<b>TCP</b>	transmission control protocol.
<b>TCP</b>	tool center point.
<b>TD</b>	temporal difference.
<b>TD3</b>	twin delayed deep deterministic policy gradient.
<b>ToM</b>	theory of mind.
<b>TRPO</b>	trust region policy optimization.
<b>UCT</b>	upper confidence bound applied to trees.
<b>UMDP</b>	universal Markov decision process.
<b>URDF</b>	unified robot description format.
<b>UVFA</b>	universal value function approximation.
<b>vel-based</b>	velocity-based strategy in Section 8.4.

## Notation

$\mathbf{p}$	placeholder variable in notation.
$(\cdot)$	blank input to a function, where $\cdot$ represents an arbitrary value of the input.
$\perp$	Boolean <i>false</i> .
$\top$	Boolean <i>true</i> .
$ \mathbf{p} $	obtain the cardinality of a set or vector or space.
$\mathbf{p}^{\otimes}$	best observed data-sample from collected experience data.
$\mathbf{p}^{\ominus}$	worst observed data-sample from collected experience data.
$m, n$	scalar variables to denote dimensions, usually in $\mathbb{N}^+$ .
$:=$	equal by definition.
$\mathcal{N}_{\mathbf{p}}$	Gaussian distribution representing function or variable $\mathbf{p}$ .
$\mathbf{p}^*$	ground truth data within a regression problem or experiment.
$\{^k\}\mathbf{p}$	hierarchical layer-indexing.
$\kappa$	hyper-parameter; indexing defines actual meaning.
$\mathbf{p}_{[k]}$	value of $\mathbf{p}$ at iteration $k$ .
$\mathbf{p}_t$	value of $\mathbf{p}$ at time $t$ .
$i, j, k$	scalar indexing variables, usually in $\mathbb{N}$ .
$\mathbf{p}_i$	content indexing of vectors, lists, sets, e.g., $\mathbf{p}_2$ denotes the second value of $\mathbf{p}$ .
$\wedge$	logical <i>AND</i> operation.
$\vee$	logical <i>OR</i> operation.
$\mathbf{lb}$	lower limit; indexing defines actual meaning.
$\mathbb{1}^{\mathbf{p} \times \mathbf{p}}$	identity matrix of dimension $\mathbf{p} \times \mathbf{p}$ .
$\mathbb{0}^{\mathbf{p} \times \mathbf{p}}$	zero matrix of dimension $\mathbf{p} \times \mathbf{p}$ .
$\mathfrak{P}$	matrix in $\mathbb{R}^{m \times n}$ .
$[\mathfrak{P}]_{(i,j)}$	matrix element, usually $\mathbb{R}^1$ .
$\mathbf{p}^{(i)}$	variable $\mathbf{p}$ assigned to agent $i$ .
$\underline{\mathbf{p}}$	variable $\mathbf{p}$ assigned to all agents.
$\underline{\mathbf{p}}^{(-i)}$	variable $\mathbf{p}$ assigned to all agents except $i$ .
$\mathbb{N}$	natural numbers.
$\mathbb{N}^+$	positive natural numbers.

$\ \mathbf{p}\ _i$	$L_i$ -norm, where $i$ is usually 1 (sum of absolutes) or 2 (euclidean).
$\mathbf{p}^*$	optimal or true value of $\mathbf{p}$ .
$\mathbf{p}^+$	posterior (belief) within Bayesian belief theory, e.g., a Bayesian-filter.
$\mathbf{p}^-$	temporal predecessor of $\mathbf{p}$ , i.e., in a discrete setting $\mathbf{p}^- = \mathbf{p}_{t-1}$ .
$\mathbb{R}$	rational numbers.
$\mathbb{R}^+$	positive non-negative rational numbers.
$\bar{\mathbf{p}}$	set of individual elements $\mathbf{p}_i$ , i.e., $\bar{\mathbf{p}} = \{\mathbf{p}_1, \mathbf{p}_2, \dots\}$ .
$\setminus$	set difference.
$\emptyset$	empty set.
$\cap$	intersection of two sets.
$\cup$	union of two sets.
$\hat{\mathbf{p}}$	estimated value of $\mathbf{p}$ .
$\mathbf{p}'$	temporal successor of $\mathbf{p}$ , i.e., in a discrete setting $\mathbf{p}' = \mathbf{p}_{t+1}$ .
$\dagger\mathbf{p}$	target network of $\mathbf{p}$ , where $\mathbf{p}$ is a neural network.
$\dot{\mathbf{p}}$	temporal derivative of $\mathbf{p}$ .
$\zeta$	threshold-value; indexing defines actual meaning.
$t$	current time or temporal indexing variable.
$T_{\max}$	maximum run-time (continuous) or number of time steps (discrete) in $\mathbb{R}^1$ .
$\vec{\mathbf{p}}$	trajectory as a sequence of $T$ variables $\mathbf{p}$ in $T \times \mathbb{R}^n$ .
$\equiv$	equality of two sets, i.e., $\{\mathbf{p}\}_i \equiv \{\mathbf{p}\}_j := \forall \mathbf{p} \in \{\mathbf{p}\}_i \Leftrightarrow \mathbf{p} \in \{\mathbf{p}\}_j$ .
ub	upper limit; indexing defines actual meaning.
$\mathfrak{A}^T, \mathbf{p}^T$	transpose of a matrix or vector.
$\mathbf{p}$	vector in $\mathbb{R}^n$ .
$\mathbb{1}^{\mathbf{p}}$	identity vector of dimension $\mathbf{p}$ .
$\mathbb{0}^{\mathbf{p}}$	zero vector of dimension $\mathbf{p}$ .



## List of Symbols

### Automated Planning

$K$	planning problem that needs to be solved within the current planning domain.
$P_{\text{prim}}$	finite set of atomic actions within a planning domain.
$E$	entity within planning domain.
$M$	methods, i.e., applicable first-order-logic-functions within planning domain.
$D$	planning domain.

### Control-Theory and System-Theory

$\varepsilon_{\text{ang}}$	angular error.
$c$	scalar constraint value in $\mathbb{R}^1$ .
$\mathbf{c}$	constraint value as a vector in $\mathbb{R}^n$ .
$\mathcal{C}$	set of constraints.
$u$	control input signal.
$\mathbf{C}_{\text{inp}}$	quadratic cost-weight matrix to penalize input-costs, most commonly diagonal.
$\mathbf{C}_{\text{sys}}$	quadratic cost-weight matrix to penalize state-costs, most commonly diagonal.
$v$	damping constant.
$\mathbf{n}$	disturbance / noise signal.
$\varepsilon$	error-vector of desired and current value in $\mathbb{R}^n$ .
$\mathbf{K}_{\text{dmp}}$	impedance controller – damping gain matrix (quadratic, positive semi-definite).
$\mathbf{K}_{\text{I}}^{\text{frc}}$	integral force-controller gain matrix (quadratic, positive semi-definite).
$\mathbf{K}_{\text{P}}^{\text{frc}}$	proportional force-controller gain matrix (quadratic, positive semi-definite).
$\mathbf{K}_{\text{stif}}$	impedance controller – stiffness gain matrix (quadratic, positive semi-definite).
$\mathbf{A}$	linear system dynamics matrix in $\mathbb{R}^{n \times n}$ for a system of state-dimension $n$ .
$\mathbf{B}$	linear system input matrix in $\mathbb{R}^{n \times m}$ , i.e., state-dimension $n$ and control-dimension $m$ .
$m$	mass of an physical entity.
$\mathbf{z}$	measurement signal.
$\mathbf{S}$	hybrid force-/position controller selection matrix.
$s$	diagonal element of the force-/position controller selection matrix $\mathbf{S}$ .

$\ell$  single step cost metric for an iterative optimization objective in  $\mathbb{R}^1$ .

$s_\tau$  time scaling term.

$\delta_t$  update time-step for discrete control processes in  $\mathbb{R}^1$ .

## Graph-Theory

$A_G$  adjacency matrix of a graph  $G$ , where  $[a]_{(i,j)} = 1 \Leftrightarrow \exists e_{i,j} \in \mathcal{E}$ .

$G$  arbitrary graph consisting of vertices  $\mathcal{V}$  and edges  $\mathcal{E}$ .

$e$  specific edge of the edges-set  $\mathcal{E}$  of a graph  $G$ .

$\mathcal{E}$  edge-set of a graph  $G$ .

$v$  specific vertex of the vertex-set  $\mathcal{V}$  of a graph  $G$ .

$\mathcal{V}$  vertex-set of a graph  $G$ .

## Markov Game Variables

$\mathbf{a}$  Action of agent  $i$ .

$\pi$  Action assignment policy that maps a state to an action  $\mathbf{a}$ .

$d$  binary done flag, symbolizing the end of a task.

$\mathbf{r}$  single step reward return.

$o$  environment observation.

$\mathcal{A}$  player or (artificial) agent.

$s$  state value in  $\mathcal{S}$ .

$\mathbf{x}$  fully observable state.

$\mathbf{y}$  hidden state.

$\gamma$  temporal discount factor.

## Machine-Learning and Stochastics

- $\mathbf{b}$  belief, i.e., current PDF of a random variable.
- $\beta$  Boltzmann constant within a Boltzmann distribution.
- $O$  algorithmic complexity (*Big O* - convention).
- $\Sigma$  covariance matrix of a multi-variate probability density function (PDF).
- $\mathcal{D}$  data buffer containing experiences usable for RL.
- $\mathfrak{X}$  observed data samples.
- $\alpha$  entropy temperature paramter for SAC.
- $\Phi$  scalar feature in  $\mathbb{R}^1$ .
- $\Phi$  feature vector or mapping in  $\mathbb{R}^n$ .
- $\mathfrak{K}$  Gram matrix, where  $[\mathfrak{K}]_{(i,j)} = \mathfrak{k}(\xi_i, \xi_j)$ .
- $\mathfrak{k}$   $\mathfrak{k}$  applied on a batch of samples  $\mathbf{p}$ , and  $\xi$ , such that  $[\mathfrak{k}_{\mathbf{p}}(\xi)]_{(i)} = \mathfrak{k}(p_i, \xi)$ .
- $D_{\text{KL}}$  Kullback–Leibler-divergence of two PDFs.
- $\iota$  learning rate for SGD.
- $\mu$  mean of a PDF.
- $\Phi$  normal cumulative distribution function.
- $\nu$  normalizing constant, i.e., for none-zero  $\mathbf{p}$  it holds  $\|\frac{1}{\nu}\mathbf{p}\|_2 = 1$  in  $\mathbb{R}^1$ .
- $\xi$  unknown parameter-vector in  $\mathbb{R}^n$ , obtained by means of regression / optimization.
- $\xi$  unknown scalar parameter in  $\mathbb{R}^1$ ,  $\xi \in \xi$ .
- $\nu$  polyak-averaging weight, e.g., used to update target-network.
- $\delta_{\text{TD}}$  temporal difference-residual as used in GAE.
- $\sigma$  variance of a one-dimensional PDF.

## Constant Numbers

- $N_{\mathfrak{A}}$  number of agents.
- $N_{\text{cell}}$  number of cells.
- $N_{\text{cont}}$  number of evaluation measurements to check against environment contact.
- $N_{\text{cnt}}$  counting variable, i.e., the number a sample has appeared in an iterative algorithm.
- $N_{\text{eps}}$  number of episodes.
- $N_{\text{I}}$  size of sliding window for integration component of PID-control.
- $N_{\text{iter}}$  maximum number of steps within an iterative algorithm.
- $N_k$  number of  $k$ -level decisions.
- $N_{\text{mtrl}}$  number of material types.
- $N_{\text{link}}$  number of link samples.
- $N_{\text{spl}}$  number of samples.
- $N_{\text{step}}$  number of steps, e.g., within an episode.
- $N_{\text{train}}$  number of training-steps.
- $N_{\text{FT}}$  size of sliding window to evaluate data obtained from a FT-sensor.

## Robot-Specific Variables

- $\mathbf{g}$  goal of a given task  $\mathfrak{T}$ , where  $\mathbf{g} \in \mathcal{S}$ .
- $\text{H}$  Human agent.
- $\mathbf{q}$  joint angles of a robot in  $\mathbb{R}^n$ , where  $n$  are the DoFs of the robot.
- $\text{R}$  Robotic agent.
- $\boldsymbol{\tau}$  torque input active on all joints of a robotic system.
- $\mathfrak{T}$  (manipulation / experimental) task.

## State Spaces and Finite Sets

- $\mathcal{X}_{\text{des}}$  desired state space or sub-space of the state space  $\mathcal{X}$ .
- $\mathcal{X}$  fully observable state space.
- $\mathcal{G}$  goal space for the goals of a task or process, where  $\mathcal{G} \in \mathcal{S}$ .
- $\mathcal{Y}$  hidden state space.
- $\mathcal{O}$  observation state space.
- $\mathcal{A}$  generic action space, often discretized.
- $\mathcal{S}$  generic state space, often discretized.

## SE(3)-Variables

- $\delta_x$  Cartesian displacement in SE(3), i.e., in  $\mathbb{R}^1$ .
- $x$  Cartesian pose of and object or the end-effector in SE(3).
- $p$  Cartesian position in  $\mathbb{R}^3$ .
- $F$  Cartesian wrench as force-torque measures in SE(3).
- $e$  SE(3) Coordinate system axis in  $\mathbb{R}^3$ , where  $\|e\|_2 = 1$  holds.
- $x$  Cartesian  $x$ -coordinate in  $\mathbb{R}^1$ . If not phrased explicitly,  ${}^{\text{ba}}x$  is assumed.
- $y$  Cartesian  $y$ -coordinate in  $\mathbb{R}^1$ . If not phrased explicitly,  ${}^{\text{ba}}y$  is assumed.
- $z$  Cartesian  $z$ -coordinate in  $\mathbb{R}^1$ . If not phrased explicitly,  ${}^{\text{ba}}z$  is assumed.
- $e_x$  SE(3) Coordinate system  $x$ -axis in  $\mathbb{R}^3$ .
- $e_y$  SE(3) Coordinate system  $y$ -axis in  $\mathbb{R}^3$ .
- $e_z$  SE(3) Coordinate system  $z$ -axis in  $\mathbb{R}^3$ .
- $f$  force magnitude or scalar force-component of translational component of  $F$  in  $\mathbb{R}^1$ .
- $n$  normal vector of a surface / object in Cartesian space in  $\mathbb{R}^3$ .
- $R$  rotation matrix in SO(3), i.e., in  $\mathbb{R}^{3 \times 3}$ .
- $R_\varphi$  rotation matrix around  $e_x$  in  $\mathbb{R}^{3 \times 3}$ .
- $R_\theta$  rotation matrix around  $e_y$  in  $\mathbb{R}^{3 \times 3}$ .
- $R_\psi$  rotation matrix around  $e_z$  in  $\mathbb{R}^{3 \times 3}$ .
- $\varphi$  roll angle in  $\mathbb{R}^1$ , i.e., angular rotation around  $e_x$ .
- $\theta$  pitch angle in  $\mathbb{R}^1$ , i.e., angular rotation around  $e_y$ .

$\psi$	yaw angle in $\mathbb{R}^1$ , i.e., angular rotation around $e_z$ .
$\tau$	torque magnitude or scalar force-component of rotational component of $\mathbf{F}$ in $\mathbb{R}^1$ .
$\mathbf{T}$	coordinate transformation matrix using homogeneous transformation in SE(3).
$v$	scalar speed, i.e., the magnitude of $\mathbf{v}$ in $\mathbb{R}^1$ .
$\mathbf{v}$	translational velocity in SE(3).

### Variables for Part I

$d$	distance measure in $\mathbb{R}^1$ .
$\alpha_{\text{DMP}}$	$\alpha$ -parameter for dynamic movement primitive (DMP).
$\beta_{\text{DMP}}$	$\beta$ -parameter for DMP.
$\mathcal{F}_{\text{shp}}$	shaping term for a DMP.
$\varepsilon_{\text{br}}$	$\varepsilon$ -value for an $\varepsilon$ -best-response.
$p$	$p$ -value for statistical analysis.
$T_{\text{spl}}$	sampling time.
$\mathbf{t}$	task component.
$\mathbf{\bar{t}}$	task component set.

### Variables for Part II

$\pi_{\text{br}}^{(j)}$	dyadic best-response-policy of agent $j$ to the action of agent $i$ .
$\underline{\pi}_{\text{br}}^{(-i)}$	dyadic best-response-policy of agent <i>nature</i> to the action of agent $i$ .
$\mathbf{e}$	environment-layer index.
$\Theta$	function-approximation parameterization for a critic $\chi$ .
$\Pi$	function-approximation parameterization for a policy $\pi$ .
$\Xi$	function-approximation parameterization for $\pi_{\text{br}}^{(j)}$ or $\underline{\pi}_{\text{br}}^{(-i)}$ .
$\mathbf{i}$	interaction-layer index.

