ROBOT CONTROLLERS, GAZE BEHAVIORS AND HUMAN MOTION DATASETS FOR OBJECT HANDOVERS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Alap Kshirsagar

August 2022

[1]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Cornell University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `https://www.ieee.org/publications/rights/rights-link.html` to learn how to obtain a License from RightsLink.

ROBOT CONTROLLERS, GAZE BEHAVIORS AND HUMAN MOTION DATASETS FOR OBJECT HANDOVERS

Alap Kshirsagar, Ph.D.

Cornell University 2022

We investigate the collaborative task of object handovers between a human and a robot. Object handover is one of the most common skills required for a collaborative or an assistive robot. Tasks such as surgical assistance, housekeeping, rehabilitation assistance, and collaborative assembly require a robot to give objects to a human (robot-to-human handover) and take objects from a human (human-to-robot handover).

First, we design robot controllers for previously unexplored human-robot handover scenarios involving a) robot behavior specification by end-users, b) flexibility to change handover strategies, c) unknown robot dynamics, and d) human-like bi-manual handovers. For "a", we propose two receding horizon controllers that utilize timing parameters and provide failure feedback to the user. We find that these two controllers offer contrasting user experience and task performance, vis-à-vis a baseline controller, in an industrial human-robot collaborative task. For "b", we present a controller that uses automated synthesis from specifications in Signal Temporal Logic, and illustrate the flexibility of our approach by reproducing, in a simulation environment, existing human-robot handover strategies found in the literature. For "c", we evaluate the potential of a reinforcement learning method, Guided Policy Search (GPS), both in a simulation environment and on a physical robot arm. Our evaluations provide new insights into the capabilities and limitations of GPS. For "d", we evaluate

the potential of an imitation learning technique, Bayesian Interaction Primitives (BIPs), and discuss two adaptations of BIPs for generating humanlike bimanual reaching motions.

Second, we investigate robot gazes in human-to-robot handovers. We design human-inspired robot gaze behaviors by analyzing the receiver's gaze patterns in human-to-human handovers. We conduct four user studies with two robot platforms to evaluate human preferences for the robot receiver's gaze behavior in handovers. Our studies reveal that the most preferred robot receiver's gaze behavior in a human-to-robot handover consists of a combination of face-oriented gaze for social engagement and task-oriented gaze directed at the giver's hand.

Third, we develop two datasets of human-to-human handovers: bimanual handovers, and multiple sequential handovers in a shelving task. These datasets could help build human motion models and robot controllers for those scenarios of human-robot handovers.

BIOGRAPHICAL SKETCH

Alap Kshirsagar is a doctoral candidate in Mechanical Engineering at Cornell University, USA. He has a Master of Technology degree in Mechanical Engineering from the Indian Institute of Technology (IIT) Madras and a Bachelor of Technology degree in Mechanical Engineering with Honours and with Minor in Aerospace Engineering from IIT Bombay. He was a visiting student researcher at the University of California-Berkeley in 2016, Rheinisch-Westfälische Technische Hochschule Aachen in 2017 and Ben-Gurion University of the Negev in 2020.

This dissertation is dedicated to all human beings and artificially intelligent agents.

ACKNOWLEDGEMENTS

As I come to the end of my Ph.D. journey, I would like to tell the new Ph.D. students that this journey is full of excitement, surprises, and challenges. Thanks to my beloved and supportive network of committee members, collaborators, friends, and family, I have thoroughly enjoyed this journey.

First, I owe my deepest gratitude to my advisor Prof. Guy Hoffman. You rightly identified my strengths and weaknesses, and always found ways to help me overcome my weaknesses. Your enormous support allows me the courage to explore uncharted territories. Second, I would like to thank collaborators Prof. Hadas Kress-Gazit, Dr. Armin Biess, and Prof. Yael Edan. Your expert guidance and insightful suggestions have been an integral part of this journey. I would also like to thank Prof. Hadas Kress-Gazit and Prof. Mark Campbell for serving on my committee. This research is a truly collaborative effort with many graduate and undergraduate students from USA and Israel. I would like to acknowledge contributions from Rahul Kumar Ravi, Raphael Fortuna, Zhiming Xie, Melanie Lim, Shemar Christian, Song Ye, Julia Katz from the Cornell University, and Tair Sela from the Ben-Gurion University of the Negev. I had a wonderful time working with and learning from all of you. I would also like to thank Prof. Ori Heffetz, Guy Ishai, and Bnaya Dreyfuss from the Hebrew University of Jerusalem, who introduced me to Behavioural Economics, another world almost as fascinating as Robotics.

Finally, I would like to thank my wife Ketaki, parents, brother, and friends for the unconditional love and support they have bestowed upon me.

This research was supported in part by the Ben-Gurion University of the Negev through the Agricultural, Biological and Cognitive (ABC) Robotics Initiative, the Helmsley Charitable Trust, the Marcus Endowment Fund and the

Rabbi W. Gunther Plaut Chair in Manufacturing Engineering, the Israel Science Foundation (grant no. 1627/17), and Hahn-Robotics through their donation of the Sawyer robot.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this work, we develop robot controllers, robot gaze behaviors, and human motion datasets for the collaborative task of object handovers. This work is motivated by the increasing shift toward a human-robot joint workforce [35, 68], and the projections of more and more robots being used in the future for assistive and domestic purposes [73, 54].

Object handovers are a central aspect of human-robot collaboration in both industrial and domestic environments. Examples include a collaborative factory robot exchanging parts with a human co-worker, a surgical assistant robot transferring instruments to or from a surgeon, a warehouse robot helping a human shelve items, a domestic robot unloading a dishwasher, and a caregiver robot providing food or medicine to bedridden people. These tasks include both human-to-robot and robot-to-human handovers.

A handover typically consists of three phases: a reach phase in which both actors extend their arms towards the handover location, a transfer phase in which the object is transferred from the giver's hand to the receiver's hand, and a retreat phase in which the actors exit the interaction. These phases involve coordination in both time and space of hand movements, grip forces, body postures, and non-verbal cues like gazes.

The importance and complexity of handovers have resulted in a growing body of scientific research on this topic. However, we identify some gaps in the existing work and address these gaps in this dissertation.

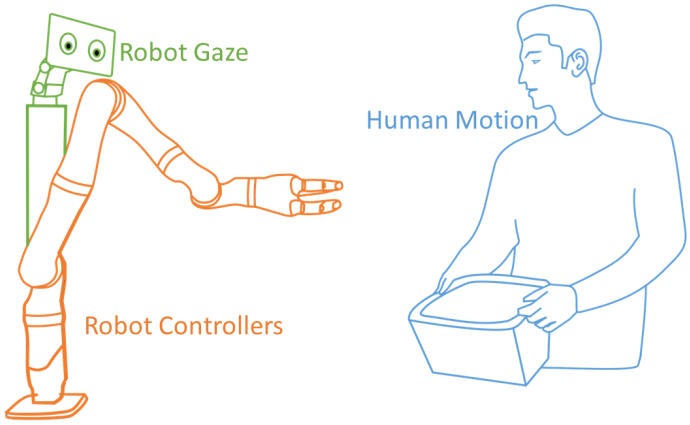


Figure 1.1: This work focuses on developing robot motion controllers and robot gaze behaviors for human-robot handovers, and investigating human motion in handovers.

1.1 Research Questions

In this work, we seek to address three research questions related to object handovers as depicted in Figure 1.1:

- How to control a robot to perform handovers with humans? We focus on four previously unexplored scenarios: controller design by end-users, unified framework for different handover strategies, unknown robot dynamics, and human-like bimanual handovers.
- Which gaze behavior should a robot utilize while receiving an object from a human?
- How do humans perform bimanual and multiple sequential handovers with each other?

Tables 1.1, 1.2, and 1.3 show comparisons of the features of some of the existing works and the research directions considered in this work. A detailed literature review is presented in Chapter 2.

Table 1.1: Comparison of features of human-robot handover controllers

	Online Target Update	User Specified Parameters	Automated Controller Synthesis	Bimanual Handovers	Humanlike Reaching Motion	Learn Dynamics
[112]	✓	-	-	-	-	-
[102]	-	-	-	-	-	-
[132]	✓	-	-	-	✓	-
[180]	✓	-	-	-	✓	-
[4]	-	-	-	-	-	-
[100]	✓	-	-	-	✓	-
[63]	-	-	-	-	-	-
[110]	✓	-	-	-	-	-
[149]	✓	-	-	✓	-	-
[169]	✓	-	-	-	✓	-
1.2.1	✓	✓	-	-	-	-
1.2.2	✓	-	✓	-	-	-
1.2.3	✓	-	-	-	-	✓
1.2.4	✓	-	-	✓	✓	-

Table 1.2: Comparison of studies of robot gaze in human-robot handovers

	Robot-to-Human Handovers	Human-to-Robot Handovers	Different Objects	Different Postures
[115]	✓	-	-	-
[188]	✓	-	-	-
[83]	✓	-	-	-
1.3	-	✓	✓	✓

Table 1.3: Comparison of datasets of human-to-human handovers

	Bimanual Handovers	Mocap Data	RGB-D Data	Multiple Viewpoints	Sequential Handovers
[25]	-	✓	-	-	-
[28]	-	✓	-	-	-
[183]	-	-	✓	✓	-
1.4	✓	✓	✓	✓	✓

1.2 Research Direction 1: Controllers for Human-Robot Handovers

The first research direction of this work focuses on the design of robot controllers for previously unexplored human-robot handover scenarios.

1.2.1 Controller Design by End-Users (Chapter 3)

Prior studies of human-robot handovers did not consider scenarios where participants need to design the robot's handover controller for a collaborative task with timing constraints. Such a flexibility is important in industrial settings with dynamically changing task requirements, and it can increase the acceptance of robots as coworkers. A majority of existing handover controllers require tuning non-intuitive controller parameters to specify the robot's behavior, for example, proportional gain [112], Dynamic Movement Primitive (DMP) parameters [132]. Some researchers have proposed handover controllers with intuitive parameters, for example delay/reach timings [125], but these controllers were not evaluated in scenarios where users can specify these parameters.

We propose two receding horizon controllers (RHC) for human-robot object handovers and evaluate them in two human-participant studies in which users design a collaborative robot's behavior for an industrial task with timing constraints, simulated in a lab setting. Our proposed controllers allow users to specify timing parameters for the reach-to-handover motion of the robot, and provide feedback if the robot cannot satisfy those constraints. These features make such a controller useful for settings where end-users need to tune the be-

havior of the robot without programming knowledge. However, our studies show that the two proposed controllers have contrasting effects on the user experience and task performance, even though they both have intuitive timing parameters and provide failure feedback. Our findings could shed light towards the design of better user-defined controllers for human-robot handovers.

1.2.2 Unified Framework for Different Handover Strategies (Chapter 4)

A multitude of handover strategies have been proposed in the literature for human-robot handovers [163, 115, 63, 189, 18, 110, 161]. But there is no unified framework to easily switch between these strategies. Also, the existing controllers are programmed manually, which is an error-prone process. To overcome these shortcomings, we develop a controller for human-robot handovers that is automatically synthesized from high-level specifications in Signal Temporal Logic (STL). We develop specification templates for human-to-robot and robot-to-human handovers, and illustrate the flexibility of our approach by reproducing, in simulation, existing human-robot handover strategies found in the literature.

1.2.3 Unknown Robot Dynamics (Chapter 5)

Recently, a reinforcement learning algorithm called "Guided Policy Search" (GPS) [92] has been successfully demonstrated for autonomous manipulation [32, 91, 96, 92], and locomotion tasks [185, 91, 93, 94] without prior knowl-

edge of the robot or the environment dynamics. However, these existing evaluations differ from human-robot handovers because handovers involve large spatial variations in reach locations, moving targets, and changes in the robot's mass induced by different objects being handed over. We evaluate GPS, both in a simulation environment and on a physical robot, for generating robot reaching motions in a handover. The key findings are that GPS has good generalizability over changes in the robot end-effector's mass but poor spatial generalizability. Our attempts to replicate these findings in the real environment, provide new insights into the challenges associated with the use of GPS on a physical collaborative robot with safety constraints.

1.2.4 Humanlike Bimanual Handovers (Chapter 7)

There are numerous applications where bimanual handovers are useful or even necessary. Bimanual handovers are necessary when handing over large rigid objects such as crates, deformable objects such as folded clothes, spherical objects such as a basketball, and delicate objects like crockery. Also in some cultures, it is a rule of etiquette to hand over objects with two hands. However, there is very little work on robot controllers for bimanual human-robot handovers [12, 159, 149, 59]. Further, these existing approaches do not generate a human-like robot motion, which is desired for humanoid robots. For single-handed or uni-manual handovers, several researchers have proposed robot controllers for generating human-like reaching motions [132, 100, 84, 180, 182]. But, to the best of our knowledge, there is no prior work on human-like robot motion in bimanual handovers. We seek to address this gap by developing robot controllers for human-like bimanual handovers. We evaluate Bayesian Interac-

tion Primitives (BIPs) [22], an imitation learning technique, for training a robot to perform bimanual reach-to-handover motions from human-human handover demonstrations. We find that the existing formulation of BIPs does not lend itself readily to bimanual reach-to-handover motions. We discuss two possible solutions to adapt BIPs for this task.

1.3 Research Direction 2: Robot Gaze Behaviors in Human-to-Robot Handovers (Chapter 6)

The second research direction of this work investigates human preferences for robot gaze behaviors when the robot is receiving an object from a human. Past research has shown that the robot's head gaze behaviors affect the subjective experience and timing of robot-to-human object handovers [115, 188, 48, 83]. However, all of these works only studied robot-to-human handovers, where the robot was the giver. Human-to-robot handovers, where the robot is the receiver, are equally important with many applications in various domains. Therefore, we investigate robot gaze behaviors in human-to-robot handovers.

To find candidates for robot gaze behaviors, we analyze gaze behaviors of human receivers in human-to-human handovers by annotating gaze locations in over 14000 frames of a public dataset of handovers [25] with the giver's and the receiver's gaze locations. We find that the most common gaze behavior employed by a receiver consists of continuously looking at the other person's hand during the reach and the transfer phases, and then looking at the other person's face during the retreat phase.

We then implement and compare four human-inspired robot head gaze behaviors during the reach phase of human-to-robot handovers. Our studies find that observers of a handover perceive a *Face-Hand transition* gaze, in which the robot initially looks at the giver's face and then at the giver's hand, as more anthropomorphic, likable, and communicative of timing compared to continuously looking at the giver's face (*Face* gaze) or hand (*Hand* gaze). Participants in a handover perceive *Face* gaze or *Face-Hand transition* gaze as more anthropomorphic and likable compared to *Hand* gaze. However, these results are limited to a specific object and the standing posture of the giver, a common limitation of other prior studies as well.

To expand and generalize our findings, we study human preferences towards robot gaze behaviors in human-to-robot handovers, instead of only the reach phase, for four different object types (large, small, fragile, non-fragile) and two human postures (standing, sitting). We find that, for both observers and participants in a handover, when the robot exhibits a *Face-Hand-Face* gaze (gazing at the giver's face and then at the giver's hand during the reach phase and back at the giver's face during the retreat phase), participants consider the handover to be more likable, anthropomorphic, and communicative of timing. We do not find evidence of any effect of the object's size or fragility or the giver's posture on the gaze preference.

1.4 Research Direction 3: Human-Human Bimanual and Sequential Handovers (Chapter 7)

The third research direction of this work focuses on investigating human-to-human bimanual and sequential handovers. Humans accomplish the challenging task of object handovers seamlessly. In the past, researchers have studied various aspects of human-to-human handovers to understand how people perform this complex maneuver. Some studies investigated different phases of handovers, such as approach [7], reach [66, 128, 107], and transfer [30, 42]. Others have built datasets of human-to-human handovers [25, 28, 183]. However, all of these works have studied handovers of objects with single-handed grasps i.e. uni-manual handovers.

To the best of our knowledge, there is no public dataset of human-to-human handovers of objects requiring bimanual grasps, even though such handovers are equally important and even more challenging. Further, several collaborative tasks require multiple sequential handovers. For example, shelving in a warehouse, automotive assembly, and unloading dishes or groceries. In these tasks some objects need to be handed over with two hands, owing to their shape, size, or fragility, while others can be handed over with one hand. Sometimes the giver and/or the receiver need/s to perform a self-handover, i.e. transferring the object from one hand to another, to adjust the grasp. When people perform tasks requiring multiple handovers, they effortlessly utilize these different types of handovers.

To help design robot controllers and human motion models for bimanual and multiple sequential handovers, we build two multi-sensor datasets of

human-to-human handovers. The first dataset consists of 360 handover recordings generated from 12 pairs of participants performing both unimanual and bimanual handovers with 5 objects, and only bimanual handovers with 5 additional objects. The second dataset consists of 1440 handover recordings generated from the same 12 pairs of participants performing shelving and unshelving tasks involving object handovers with 30 objects. Both datasets include the giver's and receiver's upper-body skeleton tracking data, two RGB-D datastreams, annotations of the handover phase, and anthropometric measurements of the participants.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a summary of the existing body of work related to object handovers between two agents, including handovers between two humans and between a human and a robot. This is not an exhaustive survey, and we refer the reader to a recent review article on object handovers [122]. We begin with a brief survey on the roles that collaborative robots are currently playing in the workforce, and are envisioned to play in the future. Object handover being one of the most important collaborative tasks, this background serves as a motivation to study handovers and to develop robot controllers for human-robot handovers. We then present the insights from existing studies of human-human and human-robot handovers. These findings have helped researchers develop models of the human behavior in a handover, and also propose controllers for generating appropriate robot behavior in a handover. Finally, we present a survey of existing controllers for human-robot handovers.

2.1 Human-Robot Workforce

Robots are playing an increasing part in the economy. Industries are rapidly introducing small table-top robotic arms, also known as collaborative robots or cobots [35], to work alongside human workers on factory floors. The International Federation of Robotics reported that annual global robot installations increased by 6% in 2018 to 422,271 units, worth USD 16.5 billion (without software and peripherals) [68]. This report projects that in 2022 the annual sales volume will increase to 583,520 units with an increase of 12% on average per

year from 2020-22.

The main industrial applications of collaborative robots involve automotive assembly [15] [6] [113], material handling [56] and welding [33]. Though the automotive industry is the largest customer of collaborative robots, closely followed by the electrical/electronics industry, small and medium-sized companies are also increasingly moving towards a joint human-robot workforce [35]. There is a growing trend of automation and data exchange, labeled "Industry 4.0" [87]. This may lead to novel production systems characterized by individualized products under the conditions of a highly flexible mass production [168]. In such production systems, collaborative robots and intelligent decision-making software platforms can complement the flexibility and soft skills of human co-workers. Apart from civil applications, human-robot teams also find applications in urban search and rescue [70], firefighting [117] and surveillance [146]. The global military robots market is anticipated to grow at a compound annual growth rate (CAGR) of 9.0% over the forecast period 2017-2024 [121].

Object handover is one of the most common skills required for a collaborative or assistive robot. Tasks such as surgical assistance, housekeeping, rehabilitation assistance, and collaborative assembly require a robot to give objects to a human (robot-to-human handover) and take objects from a human (human-to-robot handover).

2.2 Human-Human Handovers

Humans perform object handovers seamlessly, thus human-human handovers are the benchmarks for human-robot handovers. Researchers have studied various aspects of human-human handovers to understand how people perform this complex maneuver.