## Variables for Part III

$\mathcal{D}_{\text{art}}$	databuffer with artificially generated data-samples.
$p_{\text{fail}}$	desired probability threshold for posterior of failed failures to be infeasible.
$p_{\text{safe}}$	desired probability threshold for posterior of safe failures to be safe.
$\Delta p$	displacement from first point of contact until force-measurement is received.
$\mathfrak{D}$	gripper DSA-sensor reading.
$\Delta_y$	DSA-sensor cell height.
$N_y$	DSA-sensor resolution width or columns, set to 6 for WSG 50.
$N_x$	DSA-sensor resolution height or rows, set to 14 for WSG 50.
$\Delta_x$	DSA-sensor cell width.
$y_{\text{spl}}$	evaluation of Gaussian process (GP) while applying an acquisition-function $\mathcal{F}_{\text{aqu}}$ .
$\mathbf{g}_{\text{spl}}$	episodic constraint-vector sample.
$\mathcal{J}_{\text{spl}}$	episodic objective sample.
$\mathbf{s}_{\text{spl}}$	episodic success sample, where each scalar evaluates $g_i(\xi) \leq c_i$ .
$\mathfrak{s}$	scaling term $\mathfrak{s} \in [0, 1]$ .
$\vec{c}$	trajectory of cells within an inference-grid $\mathfrak{G}^{\mathfrak{S}}$ in Chapter 7.
$\mathbf{c}$	cell values within an inference-grid $\mathfrak{G}^{\mathfrak{S}}$ in Chapter 7.
$\mathfrak{G}^{\mathfrak{S}}$	Inference grid to regress the $\mathfrak{S}$ in Chapter 7.
$\mathfrak{R}^{\mathfrak{S}}$	Shape particle in Chapter 7.
$\mathfrak{S}$	Shape parameterization in Chapter 7.
$m_{\xi, \mathfrak{G}}$	largest dimension all nodes within a graphical skill-formalism from Chapter 9.
$n_{\xi}$	dimension of the parameter of a regression problem $\xi \in \mathbb{R}^{n_{\xi}}$ .
${}^{\text{le}}\mathfrak{D}$	WSG 50 DSA sensor reading left.
${}^{\text{ri}}\mathfrak{D}$	WSG 50 DSA sensor reading right.
$w_{\text{DSA}}$	width of the DSA sensor array.
$w_{\text{grpr}}$	opening width of a parallel two-finger gripper.

## List of Indices

age	age of subject participants.
align	alignment control-related variable.
ang	angular component of current variable.
art	artificial variable.
avg	average value of current variable.
bkgd	background knowledge of subject participants.
bcnst	belief consistency, i.e., how likely is an entity to diverge from a current belief.
end	candidate(s), i.e., possible solution samples or variables.
cell	cell-related variable, where cell is an element of a grid, e.g., $\mathcal{G}^S$ .
cntr	center of e.g., rigid body, multiple points or line.
col	collision(-cost).
cont	contact with object or environment.
cnt	counting variable.
cur	current value, e.g., measured state of a plant.
dmp	damping-term of mass-spring damper systems or controller-gains.
des	desired value, e.g., a desired trajectory.
dom	dominant, e.g., an action-profile that dominates another in game-theory.
dtct	detection(-time).
dspl	displacement related variable, e.g., a maximum distance.
dst	distance related variable, e.g., distance cost.
eps	current variable is related to current or all episodes.
err	error-term for current value.
estim	estimated value, e.g., the output from a regression.
ext	extrinsic value of current variable.
fail	failed trial / sample.
frc	force / wrench-related variable.
grid	grid related related variable.
grpr	robot gripper related variable.
impls	impulse variable, e.g., force-impulse during contact.



---

insrt	insertion related variable, e.g., time needed for a screwdriver insertion.
inp	input-related variable.
I	integral component of e.g., PID-controller.
int	interactive component of current variable.
iter	iteration related variable, e.g., maximum number of steps.
leader	leader in a leader-follower concept, e.g., in an extensive form game.
leg	legibility-related value or function.
len	length related variable, e.g., path-length of a trajectory.
link	link related variable.
mtrl	material type related variable.
max	maximum value of current variable.
min	minimum value of current variable.
NE	Nash-equilibrium.
nat	native / self-reflective component of current variable.
noise	noise related variable, may be systematic or artificially injected noise.
prm	parameterization mapping of parameter regression(-function).
prnt	parent of current variable.
pareto	Pareto-optimal or dominant related variable, e.g., an action-profile in game-theory.
part	partitioned version of variable. Usually referred to a set or space.
pnlty	penalty term, usually a cost-function or weight.
PI	PI-controller.
pos	position related variable or term.
pre	pre-condition, e.g., within a planning domain.
pref	preference(-cost).
prim	primitive / atomic component of current variable.
P	proportional component of e.g., PID-controller.
rate	update rate in Hz.
rat	rationality of an agent (usually a human).
reach	reaching(-cost).
ba	reference frame – robot base frame.
ct	reference frame – control frame.
le	reference frame – left digital sensor array (DSA)-frame.

<b>ri</b>	reference frame – right DSA-frame.
<b>ec</b>	reference frame – end-effector frame.
<b>to</b>	reference frame – tool center point (TCP) frame.
<b>R</b>	rotated variable.
<b>rot</b>	rotation related variable.
<b>safe</b>	safe variable w.r.t. a constraint-metric.
<b>spl</b>	sampled version of current variable.
<b>shp</b>	shaping(-function).
<b>step</b>	step, e.g., step within a Bayesian-filter update or reinforcement learning (RL)-problem.
<b>stif</b>	stiffness of mass-spring damper systems or controller-gains.
<b>sub</b>	sub-group of current variable.
<b>suc</b>	indicating success for the current task or episode.
<b>sys</b>	system dependent variable.
<b>temp</b>	temporal related variable.
<b>time</b>	time related variable, e.g., duration of an action.
<b>train</b>	training(-step).
<b>trans</b>	translatoric related variable.
<b>travel</b>	travel(-cost).
<b>vel</b>	velocity-related variable.

## List of Operators and Functions

$\mathcal{F}_{\text{aqu}}$	acquisition function to generate new data samples in $\mathbb{R}^n$ .
<b>A</b>	advantage function as the difference of <b>Q</b> and <b>V</b> in $\mathbb{R}^1$ .
$\mathcal{F}_{\text{align}}$	alignment control function in $\mathbb{R}^n$ .
$\mathcal{S}$	black box success function in $\mathbb{R}^1$ .
$\mathcal{R}$	black box reward function in $\mathbb{R}^1$ .
$g$	scalar inequality constraint in $\mathbb{R}^1$ .
<b>g</b>	inequality constraint vector in $\mathbb{R}^n$ .
<b>diag</b>	get diagonal elements from a matrix as vector in $\mathbb{R}^n$ .
$\rho$	PDF over a random variable.

---

$\mathbb{H}$	entropy of a random variable / PDF in $\mathbb{R}^1$ .
$\mathcal{F}_{\text{equ}}$	equivalence relation in $\mathbb{R}^1$ .
$\mathcal{F}_{\text{estim}}$	estimation / regression function in $\mathbb{R}^n$ .
$\mathbb{E}$	expectation of a random variable / distribution.
$\tilde{\mathcal{F}}$	function estimation for an unknown system or process mapping to $\mathbb{R}^1$ .
$\mathcal{N}$	Gaussian PDF.
$\mathcal{F}_{\mathcal{N}}$	Gaussian shaped function in $\mathbb{R}^1$ .
$\chi$	General critic-function in $\mathbb{R}^1$ , e.g., $\mathbf{V}$ , $\mathbf{A}$ or $\mathbf{Q}$ .
$\mathcal{F}_{\mathcal{S}}$	geometric shape estimation function in $\mathbb{R}^n$ .
$\mathcal{F}_{\mathbf{g}}$	goal mapping function, maps state / observation to a goal $\mathbf{g} \in \mathcal{G} \in \mathbb{R}^n$ .
$\nabla$	gradient over a function or vector in $\mathbb{R}^n$ or matrix in $\mathbb{R}^{n \times n}$ .
$\mathcal{F}_{\mathbb{H}}$	Heaviside function in $\mathbb{R}^1$ , i.e., $\mathcal{F}_{\mathbb{H}}(\mathbf{p}) = 1 \Leftrightarrow \mathbf{p} > 0$ .
$\Lambda_{\xi}^{\vee}$	joint success-probability for a sequential manipulation task.
$\mathfrak{k}$	kernel function $\mathfrak{k}(\xi_i, \xi_j)$ in $\mathbb{R}^1$ .
$\mathcal{L}$	loss function in $\mathbb{R}^1$ .
$\tilde{\mathcal{F}}_{\mathcal{J}}$	model of the actual objective-function $\mathcal{J}$ in $\mathbb{R}^1$ .
$\tilde{\mathcal{F}}_{\mathbf{g}}$	model of a success-function, i.e., a binary function mapping $\mathbb{R}^n \mapsto \top, \perp$ .
$\mathcal{O}$	observation transition function.
$\mathcal{J}$	general objective function for an optimization problem in $\mathbb{R}^1$ .
$\mathcal{F}^{\text{pareto}}$	Evaluates if an action-profile is pareto-dominated by another action-profile.
$\mathcal{F}_{\text{dom}}^{\text{pareto}}$	Evaluates if an action-profile is pareto-dominant.
$\mathcal{F}_{\text{prm}}$	parameter regression operator in $\mathbb{R}^n$ .
$\mathbb{P}$	probability of a random variable / distribution.
$\mathbf{Q}$	Q-function in $\mathbb{R}^1$ , also known as state-action-value function.
$f_{\Pi}$	deterministic function to apply the <i>reparameterization trick</i> , cf. soft actor-critic (SAC).
$\mathcal{F}_{\text{RPY}}$	transform current orientation representation into roll pitch yaw, i.e., $\varphi, \theta, \psi$ in $\mathbb{R}^3$ .
sign	return sign of function or scalar variable in $\mathbb{R}^1$ .
$\mathcal{T}$	stochastic transition function.
$\Gamma_{\xi}^{\vee}$	success-probability of an MP-node in $\mathbb{R}^1$ .
$\mathcal{U}$	utility function in $\mathbb{R}^1$ .
$\mathcal{U}_{\mathbf{c}}$	utility function for a cell within $\mathcal{G}^{\mathbf{S}}$ in $\mathbb{R}^1$ .
$\mathcal{U}_{\mathbf{S}}$	utility function for a shape-particle $\mathcal{A}^{\mathbf{S}}$ in $\mathbb{R}^1$ .

*List of Operators and Functions*

---

$\mathcal{U}_{\text{reach}}$  reachability utility function in  $\mathbb{R}^1$ .

$V$  value function in  $\mathbb{R}^1$ .

$\text{Var}$  variance of a random variable / distribution  $\text{Var} \in \mathbb{R}^1$ .

# List of Algorithms

5.1	Approximate solution for strategic robot policy . . . . .	60
6.1	Decentralized br-policy based MARL-algorithm . . . . .	78
6.2	Proposed hierarchical MARL algorithmic skeleton . . . . .	81
9.1	Induce artificial data to constraint GP . . . . .	137
9.2	Proposed BOC-algorithm . . . . .	139

# List of Figures

1.1	Topological overview of this thesis . . . . .	8
2.1	Perception-cognition-action loop for a generic HRC-scenario . . . . .	15
2.2	Toy example for an HRC-planning problem . . . . .	24
3.1	HGMDP-DBN . . . . .	29
3.2	HRC-experiment setup . . . . .	33
3.3	Visualization of the three pick-and-place goals for the two scenarios . . . . .	34
3.4	Answers for each question grouped by three different modes: L, E and LF . . . . .	35
3.5	Quantitative measures for the three robot decision-making modes . . . . .	36
3.6	Belief settling proportion . . . . .	37
4.1	Interactive action-selection – illustrative example . . . . .	40
4.2	Overview of the proposed HRC-game decision framework . . . . .	43
4.3	Collision evaluation approximation . . . . .	48
4.4	Box-plot results of collected subjective feedback . . . . .	51
4.5	Experimental performance metrics . . . . .	52
6.1	General MARL problem . . . . .	72
6.2	Exemplary hierarchical MARL-step . . . . .	76
6.3	Results of the decentralized br-based algorithms . . . . .	84
7.1	Haptic explorer – framework overview . . . . .	96
7.2	Utility and accessibility calculation visualization . . . . .	97
7.3	Choice of next action . . . . .	99
7.4	Robot in simulation environment . . . . .	100
7.5	Estimation of $f(\kappa_{stif}^{MuJoCo})$ . . . . .	101
7.6	Evaluation objects . . . . .	102
7.7	Classification of data measurements (grid) . . . . .	103

7.8	Classification of data measurements (shape) . . . . .	103
7.9	Evolution of the estimation error for the shape-based method . . . . .	104
8.1	Adaptive grasping strategy using tactile sensor data . . . . .	108
8.2	DSA cells with exemplary pressure data . . . . .	111
8.3	Alignment error estimation visualization . . . . .	112
8.4	Schematic overview of the proposed adaptive grasping controller . . . . .	114
8.5	Empirical comparison of the available control modes w.r.t. communication speed	117
8.6	Experimental evaluation setup . . . . .	118
8.7	Cartesian wrench profiles for the lightbulb removal task . . . . .	120
9.1	Schematic skill-formalism for manipulation tasks . . . . .	130
9.2	Schematic overview of the hybrid force-velocity controller . . . . .	134
9.3	Proposed graphical skill-formalism including the success probability . . . . .	136
9.4	Screwdriver insertion skill as an MP-graph . . . . .	142
9.5	Success probability of the unscrewing skill as a factor graph . . . . .	145
9.6	Regret evolution of the experimental screw-insertion task . . . . .	147
9.7	Number of safe samples for the experimental screw-insertion task . . . . .	148
9.8	Optimal sample-values for the screw-insertion task . . . . .	149

## List of Tables

3.1	Questionnaire . . . . .	34
3.2	Subjective evaluation . . . . .	36
4.1	Questionnaire . . . . .	49
4.2	Evaluated questionnaire data . . . . .	51
6.1	Detailed performance metrics for evaluated environments . . . . .	85
7.1	Estimated stiffness for both methods and objects . . . . .	104
8.1	Average ROS-update rate in Hz . . . . .	116
8.2	Cross-comparison of the proposed grasping controller against baseline . . . . .	119
9.1	Skill-specific controller-parameters $\xi$ . . . . .	146
9.2	Skill-parameters chosen by an expert . . . . .	147
9.3	Successful data-samples . . . . .	148
B.1	Usage of open-source modules . . . . .	163
B.2	Hyper-parameters for the experimental evaluation . . . . .	164
B.3	Environment parameters for the MPE . . . . .	164
B.4	Hyper-parameters for each algorithm . . . . .	165

## References

- ABBEEL, P. AND A. Y. NG (2004): “Apprenticeship learning via inverse reinforcement learning,” in *International Conference on Machine Learning (ICML)*, ed. by C. E. Brodley, ACM, vol. 69 of *ACM International Conference Proceeding Series*.
- ABERDEEN, D. (2003): “A (revised) survey of approximate methods for solving partially observable Markov decision processes,” Tech. rep., Citeseer.
- ACHIAM, J. (2018): “Spinning Up in Deep Reinforcement Learning,” .
- ACKERMANN, J. (2018): “Hierarchical Deep Reinforcement Learning for Multi-Agent Robotic Systems,” Bachelor thesis, Technical University of Munich.
- ACKERMANN, J., V. GABLER, T. OSA, AND M. SUGIYAMA (2019): “Reducing Overestimation Bias in Multi-Agent Domains Using Double Centralized Critics,” *CoRR*, abs/1910.01465.
- AGGARWAL, J. K. AND Q. CAI (1999): “Human Motion Analysis: A Review,” *Comput. Vis. Image Underst.*, 73, 428–440.
- ALLEN, P. K. AND K. S. ROBERTS (1989): “Haptic object recognition using a multi-fingered dextrous hand,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE Computer Society, 342–347.
- ALONSO-MORA, J., J. A. DECASTRO, V. RAMAN, D. RUS, AND H. KRESS-GAZIT (2018): “Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles,” *Autonomous Robots*, 42, 801–824.
- ALT, B., D. KATIC, R. JÄKEL, A. K. BOZCUOGLU, AND M. BEETZ (2021): “Robot Program Parameter Inference via Differentiable Shadow Program Inversion,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 4672–4678.
- ALTHOFF, M. (2010): “Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars,” Dissertation, Technische Universität München, München.
- AMATO, C., D. S. BERNSTEIN, AND S. ZILBERSTEIN (2007): “Optimizing Memory-Bounded Controllers for Decentralized POMDPs,” in *Conference on Uncertainty in Artificial Intelligence*, ed. by R. Parr and L. C. van der Gaag, AUAI Press, 1–8.
- AMATO, C., G. D. KONIDARIS, L. P. KAEHLING, AND J. P. HOW (2019): “Modeling and Planning with Macro-Actions in Decentralized POMDPs,” *Journal of Artificial Intelligence Research*, 64, 817–859.
- AMBIKASARAN, S., D. FOREMAN-MACKEY, L. GREENGARD, D. W. HOGG, AND M. O’NEIL (2016): “Fast Direct Methods for Gaussian Processes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 252–265.
- AMES, B. AND G. D. KONIDARIS (2019): “Bounded-Error LQR-Trees,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 144–150.

- AMOR, H. B., G. NEUMANN, S. KAMTHE, O. KROEMER, AND J. PETERS (2014): “Interaction primitives for human-robot cooperation tasks,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2831–2837.
- AMOS, B., I. D. J. RODRIGUEZ, J. SACKS, B. BOOTS, AND J. Z. KOLTER (2018): “Differentiable MPC for End-to-end Planning and Control,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, 8299–8310.
- ANDERSON, C. A. AND L. ROSS (1980): “Perseverance of social theories: The role of explanation in the persistence of discredited information,” *Journal of Personality and Social Psychology*, 39, 1037–1049.
- ANDERSSON, J., J. ÅKESSON, AND M. DIEHL (2012): “CasADi: A Symbolic Package for Automatic Differentiation and Optimal Control,” in *Recent Adv. Algorithmic Differ.*, ed. by S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, Berlin, Heidelberg: Springer Berlin Heidelberg, 297–307.
- ANDRYCHOWICZ, M., D. CROW, A. RAY, ET AL. (2017): “Hindsight Experience Replay,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 5048–5058.
- ARGALL, B. D., S. CHERNOVA, M. M. VELOSO, AND B. BROWNING (2009): “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems*, 57, 469–483.
- ARULKUMARAN, K., M. P. DEISENROTH, M. BRUNDAGE, AND A. A. BHARATH (2017): “Deep Reinforcement Learning: A Brief Survey,” *IEEE Signal Processing Magazine*, 34, 26–38.
- ARULKUMARAN, K., N. DILOKTHANAKUL, M. SHANAHAN, AND A. A. BHARATH (2016): “Classifying Options for Deep Reinforcement Learning,” *CoRR*, abs/1604.08153.
- ASIMOV, I. (1941): “Three laws of robotics,” *Asimov, I. Runaround*.
- ATKESON, C. G. AND S. SCHAAL (1997): “Robot Learning From Demonstration,” in *International Conference on Machine Learning (ICML)*, ed. by D. H. Fisher, Morgan Kaufmann, 12–20.
- BACON, P., J. HARB, AND D. PRECUP (2017): “The Option-Critic Architecture,” in *AAAI Conference on Artificial Intelligence*, 1726–1734.
- BAHRAM, M., A. LAWITZKY, J. FRIEDRICHS, M. AEBERHARD, AND D. WOLLHERR (2015): “A Game Theoretic Approach to Replanning-aware Interactive Scene Prediction and Planning,” *IEEE Transactions on Vehicular Technology*, 65, 3981–3992.
- BAI, S., J. Z. KOLTER, AND V. KOLTUN (2019): “Deep Equilibrium Models,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, 688–699.
- BAKER, C. L., R. SAXE, AND J. B. TENENBAUM (2009): “Action understanding as inverse planning,” *Cognition*, 113, 329–349, reinforcement learning and higher cognition.
- BAKER, C. L., J. B. TENENBAUM, AND R. R. SAXE (2007): “Goal inference as inverse planning,” in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 29.



- BALANDAT, M., B. KARRER, D. R. JIANG, S. DAULTON, B. LETHAM, A. G. WILSON, AND E. BAKSHY (2019): “BoTorch: Programmable Bayesian Optimization in PyTorch,” *arxiv e-prints*.
- BARAGLIA, J., M. CAKMAK, Y. NAGAI, R. P. N. RAO, AND M. ASADA (2017): “Efficient human-robot collaboration: When should a robot take initiative?” *International Journal of Robotics Research*, 36, 563–579.
- BARI, S., V. GABLER, AND D. WOLLHERR (2021): “MS2MP: A Min-Sum Message Passing Algorithm for Motion Planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Xi’an, China: IEEE, 7887–7893.
- BAUER, A., D. WOLLHERR, AND M. BUSS (2008): “Human-Robot Collaboration: a Survey.” *International Journal of Humanoid Robotics*, 5, 47–66.
- BAUMANN, D., A. MARCO, M. TURCHETTA, AND S. TRIMPE (2021): “GoSafe: Globally Optimal Safe Robot Learning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 4452–4458.
- BEHBAHANI, F. M. P. (2016): “Reverse-engineering the visual and haptic perceptual algorithms in the brain,” Ph.D. thesis, Imperial College London, UK.
- BEHBAHANI, F. M. P., G. SINGLA-BUXARRAIS, AND A. A. FAISAL (2016): “Haptic SLAM: An Ideal Observer Model for Bayesian Inference of Object Shape and Hand Pose from Contact Dynamics,” in *International Conference on Haptics: Perception, Devices, Control, and Applications (EuroHaptics)*, ed. by F. Bello, H. Kajimoto, and Y. Visell, Springer, vol. 9774 of *Lecture Notes in Computer Science*, 146–157.
- BEHBAHANI, F. M. P., R. TAUNTON, A. A. C. THOMIK, AND A. A. FAISAL (2015): “Haptic SLAM for context-aware robotic hand prosthetics - simultaneous inference of hand pose and object shape using particle filters,” in *IEEE/EMBS Conference on Neural Engineering (NER)*, IEEE, 719–722.
- BEKIROGLU, Y., J. LAAKSONEN, J. A. JØRGENSEN, V. KYRKI, AND D. KRAGIC (2011): “Assessing Grasp Stability Based on Learning and Haptic Data,” *IEEE Transactions on Robotics*, 27, 616–629.
- BELLMAN, R. (1957): “A Markovian decision process,” *Journal of mathematics and mechanics*, 679–684.
- BELTRAN-HERNANDEZ, C. C., D. PETIT, I. G. RAMIREZ-ALPIZAR, T. NISHI, S. KIKUCHI, T. MATSUBARA, AND K. HARADA (2020): “Learning Force Control for Contact-Rich Manipulation Tasks With Rigid Position-Controlled Robots,” *IEEE Robotics and Automation Letters*, 5, 5709–5716.
- BERKENKAMP, F., A. KRAUSE, AND A. P. SCHOELLIG (2016a): “Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics,” *CoRR*, abs/1602.04450.
- BERKENKAMP, F., A. P. SCHOELLIG, AND A. KRAUSE (2016b): “Safe controller optimization for quadrotors with Gaussian processes,” in *IEEE International Conference on Robotics and Automation (ICRA)*, ed. by D. Kragic, A. Bicchi, and A. D. Luca, IEEE, 491–496.

- BEVANDA, P., S. SOSNOWSKI, AND S. HIRCHE (2021): “Koopman operator dynamical models: Learning, analysis and control,” *Annual Review of Control, Robotics, and Autonomous Systems*, 52, 197–212.
- BIOENGINEERING AND ROBOTICS RESEARCH CENTER ”E. PIAGGIO” UNIVERSITY OF PISA (2014): “KUKA LWR 4+,” <https://github.com/CentroEPiaggio/kuka-lwr/>, [Online; accessed 06-August-2022].
- BOHG, J., A. MORALES, T. ASFOUR, AND D. KRAGIC (2014): “Data-Driven Grasp Synthesis - A Survey,” *IEEE Transactions on Robotics*, 30, 289–309.
- BOURGAULT, F., A. MAKARENKO, S. B. WILLIAMS, B. GROCHOLSKY, AND H. F. DURRANT-WHYTE (2002): “Information based adaptive robotic exploration,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 540–545.
- BRETON, M., A. ALJ, AND A. HAURIE (1988): “Sequential Stackelberg equilibria in two-person games,” *Journal of Optimization Theory and Applications*, 59, 71–97.
- BROCKMAN, G., V. CHEUNG, L. PETTERSSON, J. SCHNEIDER, J. SCHULMAN, J. TANG, AND W. ZAREMBA (2016): “OpenAI Gym,” .
- BROEKENS, J., M. HEERINK, H. ROSENDAL, ET AL. (2009): “Assistive social robots in elderly care: a review,” *Gerontechnology*, 8, 94–103.
- BROZ, F., I. NOURBAKSH, AND R. SIMMONS (2013): “Planning for Human–Robot Interaction in Socially Situated Tasks,” *International Journal of Social Robotics*, 5, 193–214.
- BRUYNINCKX, H., S. DUTRÉ, AND J. D. SCHUTTER (1995): “Peg-on-Hole: A Model Based Solution to Peg and Hole Alignment,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE Computer Society, 1919–1924.
- BUSCH, B., M. TOUSSAINT, AND M. LOPES (2018): “Planning Ergonomic Sequences of Actions in Human-Robot Interaction,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 1916–1923.
- CALANDRA, R., A. SEYFARTH, J. PETERS, AND M. P. DEISENROTH (2016): “Bayesian optimization for learning gaits under uncertainty - An experimental comparison on a dynamic bipedal walker,” *Ann. Math. Artif. Intell.*, 76, 5–23.
- CHAO, C. AND A. THOMAZ (2016): “Timed Petri nets for fluent turn-taking over multimodal interaction resources in human-robot collaboration,” *International Journal of Robotics Research*, 35, 1330–1353.
- CHEN, F., K. SEKIYAMA, F. CANNELLA, AND T. FUKUDA (2014): “Optimal Subtask Allocation for Human and Robot Collaboration Within Hybrid Assembly System,” *IEEE Transactions on Automation Science and Engineering*, 11, 1065–1075.
- CHEN, M., S. NIKOLAIDIS, H. SOH, D. HSU, AND S. SRINIVASA (2018): “Planning with Trust for Human-Robot Collaboration,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, New York, NY, USA: ACM, HRI ’18, 307–315.
- CHEN, S., Y. LI, AND N. M. KWOK (2011): “Active vision in robotic systems: A survey of recent developments,” *Journal of Artificial Intelligence Research*, 30, 1343–1377.

- CHITTA, S., E. MARDER-EPPSTEIN, W. MEEUSSEN, V. PRADEEP, A. RODRÍGUEZ TSOUROUKDISSIAN, J. BOHREN, D. COLEMAN, B. MAGYAR, G. RAIOLA, M. LÜDTKE, AND E. FERNÁNDEZ PERDOMO (2017): “ros\_control: A generic and simple control framework for ROS,” *The Journal of Open Source Software*.
- CHO, N. J., S. H. LEE, J. B. KIM, AND I. H. SUH (2020): “Learning, Improving, and Generalizing Motor Skills for the Peg-in-Hole Tasks Based on Imitation Learning and Self-Learning,” *Applied Sciences*, 10.
- CHOU DHURY, R., G. SWAMY, D. HADFIELD-MENELL, AND A. D. DRAGAN (2019): “On the Utility of Model Learning in HRI,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 317–325.
- COLEMAN, D., I. A. SUCAN, S. CHITTA, AND N. CORRELL (2014): “Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study,” *CoRR*, abs/1404.3785.
- CORKE, P. AND J. HAVILAND (2021): “Not your grandmother’s toolbox—the Robotics Toolbox reinvented for Python,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 11357–11363.
- COUMANS, E. AND Y. BAI (2016–2020): “PyBullet, a Python module for physics simulation for games, robotics and machine learning,” <http://pybullet.org>, [Online; accessed 06-August-2022].
- CRAIG, J. J. AND M. H. RAIBERT (1979): “A systematic method of hybrid position/force control of a manipulator,” in *The IEEE Computer Society’s Third International Computer Software and Applications Conference, COMPSAC 1979, 6-8 November, 1979, Chicago, Illinois, USA*, IEEE, 446–451.
- CSIBRA, G. AND G. GERGELY (2007): “‘Obsessed with goals’: Functions and mechanisms of teleological interpretation of actions in humans,” *Acta psychologica*, 124, 60–78.
- DANG, H. AND P. K. ALLEN (2014): “Stable grasping under pose uncertainty using tactile feedback,” *Autonomous Robots*, 36, 309–330.
- DARVISH, K., E. SIMETTI, F. MASTROGIOVANNI, AND G. CASALINO (2021): “A Hierarchical Architecture for Human-Robot Cooperation Processes,” *IEEE Transactions on Robotics*, 37, 567–586.
- DE AVILA BELBUTE-PERES, F., K. A. SMITH, K. R. ALLEN, J. TENENBAUM, AND J. Z. KOLTER (2018): “End-to-End Differentiable Physics for Learning and Control,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, 7178–7189.
- DE SILVA, L., R. LALLEMENT, AND R. ALAMI (2015): “The HATP hierarchical planner: Formalisation and an initial study of its usability and practicality,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, Hamburg, 6465–6472.
- DECHERCHI, S., P. GASTALDO, R. S. DAHIYA, M. VALLE, AND R. ZUNINO (2011): “Tactile-Data Classification of Contact Materials Using Computational Intelligence,” *IEEE Transactions on Robotics*, 27, 635–639.

- DEISENROTH, M. P., D. FOX, AND C. E. RASMUSSEN (2015): “Gaussian Processes for Data-Efficient Learning in Robotics and Control,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 408–423.
- DELHAISSE, B., L. ROZO, AND D. G. CALDWELL (2019): “PyRoboLearn: A Python Framework for Robot Learning Practitioners,” in *Conference on Robot Learning (CoRL)*, Osaka, Japan.
- DELLAERT, F. AND M. KAESS (2017): “Factor Graphs for Robot Perception,” *Found. Trends Robotics*, 6, 1–139.
- DEMIR, S. O., U. CULHA, A. C. KARACAKOL, A. PENNA-FRANCESCH, S. TRIMPE, AND M. SITTI (2021): “Task space adaptation via the learning of gait controllers of magnetic soft millirobots,” *International Journal of Robotics Research*, 40, 1331–1351, PMID: 35481277.
- DENIŠA, M., A. GAMS, A. UDE, AND T. PETRIČ (2016): “Learning Compliant Movement Primitives Through Demonstration and Statistical Generalization,” *IEEE Transactions on Mechatronics*, 21, 2581–2594.
- DEVIN, C., A. GUPTA, T. DARRELL, P. ABBEEL, AND S. LEVINE (2017): “Learning modular neural network policies for multi-task and multi-robot transfer,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2169–2176.
- DEWOLF, T. (2012–2022): “Studywolf – a blog for things I encounter while coding and researching neuroscience, motor control, and learning,” <https://studywolf.wordpress.com>, [Online; accessed 06-June-2022].
- DHARIWAL, P., C. HESSE, O. KLIMOV, A. NICHOL, M. PLAPPERT, A. RADFORD, J. SCHULMAN, S. SIDOR, Y. WU, AND P. ZHOKHOV (2017): “OpenAI Baselines,” <https://github.com/openai/baselines>, [Online; accessed 01-June-2022].
- DIBANGOYE, J. S., C. AMATO, O. BUFFET, AND F. CHARPILLET (2016): “Optimally Solving Dec-POMDPs as Continuous-State MDPs,” *Journal of Artificial Intelligence Research*, 55, 443–497.
- DINH, K. H., O. OGUZ, G. HUBER, V. GABLER, AND D. WOLLHERR (2015): “An approach to integrate human motion prediction into local obstacle avoidance in close human-robot collaboration,” in *IEEE Workshop on Advanced Robotics and its Social Impact (ARSO)*, Lyon, France.
- DINH, K. H., O. S. OGUZ, M. ELSAYED, AND D. WOLLHERR (2019): “Adaptation and Transfer of Robot Motion Policies for Close Proximity Human-Robot Interaction,” *Frontiers in Robotics and AI*, 6, 69.
- DOLLAR, A. M., L. P. JENTOFT, J. H. GAO, AND R. D. HOWE (2010): “Contact sensing and grasping performance of compliant hands,” *Autonomous Robots*, 28, 65–75.
- DRAGAN, A. D., S. BAUMAN, J. FORLIZZI, AND S. S. SRINIVASA (2015a): “Effects of Robot Motion on Human-Robot Collaboration,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, ed. by J. A. Adams, W. D. Smart, B. Mutlu, and L. Takayama, ACM, 51–58.
- DRAGAN, A. D., R. M. HOLLADAY, AND S. S. SRINIVASA (2015b): “Deceptive robot motion: synthesis, analysis and experiments,” *Autonomous Robots*, 39, 331–345.