2.2.1 Overall Structure of the Handover Process

Strabala et al. [163] proposed that the underlying structure of the handover process consists of two channels: physical and social-cognitive. The physical channel consists of three phases, the "approach" phase in which both actors move to close proximity, the "reach" phase in which both actors extend their arms towards the handover location, and the "transfer" phase, in which the object is transferred from the giver to the receiver followed by exiting the joint activity. The social-cognitive channel consists of the context i.e. establishing which object will be handed over, before beginning the "approach" phase, and the common ground in terms of when and where the object will be handed over.

The approach, reach, and transfer phases of handovers are smooth and dynamic actions, and blend into each other, instead of being separate and sequential actions, as shown by Basili et al. [7]. However, following a reductionist approach, researchers have studied individual phases of handovers to get a systematic understanding of the parameters and actions of each phase.

2.2.2 Approach Phase

Studies of the approach phase have examined where the approach phase ends and how the object is carried in the approach phase. Basili et al. [7] found that the mean interpersonal distance between the participants at the end of the approach phase is 1.16 m, but it varies considerably for different participants. In the studies conducted by Lee et al. [90], participants playing the role of a caretaker used both hands while carrying the object in the approach phase, and then transitioned to single-handed reach and transfer movements.

2.2.3 Reach Phase

Studies of the reach phase have provided a better understanding of the velocity profiles of the giver's and/or the receiver's hand when they reach towards the handover location. Shibata et al. [156] showed that the giver's hand moves along a nearly linear path with a bell-shaped velocity profile and the peak velocity is reached slightly before the half duration. They also found that the receiver's motion either followed a minimum jerk trajectory through a via point or followed a path computed online during the handover. Huber et al. [66, 64] also found that the hand movements had bell-shaped velocity profiles, higher velocities for larger movement amplitudes, and lower velocities for more accuracy. Parastegari et al. [128] also showed that the giver's hand moves with a bell-shaped velocity profile, with the peak velocity independent of the actor's pose and/or the object type in the absence of a physical constraint.

Mason et al. [107] found that during the reach phase the giver's kinematic as well as force-dependent variables were largely insensitive to the receiver's mo-

tion, whereas the receiver's motion differed when they grasped the object from a stationary vs a moving giver. Medina et al.'s [110] studies of human-human handovers revealed that the giver leads the motion during the reach phase and the receiver's motion is coupled to it. Roy et al. [148] investigated short-cycle repetitive handover tasks in three environments, real-world supermarkets, simulation, and laboratory. They found that the giver experiences higher loads on the lower back than the receiver, and the givers are perceived as faster than the receivers.

2.2.4 Transfer Phase

The transfer phase of handovers is characterized by haptic interaction between the giver and the receiver via the object. Chan et al. [30] found that the giver takes on the responsibility of ensuring object safety whereas the receiver controls the rate of load transfer, and in turn, the efficiency of the handover. They also found that the grip force to load force ratios of the giving task are similar to a picking up task, and the ratios of receiving task are similar to a placing task. Endo et al. [42] observed that the grip force of the giver during the reach phase is relatively independent of the variations in handover locations and receiver's motions. In their study, depending on whether the transfer was done at self-selected, fixed, or random locations, the giver's grasp release timing had different variability, with random location having the highest variability. Based on their findings, they suggested that the initial contact is performed in a feedforward manner whereas subsequent grip force modulation is carried out with haptic feedback. Medina et al. [110] found that the grip force applied by humans is linearly proportional to the load share of the object. Neranon [119, 120]

found that the mean peak forces were between 0.85-3.65 N for objects of weights 0.42 kg, 0.82 kg, and 1.22 kg.

Some researchers have studied the sensory inputs used by people during the transfer phase. Mason et al. [107] demonstrated that both the agents use somatosensory control to coordinate the transfer rate. Additionally, the giver uses visual feedback-based anticipatory control to precisely time initial grip force release. In their study, during conditions where the giver moved the object toward a stationary receiver, they noted a decoupling of the temporal and magnitude characteristics of the grip and inertial forces. Parastegari et al. [130] found that people primarily use visual feedback and not haptic feedback to detect the fall of the object during the transfer phase of handovers. Controzzi et al. [36] studied the effect of the receiver's reaching out speeds and the giver's sensory input conditions on the grip force modulation by the giver in human-human handovers. They found that the giver's speed of grip force release is correlated with the receiver's speed. Even when visual feedback is removed this correlation is maintained but the release is delayed. They also found that shorter release and handover durations and more reactive release behaviors are perceived as more fluent.

Prior studies have found that the time required to transfer the object from the passer to the receiver is between 300 and 500 ms [107, 30, 42, 119, 120].

2.2.5 Handover Location and Orientation

Much work has been done on investigating the handover location and orientation in human-human handovers. Basili et al. [7] and Hansen et al. [58] found

that the handover location is typically at the midpoint between both actors. Hansen et al.'s [58] studies revealed that the interpersonal distance between the giver and the receiver is inversely proportional to the height of the handover location, regardless of the object's weight. Rasch et al. [141] observed that the object is offered for transfer in two configurations, pronation where the object is covered by the hand, and supination where the object is visible to the receiver. Kato et al. [71, 72] showed that humans are able to predict the handover locations of partners they have just met, even without any visual feedback. They also found that the handover locations not only depend on the interpersonal distance but also depend on characteristics such as height, gender, and social dominance of both actors. Chan et al. [29] examined the handover orientations of common objects during human-human handovers and noticed that the handover orientations used by people for most objects follow observable patterns. They also found that natural handover orientations may differ from orientations that are convenient to the receiver and consider the usage of the object. Cini et al. [34] studied the grasp types and grasp locations in human-human object handovers. They found that the givers grasp the object in such a way that allows the receiver to comfortably perform the subsequent task with that object.

2.2.6 Handover Strategies

Some researchers have examined the strategies that people employ while transitioning from one phase to another during a handover. Strabala et al. [162] found that in a physical collaboration task involving handovers, people start the reaching motion even before they finish the approach phase. Shi et al. [154, 153] investigated human-human handovers in the context of the distribution of ob-

jects such as flyers to pedestrians. They found that in such scenarios the givers exhibited four handover strategies: extend arm first and wait for the pedestrian; wait for the pedestrian and extend arm after they get close; extend arm first and approach the pedestrian; approach the pedestrian and extend arm after getting closer to them. Huang et al. [63] studied multiple sequential handovers between two humans where the receiver had different levels of task demands. They found that the givers employed a combination of two adaptive coordination strategies viz. waiting and slowing down. In the waiting strategy, the giver waited for the receiver to complete their task by pausing their hand movement either near the object's pick-up location or in front of the body. In the slowing-down strategy, the giver reduced their hand velocity while the receiver was occupied with their task.

2.2.7 Non-verbal Communication

Surprisingly, non-verbal communication modes such as eye gazes, head movements, or hand gestures have been rarely studied in the context of human-to-human object handovers. Flanagan et al. [50] identified that during an object passing task, human eye movements and gazes focused on the end location where objects were placed rather than the grasped object and hand in motion. They also found that the gaze reached the end location well before the hand but exited at almost the same time. Similar eye movements were observed when participants watched the other person do the task. The receivers in studies conducted by Lee et al. [90] often signaled readiness by making a "grabbing" hand gesture, especially for fragile objects. The givers looked at the object or the receiver as they approached the receiver. In the case of a collaborative task,

Macdonald et al. [99] observed that people more often gazed at the shared task than at each others' faces. Strabala et al. [162] suggested that gaze cues are used to signal intent to start a handover, but they did not find mutual gaze to be an effective predictor of handover intent. Shi et al. [154, 153] investigated human-human handovers in the context of the distribution of objects such as flyers to pedestrians. They found that the givers kept gazing at the receiver to keep eye contact until the handover finished. Moon et al. [115] studied the giver's gaze in human-to-human handovers and found shared attention and turn-taking gaze to be the most frequent gaze cues during the handover. In shared attention gaze, the giver looked at the projected handover location and in turn-taking gaze, the giver initially looked at the projected handover location, and towards the end of the handover motion looked up at the receiver's face.

2.2.8 Miscellaneous

Some studies have revealed the effects of repeated trials and object's weight on the handover interaction. Shibata et al. [156] found that the duration of handover decreased with multiple trials. Huber et al. [66, 64] found that the reaction time and duration of the reach and transfer phases of the handover decreased over repeated trials. Endo et al. [42] observed that the giver's grip force at contact and the peak interaction force during contact gradually reduced over repeated trials. Chan et al. [30] and Hansen et al. [58] found that the weight of the object affected the duration of the handover, with heavier objects having a longer duration.

2.3 Human-Robot Handovers

HRI researchers have studied human-robot handovers to understand human preferences for robot behaviors in the approach, reach and transfer phases of handovers. Some studies have also examined the robot's non-verbal behaviors in handovers, and coordination strategies for handovers.

2.3.1 Approach Phase

Studies of the approach phase have examined the robot's direction of approach and interpersonal distance at the end of the approach phase. Walters et al. [172] found that most standing participants preferred a robot approaching from either the left or the right side, instead of a frontal approach. Whereas, Koay et al. [75] found that majority of seated participants preferred a robot approaching from the front and handing over the object just below their chest level. Both these studies found the preferred interpersonal distance between the human and the robot to be within the personal distance (0.6m - 1.25m), suggesting that people may treat robots similar to other humans. Koay et al. [75] also found that participants' personality traits such as Agreeableness and Openness influenced the interpersonal distance of the handover.

2.3.2 Reach Phase

Studies of the reach phase have evaluated the effects of the robot's velocity profiles and handover configurations on the fluency and subjective expe-

rience of handovers. Shibata et al. [155] compared four velocity profiles in one-dimensional robot-to-human handovers: triangle, trapezoid, and two bell-shaped profiles, and found that participants preferred a bell shaped velocity profile with the velocity peak at 30% duration. In their studies, the preferred handover location was around 350mm from the person's shoulder. Huber et al. [66] found that in robot-to-human handovers, a minimum-jerk velocity profile in spatial coordinates was perceived as safer than a conventional trapezoidal velocity profile in joint coordinates.

Koene et al. [76] studied the effect of timing and spatial performance of the robot on the human's subjective experience of human-robot object handovers. They found that the participants' ratings of ease, satisfaction, comfort, and safety were more strongly correlated with the timing parameters as compared to the spatial parameters. Their results suggest that the speed of handover carries greater importance than the accuracy of the robot's reaching motion. Cakmak et al. [19] studied the effect of spatial and temporal contrast in object configurations for robot-to-human handovers. Spatial contrast means that the robot's handover pose is distinct from poses for other actions with that object, while temporal contrast means that the robot has different poses during the approach phase and at the end of the reach phase. They did not see any effect of spatial contrast but found that temporal contrast improved the fluency of the handover.

Cakmak et al. [18] conducted a user study to learn human preferences towards handover configurations in robot-to-human handovers. They found that participants preferred robot configurations with higher arm extension, consistency along the three coordinate axes, and higher resemblance to human configurations. Pan et al. [125] found that speeds equal to or slower than human

speeds and a human-like sensorimotor delay before beginning the handover improved the perceived comfort during the handovers.

2.3.3 Transfer Phase

Studies of the transfer phase have investigated the effect of the robot's grasping or releasing behaviors on the subjective and objective parameters of human-robot handovers. Pan et al. [124, 123] studied the effect of grasp type during human-to-robot handovers on the interaction forces and the social perception of the robot. Specifically, they studied two grasp types: quick grasp and mating grasp. In quick grasp condition, the robot activated its electromagnetic gripper as soon as it moved to within 1cm of the object's end. In the mating grasp condition, the robot activated its electromagnetic gripper only after making co-planar contact with the object. Surprisingly, the mating grasp was perceived as less competent and more discomforting. The robot's grasp type also affected the human's retraction force, with larger retraction forces for quick grasps. They also noted that the interaction forces were significantly higher than those encountered in human-human handovers. Han et al. [57] compared three release behaviors in robot-to-human handovers: "proactive" in which the robot released the object when the rate of change of force at its end-effector went beyond a certain threshold, "passive" in which the robot released the object when the force at its end-effector went below a certain threshold, "rigid" in which the robot released the object only after completing the reaching motion. They found that the proactive release behavior resulted in quicker handovers and higher perceived fluency and ease of taking.

2.3.4 Miscellaneous

Some researchers have studied the effect of participants' experience of interaction with the robot on the subjective and objective metrics of human-robot handovers. Edsinger et al. [40] found that people hand over objects to a humanoid robot without explicit instructions or expertise. They showed that people place the objects directly in the robot's hand, after it executes a predefined reaching motion, and ensure an easy-to-grasp configuration. Borgsen et al.'s studies [189] tested whether the participant's prior experience of interaction with robots had any effect on the object handovers. They found significant differences in the behavior of naive users and experienced users, with naive users expecting the robot to visually monitor the handover process instead of just using the force sensor. Pan et al. [124] studied the effect of repeated handover trials on the social perception of the robot. They found that repeated trials increased the participants' comfort and emotional warmth towards the robot. The interaction forces also decreased over repeated trials, indicating that people's trust in the robot increases, and they learn to predict the robot's behavior over time.

2.3.5 Non-verbal Communication

Non-verbal communication consists of communication through modalities such as eye gaze, head gaze, and deictic or pointing gestures. Some researchers have studied the effects of robot gaze in robot-to-human handovers [1, 115, 188, 48, 83].

Admoni et al. [1] demonstrated that deliberate delays in the robot's release of an object draw human attention to the robot's head and gaze, showing in-

tentional communication through the gaze. Also, they found that such delays increased the participants' compliance with the robot's gaze behavior. Moon et al. [115] and Zheng et al. [188] demonstrated that gaze behaviors shown by the robot giver can affect the human receiver's experience and fluency of the handover. Moon et al. compared three different gaze behaviors: the "shared-attention gaze" in which the robot continuously looked at the handover location, the "turn taking gaze" in which the robot shifted its gaze from the handover location to the receiver's face midway through the handover motion, and the baseline "no gaze" condition in which the robot kept looking down at the ground. They found that when the robot exhibited the "shared-attention gaze", participants reached for the offered object significantly earlier than the other gaze conditions. Also, they found a statistical trend ($p < 0.10$) of participants preferring the "turn-taking gaze" over the other gaze conditions. Zheng et al. [188] extended this work and compared the "shared-attention" gaze with a continual robot gaze at the human receiver's face and a transitional gaze between the receiver's face and the handover location. They found that for the continuous face gaze condition the participants reached for the object even earlier than the "shared-attention" gaze. They also found that when the robot looked at the person's face, either continuously or in combination with the "shared-attention gaze", the participants considered the handover to be more likable and anthropomorphic, at $p < 0.05$ significance level. Zheng et al. [187] analyzed the videos of Moon et al.'s [115] study of robot gaze behaviors in robot-to-human handovers. Their analysis supported Moon et al.'s findings, that the robot gaze affected the reaching time of the human. Additionally, they found that the robot's face gaze was effective in drawing people's attention to the robot's face.

Fischer et al. [48] compared two gaze behaviors of a robot tasked with retrieving parts instructed by the participants and found that when the robot looked at the person's face instead of looking at the movement of its own arm, participants engaged more with the robot and felt more responsible for the task. Kuhnlenz et al. [83] compared gaze behaviors of a humanoid robot in a fetch-and-give scenario with the robot looking either towards the planned path or towards the human's face while approaching and handing over the object to the human. They found that looking at the human's face significantly increased the anthropomorphism, animacy, perceived intelligence, and social presence of the robot.

Other non-verbal gestures such as deictic or pointing gestures have rarely been studied in the context of human-robot handovers. Borgsen et al. [189] investigated whether additional gestures with the robot's second arm help synchronization in human-robot handovers. Specifically, they implemented two gestures: turning the hand in a presenting manner below the object to signalize readiness and moving the hand in a protective way above the object to signal that the robot is not yet ready. They found that though the participants perceived the robot as more human-like when it exhibited additional gestures with its second arm, those gestures did not have any significant effect on the synchronization of the handover.

2.3.6 Handover Strategies

For tasks requiring multiple sequential handovers, such as a collaborative assembly task, it is important for the robot to predict the timing of handovers.

Then it can initiate the handovers just-in-time and contribute to the task fluently. Therefore, some researchers have investigated different strategies for predicting handover timing and coordinating multiple sequential human-to-robot and/or robot-to-human handovers. Huber et al. [65] compared four different strategies of timing in human-robot handovers: average time, in which the robot initiates the handover after a fixed duration; sensor trigger, in which the robot initiates the handover only after the human requests; component average time, in which the robot uses precomputed durations of individual assembly steps to initiate the handover; and predictive assembly duration, in which the robot uses a probabilistic Bayesian framework in the form of a Kalman Filter to continuously update the handover timing based on observed assembly cycle time. They found that the predictive method outperformed the other three methods in terms of handover efficiency and fluency. In a similar study, Tang et al. [166] compared three methods of robot-to-human handover timing prediction in a repetitive assembly task: sensor trigger, component average time, and predictive assembly duration. They found that the predictive method was better in terms of average cycle time and subjective preference than the first method, and comparable to the trigger sensor method. They suggested that the predictive method would be much easier to implement on a production line as compared to the sensor trigger method.

Huang et al. [63] compared three robot-to-human handover strategies in a scenario where the human receiver was occupied with a secondary task. Their strategies involved: "proactive strategy" in which the robot started the handover even if the receiver was not ready, "reactive strategy" in which the robot started the handover only after the receiver was available, and "adaptive strategy" in which the robot used waiting and slowing-down strategies (explained in

Section 2.2.6). They found that the proactive strategy resulted in better team performance while the reactive and adaptive strategies offered a better subjective experience to the user. Lambrecht et al. [86] compared a voice command strategy with a reactive handover strategy and a manual assembly baseline. In the voice command strategy, the human instructed the robot via voice commands to achieve a just-in-time handover. They found that the participants were able to predict the handover timing but it increased the cognitive load on them. Martinson et al. [105] investigated whether users prefer a robot with the capability to execute more than one type of handover, and which factors influence the choice of one type over the other. They specifically considered three types of handovers: the robot approaching the human to deliver the object, the robot asking the human to approach and take the object, and the robot placing the object on the closest planar surface. They found that participants preferred a robot with the capability to execute multiple types of handovers, and the factors that influenced the preferred choice of handover interaction were the human's activity and the location of the handover. Quispe et al. [136] extended this work and proposed a method to learn human preferences for the type of handover interaction. They modeled the system as a Bayesian Network with inputs consisting of a discrete set of the human's activities and locations.

2.4 Models of Human's Motion in Handovers

In order to generate appropriate robot behavior during handovers, it is important to predict the human's motion. Some researchers have proposed models of human motion in the context of handovers. These models can be categorized into three types: offline, online, and hybrid. Offline models predict the human's

motion and/or the handover location before the start of the handover and do not update those predictions based on the observed motion of the human during the handover. On the contrary, online models continuously update their predictions taking into account the observed motion of the human during the handover. Hybrid models use a combination of offline and online predictions.

Offline Models

Parastegari et al. [128, 130] proposed a model for predicting the giver's reaching motion and handover location, based on dyadic joint torques on the two actors in handovers. Rasch et al. [141] studied the motion of individual joints during a single-handed human-to-human handover. They proposed a five-degree polynomial model for the human hand's trajectory in a handover.

Online Models

Widmann and Karayiannidis [178] used Dynamic Movement Primitives (DMPs) to model the human's reach phase motion in human-robot handovers. They used an extended Kalman filter for online estimation of DMPs' point attractor and timescale. Sheikholeslami et al. [152] used a planar elliptical model to predict the human's reaching motion in human-robot handovers. Wu et al. [179] used Gaussian Process Regression (GPR) for online prediction of the human's reach phase motion during human-robot handovers. They optimized the hyperparameters of the GPR model with an offline training phase and used the observed human motion to make future motion predictions online. Medina et al. [110] estimated the handover location online by modeling the human's

motion as a Linear Dynamical System (LDS).