- DRAGAN, A. D., K. C. T. LEE, AND S. S. SRINIVASA (2013): “Legibility and predictability of robot motion,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, ed. by H. Kuzuoka, V. Evers, M. Imai, and J. Forlizzi, IEEE/ACM, 301–308.
- DRAGAN, A. D. AND S. S. SRINIVASA (2013): “Generating Legible Motion,” in *Robotics: Science and Systems (RSS)*, ed. by P. Newman, D. Fox, and D. Hsu.
- (2014): “Integrating human observer inferences into robot motion planning,” *Autonomous Robots*, 37, 351–368.
- DRAGIEV, S., M. TOUSSAINT, AND M. GIENGER (2013): “Uncertainty aware grasping and tactile exploration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 113–119.
- DRIESS, D., P. ENGLERT, AND M. TOUSSAINT (2017): “Constrained Bayesian optimization of combined interaction force/task space controllers for manipulations,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 902–907.
- DURRANT-WHYTE, H. F. AND T. BAILEY (2006): “Simultaneous localization and mapping: part I,” *IEEE Robotics and Automation Magazine*, 13, 99–110.
- DZEROSKI, S., L. D. RAEDT, AND K. DRIESSENS (2001): “Relational Reinforcement Learning,” *Mach. Learn.*, 43, 7–52.
- E. ROHMER S. P. N. SINGH, M. F. (2013): “V-REP: a Versatile and Scalable Robot Simulation Framework,” in *Proc. Int. Conf. Intell. Robot. Syst.*
- EBERMAN, B. S. AND J. K. S. JR. (1994): “Application of Change Detection to Dynamic Control Contact Sensing,” *International Journal of Robotics Research*, 13, 369–394.
- ELFES, A. (1989): “Using Occupancy Grids for Mobile Robot Perception and Navigation,” *Computer*, 22, 46–57.
- (1995): “Robot Navigation: Integrating Perception, Environmental Constraints and Task Execution Within a Probabilistic Framework,” in *Reasoning with Uncertainty in Robotics: International Workshop (RUR)*, ed. by L. Dorst, M. van Lambalgen, and F. Voorbraak, Springer, vol. 1093 of *Lecture Notes in Computer Science*, 93–130.
- EMERY-MONTEMERLO, R. (2005): “Game Theoretic Control for Robot Teams,” Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, aAI3188918.
- ENGLERT, P., I. M. RAYAS FERNÁNDEZ, R. K. RAMACHANDRAN, AND G. SUKHATME (2021): “Sampling-Based Motion Planning on Sequenced Manifolds,” in *Robotics: Science and Systems (RSS)*, Virtual.
- ENGLERT, P. AND M. TOUSSAINT (2016): “Combined Optimization and Reinforcement Learning for Manipulation Skills,” in *Robotics: Science and Systems (RSS)*, ed. by D. Hsu, N. M. Amato, S. Berman, and S. A. Jacobs.
- (2018): “Learning manipulation skills from a single demonstration,” *International Journal of Robotics Research*, 37, 137–154.
- ESPIAU, B., F. CHAUMETTE, AND P. RIVES (1992): “A new approach to visual servoing in robotics,” *IEEE Transactions on Robotics and Automation*, 8, 313–326.

- EWERTON, M., G. NEUMANN, R. LIOUTIKOV, H. B. AMOR, J. PETERS, AND G. MAEDA (2015): “Learning multiple collaborative tasks with a mixture of Interaction Primitives,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 1535–1542.
- EYSENBACH, B., R. SALAKHUTDINOV, AND S. LEVINE (2019): “Search on the Replay Buffer: Bridging Planning and Reinforcement Learning,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, 15220–15231.
- FAN, S., H. GU, Y. ZHANG, M. JIN, AND H. LIU (2018): “Research on adaptive grasping with object pose uncertainty by multi-fingered robot hand,” *International Journal of Advanced Robotic Systems*, 15, 1729881418766783.
- FENG, L., C. WILTSCHKE, L. R. HUMPHREY, AND U. TOPCU (2016): “Synthesis of Human-in-the-Loop Control Protocols for Autonomous Systems,” *IEEE Transactions on Automation Science and Engineering*, 13, 450–462.
- FERN, A., S. NATARAJAN, K. JUDAH, AND P. TADEPALLI (2014): “A Decision-Theoretic Model of Assistance,” *Journal of Artificial Intelligence Research*, 50, 71–104.
- FIKES, R. AND N. J. NILSSON (1971): “STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving,” *Artif. Intell.*, 2, 189–208.
- FINZI, A. AND T. LUKASIEWICZ (2004): “Relational Markov Games,” in *European Conference on Logics in Artificial Intelligence (JELIA)*, 320–333.
- FISAC, J. F., E. BRONSTEIN, E. STEFANSSON, D. SADIGH, S. S. SASTRY, AND A. D. DRAGAN (2019): “Hierarchical Game-Theoretic Planning for Autonomous Vehicles,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 9590–9596.
- FISHEL, J. A. AND G. E. LOEB (2012): “Bayesian Exploration for Intelligent Identification of Textures,” *Frontiers in Neurobotics*, 6, 4.
- FLACCO, F., T. KROEGER, A. D. LUCA, AND O. KHATIB (2015): “A Depth Space Approach for Evaluating Distance to Objects - with Application to Human-Robot Collision Avoidance,” *J. Intell. Robotic Syst.*, 80, 7–22.
- FLASH, T. AND N. HOGAN (1985): “The coordination of arm movements: an experimentally confirmed mathematical model,” *The journal of Neuroscience*, 5, 1688–1703.
- FOERSTER, J. N., Y. M. ASSAEL, N. DE FREITAS, AND S. WHITESON (2016): “Learning to Communicate with Deep Multi-Agent Reinforcement Learning,” *CoRR*, abs/1605.06676.
- FOERSTER, J. N., G. FARQUHAR, T. AFOURAS, N. NARDELLI, AND S. WHITESON (2018): “Counterfactual Multi-Agent Policy Gradients,” in *AAAI Conference on Artificial Intelligence*, ed. by S. A. McIlraith and K. Q. Weinberger, AAAI Press, 2974–2982.
- FOX, E. AND F. L. H. III (2020): “Soft Variable Stiffness Joints for Controllable Grasp Synergies in Underactuated Robotic Hands,” in *IEEE International Conference on Soft Robotics (RoboSoft)*, IEEE, 586–592.
- FRANS, K., J. HO, X. CHEN, P. ABBEEL, AND J. SCHULMAN (2018): “Meta Learning Shared Hierarchies,” in *International Conference on Learning Representations (ICLR)*.

- FRENCH, R. M. (2000): “The Turing Test: the first 50 years,” *Trends in Cognitive Sciences*, 4, 115–122.
- FRIDOVICH-KEIL, D., A. BAJCSY, J. F. FISAC, S. L. HERBERT, S. WANG, A. D. DRAGAN, AND C. J. TOMLIN (2020): “Confidence-aware motion prediction for real-time collision avoidance<sup>1</sup>,” *International Journal of Robotics Research*, 39.
- FRIEDL, K. E., A. R. VOELKER, A. PEER, AND C. ELIASMITH (2016): “Human-Inspired Neurobotic System for Classifying Surface Textures by Touch,” *IEEE Robotics and Automation Letters*, 516–523.
- GABLER, V., G. HUBER, M. BOSCH, AND D. GIAKOUMIS (2020a): “D6.2 - Haptic Regression Report,” Tech. rep., HR-Recycler - Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment.
- GABLER, V., G. HUBER, S. ENDO, D. WOLLHERR, A. TISSOT, AND I. FREIRE GONZALEZ (2022a): “D6.1 - Force Guided Manipulation Evaluation,” Tech. rep., HR-Recycler - Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment.
- GABLER, V., G. HUBER, AND D. WOLLHERR (2022b): “A Force-Sensitive Grasping Controller Using Tactile Gripper Fingers and an Industrial Position-Controlled Robot,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia: IEEE, 770–776.
- GABLER, V., K. MAIER, S. ENDO, AND D. WOLLHERR (2020b): “Haptic Object Identification for Advanced Manipulation Skills,” in *International Conference on Biomimetic and Biohybrid Systems (Living Machines)*, ed. by V. Vouloutsi, A. Mura, F. J. Esser, T. Speck, T. J. Prescott, and P. F. M. J. Verschure, Springer, vol. 12413 of *Lecture Notes in Computer Science*, 128–140.
- GABLER, V., T. STAHL, G. HUBER, O. OGUZ, AND D. WOLLHERR (2017): “A Game-Theoretic Approach for Adaptive Action Selection in Close Distance Human-Robot Collaboration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore: IEEE, 2897–2903.
- GABLER, V. AND D. WOLLHERR (2022): “Bayesian Optimization with Unknown Constraints in Graphical Skill-Models for Compliant Manipulation Tasks Using an Industrial Robot,” *Frontiers Robotics AI*, 9.
- (2023): “Decentralized Multi-Agent Reinforcement Learning Based on Best-Response Policies,” *Frontiers Robotics AI*, submitted.
- GARDNER, J. R., G. PLEISS, D. BINDEL, K. Q. WEINBERGER, AND A. G. WILSON (2018): “GPYtorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*.
- GARG, N. P., D. HSU, AND W. S. LEE (2019): “Learning To Grasp Under Uncertainty Using POMDPs,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2751–2757.

- GARG, S., N. SÜNDERHAUF, F. DAYOUB, D. MORRISON, A. COSGUN, G. CARNEIRO, Q. WU, T. CHIN, I. D. REID, S. GOULD, P. CORKE, AND M. MILFORD (2020): “Semantics for Robotic Mapping, Perception and Interaction: A Survey,” *Found. Trends Robotics*, 8, 1–224.
- GARRETT, C. R., R. CHITNIS, R. HOLLADAY, B. KIM, T. SILVER, L. P. KAEHLING, AND T. LOZANO-PÉREZ (2021): “Integrated Task and Motion Planning,” *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 265–293.
- GEIGER, P. AND C. STRAEHLE (2021): “Learning Game-Theoretic Models of Multiagent Trajectories Using Implicit Layers,” in *AAAI Conference on Artificial Intelligence*, AAAI Press, 4950–4958.
- GERGELY, G. AND G. CSIBRA (2003): “Teleological reasoning in infancy: The naive theory of rational action,” *Trends in Cognitive Sciences*, 7, 287–292.
- GERGELY, G., Z. NÁDASDY, G. CSIBRA, AND S. BÍRÓ (1995): “Taking the intentional stance at 12 months of age,” *Cognition*, 56, 165–193.
- GOMBOLAY, M. C., R. A. GUTIERREZ, S. G. CLARKE, D. STURL, AND J. A. SHAH (2015): “Decision-Making Authority, Team Efficiency and Human Worker Satisfaction in Mixed Human-Robot Teams,” *Autonomous Robots*.
- GOMBOLAY, M. C., R. JENSEN, J. STIGILE, T. GOLEN, N. SHAH, S. SON, AND J. A. SHAH (2018): “Human-Machine Collaborative Optimization via Apprenticeship Scheduling,” *Journal of Artificial Intelligence Research*, 63, 1–49.
- GROSZ, B. J. (1996): “AAAI-94 Presidential Address: Collaborative Systems,” *AI Magazine*, 17, 67–85.
- GUENNEBAUD, G., B. JACOB, ET AL. (2010): “Eigen v3,” <http://eigen.tuxfamily.org>, [Online; accessed 06-August-2022].
- GULLAPALLI, V., J. A. FRANKLIN, AND H. BENBRAHIM (1994): “Acquiring robot skills via reinforcement learning,” *IEEE Control Systems Magazine*, 14, 13–24.
- GULLAPALLI, V., R. A. GRUPEN, AND A. G. BARTO (1992): “Learning reactive admittance control,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE Computer Society, 1475–1480.
- GUO, D., T. KONG, F. SUN, AND H. LIU (2016): “Object discovery and grasp detection with a shared convolutional neural network,” in *IEEE International Conference on Robotics and Automation (ICRA)*, ed. by D. Kragic, A. Bicchi, and A. D. Luca, IEEE, 2038–2043.
- GUO, D., F. SUN, T. KONG, AND H. LIU (2017): “Deep vision networks for real-time robotic grasp detection,” *International Journal of Advanced Robotic Systems*, 14, 1729881416682706.
- GUO, M. AND D. V. DIMAROGONAS (2017): “Task and Motion Coordination for Heterogeneous Multiagent Systems With Loosely Coupled Local Tasks,” *IEEE Transactions on Automation Science and Engineering*, 14, 797–808.



- GUPTA, A., R. MENDONCA, Y. LIU, P. ABBEEL, AND S. LEVINE (2018): “Meta-Reinforcement Learning of Structured Exploration Strategies,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, 5307–5316.
- HAARNOJA, T. (2018): “Acquiring Diverse Robot Skills via Maximum Entropy Deep Reinforcement Learning,” Ph.D. thesis, University of California, Berkeley, USA.
- HAARNOJA, T., A. ZHOU, K. HARTIKAINEN, G. TUCKER, S. HA, J. TAN, V. KUMAR, H. ZHU, A. GUPTA, P. ABBEEL, AND S. LEVINE (2018): “Soft Actor-Critic Algorithms and Applications,” *CoRR*, abs/1812.05905.
- HADDADIN, S., A. D. LUCA, AND A. ALBU-SCHÄFFER (2017): “Robot Collisions: A Survey on Detection, Isolation, and Identification,” *IEEE Transactions on Robotics*, 33, 1292–1312.
- HADFIELD-MENELL, D., A. D. DRAGAN, P. ABBEEL, AND S. RUSSELL (2017a): “The Off-Switch Game,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, ed. by C. Sierra, ijcai.org, 220–227.
- HADFIELD-MENELL, D., E. GROSHEV, R. CHITNIS, AND P. ABBEEL (2015): “Modular task and motion planning in belief space,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 4991–4998.
- HADFIELD-MENELL, D., L. P. KAEHLING, AND T. LOZANO-PÉREZ (2013): “Optimization in the now: Dynamic peephole optimization for hierarchical planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 4560–4567.
- HADFIELD-MENELL, D., C. LIN, R. CHITNIS, S. RUSSELL, AND P. ABBEEL (2016a): “Sequential quadratic programming for task plan optimization,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 5040–5047.
- HADFIELD-MENELL, D., S. MILLI, P. ABBEEL, S. J. RUSSELL, AND A. D. DRAGAN (2017b): “Inverse Reward Design,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, 6765–6774.
- HADFIELD-MENELL, D., S. RUSSELL, P. ABBEEL, AND A. D. DRAGAN (2016b): “Cooperative Inverse Reinforcement Learning,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, 3909–3917.
- HALL, R. A. (2021): “Maschinenmensch/Maria,” *Robots in Popular Culture: Androids and Cyborgs in the American Imagination*, 204.
- HAMAYA, M., R. LEE, K. TANAKA, F. VON DRIGALSKI, C. NAKASHIMA, Y. SHIBATA, AND Y. IJIRI (2020): “Learning Robotic Assembly Tasks with Lower Dimensional Systems by Leveraging Physical Softness and Environmental Constraints,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 7747–7753.
- HARKINS, W. E. (1962): “Karel Čapek,” in *Karel Čapek*, Columbia University Press.

- HARRIS, C. R., K. J. MILLMAN, S. J. VAN DER WALT, R. GOMMERS, P. VIRTANEN, D. COURNAPEAU, E. WIESER, J. TAYLOR, S. BERG, N. J. SMITH, R. KERN, M. PICUS, S. HOYER, M. H. VAN KERKWIJK, M. BRETT, A. HALDANE, J. F. DEL RÍO, M. WIEBE, P. PETERSON, P. GÉRARD-MARCHANT, K. SHEPPARD, T. REDDY, W. WECKESSER, H. ABBASI, C. GOHLKE, AND T. E. OLIPHANT (2020): “Array programming with NumPy,” *Nature*, 585, 357–362.
- HARTMANN, V. N., O. S. OGUZ, D. DRIESS, M. TOUSSAINT, AND A. MENGES (2020): “Robust Task and Motion Planning for Long-Horizon Architectural Construction Planning,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 6886–6893.
- HAUSMAN, K., J. MÜLLER, A. HARIHARAN, N. AYANIAN, AND G. S. SUKHATME (2015): “Cooperative multi-robot control for target tracking with onboard sensing,” *International Journal of Robotics Research*, 34, 1660–1677.
- HAVRYLOV, S. AND I. TITOV (2017): “Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols,” *CoRR*, abs/1705.11192.
- HAWKINS, K. P., S. BANSAL, N. N. VO, AND A. F. BOBICK (2014): “Anticipating human actions for collaboration in the presence of task and sensor uncertainty,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2215–2222.
- HAYNE, R., R. LUO, AND D. BERENSON (2016): “Considering avoidance and consistency in motion planning for human-robot manipulation in a shared workspace,” in *IEEE International Conference on Robotics and Automation (ICRA)*, ed. by D. Kragic, A. Bicchi, and A. D. Luca, IEEE, 3948–3954.
- HE, Z., L. DONG, C. SONG, AND C. SUN (2021): “Multi-agent Soft Actor-Critic Based Hybrid Motion Planner for Mobile Robots,” *CoRR*, abs/2112.06594.
- HEGAZY, D. AND J. DENZLER (2009): “Combining Appearance and Range Based Information for Multi-class Generic Object Recognition,” in *Iberoamerican Congress on Pattern Recognition (CIARP)*, ed. by E. Bayro-Corrochano and J. Eklundh, Springer, vol. 5856 of *Lecture Notes in Computer Science*, 741–748.
- HERNANDEZ-LEAL, P., B. KARTAL, AND M. E. TAYLOR (2019): “A survey and critique of multiagent deep reinforcement learning,” *Auton. Agents Multi Agent Syst.*, 33, 750–797.
- (2020): “A Very Condensed Survey and Critique of Multiagent Deep Reinforcement Learning,” in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, ed. by A. E. F. Seghrouchni, G. Sukthankar, B. An, and N. Yorke-Smith, International Foundation for Autonomous Agents and Multiagent Systems, 2146–2148.
- HIATT, L. M., C. NARBER, E. BEKELE, S. S. KHEMLANI, AND J. G. TRAFTON (2017): “Human modeling for human-robot collaboration,” *Journal of Artificial Intelligence Research*, 36, 580–596.
- HODGKINSON, G. P. (1997): “Cognitive inertia in a turbulent market: The case of UK residential estate agents,” *Journal of Management Studies*, 34, 921–945.
- HOFF, K. A. AND M. N. BASHIR (2015): “Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust,” *Hum. Factors*, 57, 407–434.

- HOFFMAN, G. AND C. BREAZEAL (2007): “Cost-Based Anticipatory Action Selection for Human-Robot Fluency,” *IEEE Transactions on Robotics*, 23, 952–961.
- HOGAN, N. (1984): “Impedance control: An approach to manipulation,” in *IEEE American Control Conference (ACC)*, IEEE, 304–313.
- HOWE, R. D., N. POPP, P. AKELLA, I. KAO, AND M. R. CUTKOSKY (1990): “Grasping, manipulation, and control with tactile sensing,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 1258–1263.
- HSIAO, K., S. CHITTA, M. T. CIOCARLIE, AND E. G. JONES (2010): “Contact-reactive grasping of objects with partial shape information,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 1228–1235.
- HSIAO, K., L. P. KAEHLING, AND T. LOZANO-PÉREZ (2007): “Grasping POMDPs,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 4685–4692.
- HSIAO, K., P. NANGERONI, M. HUBER, A. SAXENA, AND A. Y. NG (2009): “Reactive grasping using optical proximity sensors,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2098–2105.
- HU, B. AND J. CHEN (2017): “Optimal Task Allocation for Human–Machine Collaborative Manufacturing Systems,” *IEEE Robotics and Automation Letters*, 2, 1933–1940.
- HUNTER, J. D. (2007): “Matplotlib: A 2D graphics environment,” *Computing in science & engineering*, 9, 90–95.
- IJSPEERT, A. J., J. NAKANISHI, H. HOFFMANN, P. PASTOR, AND S. SCHAAL (2013): “Dynamical movement primitives: learning attractor models for motor behaviors,” *Neural computation*, 25, 328–373.
- INOUE, T., G. D. MAGISTRIS, A. MUNAWAR, T. YOKOYA, AND R. TACHIBANA (2017): “Deep reinforcement learning for high precision assembly tasks,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 819–825.
- IQBAL, S. AND F. SHA (2019): “Actor-Attention-Critic for Multi-Agent Reinforcement Learning,” in *International Conference on Machine Learning (ICML)*, ed. by K. Chaudhuri and R. Salakhutdinov, PMLR, vol. 97 of *Proceedings of Machine Learning Research*, 2961–2970.
- JARRASSÉ, N., T. CHARALAMBOUS, AND E. BURDET (2012): “A Framework to Describe, Analyze and Generate Interactive Motor Behaviors,” *PLoS ONE*, 7.
- JIAO, Q., H. MODARES, S. XU, F. L. LEWIS, AND K. G. VAMVOUDAKIS (2016): “Multi-agent zero-sum differential graphical games for disturbance rejection in distributed control,” *Automatica*, 69, 24–34.
- JOHANNSMEIER, L., M. GERCHOW, AND S. HADDADIN (2019): “A Framework for Robot Manipulation: skill-formalism, Meta Learning and Adaptive Control,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5844–5850.
- JOHANNSMEIER, L. AND S. HADDADIN (2017): “A Hierarchical Human-Robot Interaction-Planning Framework for Task Allocation in Collaborative Industrial Assembly Processes,” *IEEE Robotics and Automation Letters*, 2, 41–48.

- JOHANSSON, R. S. AND J. R. FLANAGAN (2009): “Coding and use of tactile signals from the fingertips in object manipulation tasks,” *Nature Reviews Neuroscience*, 10, 345–359.
- JULIAN, B. J., M. ANGERMANN, M. SCHWAGER, AND D. RUS (2012): “Distributed robotic sensor networks: An information-theoretic approach,” *International Journal of Robotics Research*, 31, 1134–1154.
- KACHOUIE, R., S. SEDIGHADELI, R. KHOSLA, AND M. CHU (2014): “Socially Assistive Robots in Elderly Care: A Mixed-Method Systematic Literature Review,” *Int. J. Hum. Comput. Interact.*, 30, 369–393.
- KAEHLING, L. P., M. L. LITTMAN, AND A. W. MOORE (1996): “Reinforcement Learning: A Survey,” *Journal of Artificial Intelligence Research*, 4, 237–285.
- KAEHLING, L. P. AND T. LOZANO-PÉREZ (2013): “Integrated task and motion planning in belief space,” *International Journal of Robotics Research*, 32, 1194–1227.
- KALAKRISHNAN, M., S. CHITTA, E. A. THEODOROU, P. PASTOR, AND S. SCHAAL (2011): “STOMP: Stochastic trajectory optimization for motion planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 4569–4574.
- KAMEWARI, K., M. KATO, T. KANDA, H. ISHIGURO, AND K. HIRAKI (2005): “Six-and-a-half-month-old children positively attribute goals to human action and to humanoid-robot motion,” *Cognitive Development*, 20, 303–320.
- KEARNS, M. (2007): “Graphical Games,” *Algorithmic Game Theory*, 159–178.
- KERZ, S., J. TEUTSCH, T. BRÜDIGAM, D. WOLLHERR, AND M. LEIBOLD (2021): “Data-driven stochastic model predictive control,” *CoRR*, abs/2112.04439.
- KHATIB, O. AND J. BURDICK (1986): “Motion and force control of robot manipulators,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 1381–1386.
- KINGMA, D. P. AND J. BA (2015): “Adam: A Method for Stochastic Optimization,” in *International Conference on Learning Representations (ICLR)*, ed. by Y. Bengio and Y. LeCun.
- KIS, A., F. KOVÁCS, AND P. SZOLGAY (2006): “Grasp planning based on fingertip contact forces and torques,” in *International Conference on Haptics: Perception, Devices, Control, and Applications (EuroHaptics)*, Citeseer, 455–458.
- KLAUS GREFF, AARON KLEIN, MARTIN CHOVANEC, FRANK HUTTER, AND JÜRGEN SCHMIDHUBER (2017): “The Sacred Infrastructure for Computational Research,” in *Proceedings of the 16th Python in Science Conference*, ed. by Katy Huff, David Lippa, Dillon Niederhut, and M. Pacer, 49 – 56.
- KLUYVER, T., B. RAGAN-KELLEY, F. PÉREZ, B. GRANGER, M. BUSSONNIER, J. FREDERIC, K. KELLEY, J. HAMRICK, J. GROUT, S. CORLAY, P. IVANOV, D. AVILA, S. ABDALLA, AND C. WILLING (2016): “Jupyter Notebooks – a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, ed. by F. Loizides and B. Schmidt, IOS Press, 87 – 90.
- KOBER, J., J. A. BAGNELL, AND J. PETERS (2013): “Reinforcement learning in robotics: A survey,” *Journal of Artificial Intelligence Research*, 32, 1238–1274.

- KOENIG, N. AND A. HOWARD (2004): “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, vol. 3, 2149–2154.
- KOLTER, J. Z. AND A. Y. NG (2009): “Policy search via the signed derivative,” in *Robotics: Science and Systems (RSS)*.
- KÖNIG, C., M. KHOSRAVI, M. MAIER, R. S. SMITH, A. RUPENYAN, AND J. LYGEROS (2020): “Safety-Aware Cascade Controller Tuning Using Constrained Bayesian Optimization,” *CoRR*, abs/2010.15211.
- KOPPULA, H. S. AND A. SAXENA (2016): “Anticipating Human Activities Using Object Affordances for Reactive Robotic Response,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 14–29.
- KOUSTOUMPARDIS, P. AND N. ASPRAGATHOS (2004): “A review of gripping devices for fabric handling,” *hand*, 19, 20.
- KRAMBERGER, A., A. GAMS, B. NEMEC, C. SCHOU, D. CHRYSOSTOMOU, O. MADSEN, AND A. UDE (2016): “Transfer of contact skills to new environmental conditions,” in *IEEE-RAS International Workshop on Humanoid Robots (Humanoids)*, IEEE, 668–675.
- KROCKENBERGER, D. (2019): “Hierarchical Deep Reinforcement Learning for Multi-Agent Robotic Systems,” Master thesis, Technical University of Munich.
- KRUT, S. (2005): “A Force-Isotropic Underactuated Finger,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2314–2319.
- KSCHISCHANG, F. R., B. J. FREY, AND H. LOELIGER (2001): “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, 47, 498–519.
- KUIPERS, B., E. A. FEIGENBAUM, P. E. HART, AND N. J. NILSSON (2017): “Shakey: from conception to history,” *AI Magazine*, 38, 88–103.
- KULESHOV, V. AND O. SCHRIJVERS (2015): “Inverse Game Theory: Learning Utilities in Succinct Games,” in *Conference on Web and Internet Economics (WINE)*, ed. by E. Markakis and G. Schäfer, Springer, vol. 9470 of *Lecture Notes in Computer Science*, 413–427.
- KULIC, D. AND E. A. CROFT (2005): “Safe planning for human-robot interaction,” *J. Field Robotics*, 22, 383–396.
- (2006): “Real-time safety for human-robot interaction,” *Robotics and Autonomous Systems*, 54, 1–12.
- (2007): “Pre-collision safety strategies for human-robot interaction,” *Autonomous Robots*, 22, 149–164.
- KULKARNI, T. D., K. NARASIMHAN, A. SAEEDI, AND J. TENENBAUM (2016): “Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, 3675–3683.
- KUMAR, A., S. ZILBERSTEIN, AND M. TOUSSAINT (2015): “Probabilistic Inference Techniques for Scalable Multiagent Decision Making,” *Journal of Artificial Intelligence Research*, 53, 223–270.

- LAGRASSA, A., S. LEE, AND O. KROEMER (2020): “Learning Skills to Patch Plans Based on Inaccurate Models,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 9441–9448.
- LAGRIFFOUL, F., D. DIMITROV, J. BIDOT, A. SAFFIOTTI, AND L. KARLSSON (2014): “Efficiently combining task and motion planning using geometric constraints,” *International Journal of Robotics Research*, 33, 1726–1747.
- LAMPOR, L. (1986): “LATEX: A Document Preparation System, Addison,” .
- LANCTOT, M., V. F. ZAMBALDI, A. GRUSLYS, A. LAZARIDOU, K. TUYLS, J. PÉROLAT, D. SILVER, AND T. GRAEPEL (2017): “A Unified Game-Theoretic Approach to Multi-agent Reinforcement Learning,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, 4190–4203.
- LASOTA, P. A., T. FONG, AND J. A. SHAH (2017): “A Survey of Methods for Safe Human-Robot Interaction,” *Found. Trends Robotics*, 5, 261–349.
- LASOTA, P. A. AND J. A. SHAH (2015): “Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human-Robot Collaboration,” *Hum. Factors*, 57, 21–33.
- LAURENT, G. J., L. MATIGNON, AND N. L. FORT-PIAT (2011): “The world of independent learners is not markovian,” *Int. J. Knowl. Based Intell. Eng. Syst.*, 15, 55–64.
- LEEPER, A., K. HSIAO, E. CHU, AND J. K. SALISBURY (2010): “Using Near-Field Stereo Vision for Robotic Grasping in Cluttered Environments,” in *International Symposium on Experimental Robotics (ISER)*, ed. by O. Khatib, V. Kumar, and G. S. Sukhatme, Springer, vol. 79 of *Springer Tracts in Advanced Robotics*, 253–267.
- LEIBO, J. Z., E. D. NEZ GUZMÁN, A. S. VEZHNEVETS, J. P. AGAPIOU, P. SUNEHAG, R. KOSTER, J. MATYAS, C. BEATTIE, I. MORDATCH, AND T. GRAEPEL (2021): “Scalable Evaluation of Multi-Agent Reinforcement Learning with Melting Pot,” PMLR.
- LENTON, D., F. PARDO, F. FALCK, S. JAMES, AND R. CLARK (2021): “Ivy: Templated deep learning for inter-framework portability,” *arXiv preprint arXiv:2102.02886*.
- LEURENT, E. (2018): “An Environment for Autonomous Driving Decision-Making,” <https://github.com/eleurent/highway-env>, [Online; accessed 01-May-2022].
- LEVINE, S., C. FINN, T. DARRELL, AND P. ABBEEL (2016): “End-to-End Training of Deep Visuomotor Policies,” *Journal of Machine Learning Research*, 17, 39:1–39:40.
- LEVINE, S., N. WAGENER, AND P. ABBEEL (2015): “Learning contact-rich manipulation skills with guided policy search,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 156–163.
- LEVY, A., G. KONIDARIS, R. P. JR., AND K. SAENKO (2019): “Learning Multi-Level Hierarchies with Hindsight,” in *International Conference on Learning Representations (ICLR)*.
- LEWIS, F. L., H. ZHANG, K. HENGSTER-MOVRIC, AND A. DAS (2013): *Cooperative control of multi-agent systems: optimal and adaptive design approaches*, Springer Science & Business Media.

- LEWKOWICZ, D., Y. DELEVOYE-TURRELL, D. BAILLY, P. ANDRY, AND P. GAUSSIER (2013): “Reading motor intention through mental imagery,” *Adapt. Behav.*, 21, 315–327.
- LI, N., D. W. OYLER, M. ZHANG, Y. YILDIZ, I. V. KOLMANOVSKY, AND A. R. GIRARD (2018a): “Game Theoretic Modeling of Driver and Vehicle Interactions for Verification and Validation of Autonomous Vehicle Control Systems,” *IEEE Transactions on Cybernetics*, 26, 1782–1797.
- LI, Q., O. KROEMER, Z. SU, F. VEIGA, M. KABOLI, AND H. J. RITTER (2020): “A Review of Tactile Information: Perception and Action Through Touch,” *IEEE Transactions on Robotics*, 36, 1619–1634.
- LI, Q., L. NATALE, R. HASCHKE, A. CHERUBINI, A. V. HO, AND H. J. RITTER (2018b): “Tactile Sensing for Manipulation,” *International Journal of Humanoid Robotics*, 15, 1802001:1–1802001:3.
- LI, S., Y. WU, X. CUI, H. DONG, F. FANG, AND S. J. RUSSELL (2019): “Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient,” in *AAAI Conference on Artificial Intelligence*, 4213–4220.
- LI, Y., G. GOWRISHANKAR, N. JARRASSÉ, S. HADDADIN, A. ALBU-SCHÄFFER, AND E. BURDET (2018c): “Force, Impedance, and Trajectory Learning for Contact Tooling and Haptic Identification,” *IEEE Transactions on Robotics*, 34, 1170–1182.
- LI, Y., K. P. TEE, W. L. CHAN, R. YAN, Y. CHUA, AND D. K. LIMBU (2015): “Role Adaptation of Human and Robot in Collaborative Tasks,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 5602–5607.
- LI, Y., K. P. TEE, R. YAN, W. L. CHAN, AND Y. WU (2016): “A Framework of Human-Robot Coordination Based on Game Theory and Policy Iteration,” *IEEE Transactions on Robotics*, 32, 1408–1418.
- LILLICRAP, T. P., J. J. HUNT, A. PRITZEL, ET AL. (2016): “Continuous control with deep reinforcement learning,” in *International Conference on Learning Representations (ICLR)*.
- LING, C. K., F. FANG, AND J. Z. KOLTER (2018): “What Game Are We Playing? End-to-end Learning in Normal and Extensive Form Games,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, ed. by J. Lang, ijcai.org, 396–402.
- LIOUTIKOV, R., G. NEUMANN, G. MAEDA, AND J. PETERS (2017): “Learning movement primitive libraries through probabilistic segmentation,” *Journal of Artificial Intelligence Research*, 36, 879–894.
- LITTMAN, M. L. (1994): “Markov Games as a Framework for Multi-Agent Reinforcement Learning,” in *International Conference on Machine Learning (ICML)*, 157–163.
- LLOYD, J. W. (1987): *Foundations of Logic Programming, 2nd Edition*, Springer.
- LOWE, R., Y. WU, A. TAMAR, ET AL. (2017): “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 6379–6390.
- LUCIA, S., A. TĂTULEA-CODREAN, C. SCHOPPMAYER, AND S. ENGELL (2017): “Rapid development of modular and sustainable nonlinear model predictive control solutions,” *Control Eng. Pract.*, 60, 51–62.