Hybrid Models

Nemlekar et al. [118] developed a model for the prediction of handover location during human-to-robot handovers. They used Multi-dimensional Interaction Probabilistic Movement Primitives (Pro-MP) to model the reach phase of handovers and trained the primitives with demonstrations of human-robot handovers. At the beginning of the interaction, the handover location is predicted offline, considering the giver's kinematics, and it is updated dynamically during the interaction based on the observed motion of the human. Simmering et al. [158] developed a method to predict the handover location or object transfer point (OTP) in human-initiated human-robot handovers. Their method consists of two components: offline static OTP prediction and online dynamic OTP prediction. At the beginning of the handover, they predict a static OTP at the midpoint of the human's and the robot's position. This offline prediction is updated based on the observed motion of the human during the handover. They used a minimum jerk trajectory model for this online update of the OTP.

2.5 Controllers for Single Handed Human-Robot Handovers

Handovers being a fundamental action in physical human-robot collaboration, the importance of this task has resulted in a large body of work on robot controllers for handovers. These controllers use a variety of sensor interfaces such as physical sensors, visual sensors, and wearable devices [89]. Researchers have proposed various methods to detect the start of handovers, detect the transitions

between phases, and control the robot in each of the handover phases.

2.5.1 Detection of Intent to Perform Handovers

Some researchers have developed methods to recognize the human's intention to start the human-to-robot or robot-to-human handover from visual, tactile, or wearable sensor data. Pan et al. [126] developed a classifier to detect the giver's intent of handover for robot-to-human handovers. They used the human giver's kinematic features such as joint angles, distances of joints from each other and from the receiver, and trained a Support Vector Machine (SVM) to recognize the start of object handover. Wang et al. [175] developed a wearable sensor to detect the human's intentions during object handovers to a robot, and also enable the human to control the handover. The sensor provides measurements of the human's forearm postures and muscle activities, which are used to detect intentions such as *give*, *stop*, *continue*, *speed up* and *slow down*. The robot also uses this data to check if the object is graspable in terms of its dimensions and weight. Davari et al. [37] used tactile sensor data from the robot's fingers to detect the type of interaction during the transfer phase of a robot-to-human handover. They used the temporal tactile data stream and trained an SVM to classify whether the object in the robot's hand is being pushed, pulled, held, or bumped by the human receiver.

2.5.2 Reach Phase

Existing robot controllers for the reach phase of handovers can be broadly categorized into two groups: offline and online. Offline controllers (or open-loop controllers) compute the robot's motion plan before the start of the reach phase and do not update it during the reach phase, whereas online controllers (or closed-loop controllers) constantly update the robot's motion plan during the reach phase and take into account the observed behavior of the human.

Offline (open-loop) controllers can be further categorized into three types:

- Pre-programmed or hand coded controllers
- Optimization based motion planners
- Controllers learned from demonstrations or feedback

Online (closed-loop) controllers can be further categorized into two types:

- Controllers learned from demonstrations or feedback
- Velocity controllers towards human hand or predicted handover location

Offline: Pre-programmed or Hand-coded Controllers

In pre-programmed or hand-coded controllers, the handover configuration, and sometimes even the velocity profile of the robot, is programmed manually. For example, Aleottie et al. [4] developed a controller for robot-to-human handovers that plans robot motions to a predefined handover location and orientation. Their controller also included the pick-up phase of the handover.

As described in Section 2.3, researchers have used pre-programmed or hand-coded controllers to study different aspects of human-robot handovers like the reach phase [155, 66, 18, 19, 76], the transfer phase [57], non-verbal communication [1, 115, 187, 48, 188, 83, 189], effect of the human's expertise [40, 189], and handover timing strategies [63, 105, 86, 166].

Offline: Optimization based Motion Planners

Going beyond hand-coding the full robot trajectory, a few researchers have proposed offline motion planners for the reach phase of object handovers. These planners compute the handover configuration and the velocity profile of the robot with the help of some predefined objective functions or constraints. Some offline motion planners utilize the human kinematic or biomechanical models to pose the planning problem as a constrained optimization problem. The motion plan is computed at the beginning of a handover and not updated during the runtime.

Micelli et al. [112] used a randomized planner to produce collision-free and reasonably smooth paths towards the initial position of the human hand in robot-to-human handovers. Mainprice et al. [103] and Sisbot et al. [161] developed a motion planner for robot-to-human object handovers that calculates visible, safe and comfortable robot paths. They defined three constraints in the form of cost maps: distance, visibility, and comfort, and used them to compute the object transfer point as well as the robot's trajectory. Mainprice et al. [102] developed a motion planner for robot-to-human handovers that considers human mobility, to choose the appropriate handover configuration and robot trajectory. They evaluated this algorithm in human participant experiments and

found that accounting for the human's mobility resulted in more fluent and efficient handovers. Cakmak et al. [18] used the kinematic model of the human to compute robot motions in robot-to-human handovers, optimizing ease of taking. Aleotti et al. [3] developed a motion planner for robot-to-human handovers that calculates robot paths considering the most convenient object orientation. In their user studies, the participants reported that the robot offered objects in a comfortable way and complied with their affordances.

Suay and Sisbot [164] developed an approach to estimate personalized human biomechanical models and to compute the handover location that minimizes the static load on the human in robot-to-human handovers. Peternel et al. [131] used a human biomechanical model to compute the handover location that minimizes the load on the human's joints. Bestick et al. [10] developed a framework for selecting object handover configurations in robot-to-human handovers that minimize the ergonomic cost to the human, both during the transfer phase of the handover and placement of the object at its target. Their studies showed that participants preferred this method over handover configurations that only maximize the number of grasping locations available to the human during transfer.

Walker et al. [171] and Rahman et al. [137] developed a motion planner for robot-to-human handovers that considers the human's trust in the robot, to adapt the robot's handover motions. Specifically, for low trust, the robot performs cautious handovers characterized by reaching motions with lower speeds and handover configurations that reduce potential impact forces on the human. They evaluated this motion planner in an assembly task and found that it improved the perceived safety, human trust in the robot, handover success rate,

and overall assembly efficiency. Pyo et al. [135] developed a motion planner for a fetch-and-give task with a wagon and a service robot. Their planner calculates the motion plans for the robot to grasp the wagon, approach the human while pushing the wagon and deliver an object from the wagon to the desired position. The planner considers collision avoidance and manipulability of both the agents. Waldhart et al. [170] proposed a graph-based planner for situations requiring several sequential handovers between multiple agents (both humans and robots). Their method computes the sequence of agents, handover locations, and motion plans for the robot. The robot motions are computed with collision avoidance and considerations of human comfort and safety.

Offline: Controllers Learned from Demonstrations or Feedback

Another direction of research has focused on learning the reach phase motions from demonstrations of human-human handovers or human-guided robot motions.

Cakmak et al. [19] identified the robot's poses for the approach and reach phases of handovers from a user survey, and used an existing motion planner [9] to execute the handovers. Cakmak et al. [18] obtained robot-to-human handover configurations from demonstrated good handovers. They found this approach resulted in handovers with higher rankings of liking, naturalness, and appropriateness but less reachability, as compared to an optimization-based motion planner that used the human kinematic model to optimize ease of taking. In Koay et al.'s [75] studies investigating the location of a robot handing over objects to a seated human, the participants interactively created the robot's handing over motion. Similarly, Rasch et al. [144] asked participants to interactively

guide a humanoid robot into different handover configurations and label the configurations as proper or improper. They found that the proper handover configurations can be grouped into three clusters, each differing primarily in the rotation of the forearm. The combination of object type and grasp type was strongly correlated with the cluster of handover configurations. Also, the height of the handover configuration was found to depend on the height of the receiver and the receiver's posture (sitting or standing).

Momi et al. [38] used a neural net to generate human-like reaching motions of a robot arm given a target position. They used three feedforward perceptron neural networks (NNs) to acquire the trajectories for the x,y, and z axes. They trained the NNs with recorded trajectories of the reaching behavior of a person. Tang et al. [165] used a Continuous Time Recurrent Neural Network (CTRNN) to learn robot-to-human handover trajectories with obstacle avoidance. They trained the CTRNN with motion sequences gathered via human-guided motion for three different obstacle heights. In the testing phase, the network would generate the robot's joint trajectory for a given obstacle height. Both Momi et al.'s and Tang et al.'s controllers are black box approaches and the robot's trajectory cannot be modified without retraining the network.

Rasch et al. [142] developed a joint motion model for robot-to-human handovers from motion profiles of human joints in human-human handover demonstrations. They modeled the shoulder joint motion and adduction with a sigmoid function, and the elbow joint and pronation-supination with a polynomial function. Their studies revealed that on an industrial robot this joint motion model was perceived to be significantly safer than a linear joint space trajectory. They used a predefined handover location in their studies. Rasch et

al. [143] proposed a combined Cartesian and joint motion model for the reach phase of human-robot handovers. Their controller computes the robot's trajectory towards a predefined target with a predefined duration. Both the components of the model, Cartesian and joint motion, are learned from demonstrations of human-human handovers. They suggested that such a combined model has advantages like anthropomorphic joint-state motion as well as obstacle avoidance in the Cartesian space.

Bestick et al. [11] developed a method to learn peoples' preferences of handover configurations in robot-to-human handovers. They formulated an online Bayesian inference problem to learn the ergonomic cost function of the human associated with handover configurations. They proposed two approaches for making queries: a passive approach in which the robot always tried the most comfortable configuration for the person and an active approach in which it tried the configurations with the highest expected information gain. They found that in simulations the active approach learned faster than the passive approach, but with real humans, the difference was very subtle. Chan et al. [28] proposed a method to learn appropriate handover configurations for robot-to-human handovers from demonstrations of natural handover orientations used by people. They used the idea of affordance axes to determine the quality of a set of orientations and developed a method to compute the mean orientation of that set. Their tests on a set of handover data revealed that their method gives better results in terms of receiver-centered orientations from observations of handovers. Chan et al. [27] developed a framework for learning handover configuration i.e. grasp location and orientation, for robot-to-human handovers from human demonstrations. Their system does not require additional cameras or markers and only uses a skeleton tracker and a particle filter-based object tracker. How-

ever, the object needs to be placed on a table for initial detection and the system cannot learn multiple handover configurations for the same object. Chan et al. [26] presented a method to determine appropriate object configurations for robot-to-human handovers. During the training phase, they used affordance information from demonstrations of the object's usage, and object handover configurations from human demonstrations. They generalized the appropriate handover configurations to new objects by grouping objects according to their affordances.

Online: Controllers Learned from Demonstrations or Feedback

The offline controllers that are learned from demonstrations, as described above, do not update the robot's motion plan during the handover. To overcome this drawback, some researchers have proposed methods to learn online controllers from demonstrations of human-human handovers or from human feedback.

Some researchers have used movement primitives to imitate the demonstrated human reaching motions in handovers. Prada et al. [132] used Dynamic Movement Primitives (DMPs) to model the reaching motion of a human in object handovers. They modified the original DMP formulation of Schaal [150] by changing the weights of the two components of the DMP system, such that it resulted in a predominantly feedforward motion in the earlier phase and a predominantly goal-directed motion towards the end. They also added a velocity feedback term to consider the target's, i.e. the human hand's, velocity for smoother motion and convergence. They showed that the predictions of this model matched well with the ground truth data of human-human object

handovers. Extending this work, Prada et al. [133] implemented and evaluated a control system for human-robot handovers based on DMPs. This strategy produces robot motions similar to human demonstrations with the human hand as the target, thus maintaining a feedback mechanism for changes in handover location to provide flexibility. Maeda et al. [100, 44, 101] used Probabilistic Movement Primitives (ProMPs) for imitation learning of human-robot interaction tasks such as handovers. Their controller produces robot trajectories towards the observed location of the human hand, such that the robot motion is similar to the training demonstrations. They compared ProMPs with DMPs and found that ProMPs are better at dealing with noisy data and/or data with interruptions. Vogt et al. [169] demonstrated an imitation learning methodology to learn parameters for the reach phase trajectory of human-to-robot handovers, from a single human-to-human handover demonstration. Their method involved the generation of triadic interaction meshes (IMs) to link the interaction partners with the transferred object. During human-robot handovers, these IMs are used to synthesize the robot's motion corresponding to the observed human motion. Their controller resulted in a higher success rate, faster interaction, better spatial generalization, and lower frustration for the user, as compared to the controller used by Huang et al. [63] and traditional IMs [61]. However, this approach requires one demonstration for each object and is limited in terms of generalization of the learned handover over different locations.

Some researchers have used reinforcement learning techniques to learn a policy for the reach phase of handovers from human feedback. Kupcsik et al. [84] presented an algorithm to learn a latent reward function for robot-to-human handovers under dynamic situations and noisy human feedback. In this approach, the robot learns the policy for selecting handover controller param-

eters given the context. The latent reward model for updating the policy from human feedback incorporates both absolute feedback and preferential feedback. Riccio et al. [147] developed a method to train a policy for human-robot handovers and also learn spatio-temporal affordance maps for the robot's actions. They used a reinforcement learning approach, i.e. iteratively improve the policy, given a reward function, by executing the policy in the real world or simulation. Their method can take into account prior knowledge in the form of an initial data set of samples.

Other approaches have used dynamical systems, look-up tables, or neural networks to encode the demonstrations and generate robot motion in the reach phase. Medina et al. [110] employed a Coupled Dynamical System (CDS), learned from human-human demonstrations, to drive the robot towards the predicted handover location. Sidiropoulos et al. [157] proposed an online controller for robot-to-human handovers that utilized a Dynamical System (DS), learned from human demonstrations of handovers, for generating the reach phase motion of the robot. The DS generates the target velocities for the robot end-effector, considering the position and orientation of the human hand. Yamane et al. [180] developed a method for generating humanoid robot motions for human-to-robot handover by looking up the passer motion in a human-motion database. Zhao et al. [186] developed an online controller for human-robot handovers that used Recurrent Neural Network with Long Short-Term Memory units (LSTM-RNN). The network learns the interaction model through offline training with demonstrated human-robot handovers. The robot is operated via a kinesthetic teaching interface during the training phase. Their user studies showed that this controller is more accurate than a controller based on ProMPs [100]. Yang et al. [182] learned a deep model of the grasps from a dataset

of human grasps, and used it to plan the robot's reach phase motion. Their user study revealed that their controller resulted in more fluent handovers than baselines which used a fixed grasp configuration and a grasp position inferred from the human hand pose.

Online: Velocity Control towards Human Hand or Predicted Handover Location

The simplest online controllers for the reach phase of handovers have used a visual servoing approach i.e. driving the robot towards the human hand (for robot-to-human handover) or the object (for human-to-robot handover). Micelli et al. [112] presented a visual servoing controller for human-to-robot handovers. This controller updates the robot's motion plan continuously by generating velocities proportional to the error between the human hand's position and the robot gripper's position. In their user studies, they found that this controller had a higher handover rate as compared to an offline motion planner (mentioned in Section 2.5.2), but it had a lower handover success rate. Also, the offline motion planner was perceived as more aggressive, in comparison to the online velocity controller. Bdiwi et al. [8] also used visual servoing towards the human hand (human-to-robot handovers) or the object (robot-to-human handovers) during the reach phase of handovers. They included collision avoidance with the human body as a safety measure by combining the visual and force sensor data. Their controller also performed the pick-up or drop-off phases of handovers with visual servoing. Pan et al. [124, 123] and Melchiorre et al. [111] also used proportional velocity control to drive the robot towards the human hand or the object in the reach phase.

Instead of using a proportional velocity controller, some researchers have used other velocity profiles to drive the robot towards the human hand/object or the predicted handover location. He et al. [60] developed an online controller for the reach phase of human-robot handovers. Their approach generates robot motion online by joining up the target trajectory or a target point, provided by a human-aware offline motion planner, from the current state. Their controller can perform target tracking, trajectory tracking in task frame, and path re-planning and trajectory switch, in case of movement of obstacles. Fishman et al. [49] developed an online trajectory planner for the reach phase of human-robot handovers. They used Model Predictive Control (MPC) algorithm to simultaneously predict the human's reaching motion and plan the robot's movement towards the human's predicted position. Pan et al. [125] proposed an online controller for bi-directional human-robot handovers. They used Bézier curves to generate smooth minimum-jerk motions during the reach phase, initially towards an expected handover location and eventually converging to the human hand. Scimmi et al. [151] developed an online controller for the reach phase of human-robot handovers. They used a smooth predefined velocity profile for the robot end-effector, as a function of the distance between the end-effector and a location close to the human's hand.

Miscellaneous

Marturi et al. [106] developed a two-layered controller for the reach phase of human-to-robot handovers. Their controller generates feasible grasp locations for unknown objects offline and tracks the object online to plan robot motion towards a grasp location that minimizes task space error. The controller

switches between two planners: a local planner that continuously plans robot motion towards the current grasp location, and a global planner that switches the grasp location if the current grasp becomes infeasible. Erden et al. [43] developed a fuzzy-logic-based controller for the approach and reach phases of two-dimensional human-robot handovers. Their controller used a multi-agent system with individual joints treated as different agents. They tuned the parameters of the controller with the genetic algorithm to minimize deviation from a direct path, time duration, and energy consumption during the handover. Wu et al. [179] developed an impedance controller for generating the robot's reaching motion in response to the human's motion.

2.5.3 Transfer Phase

Existing controllers for the transfer phase of handovers can be broadly categorized into four groups: position controllers with position feedback, position controllers with force feedback, force controllers with force feedback, and force controllers with multiple feedback modalities. Position controllers either use position feedback or force feedback and control the gripper fingers' positions to trigger the grasp/release of the object during the transfer phase. While these approaches are simple to implement, they are limited in terms of the haptic interaction that characterizes the transfer phase. Force controllers overcome these limitations by controlling the grasping force applied by the robot on the object, with feedback from force sensors or a combination of multiple tactile and position sensors.

Position Controller with Position Feedback

Kim et al. [74] developed a grasp planner for the transfer phase of human-robot handovers, which controls the position of the gripper with visual feedback of the object. Their planner works for parallel jaw grippers and polyhedral objects. Their controller takes into account etiquettes such as object shapes, object functions, and the safety of the human. Aleottie et al. [4] used feedback from a range sensor to detect the human grasp and trigger the release of the object.

Position Controller with Force Feedback

Many researchers have used a position controller that triggers the opening/closing of the gripper when the external force on the gripper exceeds a threshold [8, 77, 189, 57, 125, 157]. The threshold for triggering the action is typically derived from experimentally collected data for the specific robot.

Force Controller with Force Feedback

Some researchers have proposed force controllers to modulate the grasping force as a function of the external force on the gripper [40, 31, 134, 41] or the load-share i.e. the proportion of the object's weight supported by the robot [110, 84]. Edsinger et al. [40] used a force controller to compliantly grasp the object handed over by the human. Chan et al.'s [31] controller modulated the applied grip force as a linear function of the measured load on the robot's gripper. The parameters of the function were obtained from their previous studies of human-human handovers [30]. Their user studies revealed that participants preferred this controller as compared to a constant grip-force controller. Psomopoulou

and Doulgeri [134] used fingertip rolling to grasp the object and developed a force controller to modulate the grasp force with a rolling contact as a function of the estimated load of the object's weight. The object's weight and shape are not known apriori. However, their approach applies only to two-dimensional grasps. Eguíluz et al. [41] controlled the grasping force as a function of the applied force perturbation direction. They used tactile sensing to detect the direction of external forces and joint effort control to manipulate the grasping force. The parameters for their algorithm were obtained from controlled experiments. The parameters need to be tuned for the specific handover scenario and individual human. Medina et al. [110] used insights from human-human handovers to control the robot's grip force as a function of the load-share estimate. They found that this approach reduced the transfer duration and internal forces between the robot and the human, as compared to a force threshold-based controller. Kupcsik et al. [84] used reinforcement learning to tune the parameters for their force controller, which modulated the grip force as a linear function of the load-share estimate.

Force Controller with Multiple Feedback Modalities

Some researchers have proposed to use feedback from multiple types of sensors to modulate the grip force during the transfer phase of handovers. Grigore et al. [55] proposed a two-layered controller which included the human's eye gaze and head orientation, in addition to the force applied by them, to trigger the release of the object. Their experiments demonstrated that this approach improved the success rate and safety of handovers. Parastegari et al. [129] proposed a controller with re-grasping capability to ensure fail-safe human-robot

handovers. They used an optical sensor to monitor the object's acceleration and to detect object-fall resulting from an imperfect handover. Their controller combined the acceleration sensing with a force sensor attached to the wrist of the robot and adjusted the grip force to ensure a safe transfer of the object to the human.

Miscellaneous

Other miscellaneous controllers for the transfer phase of handovers have used electromagnets [124, 123], soft gripper [13], and vibration sensing [160]. Pan et al. [124, 123] used an electromagnet to grasp the object in the transfer phase. They used both force and position feedback to activate the electromagnet. Bianchi et al. [13] developed a controller for grasping the object with a soft gripper during the transfer phase of human-to-robot handovers. They created grasp primitives from human-guided demonstrations of object grasps with a soft robotic gripper. Their method used acceleration contact signals from the robot's fingertips, wrist motion, and hand closure commands to train the grasp controller. Singh et al. [160] developed a sensor for detecting interaction forces during the transfer phase of a robot-to-human handover. Their system consisted of a vibrator on one finger of the gripper and an accelerometer on the other finger. Based on the vibration analysis, this setup detected the type of interaction among grasp, touch, rest, and hit.

2.6 Two Handed Human-Robot Handovers

Some researchers have studied human-robot handovers of objects requiring two-handed grasps. Kim et al. [74] developed a grasp planner for the transfer phase of two-handed human-robot handovers. They control the position of the gripper with visual feedback of the object taking into account etiquettes such as object shapes, object functions, and the safety of the human. However, their planner works only for parallel jaw grippers and polyhedral objects. Bestick et al. [12] developed an approach to estimate personalized human kinematic models, and used these models to compute the handover location in two-handed robot-to-human handovers. Salehian et al. [159] developed a controller for bimanual coordinated reaching behaviors of a robot towards a moving object. They proposed a linear parameter varying dynamical system for generating the robots' motions towards a virtual object that asymptotically converges to the real object. The parameters of the dynamical system are estimated from kinematically feasible demonstrations with gaussian mixture regression. They extended this approach to include both asynchronous (independent) and synchronous (co-ordinated) bimanual motions [149]. They also devised a data-driven centralized IK solver for self-collision avoidance. The drawback of this approach is that the motion of the robots is not human-like. Also, the feasible reaching points on the object have to be manually defined by the user.

CHAPTER 3

TIMING-SPECIFIED CONTROLLERS WITH FEEDBACK FOR HUMAN-ROBOT HANDOVERS

3.1 Introduction

We present two controllers for human-robot handovers, in which end-users can specify the robot's behavior in terms of reach-to-handover timing parameters, along with two studies evaluating the controllers' performance. Prior studies of human-robot handovers did not consider scenarios where participants need to design the robot's handover controllers for a collaborative task with timing constraints. Such flexibility is important in industrial settings with dynamically changing task requirements, and it can increase the acceptance of robots as coworkers. The majority of existing handover controllers require tuning non-intuitive controller parameters to specify the robot's behavior, for example, proportional gain [112, 8, 111, 110] or Dynamic Movement Primitive (DMP) parameters [132, 76]. Some researchers have proposed handover controllers with intuitive parameters, for example, timings of different phases of handovers [80] and delay/reach timings [125], but these controllers were not evaluated experimentally in scenarios where end-users can specify these parameters.

Our proposed controllers not only allow users to specify intuitive timing constraints for the object handover but also provide feedback to the end-user if they cannot satisfy those constraints, which can be used to better tune the

A substantial portion of this chapter is taken from [82] ©2022 IEEE. Reprinted, with permission, from A. Kshirsagar*, R. Ravi*, H. Kress-Gazit and G. Hoffman, "Timing-Specified Controllers with Feedback for Human-Robot Handovers", IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Naples, Italy, 29 August-2 September 2022

controller. We use Receding Horizon Control (RHC), also known as Model Predictive Control (MPC), to generate the robot's closed-loop motion towards the human's hand from the user-specified timing constraints. We implement the controllers on a collaborative robot with two different motion strategies, encoded as objective functions and timing constraints: minimum cumulative jerk (MCJ) and minimum cumulative error (MCE). In the MCJ strategy, the users can specify the maximum reaching time of the robot and the controller generates a minimum jerk trajectory towards the human hand within the specified time limit. In the MCE strategy, the users can specify a stall time along with the maximum reaching time of the robot and the controller generates a trajectory with a minimum cumulative error from the human hand.

We conduct two studies to evaluate the user experience and task performance with each motion strategy. In our studies, users interact with a physical robot to handle time-sensitive "vaccine vials," which have a minimum and a maximum air exposure time. The vials need to be packaged at a variety of locations, making the robot's handover strategy sensitive to the object and the final location. Participants design the robot's handover behavior by specifying the controller parameters for each controller and then have to use the controllers they tuned to perform the collaborative vaccine packaging task. Motivated by the fixed-pay and incentive-pay compensation schemes, we consider two scenarios of the task: first, users tune the controller parameters only for successfully performing the task, and second, they optimize the controller parameters to reduce the task duration and earn additional monetary rewards.

Each of the two controllers is compared to a baseline proportional velocity (PV) controller, a commonly used closed-loop controller for generating reach-

to-handover motions [112, 111]. In the PV controller, the robot's instantaneous cartesian velocity is proportional to the distance from the human's hand, and the users can tune the proportional gain.

We find that the timing controller with the MCE strategy can provide a better user experience and fewer failures compared to the PV controller, but we did not find any evidence to support that the timing controller with the MCJ strategy performs better than the baseline PV controller.

Note on contributions

This work was done in collaboration with Rahul Ravi (Master of Science, Mechanical Engineering, Cornell University, 2021). Both collaborators had an equal contribution in the experiment design and data collection. The controllers and experiment design is described in detail in [145], and this chapter provides a shortened description for completeness. This chapter presents additional results that are not in [145].

3.2 Handover Controllers with Timing Parameters

We hope to understand how to design handover controllers that can be tuned intuitively by end-users. We seek to investigate how user-specified timing constraints should translate into controller parameters in a way that results in a high user experience and an efficient and useful controller. Our controllers in this study focus on the "reach" phase of the robot-to-human handover.

We propose two MPC [20] closed-loop controllers with reach time param-

eters for generating reach-to-handover motions of a robot. The MPC framework computes control inputs $\mathbf{u}_{t:t+L-1}$ for a given time horizon of length L, so that these controls optimize an objective function J under a set of N constraints $\{h_1, \ldots, h_N\}$. The objective function and constraints are defined over the system state $\mathbf{x}_t$ at time step t and the control input. Formally, MPC solves

$$\min_{\mathbf{u}_{t:t+L-1}} J_{t:t+L-1}(\mathbf{x}_t, \mathbf{u}_{t:t+L-1}), \tag{3.1a}$$

$$\text{s.t. } h_n(\mathbf{x}_i, \mathbf{u}_i) \leq 0,\ \forall i \in [t, t+L-1],\ \forall n \in [1, N] \tag{3.1b}$$

We formulate two instances of the MPC optimization problem for a robot's reach-to-handover motion generation. The two instances differ in terms of the objective functions and timing parameters that are specified by the user. We convert these timing parameters into MPC constraints, and the constrained optimization problem is solved at each time step in a receding horizon mode. The control input is given as the robot's end-effector velocity. If there is no feasible solution to the optimization problem at any time step, feedback is provided to the user that the robot is unable to reach the handover location within the specified constraints.

3.2.1 Minimum Cumulative Jerk (MCJ) Controller

In our first MPC implementation, users can specify the maximum reaching time of the robot and the controller generates a minimum jerk trajectory towards the human hand. Minimum Jerk is one of the most prominent models of point-to-point human arm movements [51, 156, 173]. Given a maximum reaching time, this controller should generate a smooth trajectory toward the human's hand.

To maintain smoothness (*i.e.,* minimize jerk), this strategy tends to generate a slow robot motion that still stays within the timing constraint. Formally, the MPC objective function and constraints for this strategy are given by:

$$\min \sum_{i=0}^{L-1} \ddot{\mathbf{u}}_{t+i}, \tag{3.2}$$

s.t.

$$\mathbf{x}_{t+i+1} = \mathbf{x}_{t+i} + \mathbf{u}_{t+i}\Delta t \; \forall i \in [0, L-1], \tag{3.3a}$$

(System Dynamics)

$$||\mathbf{u}_{t+i}|| \leq u_{max} \; \forall i \in [0, L-1], \tag{3.3b}$$

(Velocity Limit)

$$\mathbf{x}_0 = [p_x, p_y, p_z]\,, \tag{3.3c}$$

(Initial End Effector Position)

$$\mathbf{u}_0 = [0, 0, 0]^T\,, \tag{3.3d}$$

(Zero Initial Velocity)

$$||\mathbf{x}_{i_{reach}-1} - \mathbf{h}_{i_{reach}-1}|| \leq \epsilon_x, \tag{3.3e}$$

(Get Near Human Hand at Reach Time)

$$||\mathbf{u}_{i_{reach}-1}|| \leq \epsilon_u. \tag{3.3f}$$

(Reach Velocity near Zero)

In the above formulation, the current time step is t, the prediction horizon is L, Δt is the time step size, $\mathbf{u}$ is the control input i.e. robot end-effector's velocity, $\mathbf{u_{max}}$ is the maximum permissible speed of the robot's end-effector, $\mathbf{x}$ is the position of the robot's end-effector, $[p_x, p_y, p_z]$ is the initial Cartesian position of the robot's end-effector, i_{reach} is the time step corresponding to the

user-specified reach time t_r, $\mathbf{h}$ is the position of the user's hand, ϵ_x and ϵ_u are the position and velocity tolerances.

3.2.2 Minimum Cumulative Error (MCE) Controller

Our second MPC implementation uses a minimum cumulative error (MCE) optimization strategy, meaning that the robot will try to minimize the total distance from the human hand over the whole trajectory. In contrast to the smooth but slower trajectories produced by the MCJ controller, we expect the MCE controller to generate fast robot trajectories, as these will have the lowest cumulative error. To allow users to also tune the lower bound of the reach time, users can specify a stall time along with the maximum reaching time of the robot and the controller generates an MCE trajectory after stalling for the specified amount of time.

Formally, the MPC objective function for this strategy is represented by,

$$\min \sum_{i=0}^{L-1} ||\mathbf{x}_{t+i} - \mathbf{h}_{t+i}||^2. \tag{3.4}$$

The constraints are identical to the MCJ formulation, with one exception: Eq. 3.3d is replaced by,

$$\mathbf{u}_i = [0, 0, 0]^T \ \forall i \in [0, i_{stall} - 1]\,, \tag{3.5}$$

which means that the robot end-effector's velocity is 0 for the user specified time steps i_{stall}.

3.2.3 Baseline Controller

To evaluate the user experience and task performance of the two timing controllers, we compare them with a proportional velocity (PV) baseline controller. PV is a widely used closed-loop controller for generating reach-to-handover robot motions [112, 111]. This controller commands the robot's instantaneous Cartesian velocity $\mathbf{u}_t$ to be proportional to the distance between the robot end-effector's position $\mathbf{x}_t$ and the position of the human's hand $\mathbf{h}_t$ *i.e.*,

$$\mathbf{u}_t = k_p||\mathbf{x}_t - \mathbf{h}_t||, \tag{3.6}$$

where k_p is the user specified proportional gain. The control input is limited by the maximum permissible velocity of the end-effector, *i.e.*, $||\mathbf{u}_t|| \leq u_{max}$.

3.3 Study Design

We conducted two studies to compare the user experience and task performance of our proposed timing controllers with the baseline PV controller in an industrial task simulated in a lab setting. The first study evaluated the MCJ controller and the PV controller, and the second study evaluated the MCE controller and the PV controller. In each study, we performed this comparison in two scenarios: first, a "non-optimization" scenario where users need to tune the controller parameters to successfully perform the task, but not necessarily in the shortest possible time, and second, an "optimization" scenario where they need to optimize the controller parameters to reduce the task duration and earn additional money. Our study design was motivated by these research questions: Whether and to what extent do the proposed controllers offer a **better user experience**

than a baseline controller? Whether and to what extent do the proposed controllers **improve efficiency** of handovers as compared to a baseline controller?

3.3.1 Hypotheses

Since the timing controllers allow users to specify intuitive parameters and provide feedback to the user, we hypothesize that **H1:** *A timing controller will provide a better user experience than a PV controller*. We formulate similar hypotheses for three components of user experience: efficiency, perspicuity, and dependability. **H1a:** *A timing controller will have higher efficiency than a PV controller*. **H1b:** *A timing controller will have higher perspicuity than a PV controller*. **H1c:** *A timing controller will have higher dependability than a PV controller*. Since the timing controllers allow users to specify time constraints on the robot's motion, we hypothesize that **H2:** *A timing controller will have lower failures in a time-constrained task than a PV controller*. We do not expect the timing controller to generate slower trajectories than a PV controller. Thus, we hypothesize that **H3:** *A timing controller will not have a higher task duration in a time-constrained task than a PV controller*. **H4:** *A timing controller will not have higher human idle time in a time-constrained task than a PV controller*. **H5:** *A timing controller will not have higher robot idle time in a time-constrained task than a PV controller*. We preregistered on a public experimental registry[1]. The experiment protocol was approved by Cornell University's Institutional Review Board for Human Participants.

[1]https://aspredicted.org/blind.php?x=5uk95q

3.3.2 Collaborative Task with Time Constraints

We tested the different robot controllers in a human-robot collaboration task consisting of object handovers from the robot to the human. The task simulates a scenario of vaccine packaging in the pharmaceutical industry: A person and a robot work together to transfer vaccine vials from a storage location to different packages while ensuring that vaccines do not get over-exposure or under-exposure to air, *i.e.*, the handover time of the vial from the robot to the human falls within acceptable bounds. The experiment setup, as shown in Fig. 3.2, consists of three types of vaccines, G (green), Y (yellow), and R (red), and three packaging stations at different distances from the storage rack. The robot arm picks up the vials sequentially from the storage rack and hands them over to the person who has to place the vials in the packaging stations. One limitation of the MCE, MCJ, and PV controllers is that they do not consider the 3D orientation information, and the robot always has a fixed orientation in our studies. The person is not allowed to enter the refrigeration zone around the storage stack, and if their hand is in this region the system generates a loud beeping sound. The robot starts the handover only if the person's hand is in the reachable workspace of the robot. The handover time for each vial depends on the parameters of the robot's controller which are specified by the user before performing the packaging task with the robot. Each vaccine has a different range of permissible exposure constraints, shown in Fig. 3.1, therefore requiring different time spans for the handover. For example, the green vaccine needs to be handed over within 1 to 4 seconds.

In each study session, there were two types of rounds for each controller: design and test. Users tuned the controller parameters in the "design" round

Vaccine	Minimum Exposure Time (s)	Maximum Exposure Time (s)
G	1	4
Y	2	5
R	3	6

Figure 3.1: Graphical Interface showing the time constraints for each vaccine

and performed the vaccine packaging task in the "test" round. In the design round, users could tune the controller parameters by performing handovers with different vaccine vials at the different packaging stations and observing the handover time. Once the design round was over, the controller parameters were locked-in for the subsequent test round. In the test round, users needed to place a set of vaccine vials, handed over by the robot, in three different packaging stations following a predefined sequence. In each study session, there were two types of test rounds: non-optimization and optimization. In the design round before the non-optimization test round, users tuned the controller parameters so that the vials are handed over within the timing constraints, but not necessarily in the shortest possible time. In the design round before the optimization test round, they re-tuned those controller parameters to perform the vaccine packaging task in the shortest time.

3.3.3 Monetary Incentive Scheme

For more external validity, we designed a monetary reward system to incentivize participants in the optimization test round: If they performed the task faster than a specified time limit, participants would get $1 per second less than the specified time limit. For each second spent by the user in the unsafe refriger-

ation zone, they were penalized by $1. Each failed handover incurred a penalty of $2.

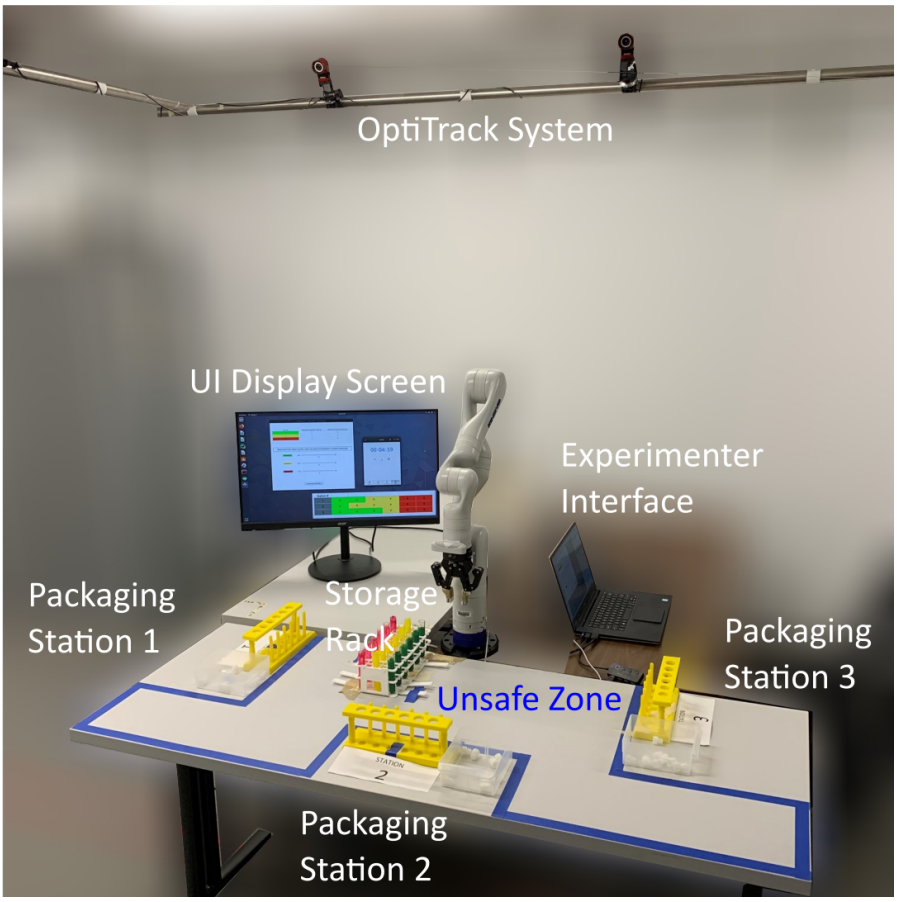


Figure 3.2: The experiment setup consisted of a stack of vaccine vials, three packaging stations, a Kinova Gen3 robot arm, a screen to display UIs, and an OptiTrack system. The participants were penalized if they entered the area inside the blue tape *i.e.* the unsafe zone.