- LUO, J., E. SOLOWJOW, C. WEN, J. A. OJEA, A. M. AGOGINO, A. TAMAR, AND P. ABBEEL (2019): “Reinforcement Learning on Variable Impedance Controller for High-Precision Robotic Assembly,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 3080–3087.
- LUO, S., J. BIMBO, R. DAHIYA, AND H. LIU (2017): “Robotic tactile perception of object properties: A review,” *Mechatronics*, 48, 54–67.
- MA, L., Z. WANG, Q. HAN, AND Y. LIU (2017): “Consensus control of stochastic multi-agent systems: a survey,” *Sci. China Inf. Sci.*, 60, 120201:1–120201:15.
- MA, R. R., L. ODHNER, AND A. M. DOLLAR (2013): “A modular, open-source 3D printed underactuated hand,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2737–2743.
- MACKENZIE, C. L. AND T. IBERALL (1994): *The grasping hand*, Elsevier.
- MAEDA, G., G. NEUMANN, M. EWERTON, R. LIOUTIKOV, O. KROEMER, AND J. PETERS (2017): “Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks,” *Autonomous Robots*, 41, 593–612.
- MAIER, K. (2019): “Object Identification for Advanced Manipulation Skills Using Haptic SLAM,” Master thesis, Technical University of Munich.
- MAINPRICE, J. AND D. BERENSON (2013): “Human-robot collaborative manipulation planning using early prediction of human motion,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, 299–306.
- MAINPRICE, J., M. GHARBI, T. SIMÉON, AND R. ALAMI (2012): “Sharing effort in planning human-robot handover tasks,” in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, IEEE, 764–770.
- MAINPRICE, J., R. HAYNE, AND D. BERENSON (2016): “Goal Set Inverse Optimal Control and Iterative Replanning for Predicting Human Reaching Motions in Shared Workspaces,” *IEEE Transactions on Robotics*, 32, 897–908.
- MAINPRICE, J., E. A. SISBOT, L. JAILLET, J. CORTÉS, R. ALAMI, AND T. SIMÉON (2011): “Planning human-aware motions using a sampling-based costmap planner,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5012–5017.
- MALIK, D., M. PALANIAPPAN, J. F. FISAC, D. HADFIELD-MENELL, S. RUSSELL, AND A. D. DRAGAN (2018): “An Efficient, Generalized Bellman Update For Cooperative Inverse Reinforcement Learning,” in *International Conference on Machine Learning (ICML)*, ed. by J. G. Dy and A. Krause, PMLR, vol. 80 of *Proceedings of Machine Learning Research*, 3391–3399.
- MARCO, A., D. BAUMANN, M. KHADIV, P. HENNIG, L. RIGHETTI, AND S. TRIMPE (2021): “Robot Learning With Crash Constraints,” *IEEE Robotics and Automation Letters*, 6, 1439–1446.
- MARTÍN-MARTÍN, R., M. A. LEE, R. GARDNER, S. SAVARESE, J. BOHG, AND A. GARG (2019): “Variable Impedance Control in End-Effector Space: An Action Space for Reinforcement Learning in Contact-Rich Tasks,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 1010–1017.



- MARTINEZ-HERNANDEZ, U., G. METTA, T. J. DODD, T. J. PRESCOTT, L. NATALE, AND N. F. LEPORA (2013): “Active contour following to explore object shape with robot touch,” in *World Haptics Conference (WHC)*, IEEE, 341–346.
- MASON, M. T., A. RODRIGUEZ, S. S. SRINIVASA, AND A. S. VÁZQUEZ (2012): “Autonomous manipulation with a general-purpose simple hand,” *International Journal of Robotics Research*, 31, 688–703.
- MASON, M. T., S. S. SRINIVASA, AND A. S. VÁZQUEZ (2009): “Generality and Simple Hands,” in *International Symposium on Robotics Research (ISRR)*, ed. by C. Pradalier, R. Siegwart, and G. Hirzinger, Springer, vol. 70 of *Springer Tracts in Advanced Robotics*, 345–361.
- MAUSAM AND D. S. WELD (2008): “Planning with Durative Actions in Stochastic Domains,” *Journal of Artificial Intelligence Research*, 31, 33–82.
- MCCLOSKEY, M. AND N. J. COHEN (1989): “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of Learning and Motivation*, Elsevier, vol. 24, 109–165.
- MCKINNEY, W. ET AL. (2010): “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, Austin, TX, vol. 445, 51–56.
- MELO, F. S. AND M. I. RIBEIRO (2006): “Transition Entropy in Partially Observable Markov Decision Processes,” in *International Conference on Intelligent Autonomous Systems*, ed. by T. Arai, R. Pfeifer, T. R. Balch, and H. Yokoi, IOS Press, 282–289.
- MERCADO, V., M. MARCHAL, AND A. LÉCUYER (2021): “Haptics On-Demand”: A Survey on Encountered-Type Haptic Displays,” *IEEE Transactions on Haptics*, 14, 449–464.
- MERKEL, D. (2014): “Docker: lightweight linux containers for consistent development and deployment,” *Linux journal*, 2014, 2.
- MERZIC, H., M. BOGDANOVIC, D. KAPPLER, L. RIGHETTI, AND J. BOHG (2019): “Leveraging Contact Forces for Learning to Grasp,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 3615–3621.
- MEURER, A., C. P. SMITH, M. PAPROCKI, O. ČERTÍK, S. B. KIRPICHEV, M. ROCKLIN, A. KUMAR, S. IVANOV, J. K. MOORE, S. SINGH, ET AL. (2017): “SymPy: symbolic computing in Python,” *PeerJ Computer Science*, 3, e103.
- MIDDLETONE, R. AND G. C. GOODWIN (1986): “Adaptive computed torque control for rigid link manipulators,” in *IEEE Conference on Decision and Control (CDC)*, IEEE, 68–73.
- MISSIONI, I., P. TAJVAR, D. KRAGIC, J. TUMOVA, AND C. PEK (2021): “Safe Data-Driven Contact-Rich Manipulation,” in *IEEE-RAS International Workshop on Humanoid Robots (Humanoids)*, IEEE, 120–127.
- MNIH, V., K. KAVUKCUOGLU, D. SILVER, A. A. RUSU, J. VENESS, M. G. BELLEMARE, A. GRAVES, M. A. RIEDMILLER, A. FIDJELAND, G. OSTROVSKI, S. PETERSEN, C. BEATTIE, A. SADIK, I. ANTONOGLU, H. KING, D. KUMARAN, D. WIERSTRA, S. LEGG, AND D. HASSABIS (2015): “Human-level control through deep reinforcement learning,” *Nature*, 518, 529–533.

- MOHAMMADI, A., J. LAVRANOS, R. HOWE, P. CHOONG, AND D. OETOMO (2017): “Grasp specific and user friendly interface design for myoelectric hand prostheses,” in *International Conference on Rehabilitation Robotics (ICORR)*, IEEE, 1621–1626.
- MONTEMERLO, M., S. THRUN, D. KOLLER, AND B. WEGBREIT (2002): “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem,” in *AAAI Conference on Artificial Intelligence*, ed. by R. Dechter, M. J. Kearns, and R. S. Sutton, AAAI Press / The MIT Press, 593–598.
- MORDATCH, I. AND P. ABBEEL (2017): “Emergence of Grounded Compositional Language in Multi-Agent Populations,” *CoRR*, abs/1703.04908.
- MORITZ, P., R. NISHIHARA, S. WANG, A. TUMANOV, R. LIAW, E. LIANG, M. ELIBOL, Z. YANG, W. PAUL, M. I. JORDAN, AND I. STOICA (2018): “Ray: A Distributed Framework for Emerging AI Applications,” in *13th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2018, Carlsbad, CA, USA, October 8-10, 2018*, ed. by A. C. Arpaci-Dusseau and G. Voelker, USENIX Association, 561–577.
- MUKADAM, M., J. DONG, X. YAN, F. DELLAERT, AND B. BOOTS (2018): “Continuous-time Gaussian process motion planning via probabilistic inference,” *International Journal of Robotics Research*, 37.
- MUNZER, T., M. TOUSSAINT, AND M. LOPES (2017): “Efficient behavior learning in human–robot collaboration,” *Autonomous Robots*.
- NASH, J. (1950): “Equilibrium Points in N-Person Games,” *Proceedings of the National Academy of Sciences*, 36, 48–49.
- (1951): “Non-Cooperative Games,” *Annals of Mathematics*, 286–295.
- NATALE, L. AND E. TORRES-JARA (2006): “A sensitive approach to grasping,” in *International Workshop on Epigenetic Robotics*, Citeseer, 87–94.
- NATARAJAN, S., S. JOSHI, P. TADEPALLI, K. KERSTING, AND J. W. SHAVLIK (2011): “Imitation Learning in Relational Domains: A Functional-Gradient Boosting Approach,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1414–1420.
- NAU, D., M. GHALLAB, AND P. TRAVERSO (2004): *Automated Planning: Theory & Practice*, Morgan Kaufmann Publishers Inc.
- NAVARRO, S. E., N. GORGES, H. WÖRN, J. SCHILL, T. ASFOUR, AND R. DILLMANN (2012): “Haptic object recognition for multi-fingered robot hands,” in *IEEE Haptics Symposium*, IEEE, 497–502.
- NEMEC, B., F. J. ABU-DAKKA, B. RIDGE, A. UDE, J. A. JØRGENSEN, T. R. SAVARIMUTHU, J. JOUFFROY, H. G. PETERSEN, AND N. KRÜGER (2013): “Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile,” in *IEEE International Conference onn Advanced Robotics (ICAR)*, IEEE, 1–7.
- NG, A. Y., A. COATES, M. DIEL, ET AL. (2004): “Autonomous Inverted Helicopter Flight via Reinforcement Learning,” in *International Symposium on Experimental Robotics (ISER)*, 363–372.

- NGUYEN, T. T., N. D. NGUYEN, AND S. NAHAVANDI (2020): “Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications,” *IEEE Transactions on Cybernetics*, 50, 3826–3839.
- NIKOLAIDIS, S., D. HSU, AND S. S. SRINIVASA (2017a): “Human-robot mutual adaptation in collaborative tasks: Models and experiments,” *International Journal of Robotics Research*, 36, 618–634.
- NIKOLAIDIS, S., P. A. LASOTA, R. RAMAKRISHNAN, AND J. A. SHAH (2015a): “Improved human-robot team performance through cross-training, an approach inspired by human team training practices,” *International Journal of Robotics Research*, 34, 1711–1730.
- NIKOLAIDIS, S., S. NATH, A. D. PROCACCIA, AND S. SRINIVASA (2017b): “Game-Theoretic Modeling of Human Adaptation in Human-Robot Collaboration,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, New York, NY, USA: ACM, HRI ’17, 323–331.
- NIKOLAIDIS, S., R. RAMAKRISHNAN, K. GU, AND J. A. SHAH (2015b): “Efficient Model Learning from Joint-Action Demonstrations for Human-Robot Collaborative Tasks,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, ed. by J. A. Adams, W. D. Smart, B. Mutlu, and L. Takayama, ACM, 189–196.
- OGENYI, U. E., J. LIU, C. YANG, Z. JU, AND H. LIU (2021): “Physical Human-Robot Collaboration: Robotic Systems, Learning Methods, Collaborative Strategies, Sensors, and Actuators,” *IEEE Transactions on Cybernetics*, 51, 1888–1901.
- OGUZ, O. S., B. M. PFIRRMANN, M. GUO, AND D. WOLLHERR (2018a): “Learning Hand Movement Interaction Control Using RNNs: From HHI to HRI,” *IEEE Robotics and Automation Letters*, 3, 4100–4107.
- OGUZ, O. S., O. C. SARI, K. H. DINH, AND D. WOLLHERR (2017): “Progressive stochastic motion planning for human-robot interaction,” in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, IEEE, 1194–1201.
- OGUZ, O. S., Z. ZHOU, AND D. WOLLHERR (2018b): “A Hybrid Framework for Understanding and Predicting Human Reaching Motions,” *Frontiers in Robotics and AI*, 5, 27.
- OMIDSHAFIEL, S., A. AGHA-MOHAMMADI, C. AMATO, S. LIU, J. P. HOW, AND J. VIAN (2017): “Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions,” *International Journal of Robotics Research*, 36, 231–258.
- ONG, S. C. W., S. W. PNG, D. HSU, AND W. S. LEE (2010): “Planning under Uncertainty for Robotic Tasks with Mixed Observability,” *Journal of Artificial Intelligence Research*, 29, 1053–1068.
- OSA, T. (2020): “Multimodal trajectory optimization for motion planning,” *International Journal of Robotics Research*, 39.
- OZGUR, O., V. GABLER, G. HUBER, Z. ZHOU, AND D. WOLLHERR (2016): “Hybrid Human Motion Prediction for Action Selection Within Human-Robot Collaboration,” in *International Symposium on Experimental Robotics (ISER)*, Cham: Springer International Publishing, 289–298.

- PANDEY, A. K. (2012): “Towards Socially Intelligent Robots in Human Centered Environment,” Ph.D. thesis, University of Toulouse.
- PANERATI, J., H. ZHENG, S. ZHOU, J. XU, A. PROROK, AND A. P. SCHOELLIG (2021): “Learning to Fly—a Gym Environment with PyBullet Physics for Reinforcement Learning of Multi-agent Quadcopter Control,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- PARASCHOS, A., E. RUECKERT, J. PETERS, AND G. NEUMANN (2018): “Probabilistic movement primitives under unknown system dynamics,” *Advanced Robotics*, 32, 297–310.
- PASZKE, A., S. GROSS, F. MASSA, A. LERER, J. BRADBURY, G. CHANAN, T. KILLEEN, Z. LIN, N. GIMELSHEIN, L. ANTIGA, A. DESMAISON, A. KOPF, E. YANG, Z. DEVITO, M. RAISON, A. TEJANI, S. CHILAMKURTHY, B. STEINER, L. FANG, J. BAI, AND S. CHINTALA (2019): “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 8024–8035.
- PEDERSEN, O., E. MISIMI, AND F. CHAUMETTE (2020): “Grasping Unknown Objects by Coupling Deep Reinforcement Learning, Generative Adversarial Networks, and Visual Servoing,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5655–5662.
- PELLEGRINELLI, S., H. ADMONI, S. JAVDANI, AND S. S. SRINIVASA (2016): “Human-robot shared workspace collaboration via hindsight optimization,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 831–838.
- PEREIRA, A. AND M. ALTHOFF (2018): “Overapproximative Human Arm Occupancy Prediction for Collision Avoidance,” *IEEE Transactions on Automation Science and Engineering*, 15, 818–831.
- PEREPELITSA, R. (2019): “Powerlevel10k,” [Online; accessed 01-May-2022].
- PESHKIN, M. A., J. E. COLGATE, W. WANNASUPHOPRASIT, C. A. MOORE, R. B. GILLESPIE, AND P. AKELLA (2001): “Cobot architecture,” *IEEE Transactions on Robotics and Automation*, 17, 377–390.
- PETRIC, T., A. GAMS, L. COLASANTO, A. J. IJSPEERT, AND A. UDE (2018): “Accelerated Sensorimotor Learning of Compliant Movement Primitives,” *IEEE Transactions on Robotics*, 34, 1636–1642.
- PEZZEMENTI, Z. A., E. PLAKU, C. REYDA, AND G. D. HAGER (2011): “Tactile-Object Recognition From Appearance Information,” *IEEE Transactions on Robotics*, 27, 473–487.
- PINTO, L. AND A. GUPTA (2016): “Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours,” in *IEEE International Conference on Robotics and Automation (ICRA)*, ed. by D. Kragic, A. Bicchi, and A. D. Luca, IEEE, 3406–3413.
- QIAN, K., X. JING, Y. DUAN, B. ZHOU, F. FANG, J. XIA, AND X. MA (2020): “Grasp Pose Detection with Affordance-based Task Constraint Learning in Single-view Point Clouds,” *J. Intell. Robotic Syst.*, 100, 145–163.

- QUIGLEY, M., K. CONLEY, B. P. GERKEY, J. FAUST, T. FOOTE, J. LEIBS, R. WHEELER, AND A. Y. NG (2009): “ROS: an open-source Robot Operating System,” in *IEEE ICRA - Workshop Open Source Softw.*
- RAESSA, M., J. C. Y. CHEN, W. WAN, AND K. HARADA (2020): “Human-in-the-Loop Robotic Manipulation Planning for Collaborative Assembly,” *IEEE Transactions on Automation Science and Engineering*, 17, 1800–1813.
- RAKICEVIC, N. AND P. KORMUSHEV (2019): “Active learning via informed search in movement parameter space for efficient robot task learning and transfer,” *Autonomous Robots*, 43, 1917–1935.
- RANA, K., V. DASAGI, J. HAVILAND, B. TALBOT, M. MILFORD, AND N. SÜNDERHAUF (2021): “Bayesian Controller Fusion: Leveraging Control Priors in Deep Reinforcement Learning for Robotics,” *arXiv preprint arXiv:2107.09822*.
- RASHID, T., M. SAMVELYAN, C. S. DE WITT, G. FARQUHAR, J. N. FOERSTER, AND S. WHITESON (2018): “QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning,” in *International Conference on Machine Learning (ICML)*, ed. by J. G. Dy and A. Krause, PMLR, vol. 80 of *Proceedings of Machine Learning Research*, 4292–4301.
- RASMUSSEN, C. E. AND C. K. I. WILLIAMS (2006): *Gaussian processes for machine learning*, Adaptive computation and machine learning, MIT Press.
- RAVICHANDAR, H., A. S. POLYDOROS, S. CHERNOVA, AND A. BILLARD (2020): “Recent advances in robot learning from demonstration,” *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 297–330.
- REIST, P., P. PREISWERK, AND R. TEDRAKE (2016): “Feedback-motion-planning with simulation-based LQR-trees,” *International Journal of Robotics Research*, 35, 1393–1416.
- ROHRMÜLLER, F. (2011): “Action Selection in Cooperative Multi-Robot Systems,” Dissertation, Technische Universität München, München.
- ROMANO, J. M., K. HSIAO, G. NIEMEYER, S. CHITTA, AND K. J. KUCHENBECKER (2011): “Human-Inspired Robotic Grasp Control With Tactile Sensing,” *IEEE Transactions on Robotics*, 27, 1067–1079.
- RUDENKO, A., L. PALMIERI, M. HERMAN, K. M. KITANI, D. M. GAVRILA, AND K. O. ARRAS (2020): “Human motion trajectory prediction: a survey,” *Journal of Artificial Intelligence Research*, 39.
- RUSSELL, R. (2009): “Oh My Zsh,” [Online; accessed 01-May-2022].
- RUSU, R. B. AND S. COUSINS (2011): “3D is here: Point Cloud Library (PCL),” in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.
- RUSU, R. B., A. HOLZBACH, R. DIANKOV, G. R. BRADSKI, AND M. BEETZ (2009): “Perception for mobile manipulation and grasping using active stereo,” in *IEEE-RAS International Workshop on Humanoid Robots (Humanoids)*, IEEE, 632–638.
- RYU, H., H. SHIN, AND J. PARK (2020): “Multi-Agent Actor-Critic with Hierarchical Graph Attention Network,” in *AAAI Conference on Artificial Intelligence*, AAAI Press, 7236–7243.