3.3.4 Robot and User Interface

We implemented the MCE, MCJ, and PV controllers in the Python programming language, and programmed a Kinova Gen 3 robot using the Robot Operating System (ROS) to perform the vaccine packaging task autonomously with a human. We used an OptiTrack system to track the human hand and the robot's

gripper. The robot released the vial if the human's hand was within a certain distance from the robot's gripper. We developed graphical interfaces in Qt for each controller. The timing constraints and the controller parameters of each vaccine are shown in Fig. 3.3 (PV controller), Fig. 3.4 (MCJ controller), and Fig. 3.5 (MCE controller). During the design rounds, the UI display screen showed the controller interface, current handover timer, design round countdown timer, packaging sequence, and time constraints for each vaccine. In the test rounds, only the controller interface, vaccine packaging sequence, and time constraints for each vaccine were shown.

3.3.5 Human Motion Model

The MCE and MCJ controllers can utilize a prediction model of the human hand's motion in the reach phase of a handover. In our pilot trials, we observed that the users reached the handover location much faster than the robot, and waited for the robot at the handover location. Therefore in our studies, we used a static model of the human hand's position $\mathbf{h}$ over the prediction horizon L, given by,

$$\mathbf{h}_{t+i} = \mathbf{h}_t \forall i \in [0, L-1] \tag{3.7}$$

The human hand's position is updated at each time step using the observations from the OptiTrack system.

3.3.6 Experiment Procedure

Participants signed a consent form, read study instructions, and then performed an unrecorded practice session consisting of a design and a test round with a dummy controller to get familiarized with the experiment setup. The main part of the experiment consisted of two pairs of design and test rounds, one for each controller. The first test round was the non-optimization round and the second test round was the optimization round, as described in Section 3.3.2 and Section 3.3.3. The participants did not know about the monetary incentive a-priori and an experimenter explained the reward system only after the participants finished the non-optimization rounds with each of the two controllers. The order of controllers was randomized and counterbalanced to avoid recency effects.

In each design round, the participants were given 5 minutes to tune the robot's controller parameters. Participants specified the controller parameters verbally to the experimenter, who activated the the UIs shown in Fig. 3.3 (PV), Fig. 3.4 (MCJ), and Fig. 3.5 (MCE). Participants would see the UI screen from their workspace position, as shown in Fig. 3.2. This choice was made to save participants the time to move back and forth between the workspace area and the keyboard and mouse area of the UI computer. Participants could test their specified parameters by verbally requesting the robot to hand over a vaccine vial, and updating the parameters after observing the handover time for that vaccine. They could do so as many times as they wanted within the 5-minute design round. The participants' verbal requests were relayed to the robot by the experimenter, to avoid speech processing errors.

Each design round was followed by a test round in which the participants

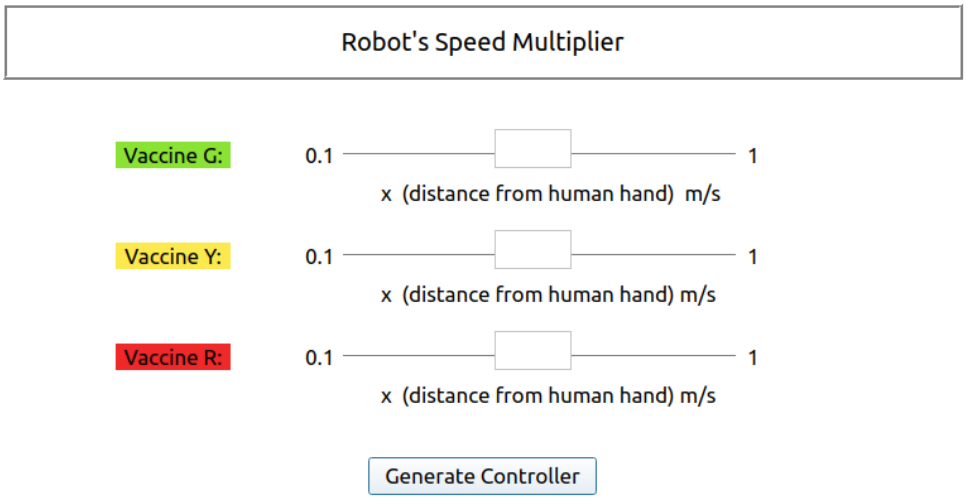


Figure 3.3: Graphical Interface for the PV controller

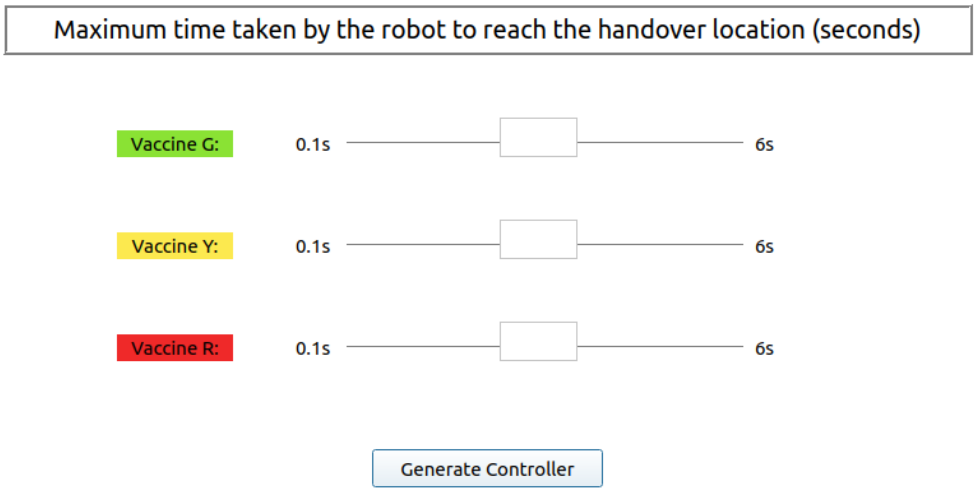


Figure 3.4: Graphical Interface for the MCJ controller

had to perform 18 sequential handovers with the robot, which used the final controller parameters from the design round. The robot autonomously handed over the vials, one by one, in the following order: six green vaccines, six yellow vaccines, and six red vaccines. The participants had to receive the vials and place them in a pre-defined packaging sequence shown in Fig. 3.6. The participants were asked to move to the next packaging station before receiving the vaccine vial. This rule forced participants to perform handovers at different

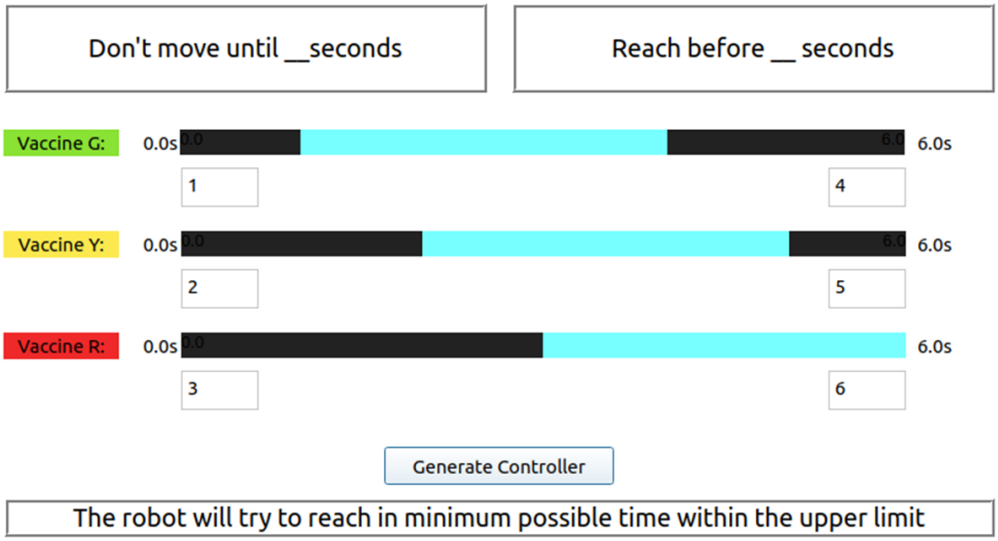


Figure 3.5: Graphical Interface for the MCE controller

packaging stations, and the pre-defined packaging sequence ensured that the three vaccine types were handed over at each of the three stations.

Station #						
1	G	G	Y	Y	R	R
2	G	Y	Y	Y	R	R
3	G	G	G	Y	R	R

Figure 3.6: Sequence of vaccines to be placed at each packaging station

After finishing each of the non-optimization and optimization test rounds with both controllers, participants reported their experience of working with each controller. We used Laugwitz et al.'s [88] user experience questionnaire to elicit the efficiency, perspicuity, and dependability ratings of each of the controllers. In Study 1, we also collected responses to these measures of workload: mental demand, performance, effort, and frustration, using the NASA TLX questionnaire, but in Study 2 we did not measure these variables (see: Section 3.3.7). Additionally, at the end of the experiment participants were asked to provide a written response to the prompt: "Please write a few sentences com-

paring the two controllers based on your experience of designing and using them in this experimental task".

We recorded the duration of the test rounds, the robot's and the human's idle time during the test rounds, and the number of handover failures in the test rounds. In the first study, failures include handovers with timing constraint violations, and—for the timing controller—also cases in which the controller could not find a feasible solution. In the second study, we also counted instances of vials falling from users' hand during the handover as failures (see: Section 3.3.7).

Table 3.1: Summary of Quantitative Results of Study 1

	Non-optimization Round		Optimization Round	
Variable	**MCJ**	**PV**	**MCJ**	**PV**
Efficiency	4.72(1.05)	4.64(1.09)	4.73(0.99)	5.08(1.01)
Perspicuity	4.79(1.34)	4.82(1.42)	4.39(1.37)	4.83(1.45)
Dependability	4.92(1.01)	4.97(1.24)	4.58(1.21)	5.22(1.00)
Overall UE	4.81(0.98)	4.81(1.16)	4.56(1.02)	5.04(1.01)
Failures	2.77(2.67)	1.87(1.83)	2.43(2.57)	1.43(1.48)
Task Duration	286.07(15.00)	250.00(6.48)	281.08(9.96)	245.52(5.25)
HIT	198.29(21.38)	163.71(20.32)	200.85(22.13)	168.88(17.25)
RIT	165.21(12.58)	142.56(7.70)	162.67(10.77)	138.08(5.77)

HIT: Human Idle Time, RIT: Robot Idle Time

3.3.7 Design Differences Between Study 1 and Study 2

The experiment design of both of our studies was similar except for the following differences:.

- In Study 1, we asked participants to report their experience of "utilizing this controller for performing this task". We realized that this ques-

Table 3.2: Summary of Quantitative Results of Study 2

Variable	Non-optimization Round MCE	Non-optimization Round PV	Optimization Round MCE	Optimization Round PV
Efficiency (P)	5.38(1.21)	4.53(1.17)	5.06(1.31)	4.65(1.37)
Efficiency (C)	5.45(1.15)	4.52(1.14)	5.23(1.37)	4.70(1.44)
Perspicuity (P)	5.45(1.37)	4.89(1.44)	5.06(1.97)	4.60(1.48)
Perspicuity (C)	5.95(1.08)	5.11(1.23)	5.45(1.72)	4.49(1.54)
Dependability (P)	5.56(1.25)	4.79(1.27)	5.08(1.58)	4.48(1.26)
Dependability (C)	5.77(1.11)	4.73(1.04)	5.42(1.31)	4.43(1.42)
Overall UE (P)	5.46(1.21)	4.74(1.19)	5.02(1.71)	4.57(1.31)
Overall UE (C)	5.72(1.01)	4.78(1.03)	5.42(1.30)	4.43(1.60)
Failures	1.93(3.93)	2.97(2.82)	1.63(3.22)	3.10(3.06)
Task Duration	268.25(28.25)	246.42(10.87)	259.63(12.08)	240.08(10.04)
HIT	178.28(34.96)	156.44(21.42)	180.80(19.51)	159.34(24.63)
RIT	166.40(34.36)	151.83(15.60)	159.81(17.52)	148.91(14.35)

P: Programming, C: Collaborating, HIT: Human Idle Time, RIT: Robot Idle Time

tion combines two different user experiences: one of specifying the controller parameters and one of physically performing the handovers with the robot. To measure these two variables separately, in Study 2, we asked participants to report their experience of "programming the robot (setting the parameters) with this controller", and additionally, their experience of "physically collaborating with the robot (handling the vaccines) with this controller".

- In Study 1, we collected responses to the NASA TLX questionnaire to measure user workload. We found that these scores were correlated with the user experience (UE) scores. The Pearson's correlation scores are provided in Appendix A. Therefore, in Study 2 we did not use the NASA TLX questionnaire.
- In Study 1, the feedback from the MCJ timing controller was "Couldn't reach your hand in time" and the experimenter verbally told participants that they need to increase the reach time. In Study 2, we changed the feed-

back from the MCE controller to a more comprehensible prompt: "I will not be able to reach in the specified time limit", and to maintain uniformity across participants, the experimenter did not provide any additional prompts.

- In Study 1, we noticed some instances of vials falling from the participants' hands during the handover, but we could not record these instances. In Study 2, we decided to record these instances since such falls are a mode of failure of the robot's handover controller.
- In the optimization test rounds of Study 1, the time limit for earning the monetary rewards described in Section 3.3.3 was: 279.19 seconds (SD=3.96) for the MCJ controller and 245.16 seconds (SD=3.34) for the PV controller. In Study 2, this time limit was: 260 seconds (SD=0.0) for the MCE controller and 245 seconds (SD=0.0) for the PV controller, because the MCE controller resulted in faster handovers, on average, as compared to the MCJ controller.

3.3.8 Participants

31 participants (12 engineering, 19 non-engineering) participated in Study 1, and a different set of 30 participants (2 engineering, 28 non-engineering) participated in Study 2. We did not use the same participants in both studies to avoid learning effects. In Study 1, there was an error in the data logging system for one of the participants, and we do not include this participant in the analysis of task duration and handover failures. Additionally, another participant did not fill out one of the user experience questionnaires, and we do not include this participant in the analysis of user experience. Thus we have $n = 30$ ob-

servations for all analyses. Each study session took approximately 90 minutes, and the participants were compensated with a minimum of $15, and additional rewards based on performance in the optimization round. The average payout was $18.42 for Study 1 and $19.57 for Study 2.

3.4 Results

3.4.1 Quantitative Results

Tables 3.1 and 3.2 show the summary of quantitative results of both studies, where the mean and standard deviation of the dependent variables in the non-optimization and optimization rounds are shown. To test the hypotheses **H1**–**H5**, we conducted paired sample Student's t-tests on the measured differences. In case of significant deviations from normality, checked with the Shapiro-Wilk Normality test, we used the Wilcoxon's signed-rank test. The results of these tests are shown in Tables 3.3,3.4, 3.5, 3.6.

Table 3.3: Results of Paired Sample Tests for Hypotheses **H1**–**H5** for the non-optimization round of Study 1 (MCJ vs PV)

Hypothesis	**Variable**	**Test**	**Statistic**	p-**Value**	**Effect Size**
H1a	Efficiency	S	-0.256	0.4	-0.047
H1b	Perspicuity	S	0.065	0.526	0.012
H1c	Dependability	S	0.170	0.567	0.031
H1	Overall UE	S	0.0	0.5	0.000
H2	Failures	S	-1.451	0.921	-0.265
H3	Task Duration	W	0.0	1.000	-1.000
H4	Human Idle Time	W	2.0	1.000	-0.991
H5	Robot Idle Time	S	-11.187	1.000	-2.042

S: Student's t-test, W: Wilcoxon's signed-rank test, (P): Programming, (C): Collaborating

$^{*}\ p < 0.05$, $^{**}\ p < 0.005$, $^{***}\ p < 0.001$

Table 3.4: Results of Paired Sample Tests for Hypotheses **H1–H5** for the optimization round of Study 1 (MCJ vs PV)

Hypothesis	Variable	Test	Statistic	p-Value	Effect Size
H1a	Efficiency	S	1.696	0.950	0.310
H1b	Perspicuity	S	1.144	0.869	0.209
H1c	Dependability	S	2.402	0.989	0.439
H1	Overall UE	S	1.848	0.963	0.337
H2	Failures	S	−1.912	0.967	−0.349
H3	Task Duration	S	−18.260	1.000	−3.334
H4	Human Idle Time	S	−15.455	1.000	−2.822
H5	Robot Idle Time	S	−12.236	1.000	−2.234

S: Student's t-test, W: Wilcoxon's signed-rank test, (P): Programming, (C): Collaborating
$^{*}\ p < 0.05$, $^{**}\ p < 0.005$, $^{***}\ p < 0.001$

Table 3.5: Results of Paired Sample Tests for Hypotheses **H1–H5** for the non-optimization round of Study 2 (MCE vs PV)

Hypothesis	Variable	Test	Statistic	p-Value	Effect Size
H1a	Efficiency (P)	S	−2.658	0.006^{**}	−0.485
H1a	Efficiency (C)	S	−3.102	0.002^{**}	−0.566
H1b	Perspicuity (P)	S	−1.556	0.065	−0.284
H1b	Perspicuity (C)	S	−2.844	0.004^{**}	−0.519
H1c	Dependability (P)	S	−2.776	0.005^{*}	−0.507
H1c	Dependability (C)	S	−3.679	$< 0.001^{***}$	−0.672
H1	Overall UE (P)	S	−2.426	0.011^{*}	−0.443
H1	Overall UE (C)	S	−3.547	$< 0.001^{***}$	−0.648
H2	Failures	W	278.5	0.016^{*}	0.474
H3	Task Duration	W	13.0	1.000	−0.944
H4	Human Idle Time	W	38.0	1.000	−0.837
H5	Robot Idle Time	W	46.0	1.000	−0.789

S: Student's t-test, W: Wilcoxon's signed-rank test, (P): Programming, (C): Collaborating
$^{*}\ p < 0.05$, $^{**}\ p < 0.005$, $^{***}\ p < 0.001$

Table 3.6: Results of Paired Sample Tests for Hypotheses **H1–H5** for the optimization round of Study 2 (MCE vs PV)

Hypothesis	Variable	Test	Statistic	p-Value	Effect Size
H1a	Efficiency (P)	S	-1.307	0.101	-0.239
H1a	Efficiency (C)	S	-1.859	0.037^{*}	-0.339
H1b	Perspicuity (P)	S	-1.002	0.162	-0.183
H1b	Perspicuity (C)	S	-3.245	0.001^{**}	-0.592
H1c	Dependability (P)	S	-1.635	0.056	-0.298
H1c	Dependability (C)	S	-3.452	0.001^{***}	-0.630
H1	Overall UE (P)	S	-1.197	0.120	-0.219
H1	Overall UE (C)	S	-3.307	0.001^{**}	-0.604
H2	Failures	S	2.068	0.024^{*}	0.378
H3	Task Duration	W	6.0	1.000	-0.974
H4	Human Idle Time	S	-6.291	1.000	-1.149
H5	Robot Idle Time	S	-3.053	0.998	-0.557

S: Student's t-test, W: Wilcoxon's signed-rank test, (P): Programming, (C): Collaborating
$^{*}\ p < 0.05$, $^{**}\ p < 0.005$, $^{***}\ p < 0.001$

3.4.2 Qualitative Feedback

At the end of the study session, we had asked participants to provide qualitative feedback, with the prompt "Please write a few sentences comparing the two controllers based on your experience of designing and using them in this experimental task". Two coders independently analyzed the comments to find which controller each participant preferred in terms of efficiency, perspicuity, dependability, and handover quality. The inter-coder reliability was 87.1% in Study 1 and 84.2% in Study 2.

The majority of the participants who wrote comments related to perspicuity preferred the perspicuity of the timing controllers over that of the baseline PV controller: 18 out of 28 in Study 1 and 17 out of 26 in Study 2.

> Study 1, P22: *"The speed multiplier [PV] controller was a little more difficult to learn because I did not have a great idea of what the base speed*

was or what speed a number would produce until after practicing with the controller."

Study 2, P28: *"Personally, I like the time [MCE] controller compared to the second one [PV] because the multiplier [PV] is kind of confusing. Although there is pattern and relation between multiplier and speed of the robot reaching my hand, it takes me some time to think about what it means and how it is related to the robot's speed."*

Study 2, P04: *"[PV] was much easier to use since it was more straight forward and I only had to worry about one parameter instead of 2."*

The majority of the participants who wrote comments related to the dependability of the controllers felt that the PV controller was more dependable than the MCJ timing controller (10 out of 12 participants), but the MCE timing controller was more dependable than the PV controller (4 out of 5 participants).

Study 1, P26: *"With the time controller [MCJ], I faced a problem of the controller not being able to meet the set time which felt like it needed more tuning of parameters than it seems on paper."*

Study 2, P09: *"The time parameter [MCE] was better because I felt the time could be more predictable rather than depending on the placement of your hand."*

Study 2, P10: *"With the time [MCE], I could select a concrete amount of time and expect the robot arm not to move until that set amount of time had passed."*

Four participants commented on the efficiency of the MCJ timing controller vs PV controller, and all of them preferred the latter, and five participants out of the seven that commented on the efficiency of the MCE timing controller vs PV controller preferred the latter.

> Study 1, P01: *"Since the three stations were at different distances from the robot, I felt that the multiplier [PV] worked better."*

> Study 2, P18: *"But, the multiplier [PV] may be more efficient since you can change the speeds to deliver the vials faster."*

3.5 Discussion

We developed handover controllers that allow users to specify a robot's behavior in terms of intuitive timing parameters and provide failure feedback. Our hypothesis was that—compared to a baseline PV controller—the timing controllers would result in a better user experience and lower failure rates, while not negatively affecting the handover timing efficiency of the robot.

Our results suggest that "the devil is in the details." Two instances of a timing-with-feedback controller can result in very different, and even opposite, user experience and task performance. The instances developed for this study differ in terms of the motion strategies of the robot, in turn, encoded as objective functions and constraints of the MPC problem.

Our findings indicate that the MCJ strategy, which we initially thought would be preferred by users, results in a lower user experience compared to

a PV controller, more prominently when the users have to optimize the controller parameters for faster handovers, whereas the MCE strategy results in a higher user experience compared to the baseline. One possible cause for this is that the robot's reach time when using the MCJ controller is more sensitive to target locations due to a gradual starting motion. This leads to a higher number of instances of infeasible solutions to the MPC problem for MCJ (mean: 2.03, SD: 1.86) than MCE (mean: 1.20, SD: 2.11). Another possible cause is that the MCJ controller produces slower robot motions than the MCE and PV controllers. Finally, the prompts of the controller parameters and the feedback could have also affected the user experience of designing the controllers. The MCJ controller's UI had a more descriptive prompt about the controller parameter ("Maximum time taken by the robot to reach the handover location") while the MCE controller had command-like prompts for the controller parameters ("Don't move until" and "Reach before").

We found that the MCE controller resulted in lower vial falls from the human hand during the handover than the PV controller[2]. This could be attributed to the fact that the PV controller slows down the robot as it approaches the human hand, and sometimes stops short of the human hand due to a steady state error. Thus, users find it difficult to gauge exactly when the object will be released by the gripper. In contrast, the MCE controller often overshoots the target position and thus thrust the object into the human hand, which may be preferred by users. In Study 2, P14 commented, for example: *"I like how it got really close to my hand and basically put the tubes into my hand."*

Both instances of our timing controller resulted in higher test-round task duration, robot idle time, and human idle time compared to the baseline con-

[2]MCE mean falls: 0.07, PV mean falls: 0.4, $W = 105.5,\ p = 0.005,\ r = 0.758$

troller. One possible cause is that users were more conservative in setting the parameters for the timing controller. For example, in the case of the MCE controller even though the optimal stall time for the green vaccine was 0 seconds, 13 participants set a non-zero value in the optimization round. Second, there is an inherent sub-optimality with both instances of the timing controller for our setup shown in Fig. 3.2. A reach time that is optimal for handovers near packaging station 1 might be infeasible for handovers near packaging station 3 because of the longer travel distance from the storage rack. Thus, the user has to select a higher reach time to have successful handovers at all the packaging stations. Additionally, the MPC controller has a higher sample time increasing its response time by approximately 0.2 seconds per handover.

One surprising result was that the participants rated the MCE controller to be more efficient for physically collaborating with the robot, compared to the PV controller, even though the MCE controller resulted in higher task duration than the PV controller. Based on the qualitative feedback of some participants this could be attributed to the faster robot motion produced by the MCE controller. For example, in Study 2, P30 commented: *"I also found it [PV] the most annoying because it felt like it was moving so slowly. I actually like the second one [MCE] better because it feels like you are going faster."*, and P28 commented: *"I actually like the second one [MCE] better because it feels like you are going faster, I know it's not necessarily [so], but it feels that way."*

A few participants had more than 50% failures in the test rounds. In Study 1, one participant had 11 failures with the MCJ controller in the first test round. In Study 2, three participants had more than 10 failures with PV, and four participants had more than 10 failures with the MCE controller. In their qualitative

feedback, some participants attributed these failures to their lack of experience with robots. Study 2, P9: *"I was a little slow at understanding how the concept applied to the programming. I am not familiar with robotics or programming them. Once I got the idea, it made logical sense. But it took me a while to get there."*

3.6 Conclusion and Future Work

We hope that our timing-with-feedback controllers will be useful in scenarios where end-users need to design a robot's handover behavior. We investigated one such task in this work, and our studies revealed the advantages and disadvantages of our proposed controllers vis-à-vis a baseline controller. Our work can motivate further research toward the design of better user-defined controllers for human-robot handovers. This includes investigating other motion strategies and constraints, designing corrective feedback, and testing in real industrial settings.

CHAPTER 4

SPECIFYING AND SYNTHESIZING HUMAN-ROBOT HANDOVERS

4.1 Introduction

We present a controller for human-robot handovers that is automatically synthesized from high-level specifications in Signal Temporal Logic (STL). In contrast to existing controllers, this approach can provide formal guarantees on the timing of each of the handover phases. Using synthesis also allows end-users to specify and dynamically change the robot's behaviors using high-level requirements of goals and constraints rather than by tuning low-level controller parameters. We illustrate the proposed approach by replicating the behavior of existing handover strategies from the literature. We also identify specification parameters that are likely to lead to successful handovers using a public database of human-human handovers.

As described in Section 2.5.2, the offline controllers of human-robot handovers plan the robot's motion before the start of a handover. These approaches do not take into account the observed behavior of the human during a handover, and thus, lack adaptability to the human's behavior. The online controllers proposed in the literature take into account the observed human motion. Though these approaches enable the robot to adapt to the human's motion, they require tuning non-intuitive controller parameters (e.g., the weights of DMP terms [133] or the velocity-tracking gain[112]) to achieve a specific robot behavior. Also,

A substantial portion of this chapter is taken from [80] ©2019 IEEE. Reprinted, with permission, from A. Kshirsagar, H. Kress-Gazit and G. Hoffman, "Specifying and Synthesizing Human-Robot Handovers", IEEE/RSJ International Conference on Intelligent Systems and Robots (IROS), Macau, 4-8 November 2019

none of these provide timing guarantees on different stages of a handover. Such timing guarantees may be crucial in productivity-oriented industrial tasks and fast-paced life-critical scenarios like surgery. Finally, although several human-inspired strategies have been proposed in the literature for human-robot handovers, there is no unified framework to easily switch between those strategies. Our proposed approach, shown in Fig. 4.1, tries to address these limitations.

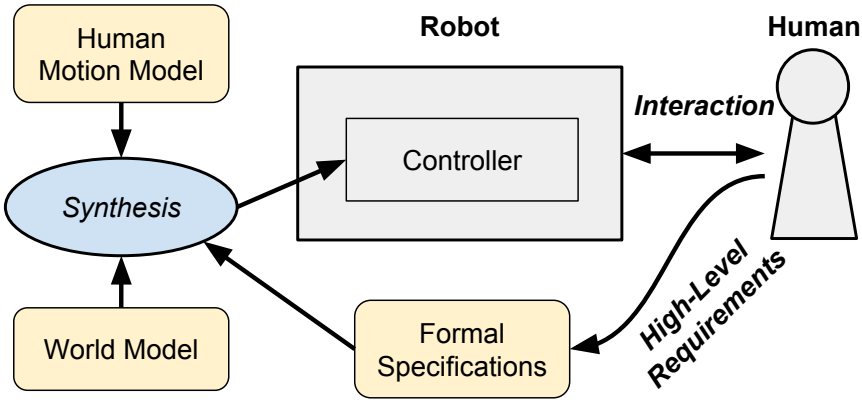


Figure 4.1: Our approach to human-robot handovers uses the automatic synthesis of a robot controller from formal specifications written in Signal Temporal Logic. Users can change the robot's behavior with high-level requirements of goals and intuitive parameters such as the timing of different stages of handovers. The controller is then synthesized online based on a given human-motion prediction model and a world dynamics model.

We exemplify the possibility of automatically synthesizing handovers by specifying the robot's handover behavior using Signal Temporal Logic (STL) [104] formulae. STL is well-suited to the handover domain as it enables reasoning about timings and distances. Using STL, the user can change the robot's behavior by specifying intuitive parameters like the timing of different phases of handovers and safety constraints. We present specification templates both for human-to-robot and robot-to-human handovers and illustrate the flexibility of our approach by reproducing existing human-robot handover strategies found in the literature. We then use the same specifications in conjunction

with a human-human handover dataset to identify parameters that can lead to successful human-robot handovers.

Automatic synthesis of robot controllers outside of HRI has been gaining interest in the past several years [78] and has been demonstrated on a variety of platforms including mobile robots, UAVs, autonomous cars, modular robots, mobile manipulators, swarm robots, and humanoids. Still, very little work has been done regarding synthesis for HRI. In the area of formal methods, Araiza-Illan et al. [5] and Webster et al. [176] applied probabilistic model-checking in a human-robot handover task, verifying a controller with respect to safety and liveness specifications. But these works consider a given controller and do not deal with the synthesis of one. Li et al. [97] presented a formalism for human-in-the-loop control synthesis and synthesized a semi-autonomous controller from temporal logic specifications, but the HRI aspects were limited to human intervention and correction of the robot's operation, and not the actual handover interaction. To the best of our knowledge, there is no prior work on synthesis for human-robot handovers. We try to address this gap in this work.

4.2 Signal Temporal Logic for Robot Control

Automatic synthesis of robot control [78] has been demonstrated using several formalisms, including discrete logic such as Linear Temporal Logic (LTL), probabilistic logic such as Probabilistic Computation Tree Logic (PCTL), and metric logic such as Signal Temporal Logic (STL). We choose STL for human-robot handovers, as it allows specifications that include distances, timing, and object states.

4.2.1 Signal Temporal Logic

STL formulae are composed of predicates $\pi^\gamma : \mathbb{R}^n \to \mathbb{B}$ over continuous signals. The truth value of a predicate π^γ is determined by the sign of a function $\gamma : \mathbb{R}^n \to \mathbb{R}$ of an underlying signal $\mathbf{s}$. The predicates can be combined with Boolean operators ["NOT" ($\neg$), "AND" ($\wedge$), "OR" ($\vee$)], and temporal operators ["Until" ($\mathcal{U}$), "Always" ($\Box$), and "Eventually" ($\Diamond$)]. STL formulae are defined recursively according to the following grammar:

$$\varphi ::= \pi^\gamma \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Box_{[a,b]}\varphi \mid \varphi_1 \mathcal{U}_{[a,b]} \varphi_2 \mid \Diamond_{[a,b]}\varphi$$

where $a, b \in \mathbb{R}_{\geq 0}$. The operators $\Box$ ("Always"), $\Diamond$ ("Eventually") and $\mathcal{U}$ ("Until") can also be unbounded.

The satisfaction of an STL formula φ with respect to the signal $\mathbf{s}$ at time t is defined inductively as follows:

$$\begin{aligned}
(\mathbf{s}, t) \models \pi^\gamma &\iff \gamma(\mathbf{s}(t)) > 0 \\
(\mathbf{s}, t) \models \neg\pi^\gamma &\iff \neg((\mathbf{s}, t) \models \pi^\gamma) \\
(\mathbf{s}, t) \models \varphi_1 \wedge \varphi_2 &\iff (\mathbf{s}, t) \models \varphi_1 \wedge (\mathbf{s}, t) \models \varphi_2 \\
(\mathbf{s}, t) \models \varphi_1 \vee \varphi_2 &\iff (\mathbf{s}, t) \models \varphi_1 \vee (\mathbf{s}, t) \models \varphi_2 \\
(\mathbf{s}, t) \models \Box_{[a,b]}\varphi &\iff \forall t' \in [t+a, t+b], (\mathbf{s}, t') \models \varphi \\
(\mathbf{s}, t) \models \varphi_1 \mathcal{U}_{[a,b]} \varphi_2 &\iff \exists t' \in [t+a, t+b] \text{ s.t. } (\mathbf{s}, t') \models \varphi_2 \\
&\qquad \wedge \forall t'' \in [t, t'], (\mathbf{s}, t'') \models \varphi_1 \\
(\mathbf{s}, t) \models \Diamond_{[a,b]}\varphi &\iff \exists t' \in [t+a, t+b] \text{ s.t. } (\mathbf{s}, t') \models \varphi
\end{aligned}$$

A signal $\mathbf{s} = s_0 s_1 s_2 \ldots$ satisfies φ, denoted by $\mathbf{s} \models \varphi$, if $(\mathbf{s}, 0) \models \varphi$. Informally, $\mathbf{s} \models \Box_{[a,b]}\varphi$ if φ holds at each time between a and b. $\mathbf{s} \models \varphi_1 \mathcal{U}_{[a,b]} \varphi_2$ if φ_2 holds at some time c between a and b, and φ_1 holds at each time between a and c. $\mathbf{s} \models \Diamond_{[a,b]}\varphi$ if φ holds at some time between a and b.

4.2.2 Receding Horizon Control Synthesis from STL

For controller synthesis, STL specifications can be automatically transformed into mixed integer linear programs (MILP) and solved either globally for a sequence of control actions or iteratively in a receding horizon manner [139] [140]. In this work, we use the receding horizon control (RHC) synthesis algorithm presented by Raman et al. [139]: Given a system of the form $x_{t+1} = f(x_t, u_t)$, initial state x_0, STL formula φ, cost function J and finite horizon H, the following optimization problem is solved at each time-step t:

$$\begin{aligned} \text{argmin}_{\mathbf{u}^{H,t}} \quad & J(\mathbf{x}(x_t, \mathbf{u}^{H,t}), \mathbf{u}^{H,t}) \\ \text{s.t.} \quad & \mathbf{x}(x_0, \mathbf{u}_0) \models \varphi \end{aligned} \tag{4.1}$$

where $\mathbf{u}^{H,t} = u_0^t, u_1^t, u_2^t ... u_{H-1}^t$ is the horizon-H control input sequence computed at each time step and $\mathbf{x}(x_t, \mathbf{u}^{H,t}) = x_t, x_{t+1}, x_{t+2} ... x_{t+H-1}$ is the sequence of system states over the horizon H. Though the control input is computed over the horizon H, only the first control is applied at each time-step and the horizon is calculated again. Thus in Eq. 4.1, $\mathbf{u}_0 = u_0^0, u_0^1, u_0^2, ...$ is the applied control sequence consisting of only the first control inputs of $\mathbf{u}^{H,t}$ with $\mathbf{x}(x_0, \mathbf{u}_0) = x_0, x_1, x_2 ...$ being the sequence of system states as a result of the input $\mathbf{u_0}$.

4.3 Formulation of Handovers in Signal Temporal Logic (STL)

To specify human-robot handovers in STL, we first define the following variables. Bold letters denote vectors.

Pose:

- $\mathbf{p_i} = [x_i, y_i, z_i]$ and $\mathbf{q_i} = [q_{x,i}, q_{y,i}, q_{z,i}, q_{w,i}]$ represent the position and orientation vectors in 3D space, jointly called 'pose'. Subscript $i \in \{r, h, o, l, d, \eta\}$ refers to: r = robot, h = human, o = object, l = handover location, d = object drop-off destination, η = robot home pose.
- $\mathbf{q}_\delta$ is the offset between the robot end-effector's orientation and the human hand's orientation, when they both hold the object.
- ϵ_p and ϵ_q are the permitted tolerances (errors) in the robot end-effector's position and orientation from any desired position and orientation.

Gripper and Hand:

- g_r and $g_h \in [0, 1]$ are the robot gripper and human hand "openness" (0 for fully closed, 1 for fully open).
- $g_* \in [0, 1]$ is the grip-width of the object.
- ϵ_g is the permitted tolerance (error) between a gripper/hand openness from a desired level of openness.

Force Sensing:

- $\alpha_r \in [0, 1]$ is the proportion of the weight (or the "load-share") of the object supported by the robot.
- δ_1, δ_2 are thresholds on the load share specifying holding and releasing states.

Other:

- l_h is the radius of a "handover zone" centered around the robot's base.

- e is a boolean variable used to specify that the handover has started.
- o_s, o_r, o_h, o_g are discrete object states (defined below).

Assumptions: We represent all the poses in the frame attached to the base of the robot, as shown in Fig. 4.2. We assume that the human hand always remains in the dextrous workspace of the robot. We consider that the human is ready for the handover if the human hand is within a region of radius l_h centered at the robot's base, which we call the *handover zone*. For human-to-robot handovers, we assume that the human hand contains the object at the start of the handover, and also assume that the object's drop-off destination location is in the dextrous workspace of the robot. For robot-to-human handovers, we assume that the object is initially in the dextrous workspace of the robot. For simplicity, we consider that there is only one object in the workspace, but the formulation can be easily extended to multiple objects.