- SADIGH, D., S. SASTRY, S. A. SESHIA, AND A. D. DRAGAN (2016a): “Planning for Autonomous Cars that Leverage Effects on Human Actions,” in *Robotics: Science and Systems (RSS)*.
- SADIGH, D., S. S. SASTRY, S. A. SESHIA, AND A. D. DRAGAN (2016b): “Information gathering actions over human internal state,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 66–73.
- SAVITZKY, A. AND M. J. GOLAY (1964): “Smoothing and differentiation of data by simplified least squares procedures.” *Analytical chemistry*, 36, 1627–1639.
- SAXENA, A., J. DRIEMEYER, AND A. Y. NG (2008): “Robotic Grasping of Novel Objects using Vision,” *International Journal of Robotics Research*, 27, 157–173.
- SAXENA, S., A. LAGRASSA, AND O. KROEMER (2021): “Learning Reactive and Predictive Differentiable Controllers for Switching Linear Dynamical Models,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 7563–7569.
- SCHAEFFER, M. A. AND A. M. OKAMURA (2003): “Methods for intelligent localization and mapping during haptic exploration,” in *IEEE International Conference on Systems, Man & Cybernetics (SMC)*, IEEE, 3438–3445.
- SCHAUL, T., D. HORGAN, K. GREGOR, AND D. SILVER (2015): “Universal Value Function Approximators,” in *International Conference on Machine Learning (ICML)*, 1312–1320.
- SCHERZINGER, S., A. ROENNAU, AND R. DILLMANN (2019a): “Contact Skill Imitation Learning for Robot-Independent Assembly Programming,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 4309–4316.
- (2019b): “Inverse Kinematics with Forward Dynamics Solvers for Sampled Motion Tracking,” in *International Conference on Advanced Robotics (ICAR)*, IEEE, 681–687.
- SCHERZINGER, S., A. RÖNNAU, AND R. DILLMANN (2017): “Forward Dynamics Compliance Control (FDCC): A new approach to cartesian compliance for robotic manipulators,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 4568–4575.
- SCHREMPF, O. C., U. D. HANEBECK, A. J. SCHMID, AND H. WORN (2005): “A Novel Approach to Proactive Human-Robot Cooperation,” in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, IEEE, 555–560.
- SCHULMAN, J., Y. DUAN, J. HO, A. X. LEE, I. AWWAL, H. BRADLOW, J. PAN, S. PATIL, K. GOLDBERG, AND P. ABBEEL (2014): “Motion planning with sequential convex optimization and convex collision checking,” *International Journal of Robotics Research*, 33, 1251–1270.
- SCHULMAN, J., S. LEVINE, P. ABBEEL, M. I. JORDAN, AND P. MORITZ (2015a): “Trust Region Policy Optimization,” in *International Conference on Machine Learning (ICML)*, ed. by F. R. Bach and D. M. Blei, JMLR.org, vol. 37 of *JMLR Workshop and Conference Proceedings*, 1889–1897.
- SCHULMAN, J., P. MORITZ, S. LEVINE, M. I. JORDAN, AND P. ABBEEL (2015b): “High-Dimensional Continuous Control Using Generalized Advantage Estimation,” *CoRR*, abs/1506.02438.

- SCHULMAN, J., F. WOLSKI, P. DHARIWAL, A. RADFORD, AND O. KLIMOV (2017): “Proximal Policy Optimization Algorithms,” *CoRR*, abs/1707.06347.
- SCHULZ, R., P. KRATZER, AND M. TOUSSAINT (2018): “Preferred Interaction Styles for Human-Robot Collaboration Vary Over Tasks With Different Action Types,” *Frontiers in Neurorobotics*, 12, 36.
- SEBANZ, N., H. BEKKERING, AND G. KNOBLICH (2006): “Joint Action: Bodies and Minds Moving Together,” *Trends in Cognitive Sciences*, 10, 70–76.
- SEBANZ, N. AND C. FRITH (2004): “Beyond Simulation? Neural Mechanisms for Predicting the Actions of Others,” *Nature Neuroscience*, 7, 5–6.
- SEKIGUCHI, S. (2018-2022): “PythonLinearNonlinearControl,” <https://github.com/Shunichi09/PythonLinearNonlinearControl>, [Online; accessed 01-June-2022].
- SERAJI, H. (1994): “Adaptive Admittance Control: An Approach to Explicit Force Control in Compliant Motion,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE Computer Society, 2705–2712.
- SHAPLEY, L. S. (1952): *A Value for N-Person Games*, Santa Monica, CA: RAND Corporation.
- (1953): “Stochastic games,” *Proceedings of the National Academy of Sciences*, 39, 1095–1100.
- SHARMA, M., J. LIANG, J. ZHAO, A. LAGRASSA, AND O. KROEMER (2020): “Learning to Compose Hierarchical Object-Centric Controllers for Robotic Manipulation,” in *Conference on Robot Learning (CoRL)*, ed. by J. Kober, F. Ramos, and C. J. Tomlin, PMLR, vol. 155 of *Proceedings of Machine Learning Research*, 822–844.
- SHEIKH, H. U. AND L. BÖLÖNI (2020): “Multi-Agent Reinforcement Learning for Problems with Combined Individual and Team Reward,” in *IEEE International Joint Conference on Neural Networks (IJCNN)*, IEEE, 1–8.
- SHIMOGA, K. B. (1996): “Robot Grasp Synthesis Algorithms: A Survey,” *International Journal of Robotics Research*, 15, 230–266.
- SHOHAM, Y. AND K. LEYTON-BROWN (2008): *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, New York, NY, USA: Cambridge University Press.
- SILVER, D., A. HUANG, C. J. MADDISON, A. GUEZ, L. SIFRE, G. VAN DEN DRIESSCHE, J. SCHRITTWIESER, I. ANTONOGLOU, V. PANNEERSHELVAM, M. LANCTOT, S. DIELEMAN, D. GREWE, J. NHAM, N. KALCHBRENNER, I. SUTSKEVER, T. LILLICRAP, M. LEACH, K. KAVUKCUOGLU, T. GRAEPEL, AND D. HASSABIS (2016): “Mastering the game of Go with deep neural networks and tree search,” *Nature*, 529, 484–489.
- SILVER, D., G. LEVER, N. HEES, ET AL. (2014): “Deterministic Policy Gradient Algorithms,” in *International Conference on Machine Learning (ICML)*, 387–395.
- SILVER, D., H. VAN HASSELT, M. HESSEL, T. SCHAUL, A. GUEZ, T. HARLEY, G. DULAC-ARNOLD, D. P. REICHERT, N. C. RABINOWITZ, A. BARRETO, AND T. DEGRIS (2017): “The Predictron: End-To-End Learning and Planning,” in *International Conference on Machine Learning (ICML)*, ed. by D. Precup and Y. W. Teh, PMLR, vol. 70 of *Proceedings of Machine Learning Research*, 3191–3199.

- SISBOT, E. A., L. F. MARIN-URIAS, R. ALAMI, AND T. SIMÉON (2007): “A Human Aware Mobile Robot Motion Planner,” *IEEE Transactions on Robotics*, 23, 874–883.
- SOBOL’, I. M. (1967): “On the distribution of points in a cube and the approximate evaluation of integrals,” *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki*, 7, 784–802.
- SON, K., D. KIM, W. J. KANG, D. HOSTALLERO, AND Y. YI (2019): “QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning,” in *International Conference on Machine Learning (ICML)*, ed. by K. Chaudhuri and R. Salakhutdinov, PMLR, vol. 97 of *Proceedings of Machine Learning Research*, 5887–5896.
- SONG, H. F., A. ABDOLMALEKI, J. T. SPRINGENBERG, A. CLARK, H. SOYER, J. W. RAE, S. NOURY, A. AHUJA, S. LIU, D. TIRUMALA, N. HEES, D. BELOV, M. A. RIEDMILLER, AND M. M. BOTVINICK (2020): “V-MPO: On-Policy Maximum a Posteriori Policy Optimization for Discrete and Continuous Control,” in *International Conference on Learning Representations (ICLR)*, OpenReview.net.
- STAHL, T. (2016): “Development of a Human-Robot Collaboration Algorithm for Industrial Assamby Based on Game Theory,” Master thesis, Technical University of Munich.
- STEFFEN, J., R. HASCHKE, AND H. J. RITTER (2007): “Experience-based and tactile-driven dynamic grasp control,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 2938–2943.
- STENGER, D., M. NITSCH, AND D. ABEL (2022): “Joint Constrained Bayesian Optimization of Planning, Guidance, Control, and State Estimation of an Autonomous Underwater Vehicle,” *CoRR*, abs/2205.14669.
- STOLT, A., F. B. CARLSON, M. M. G. ARDAKANI, I. LUNDBERG, A. ROBERTSSON, AND R. JOHANSSON (2015): “Sensorless friction-compensated passive lead-through programming for industrial robots,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 3530–3537.
- STOLT, A., M. LINDEROTH, A. ROBERTSSON, AND R. JOHANSSON (2012): “Force controlled robotic assembly without a force sensor,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 1538–1543.
- STRESE, M. (2021): “Haptic Material Acquisition, Modeling, and Display,” Ph.D. thesis, Technical University of Munich, Germany.
- STROUSTRUP, B. (2000): *The C++ programming language*, Pearson Education India.
- STULP, F., J. GRIZOU, B. BUSCH, AND M. LOPES (2015): “Facilitating intention prediction for humans by optimizing robot motions,” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 1249–1255.
- STULP, F. AND O. SIGAUD (2012): “Policy Improvement Methods: Between Black-Box Optimization and Episodic Reinforcement Learning,” 34 pages.
- STULP, F., E. A. THEODOROU, J. BUCHLI, AND S. SCHAAL (2011): “Learning to grasp under uncertainty,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5703–5708.



- SU, Z., K. HAUSMAN, Y. CHEBOTAR, A. MOLCHANOV, G. E. LOEB, G. S. SUKHATME, AND S. SCHAAL (2015): “Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor,” in *IEEE-RAS International Workshop on Humanoid Robots (Humanoids)*, IEEE, 297–303.
- SUI, Y., A. GOTOVOS, J. W. BURDICK, AND A. KRAUSE (2015): “Safe Exploration for Optimization with Gaussian Processes,” in *International Conference on Machine Learning (ICML)*, ed. by F. R. Bach and D. M. Blei, JMLR.org, vol. 37 of *JMLR Workshop and Conference Proceedings*, 997–1005.
- SUN, Y., J. LAI, L. CAO, X. CHEN, Z. XU, AND Y. XU (2020): “A Novel Multi-Agent Parallel-Critic Network Architecture for Cooperative-Competitive Reinforcement Learning,” *IEEE Access*, 8, 135605–135616.
- SUTTON, R. S., D. A. MCALLESTER, S. SINGH, AND Y. MANSOUR (1999a): “Policy Gradient Methods for Reinforcement Learning with Function Approximation,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by S. A. Solla, T. K. Leen, and K. Müller, The MIT Press, 1057–1063.
- SUTTON, R. S., D. PRECUP, AND S. SINGH (1999b): “Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning,” *Artif. Intell.*, 112, 181–211.
- TAN, M. (1993): “Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents,” in *International Conference on Machine Learning (ICML)*, ed. by P. E. Utgoff, Morgan Kaufmann, 330–337.
- TANG, H., J. HAO, T. LV, ET AL. (2018): “Hierarchical Deep Multiagent Reinforcement Learning,” *CoRR*, abs/1809.09332.
- TANG, H., A. WANG, F. XUE, J. YANG, AND Y. CAO (2021): “A Novel Hierarchical Soft Actor-Critic Algorithm for Multi-Logistics Robots Task Allocation,” *IEEE Access*, 9, 42568–42582.
- TEDRAKE, R., I. R. MANCHESTER, M. M. TOBENKIN, AND J. W. ROBERTS (2010): “LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification,” *International Journal of Robotics Research*, 29, 1038–1052.
- THRUN, S., W. BURGARD, AND D. FOX (2005): *Probabilistic robotics*, Intelligent robotics and autonomous agents, MIT Press.
- TIAN, Z., Y. WEN, Z. GONG, F. PUNAKKATH, S. ZOU, AND J. WANG (2019): “A Regularized Opponent Model with Maximum Entropy Objective,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, ed. by S. Kraus, ijcai.org, 602–608.
- TIZIANI, L. O., A. M. HART, T. W. CAHOON, F. WU, H. H. ASADA, AND F. L. HAMMOND (2017): “Empirical characterization of modular variable stiffness inflatable structures for supernumerary grasp-assist devices,” *International Journal of Robotics Research*, 36, 1391–1413.
- TODOROV, E., T. EREZ, AND Y. TASSA (2012): “MuJoCo: A physics engine for model-based control.” in *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, IEEE, 5026–5033.

- TORVALD, L., J. HAMANO, ET AL. (2005): ““Initial revision of ”git”, the information manager from hell”,” .
- TORVALDS, L. ET AL. (1991): “Linux,” [Online (Archived); accessed 01-May-2022].
- TOUSSAINT, M. (2015): “Logic-Geometric Programming: An Optimization-Based Approach to Combined Task and Motion Planning,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, ed. by Q. Yang and M. J. Wooldridge, AAAI Press, 1930–1936.
- TOUSSAINT, M., T. MUNZER, Y. MOLLARD, L. Y. WU, N. A. VIEN, AND M. LOPES (2016): “Relational activity processes for modeling concurrent cooperation,” in *IEEE International Conference on Robotics and Automation (ICRA)*, ed. by D. Kragic, A. Bicchi, and A. D. Luca, IEEE, 5505–5511.
- TURNWALD, A., D. ALTHOFF, D. WOLLHERR, AND M. BUSS (2016): “Understanding Human Avoidance Behavior: Interaction-Aware Decision Making Based on Game Theory,” *International Journal of Social Robotics*, 8, 331–351.
- TURNWALD, A. AND D. WOLLHERR (2019): “Human-Like Motion Planning Based on Game Theoretic Decision Making,” *International Journal of Social Robotics*, 11, 151–170.
- UMLAUFT, J., T. BECKERS, A. CAPONE, A. LEDERER, AND S. HIRCHE (2020): “Smart Forgetting for Safe Online Learning with Gaussian Processes,” in *Conference on Learning for Dynamics and Control (L4DC)*, ed. by A. M. Bayen, A. Jadbabaie, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, and M. N. Zeilinger, PMLR, vol. 120 of *Proceedings of Machine Learning Research*, 160–169.
- VAMVOUDAKIS, K. G. AND F. L. LEWIS (2011): “Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton-Jacobi equations,” *Automatica*, 47, 1556–1569.
- VAMVOUDAKIS, K. G., F. L. LEWIS, AND G. R. HUDAS (2012): “Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality,” *Automatica*, 48, 1598–1611.
- VAN DER WAL, J. (1980): “Stochastic dynamic programming,” Ph.D. thesis, Mathematisch Centrum.
- VAN HASSELT, H. (2010): “Double Q-learning,” in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, ed. by J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Curran Associates, Inc., 2613–2621.
- VAN ROSSUM, G. AND F. L. DRAKE (2009): *Python 3 Reference Manual*, Scotts Valley, CA: CreateSpace.
- VAN ROSSUM, G. AND F. L. DRAKE JR (1995): *Python reference manual*, Centrum voor Wiskunde en Informatica Amsterdam.
- VANDERBORGH, B., A. ALBU-SCHÄFFER, A. BICCHI, E. BURDET, D. G. CALDWELL, R. CARLONI, M. G. CATALANO, O. EIBERGER, W. FRIEDL, G. GANESH, M. GARABINI, M. GREBENSTEIN, G. GRIOLI, S. HADDADIN, H. HÖPPNER, A. JAFARI, M. LAFFRANCHI, D. LEFEBER, F. PETIT, S. STRAMIGIOLI, N. G. TSAGARAKIS, M. V. DAMME, R. V. HAM, L. C. VISSER, AND S. WOLF (2013): “Variable impedance actuators: A review,” *Robotics and Autonomous Systems*, 61, 1601–1614.