4.3.1 System Representation

We model the robot's end-effector's motion as a linear system with its pose and gripper openness making up the system state $\mathbf{x}$. The end-effector velocity and the gripper's opening velocity are the control input $\mathbf{u}$, constrained by $\mathbf{u_{min}} \leq \mathbf{u} \leq \mathbf{u_{max}}$.

$$\dot{\mathbf{x}} = \left[\dot{\mathbf{p}}_\mathbf{r}, \dot{\mathbf{q}}_\mathbf{r}, \dot{g}_r\right]^T = \mathbf{u} \tag{4.2}$$

We choose velocity control as it enables control over the timing of the pose

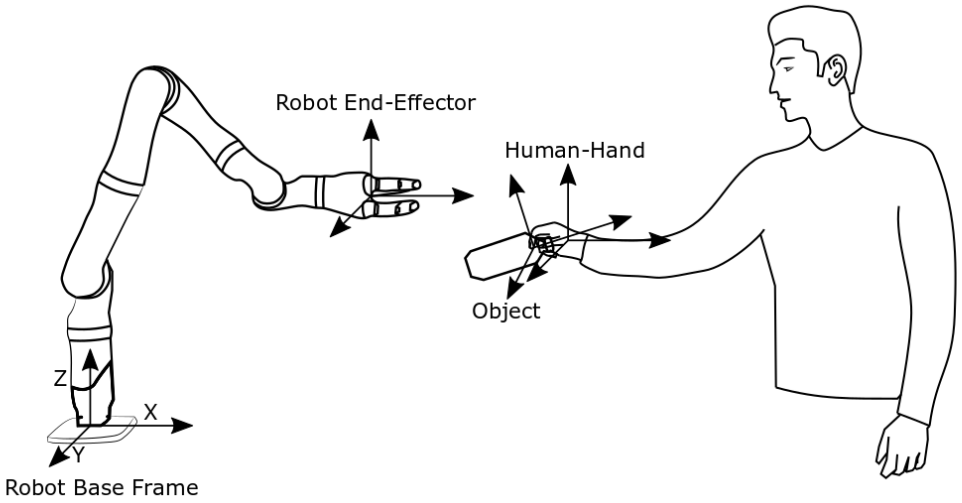


Figure 4.2: Human-Robot object handover reference frames. All poses are expressed in the frame attached to the base of the robot.

trajectory. Also, velocity control is the most suitable choice for reactive trajectory modification and online motion planning [184]. We represent the world state $\mathbf{w}$ as the pose of the human's hand, the human's hand's openness, and the pose of the object.

$$\mathbf{w} = \left[\mathbf{p_h}, \mathbf{q_h}, g_h, \mathbf{p_o}, \mathbf{q_o}\right]^T \tag{4.3}$$

To simplify the specification formulae, we create sets of the robot, human, and object states and represent them using the discrete variable $o \in \{o_s, o_r, o_h, o_g\}$, defined in Table 4.1. If the robot's gripper is equipped with a force sensor, the load-share estimate α_r is used to determine o, similar to Medina et al. [110]. Otherwise, we use the human's hand state g_h and the robot end-effector state g_r.

Table 4.1: Discrete object states defined for robots without and with a gripper force sensor.

State	Formula (without Force Sensor)	Formula (with Force Sensor)
o_r	$(\|\mathbf{p_o} - \mathbf{p_r}\| \leq \epsilon_p \wedge \|\|g_r - g_*\|\| \leq \epsilon_g)$	$\alpha_r > \delta_2$
o_h	$\neg(\|\mathbf{p_o} - \mathbf{p_r}\| \leq \epsilon_p \wedge \|\|g_r - g_*\|\| \leq \epsilon_g)$	$\alpha_r \leq \delta_1 \wedge \|\mathbf{p_o} - \mathbf{p_h}\| \leq \epsilon_p$
o_s	$(\|\mathbf{p_o} - \mathbf{p_r}\| \leq \epsilon_p \wedge \|\|g_r - g_*\|\| \leq \epsilon_g)$	$\delta_2 \geq \alpha_r > \delta_1$
o_g	$\neg(\|\mathbf{p_o} - \mathbf{p_r}\| \leq \epsilon_p \wedge \|\|g_r - g_*\|\| \leq \epsilon_g)$	$\alpha_r \leq \delta_1 \wedge \|\mathbf{p_o} - \mathbf{p_h}\| > \epsilon_p$

o_r (object is with the robot), o_h (object is with the human), o_s (object is shared by both), o_g (object is on the ground)

In the following sections, we deal with human-to-robot handovers and robot-to-human handovers separately.

4.3.2 Specifications for Human-to-Robot Handovers

From a receiver's perspective, a handover consists of three phases: a reach phase in which the receiver moves to the handover location, a transfer phase in which the receiver grasps the object and the giver releases the object, and a retreat phase in which the receiver moves to the object's drop-off destination location. When the robot is the receiver, we specify its behavior in terms of timing constraints on these three phases (Table 4.2 top).

- Reach: The robot should reach the handover location $[\mathbf{p_l}, \mathbf{q_l}]$ within t_1 seconds after the handover signal e.
- Transfer: The robot should grasp the object within t_2 seconds after it reaches the object's location.
- Retreat: The robot should retreat to the object's drop-off location $[\mathbf{p_d}, \mathbf{q_d}]$ within t_3 seconds after it has the object and release the object in t_4 seconds after reaching the object's drop-off location.

4.3.3 Specifications for Robot-to-Human Handovers

From a giver's perspective, a handover consists of four phases: a pick-up phase in which the giver moves to the object's location and grasps the object, a reach phase in which the giver moves the object to the handover location, a transfer

phase in which the giver releases the object and the receiver grasps the object, and a retreat phase in which the giver moves back to its original position. When the robot is the giver, we specify its behavior in terms of timing constraints on these four phases (Table 4.2 bottom).

- Pick-up: The robot should reach the object's location within t_5 seconds and grasp the object within t_6 seconds after reaching the object's location.
- Reach: The robot should take the object to the handover location $[\mathbf{p}_\mathbf{l}, \mathbf{q}_\mathbf{l}]$ within t_7 seconds after the handover signal e.
- Transfer: The robot should release the object within t_8 seconds after the object is shared by both.
- Retreat: The robot should retreat to a home position $[\mathbf{p}_\eta, \mathbf{q}_\eta]$ within t_9 seconds after the human has received the object.

4.3.4 Specifying Different Handover Strategies

To illustrate the flexibility of our approach, we list specifications for four different handover strategies which are found in the literature. These differ only in the way the reach phase is specified. For each of these strategies, the specification in Table 4.3 replaces the reach phase specification in Table 4.2.

- Proactive, Predetermined: The robot should reach a predefined or offline computed handover location $[\mathbf{p}_*, \mathbf{q}_*]$ without waiting for the human's hand to enter the handover zone. This behavior is similar to the robot's behavior presented by several researchers (see "Offline:Pre-programmed or

Table 4.2: STL Specifications for Human-Robot Handovers

Robot's Role	Phase	Specification
Receiver	Reach	$\square(e \Rightarrow \lozenge_{[0,t_1]}(\|\|\mathbf{p_l} - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q_l} - \mathbf{q_r}\|\| < \epsilon_q))$
	Transfer	$\square((\|\|\mathbf{p_o} - \mathbf{p_r}\|\| < \epsilon_p \wedge (o == o_h \vee o == o_s)) \Rightarrow \lozenge_{[0,t_2]}(\|\|g_r - g_*\|\| < \epsilon_g))$
		$\square((o == o_r) \Rightarrow \lozenge_{[0,t_3]}(\|\|\mathbf{p_r} - \mathbf{p_d}\|\| < \epsilon_p \wedge \|\|\mathbf{q_r} - \mathbf{q_d}\|\| < \epsilon_q))$
	Retreat	$\square((\|\|\mathbf{p_r} - \mathbf{p_d}\|\| < \epsilon_p \wedge \|\|\mathbf{q_r} - \mathbf{q_d}\|\| < \epsilon_q \wedge o == o_r) \Rightarrow \lozenge_{[0,t_4]}(\|\|g_r - 1\|\| < \epsilon_g))$
Giver	Pick-up	$\square((o == o_g) \Rightarrow \lozenge_{[0,t_5]}(\|\|\mathbf{p_o} - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q_o} - \mathbf{q_r}\|\| < \epsilon_q))$
		$\square((\|\|\mathbf{p_o} - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q_o} - \mathbf{q_r}\|\| < \epsilon_q \wedge o == o_g) \Rightarrow \lozenge_{[0,t_6]}(\|\|g_r - g_*\|\| < \epsilon_g))$
	Reach	$\square(e \Rightarrow \lozenge_{[0,t_7]}(\|\|\mathbf{p_l} - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q_l} - \mathbf{q_r}\|\| < \epsilon_q))$
	Transfer	$\square((o == o_s) \Rightarrow \lozenge_{[0,t_8]}(\|\|g_r - 1\|\| < \epsilon_g))$
	Retreat	$\square((o == o_h) \Rightarrow \lozenge_{[0,t_9]}(\|\|\mathbf{p_r} - \mathbf{p_\eta}\|\| < \epsilon_p \wedge \|\|\mathbf{q_r} - \mathbf{q_\eta}\|\| < \epsilon_q))$

Hand-coded Controllers" in Section 2.5.2, such as Moon et al. [115] and the "proactive" strategy presented by Huang et al. [63].

- Proactive, Towards Human: The robot should reach the human's hand without waiting for the human's hand to enter the handover zone. This behavior is similar to the controller of Medina et al. [110].
- Reactive, Predetermined: The robot should reach a predefined or offline computed handover location $[\mathbf{p}_*, \mathbf{q}_*]$, only when the human hand is in the handover zone. This is similar to the "reactive" strategy proposed by Huang et al. [63].
- Reactive, Towards Human: The robot should reach the human's hand, only when the human's hand is in the handover zone. This behavior is similar to the behaviors presented by several researchers (see "Velocity controllers towards human hand" in Section 2.5.2) such as Micelli et al. [112].

4.4 Evaluation

In this section, we present simulated runs to demonstrate the different robot behaviors synthesized from the specifications described in Section 4.3. We also show how different choices of reach-time parameters affect the expected success rate of the handover, as evaluated on a public dataset of human-human handovers. We only present the simulations of the reach phase since the other phases have the same specifications in all the presented strategies. Also, we only consider the position trajectory of the human hand and synthesize the position

Table 4.3: STL Specifications for Reach-Phase Strategies

Robot's Role	Strategy	Target	Specification
Receiver	Proactive	P	$\square(\neg(o == o_r) \Rightarrow \Diamond_{[0,t_1]}(\|\|\mathbf{p}_* - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q}_* - \mathbf{q_r}\|\| < \epsilon_q))$
		T	$\square(\neg(o == o_r) \Rightarrow \Diamond_{[0,t_1]}(\|\|\mathbf{p_h} - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q_h} - \mathbf{q_r} - \mathbf{q}_\delta\|\| < \epsilon_q))$
	Reactive	P	$\square((\neg(o == o_r) \wedge \|\|\mathbf{p_h}\|\| \leq l_h) \Rightarrow \Diamond_{[0,t_1]}(\|\|\mathbf{p}_* - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q}_* - \mathbf{q_r}\|\| < \epsilon_q))$
		T	$\square((\neg(o == o_r) \wedge \|\|\mathbf{p_h}\|\| \leq l_h) \Rightarrow \Diamond_{[0,t_1]}(\|\|\mathbf{p_h} - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q_h} - \mathbf{q_r} - \mathbf{q}_\delta\|\| < \epsilon_q))$
Giver	Proactive	P	$\square((o == o_r) \Rightarrow \Diamond_{[0,t_7]}(\|\|\mathbf{p}_* - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q}_* - \mathbf{q_r}\|\| < \epsilon_q))$
		T	$\square((o == o_r) \Rightarrow \Diamond_{[0,t_7]}(\|\|\mathbf{p_h} - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q_h} - \mathbf{q_r} - \mathbf{q}_\delta\|\| < \epsilon_q))$
	Reactive	P	$\square((o == o_r \wedge \|\|\mathbf{p_h}\|\| \leq l_h) \Rightarrow \Diamond_{[0,t_7]}(\|\|\mathbf{p}_* - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q}* - \mathbf{q_r}\|\| < \epsilon_q))$
		T	$\square((o == o_r \wedge \|\|\mathbf{p_h}\|\| \leq l_h) \Rightarrow \Diamond_{[0,t_7]}(\|\|\mathbf{p_h} - \mathbf{p_r}\|\| < \epsilon_p \wedge \|\|\mathbf{q_h} - \mathbf{q_r} - \mathbf{q}_\delta\|\| < \epsilon_q))$

T: Towards Human, P: Predetermined

trajectory of the robot, since the orientation trajectory of the robot depends on the offset $\mathbf{q}_\delta$ specific to the object being handed over.

4.4.1 Implementation

We use the MATLAB 'BluSTL' toolbox [138] to synthesize controllers from the system dynamics and specifications. This toolbox implements the RHC synthesis algorithm described in Section 4.2.2. Since this toolbox accepts only linear and affine predicates, we use the l_1 norm in the specifications. For the objective function, J in Eq. 4.1, we use

$$J(\mathbf{x}(x_t, \mathbf{u}^{H,t}), \mathbf{u}^{H,t}) = \sum_{k=0}^{H} ||\mathbf{u}_{t+k}|| \tag{4.4}$$

to suggest that the robot move with minimum mean velocity.

The receding horizon control synthesis from the STL specifications depends on the predicted behavior of the environment w over the horizon H. Specifically, the prediction horizon H has to be greater than the maximum t_i value in the specifications. At each time-step, we predict the motion of the human by a Linear Dynamical System (LDS) $\dot{\mathbf{p}}_\mathbf{h} = \mathbf{A}\mathbf{p}_\mathbf{h}$. Similar to the approach used by Medina et al. [110], we use the pose data of the human for a pre-defined time interval before the current time-step and estimate the matrix $\mathbf{A}$ using least squares approximation. Then the predicted motion of the human is given by:

$$\mathbf{p}_\mathbf{h}(t_0 + t) = \mathbf{p}_\mathbf{h}(t_0) + (t\delta_t)\mathbf{A}\mathbf{p}_\mathbf{h}(t_0) \; \forall \; t \in [0, H] \tag{4.5}$$

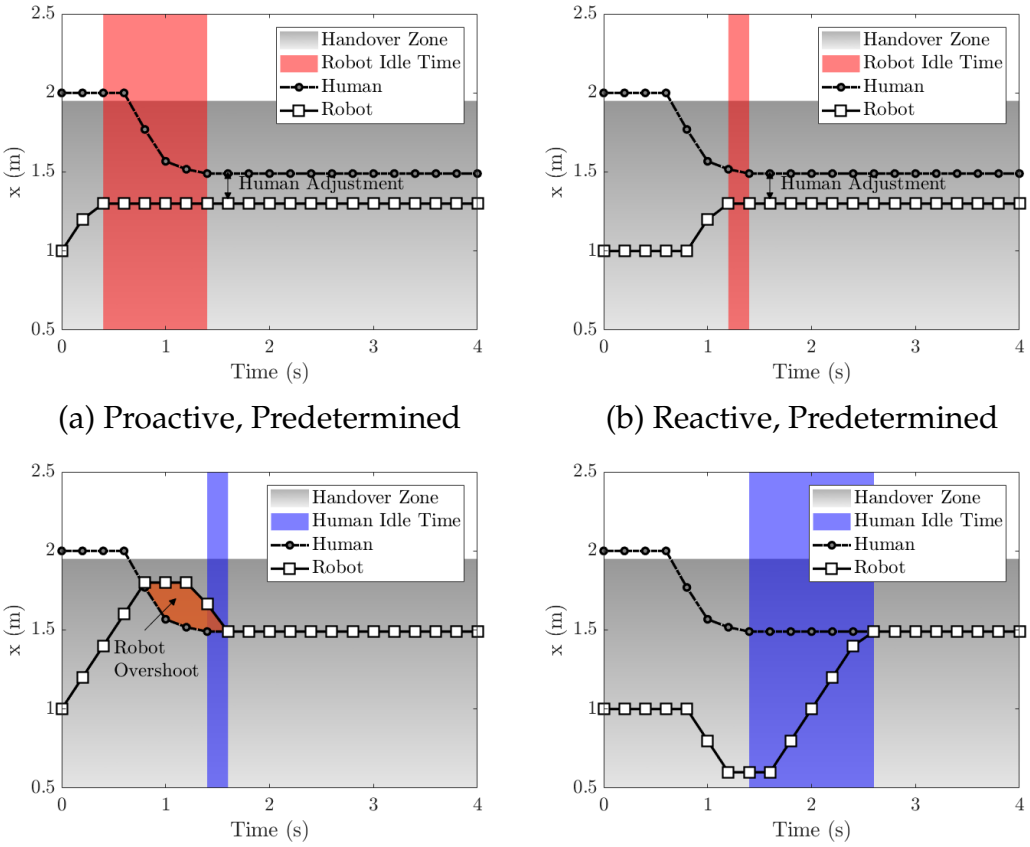
where H is the prediction horizon and δ_t is the sampling time. We update this estimate at each time-step using the position of the human hand and generate

the control input for the next time-step. If no feasible control input is found, the robot stops.

4.4.2 Simulations

Fig. 4.3 shows the simulated robot end-effector's trajectories for each of the four reach-phase strategies for the same human hand motion. The values of the STL parameters are: t_1 (reach time) $= 3s$, $\epsilon_p = 0.01m$, $\mathbf{u_{min}} = [-1.0, -1.0, -1.0]m/s$, $\mathbf{u_{max}} = [1.0, 1.0, 1.0]m/s$, δ_t (control input sampling interval) $= 0.2s$, $H = 15$ time-steps. For readability, we only show the x axis projection. Each strategy results in a different robot trajectory. In the "Predetermined" strategies (Fig. 4.3, top row), the robot goes to a fixed handover location regardless of the human's motion and thus may require adjustment on the human's part to reach the handover location. In the "Towards Human" strategies (Fig. 4.3, bottom row), the robot moves to the human hand's location. In the "Proactive" strategies (Fig. 4.3, left column), the robot starts to move towards the handover location even if the human hand is not in the handover zone, while in "Reactive" strategies (Fig. 4.3, right column) the robot starts to move towards the handover location only when the human hand is in the handover zone. These strategies result in different values for human and robot idle times, which are indicators of perceived human-robot fluency [62].

In addition to specifying overall strategies, the STL formulation allows users to specify high-level timing constraints for the handover, resulting in different controllers. To illustrate, Fig. 4.4 shows simulated robot trajectories for different values of t_1. The values of the other parameters are: $\epsilon_p = 0.01m$,



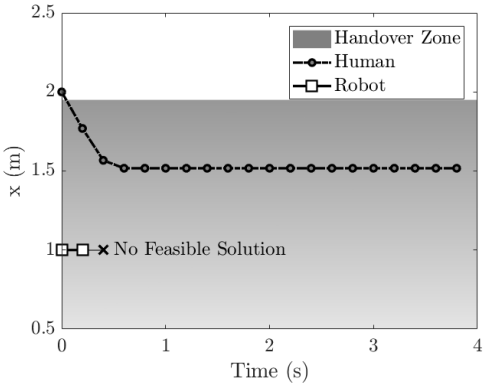
(a) Proactive, Predetermined

(b) Reactive, Predetermined

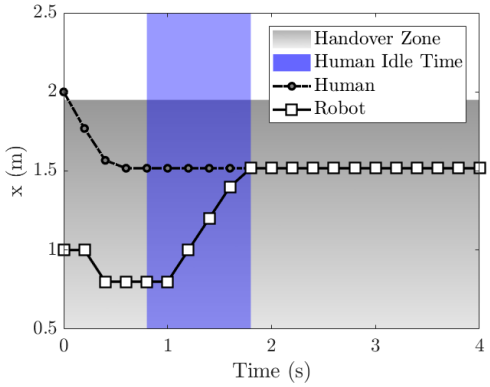
(c) Proactive, Towards Human

(d) Reactive, Towards Human

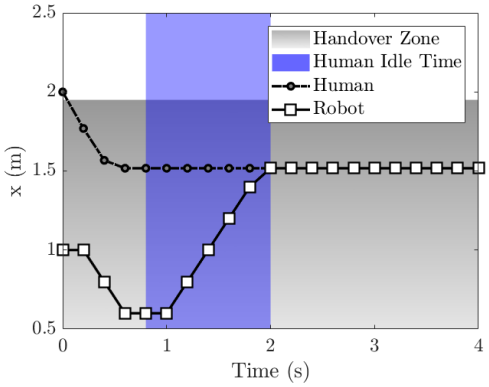
Figure 4.3: Simulation runs of different handover strategies. "Proactive" strategies (left column) may result in shorter human idle time but longer robot idle time, while "Reactive" strategies (right column) may result in shorter robot idle time but longer human idle time. A predetermined handover location may require adjustment on the human's part (top row), while the human hand as the handover location (bottom row) does not require any adjustment on the human's part. However, this may result in robot trajectories with overshoot (bottom-left) or the robot initially moving away from the human (bottom-right), due to the prediction of the human-motion model.