- VINYALS, O., I. BABUSCHKIN, W. M. CZARNECKI, M. MATHIEU, A. DUDZIK, J. CHUNG, D. H. CHOI, R. POWELL, T. EWALDS, P. GEORGIEV, J. OH, D. HORGAN, M. KROISS, I. DANIHELKA, A. HUANG, L. SIFRE, T. CAI, J. P. AGAPIOU, M. JADERBERG, A. S. VEZHNEVETS, R. LEBLOND, T. POHLEN, V. DALIBARD, D. BUDDEN, Y. SULSKY, J. MOLLOY, T. L. PAINE, Ç. GÜLÇEHRE, Z. WANG, T. PFAFF, Y. WU, R. RING, D. YOGATAMA, D. WÜNSCH, K. MCKINNEY, O. SMITH, T. SCHAUL, T. P. LILLICRAP, K. KAVUKCUOGLU, D. HASSABIS, C. APPS, AND D. SILVER (2019): “Grandmaster level in StarCraft II using multi-agent reinforcement learning,” *Nature*, 575, 350–354.
- VIRTANEN, P., R. GOMMERS, T. E. OLIPHANT, M. HABERLAND, T. REDDY, D. COURNAPEAU, E. BUROVSKI, P. PETERSON, W. WECKESSER, J. BRIGHT, S. J. VAN DER WALT, M. BRETT, J. WILSON, K. J. MILLMAN, N. MAYOROV, A. R. J. NELSON, E. JONES, R. KERN, E. LARSON, C. J. CAREY, İ. POLAT, Y. FENG, E. W. MOORE, J. VANDERPLAS, D. LAXALDE, J. PERKTOLD, R. CIMRMAN, I. HENRIKSEN, E. A. QUINTERO, C. R. HARRIS, A. M. ARCHIBALD, A. H. RIBEIRO, F. PEDREGOSA, P. VAN MULBREGT, AND SCI-PY 1.0 CONTRIBUTORS (2020): “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, 17, 261–272.
- WÄCHTER, A. AND L. BIEGLER (2006): “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Math. Program.*, 106, 25–57.
- WANG, Y., Y. SHENG, J. WANG, AND W. ZHANG (2018): “Optimal Collision-Free Robot Trajectory Generation Based on Time Series Prediction of Human Motion,” *IEEE Robotics and Automation Letters*, 3, 226–233.
- WANG, Z., C. R. GARRETT, L. P. KAEHLING, AND T. LOZANO-PÉREZ (2021): “Learning compositional models of robot skills for task and motion planning,” *International Journal of Robotics Research*, 40.
- WARD-CHERRIER, B., N. PESTELL, L. CRAMPORN, B. WINSTONE, M. E. GIANNACCINI, J. ROSSITER, AND N. F. LEPORA (2018): “The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies,” *Soft Robotics*, 5, 216–227.
- WASKOM, M., O. BOTVINNIK, D. O’KANE, P. HOBSON, S. LUKAUSKAS, D. C. GEMPERLINE, T. AUGSPURGER, Y. HALCHENKO, J. B. COLE, J. WARMENHOVEN, J. DE RUITER, C. PYE, S. HOYER, J. VANDERPLAS, S. VILLALBA, G. KUNTER, E. QUINTERO, P. BACHANT, M. MARTIN, K. MEYER, A. MILES, Y. RAM, T. YARKONI, M. L. WILLIAMS, C. EVANS, C. FITZGERALD, BRIAN, C. FONNESBECK, A. LEE, AND A. QALIEH (2017): “mwaskom/seaborn: v0.8.1 (September 2017),” .
- WATKINS, C. J. C. H. AND P. DAYAN (1992): “Technical Note Q-Learning,” *Mach. Learn.*, 8, 279–292.
- WAUGH, K., B. D. ZIEBART, AND D. BAGNELL (2011): “Computational Rationalization: The Inverse Equilibrium Problem,” in *International Conference on Machine Learning (ICML)*, ed. by L. Getoor and T. Scheffer, Omnipress, 1169–1176.
- WEI, E., D. WICKE, D. FREELAN, AND S. LUKE (2018): “Multiagent Soft Q-Learning,” in *AAAI Spring Symposia*, AAAI Press.
- WETTELS, N., V. J. SANTOS, R. S. JOHANSSON, AND G. E. LOEB (2008): “Biomimetic Tactile Sensor Array,” *Advanced Robotics*, 22, 829–849.

- WU, B., B. HU, AND H. LIN (2017): “A Learning Based Optimal Human Robot Collaboration with Linear Temporal Logic Constraints,” *CoRR*, abs/1706.00007.
- WU, X., X. LI, J. LI, P. C. CHING, V. C. M. LEUNG, AND H. V. POOR (2021): “Caching Transient Content for IoT Sensing: Multi-Agent Soft Actor-Critic,” *IEEE Transactions on Communications*, 69, 5886–5901.
- XU, D., G. E. LOEB, AND J. A. FISHEL (2013): “Tactile identification of objects using Bayesian exploration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 3056–3061.
- YADAN, O. (2019): “Hydra - A framework for elegantly configuring complex applications,” Github, [Online; accessed 06-August-2022].
- YANG, L., Z. LI, J. ZENG, AND K. SREENATH (2022): “Bayesian Optimization Meets Hybrid Zero Dynamics: Safe Parameter Learning for Bipedal Locomotion Control,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 10456–10462.
- YANG, Y. AND J. WANG (2020): “An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective,” *CoRR*, abs/2011.00583.
- YOSHIDA, K. AND B. WILCOX (2008): “Space Robots and Systems,” in *Springer Handbook of Robotics*, ed. by B. Siciliano and O. Khatib, Springer, 1031–1063.
- ZENG, G. AND A. HEMAMI (1997): “An overview of robot force control,” *Robotica*, 15, 473–482.
- ZHANG, H., H. JIANG, Y. LUO, AND G. XIAO (2017): “Data-Driven Optimal Consensus Control for Discrete-Time Multi-Agent Systems With Unknown Dynamics Using Reinforcement Learning Method,” *IEEE Transactions on Industrial Electronics*, 64, 4091–4100.
- ZHANG, K., Z. YANG, AND T. BAŞAR (2021a): “Multi-agent reinforcement learning: A selective overview of theories and algorithms,” *Handbook of Reinforcement Learning and Control*, 321–384.
- ZHANG, Q., H. DONG, AND W. PAN (2020a): “Lyapunov-Based Reinforcement Learning for Decentralized Multi-agent Control,” in *International Conference on Distributed Artificial Intelligence (DAI)*, ed. by M. E. Taylor, Y. Yu, E. Elkind, and Y. Gao, Springer, vol. 12547 of *Lecture Notes in Computer Science*, 55–68.
- ZHANG, X., L. SUN, Z. KUANG, AND M. TOMIZUKA (2021b): “Learning Variable Impedance Control via Inverse Reinforcement Learning for Force-Related Tasks,” *IEEE Robotics and Automation Letters*, 6, 2225–2232.
- ZHANG, Z., K. QIAN, B. W. SCHULLER, AND D. WOLLHERR (2020b): “An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory,” *IEEE Transactions on Automation Science and Engineering*, 1–13.
- ZHOU, A., D. HADFIELD-MENELL, A. NAGABANDI, AND A. D. DRAGAN (2017): “Expressive Robot Motion Timing,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, ed. by B. Mutlu, M. Tscheligi, A. Weiss, and J. E. Young, ACM, 22–31.
- ZHOU, Z., O. S. OGUZ, Y. REN, M. LEIBOLD, AND M. BUSS (2021): “Data Generation Method for Learning a Low-dimensional Safe Region in Safe Reinforcement Learning,” *CoRR*, abs/2109.05077.

- ZHU, H., V. GABLER, AND D. WOLLHERR (2017): “Legible Action Selection in Human-Robot Collaboration,” in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Lisbon: IEEE.
- ZIEBART, B. D., A. L. MAAS, J. A. BAGNELL, AND A. K. DEY (2008): “Maximum Entropy Inverse Reinforcement Learning,” in *AAAI Conference on Artificial Intelligence*, 1433–1438.
- ZUCKER, M., N. D. RATLIFF, A. D. DRAGAN, M. PIVTORAIKO, M. KLINGENSMITH, C. M. DELLIN, J. A. BAGNELL, AND S. S. SRINIVASA (2013): “CHOMP: Covariant Hamiltonian optimization for motion planning,” *International Journal of Robotics Research*, 32, 1164–1193.

*This bibliography contains 425 references.*

## Own Thesis-Related Publications

- ACKERMANN, J., V. GABLER, T. OSA, AND M. SUGIYAMA (2019): “Reducing Overestimation Bias in Multi-Agent Domains Using Double Centralized Critics,” *CoRR*, abs/1910.01465.
- BARI, S., V. GABLER, AND D. WOLLHERR (2021): “MS2MP: A Min-Sum Message Passing Algorithm for Motion Planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Xi’an, China: IEEE, 7887–7893.
- (2023): “Probabilistic Inference-based Robot Motion Planning via Gaussian Belief Propagation,” *IEEE Robotics and Automation Letters*, 8, 5156–5163.
- DANIELS, A., S. KERZ, S. BARI, V. GABLER, AND D. WOLLHERR (2023): “Grasping in Uncertain Environments: A Case Study For Industrial Robotic Recycling,” in *IEEE International Conference on Systems, Man & Cybernetics (SMC)*, accepted: IEEE.
- DINH, K. H., O. OGUZ, G. HUBER, V. GABLER, AND D. WOLLHERR (2015): “An approach to integrate human motion prediction into local obstacle avoidance in close human-robot collaboration,” in *IEEE Workshop on Advanced Robotics and its Social Impact (ARSO)*, Lyon, France.
- GABLER, V., G. HUBER, M. BOSCH, AND D. GIAKOUMIS (2020a): “D6.2 - Haptic Regression Report,” Tech. rep., HR-Recycler - Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment.
- GABLER, V., G. HUBER, S. ENDO, D. WOLLHERR, A. TISSOT, AND I. FREIRE GONZALEZ (2022a): “D6.1 - Force Guided Manipulation Evaluation,” Tech. rep., HR-Recycler - Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment.
- GABLER, V., G. HUBER, AND D. WOLLHERR (2022b): “A Force-Sensitive Grasping Controller Using Tactile Gripper Fingers and an Industrial Position-Controlled Robot,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia: IEEE, 770–776.
- GABLER, V., K. MAIER, S. ENDO, AND D. WOLLHERR (2020b): “Haptic Object Identification for Advanced Manipulation Skills,” in *International Conference on Biomimetic and Biohybrid Systems (Living Machines)*, ed. by V. Vouloutsi, A. Mura, F. J. Esser, T. Speck, T. J. Prescott, and P. F. M. J. Verschure, Springer, vol. 12413 of *Lecture Notes in Computer Science*, 128–140.
- GABLER, V., T. STAHL, G. HUBER, O. OGUZ, AND D. WOLLHERR (2017): “A Game-Theoretic Approach for Adaptive Action Selection in Close Distance Human-Robot-Collaboration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore: IEEE, 2897–2903.
- GABLER, V. AND D. WOLLHERR (2022): “Bayesian Optimization with Unknown Constraints in Graphical Skill-Models for Compliant Manipulation Tasks Using an Industrial Robot,” *Frontiers Robotics AI*, 9.

- (2023): “Decentralized Multi-Agent Reinforcement Learning Based on Best-Response Policies,” *Frontiers Robotics AI*, submitted.
- HUBER, G., V. GABLER, AND D. WOLLHERR (2017): “An online trajectory generator on SE(3) with magnitude constraints,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, IEEE, 6171–6177.
- KOBAYASHI, Y., T. MATSUMOTO, W. TAKANO, D. WOLLHERR, AND V. GABLER (2017): “Motion Recognition by Natural Language Including Success and Failure of Tasks for Co-Working Robot with Human,” in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Sheraton Arabella Park Hotel, Munich, Germany: IEEE, 10–15.
- OZGUR, O., V. GABLER, G. HUBER, Z. ZHOU, AND D. WOLLHERR (2016): “Hybrid Human Motion Prediction for Action Selection Within Human-Robot Collaboration,” in *International Symposium on Experimental Robotics (ISER)*, Cham: Springer International Publishing, 289–298.
- ZHU, H., V. GABLER, AND D. WOLLHERR (2017): “Legible Action Selection in Human-Robot Collaboration,” in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Lisbon: IEEE.

## Supervised Student Theses

- ACKERMANN, J. (2017): “Online Adaptation to Human Preferences in Relational Domains,” Research internship (Bachelor), Technical University of Munich.
- (2018): “Hierarchical Deep Reinforcement Learning for Multi-Agent Robotic Systems,” Bachelor thesis, Technical University of Munich.
- BUDDE GENANNT DOHMANN, P. (2016): “Implementation of a Trajectory Learning and Flexible Roll-out Module Based on Dynamic Movement Primitives on the KUKA LWR 4+,” Research internship (Bachelor), Technical University of Munich.
- DEURINGER, F. (2019): “Adaptive Impedance Control for Robotic Screwing-Skills given Partial Information,” Research internship (Master), Technical University of Munich.
- GASSE, S. (2016): “Learning Human-Human Collaborative Behavior in Pick and Place Tasks,” Bachelor thesis, Technical University of Munich.
- HAUBER, D. (2019): “Learning Joint Object Manipulation for Multi-Robot Systems Given Few Demonstrations,” Master thesis, Technical University of Munich.
- HOFMANN, M. (2016): “Improvements of a human robot collaboration framework,” Bachelor thesis, Technical University of Munich.
- HÖLZL, F. (2018): “Decentralized Mixed Observable Markov Decision Processes (DEC-MOMDPs) for Human-Robot Collaboration,” Bachelor thesis, Technical University of Munich.

- HONG YONG, T. (2015): “Creation of a Suction Gripper for LEGO,” Bachelor thesis, Technical University of Munich.
- KREUTMAYR, F. (2019): “Human Robot Collaboration as a Differential Game,” Master thesis, Technical University of Munich.
- KROCKENBERGER, D. (2019): “Hierarchical Deep Reinforcement Learning for Multi-Agent Robotic Systems,” Master thesis, Technical University of Munich.
- LI, J. (2016): “Force-Sensitive Manipulation with a KUKA Lightweight Robot,” Master thesis, Technical University of Munich.
- MAIER, K. (2019): “Object Identification for Advanced Manipulation Skills Using Haptic SLAM,” Master thesis, Technical University of Munich.
- MÜNZ, D. (2020): “Adaptive Impedance Control for Robotic Screwing-Skills given Partial Information,” Research internship (Bachelor), Technical University of Munich.
- NATZER, J. (2016): “Design of a Human-Robot Collaboration Decision Framework as a Dynamic Game,” Bachelor thesis, Technical University of Munich.
- RIEMANN, S. (2015): “Development of an hand tracking system in ROS with online data analysis in Matlab for Human-Robot collaboration,” Research internship (Bachelor), Technical University of Munich.
- SATIMUN, N., N. KEITH, AND O. POH SENG (2017): “Development of virtual reality pipeline for interactive human-robot assembly using ROS-control and Gazebo-simulator,” Bachelor thesis, Technical University of Munich.
- SEYLER, T. (2016): “Human Aware Hierarchical Task Planning Including Temporal Task-Constraints,” Master thesis, Technical University of Munich.
- SHUN FA, T. (2015): “Collision Detection and Classification,” Bachelor thesis, Technical University of Munich.
- SIEW, T. (2015): “Implementing a Robot-Human Handover Scenario,” Bachelor thesis, Technical University of Munich.
- SKURZYŃSKA, E. (2016): “Development of a Logical Reasoning Module to Determine Geometrical Constraints Within LEGO-Assembly,” Research internship (Master), Technical University of Munich.
- STAHL, T. (2016): “Development of a Human-Robot Collaboration Algorithm for Industrial Assambly Based on Game Theory,” Master thesis, Technical University of Munich.
- STEBBERGER, M. (2016): “Adapting to Human Preferences by Applying Inverse Reinforcement Learning,” Research internship (Master), Technical University of Munich.
- (2017): “Autonomous Decisions in Mixed Human-Robot Teams Derived from Graphical Games,” Master thesis, Technical University of Munich.
- UNVERRICHT, N. (2017): “Modeling and Analyzing Mixed Human-Robot Teams as a Bayesian Game,” Research internship (Master), Technical University of Munich.



- WÜNSCHE, S. (2018): “Optimal Task Allocation for Human-Robot Collaboration Using Stochastic Petri-Nets and Human Behavior Models,” Bachelor thesis, Technical University of Munich.
- YUAN, S. (2017): “Multi-Layered Task and Motion Planning for Human-Robot Collaboration,” Research internship (Master), Technical University of Munich.
- (2018): “Human Robot Collaboration as a Differential Game,” Master thesis, Technical University of Munich.
- ZHOU, Z. (2015): “Development of a Scene Interpretation and Hand Tracking System – Combining Kinect-Fusion and Model-Based Hand Tracking on Point Cloud Data,” Research internship (Master), Technical University of Munich.