(a) Reach-Time = 1s



(b) Reach-Time = 2s



(c) Reach-Time = 3s

Figure 4.4: Representative simulations of different reach-time values (reactive, towards human strategy). A timing constraint of $t_1 = 1$s is infeasible given the safety cap on the robot's velocity. Increasing the constraint to $t_1 = 2$s allows the robot end-effector's trajectory to converge to the human hand's trajectory within the specified time limit. Increasing the reach time constraint from $t_1 = 2$s to $t_1 = 3$s increases the time taken by the robot end-effector's trajectory to converge to the human hand's trajectory and thus increases human idle time. This increase is less than 1s because the timing parameter is only an upper bound.

$\mathbf{u_{min}} = [-1.0, -1.0, -1.0]m/s$, $\mathbf{u_{max}} = [1.0, 1.0, 1.0]m/s$, $\delta_t = 0.2s$, $H = 15$ time-steps. In this example, a specification of $t_1 = 1s$ (reaching the handover location within one second) results in the controller being unable to find a control input, causing the robot to stop. For $t_1 = 2s$ and $t_1 = 3s$ the robot reaches the handover location within the specified time limit, with different human idle times.

4.4.3 Inferring Timing Parameters from a Handover Database

In addition to the possibility to synthesize a single controller for a given specification, our approach provides an additional benefit: the ability to estimate constraints on the high-level behavior of the robot under different circumstances. We illustrate this here by finding feasible bounds for t_1 based on a public database of human-human handovers [25]. This database of 1000 recordings was collected from 18 volunteers in 76 test configurations with different volunteers' starting positions, roles, objects to pass, and motion strategies. In our evaluation, we only use the trajectories for "uncontrolled" (the experimenter is not one of the actors) and "no-approach" (the volunteers are already standing close to each other) configurations. This results in a dataset of 72 giving motions and 72 receiving motions.

Each trajectory recording consists of 6-axis inertial data from the smartwatches on the wrists of both participants, the upper-body models of both participants with 20 motion capture markers, and the 15-joint skeleton model of one participant with an RGB-D camera. The sampling rate of the sensors is 7 Hz. Due to the drift errors associated with IMU measurements and loss of motion-tracking when the participants' hands overlap, we find that these two sensor data are not useful for getting the trajectory of the human hand. Therefore, we only use the wrist joint tracking data from the RGB-D camera. Since we only simulate the reach phase, we first segment each trajectory into reach, transfer, and retreat phases, as shown in Fig. 4.5. The reach phase begins at the time instant after which the distance from the starting position becomes greater than $d_1 \times d_{max}$. The reach phase ends (and the transfer phase begins) when the speed drops below $v_1 \times v_{max}$. The transfer phase ends (and the retreat phase begins)

when the speed goes below $-v_1 \times v_{max}$. The retreat phase ends when the distance from the starting position becomes less than $d_1 \times d_{max}$. We obtain the parameters d_1 and v_1 through a parameter sweep over the range $[0, 0.01, 0.02...0.5]$. For each combination of d_1 and v_1 in this range, we calculate the total number of handover trajectories from the database that can be split into valid reach, transfer, and retreat phases. A phase is valid if its end is detected before the end of the trajectory and if its duration is non-zero. We find that $d_1 = 0.41$ and $v_1 = 0.43$ result in the maximum number of such valid phases and hence use these values for splitting the trajectories.

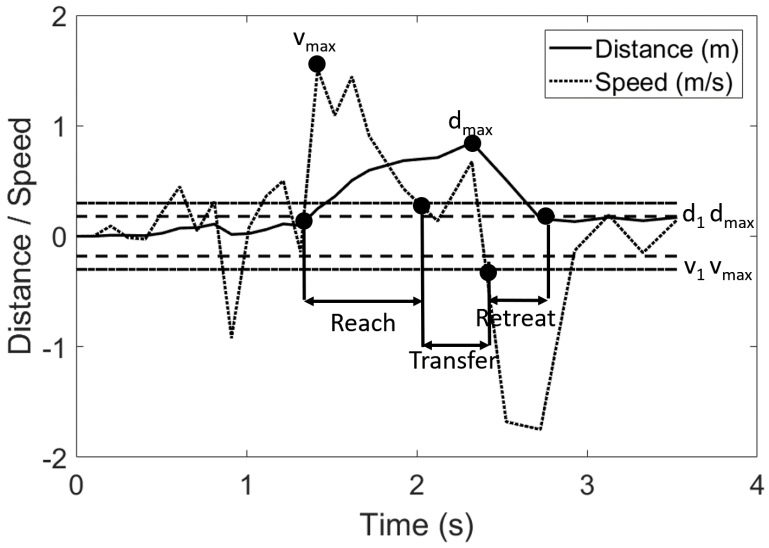


Figure 4.5: Distance from the starting position and speed of the human hand during a handover. Speed is positive if the hand is moving away from the starting position and is negative if the hand is moving towards the starting position. The trajectory is split into Reach, Transfer, and Retreat phases.

Table 4.4 shows the percentage of the human giving motions from the database for which the controller finds a feasible solution for a given reach-time value and a maximum velocity control input u_{max}. The values of the remaining STL parameters are: $\epsilon_p = 0.01m$, $\delta_t = 0.2s$, $H = 20$ time-steps. An end-user could use results like these to estimate the timing, and thus the productivity, of a handover system given a safety cap on the robot's velocity.

Table 4.4: Handover success rate for different reach-times.

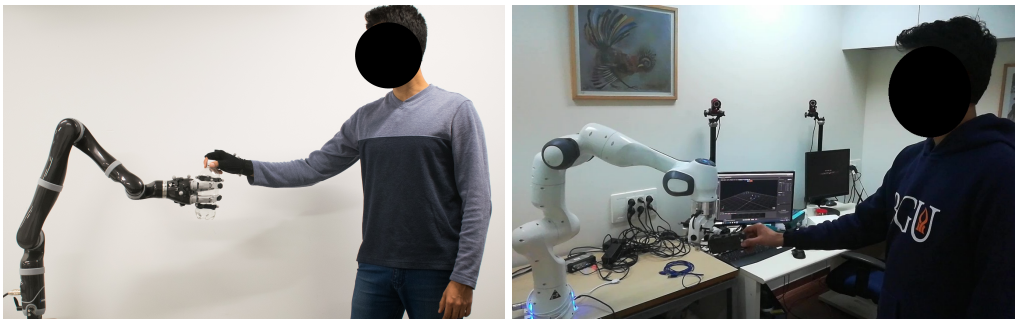
$u_{max}(m/s)$	t_1(s)	Successful Handovers
1	1s	14.3%
	2s	91.4%
	3s	100%
0.5	2s	2.9%
	3s	82.9%
	4s	97.1%

4.4.4 Implementation on a Real Robot

We attempted to implement our proposed STL controllers on two robot systems: Kinova Jaco-2 and Franka-Emika Panda. Fig. 4.6 shows the two setups. We used the OptiTrack motion tracking system to monitor the movements of the human hand, the object, and the robot end-effector. We used the BluSTL [138] toolbox in MATLAB to synthesize robot controllers from the STL specifications for the reach phase of a handover. The specifications only considered the position trajectory of the robot and the human, and the orientation of the robot was kept constant throughout the handover. The opening and closing of the robot's gripper, for robot-to-human and human-to-robot handovers respectively, and the retreat motion of the robot were commanded by another controller. The gripper's actions were triggered based on a distance threshold from the human's hand. The robot, the OptiTrack system, and the MILP solver in MATLAB were connected in a Robot Operating System (ROS) framework. We leveraged the distributed computing capabilities of ROS and used three separate computers to run these three components.

We used the *kinova_ros*[1] package to command the Kinova Jaco-2 robot. The robot end-effector's Cartesian velocity commands, obtained from BluSTL, were

[1] `https://github.com/Kinovarobotics/kinova-ros`



(a) (b)

Figure 4.6: Setup for testing STL controllers for human-robot handovers with (a) Kinova Jaco-2 collaborative robot, (b) Franka-Emika Panda collaborative robot

sent to the robot through the *"/in/cartesian_velocity"* topic. We found that the MILP solver in MATLAB required high computation times (approximately 0.5s for a prediction horizon of 16 time-steps and 3 control inputs), which severely limited the looprate of the controller. Also, the Kinova Jaco-2 robot's maximum Cartesian speed is 0.2m/s, which limited the reaching range of the robot in the prediction horizon. Therefore, we switched to a faster robot, the Franka-Emika Panda, which has a maximum Cartesian speed of 2m/s.

We used the *franka_ros*[2] package to command the Franka robot. The robot end-effector's Cartesian velocity commands, obtained from BluSTL, were sent to the robot through the *"franka_hw::FrankaVelocityCartesianInterface"* interface. Again, the looprate of the controller was limited by the high computation times of the MILP solver. Further, though the maximum Cartesian speed of the Franka robot is 2m/s, its joint/Cartesian velocity, acceleration and jerk limits[3] reduced the reaching range of the robot in the prediction horizon.

Future work can explore other computationally efficient techniques for STL controller synthesis, for example, Control Barrier Functions [98].

[2] `https://github.com/frankaemika/franka_ros`
[3] `https://frankaemika.github.io/docs/control_parameters.html`

CHAPTER 5

EVALUATING GUIDED-POLICY SEARCH FOR HUMAN-ROBOT HANDOVERS

5.1 Introduction

We develop and evaluate a robot controller that uses Guided Policy Search (GPS) to perform reaching motions for object handovers. GPS [93, 94, 95] is a reinforcement learning (RL) algorithm that has shown promising success in several autonomous tasks [91, 96, 92]. Some variants of GPS, like the one we use in this work [92], do not require prior knowledge of the robot or environment dynamics.

While GPS has been demonstrated on several autonomous manipulation and navigation tasks, it has not been tested in a physical human-robot collaborative task such as a handover. Examples of successfully learned manipulation tasks with GPS include stacking small blocks, assembling toys, inserting rings on wooden pegs, screwing bottle caps, inserting shapes into sorting cubes, and opening doors [91, 96, 92]. Common to all of these GPS applications are fixed targets, small variations in the test locations, and fixed robot dynamics. The task of object handovers has important characteristics that deviate from previous work: First, it requires a robot to plan its motion towards a moving target, i.e., the human's hand. Second, given the unpredictability of human behavior, the training and test target trajectories could be very different. Finally, due to different objects that are handed over, the robot dynamics are not fixed.

A substantial portion of this chapter is taken from [79] Reprinted, with permission, from A. Kshirsagar, G. Hoffman and A. Biess. "Evaluating Guided Policy Search for Human-Robot Handovers" IEEE Robotics and Automation Letters, 2021 (open access)

First, we evaluate GPS for handovers in simulation, and tackle previously unanswered questions such as: How does GPS perform if the training and test conditions are spatially far apart? How does GPS perform when reaching for an unpredictable moving target? How does GPS perform in the case of changes in the robot's end-effector mass? To do so, we formulate the reach phase of a handover as an RL problem and investigate the performance of GPS for the scenarios listed above. We find that the global policy learned with GPS does not perform well for target test locations spatially too distant from the target training locations but that this can be addressed with the addition of local controllers which are trained over target locations in the high error regions. In that case, the learned global policy can also handle moving targets with comparable errors, albeit with highly inefficient trajectories. Training on moving targets improves the trajectories but results in higher worst-case errors. Finally, we find that a learned global policy adapts well to changes in robot dynamics due to changes in the robot's end-effector mass. In an exploratory evaluation of different state representations, we find that a low-dimensional state representation may be more suitable for GPS-trained handover controllers.

Second, we try to replicate the findings from the simulation on a physical collaborative robot. Our investigations provide new insights on GPS when it is used on a physical robot platform. First, we find that a policy learned in simulation does not transfer readily to the physical robot due to differences in model parameters and existing safety constraints on the real robot. Second, to successfully train a GPS model, the robot's workspace needs to be severely reduced, owing to the joint-space limitations of the physical robot. Third, a policy trained with moving targets results in large worst-case errors even in regions spatially close to the training target locations. Our findings motivate further

research towards utilizing GPS in human-robot interaction settings, especially where safety constraints are imposed.

Note on contributions

The implementation and evaluation of GPS on a physical collaborative robot were carried out in collaboration with Tair Faibish (Master of Science, Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel). Both collaborators had an equal contribution. The implementation and evaluation is presented in detail in [45], and an abridged version is included in Sections 5.5 and 5.6.

5.2 Existing Evaluations of Guided Policy Search Methods

GPS was initially proposed by Levine et al. [93, 94, 95], and since then researchers have proposed several variations of the GPS algorithm [39]. Levine and Koltun [93] first proposed a GPS algorithm consisting of iterative linear quadratic regulators (iLQR) to generate locally optimal controllers, which guided the supervised learning of policies with a large number of parameters. They initialized the trajectories from demonstrations and used importance sampling to incorporate new samples from the optimized trajectories at each gradient step. They demonstrated their method by learning simulated locomotion tasks such as planar swimming, hopping, walking, and running. They improved this method by replacing importance sampling with a variational decomposition of a maximum likelihood objective [94] or a policy agreement constraint [95]. All of these GPS variations required known dynamics of the

system. Montgomery and Levine [114] proposed a variant of GPS called "Mirror Descent GPS" which used the constrained GPS formulation but instead of constraining the local policies against the previous local policies, this method constrained it directly against the global policy and used the true cost instead of the surrogate cost. This method achieved similar or better results than previous GPS algorithms on simulated peg insertion and obstacle avoidance tasks. For optimizing trajectories of systems with unknown dynamics, Levine and Abbeel [91] extended the constrained GPS algorithm of Levine and Koltun [95] with iterative refitting of locally linear dynamics models. They showed that this method requires fewer samples than model-free methods and does not need to learn global models, which are difficult to learn for complex systems. They evaluated their method on simulated robotic manipulation tasks such as peg insertion, and locomotion tasks such as swimming and walking. Levine et al. [96] extended the evaluation of the algorithm through a variety of experiments on a real robotic platform for tasks such as stacking lego blocks, assembling toys, inserting a shoe tree, inserting rings on wooden pegs, and screwing bottle caps. Building upon their prior work, they improved the GPS algorithm by adding an adaptive scheme to select step size and number of samples, along with a method of augmenting policy training with synthetic samples.

Levine et al. [92] proposed an end-to-end approach to learn policies that map raw image observations directly to robot joint torques. They used the constrained GPS algorithm and formulated it as an instance of the Bregman-Alternating Direction Method of Multipliers (BADMM). They tested this method on tasks that require close coordination between vision and control such as inserting shapes into a sorting cube, screwing a bottle cap, placing a hanger on a bar, and inserting a hammer underneath a nail. Zhang et al. [185]

augmented the original GPS algorithm with a model predictive control (MPC) scheme to generate training data without catastrophic failures. They used an instrumented setup during the training phase to get full state observations and trained a deep neural network policy with the guiding samples generated by MPC. During the testing phase, the policy could generate control inputs from partial observations of the system. They showed that this algorithm was comparable to the original GPS algorithm in the absence of model errors and outperformed the GPS algorithm when model errors were introduced. Chebotar et al. [32] augmented GPS with a model-free local optimizer based on path integral (PI) stochastic optimal control, instead of iLQR, to generate local controllers. Also, unlike the GPS algorithms of Levine and Koltun, which used the local controllers to generate training data, Chebotar et al. generated training samples by running the global policy on new sets of task instances in each iteration. They initialized local policies with kinesthetic teaching and initialized the global policy by performing several iterations of standard GPS with local policy sampling using PI. This method performed better than iLQR-based GPS on tasks with intermittent and variable contacts and discontinuous cost functions.

While researchers have tested GPS algorithms on a variety of locomotion and autonomous manipulation tasks, to the best of our knowledge, no work has evaluated GPS for tasks with large variations in target locations, moving targets, and changes in robot dynamics, as are typical in HRI scenarios such as handovers. Also, none of the prior works on GPS have evaluated the sensitivity of GPS to the system's state-space representation. We seek to address this gap in this work by evaluating a robot controller that uses GPS for the reach phase of human-robot object handovers. A large body of work has studied transfer learning [127] and domain adaptation [174] where the training and testing conditions

belong to different tasks/distributions. However, in our work, the training and testing conditions belong to the same task and are drawn from the same distribution. Therefore, our problem statement is different from transfer learning or domain adaptation.

5.2.1 Overview of the Guided Policy Search Algorithm

The goal of policy search algorithms is to find a policy $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$ that minimizes the expected cost $E_{\pi_\theta}[\sum_{t=1}^{T} l(\mathbf{x}_t, \mathbf{u}_t)]$ of executing a task. Here θ denotes the policy parameters, for example weights of a neural network, $\mathbf{u}_t$ is the control input at time t, $\mathbf{x}_t$ is the state of the system at time t, and $l(\mathbf{x}_t, \mathbf{u}_t)$ is the cost associated with the task at time t. Directly solving this minimization problem through reinforcement learning requires large amounts of training data and is susceptible to local minima. Guided policy search algorithms overcome these issues through the use of guiding distributions or "local" controllers $p_i(\mathbf{u}_t|\mathbf{x}_t)$ to train the "global" policy $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$ through supervised learning. The local controllers can be trained via trajectory optimization methods such as iLQR. Thus GPS poses the expected cost minimization problem as a constrained problem given by

$$\min_{p,\theta} E_p[\sum_{t=1}^{T} l(\mathbf{x}_t, \mathbf{u}_t)] \quad \text{s.t.} \quad p(\mathbf{u}_t|\mathbf{x}_t) = \pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\ \forall t, \tag{5.1}$$

where $p(\mathbf{u}_t|\mathbf{x}_t)$ is a mixture of guiding distributions $p_i(\mathbf{u}_t|\mathbf{x}_t)$. The expectation is taken with respect to $p(\tau) = p(\mathbf{x}_1)\prod_{t=1}^{T} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)p(\mathbf{u}_t|\mathbf{x}_t)$, where $\tau = \{\mathbf{x}_1, \mathbf{u}_1, ..., \mathbf{x}_T, \mathbf{u}_T\}$ is a trajectory and $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$ is the dynamics model of the system. As described in Section 5.2, some variants of GPS algorithms require known dynamics models while others iteratively learn locally linear dy-

namics models from the training data.

In this work, we use the Bregman-Alternating Direction Method of Multipliers (BADMM) GPS algorithm proposed by Levine et al. [92] which does not require prior knowledge of the robot dynamics. In this algorithm, the local controllers $p_i(\mathbf{u}_t|\mathbf{x}_t)$ and the dynamics $p_i(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$, $\forall i \in [1, 2, ..., N]$ where N is the number of local controllers, are represented with time-varying Linear Gaussians:

$$p_i(\mathbf{u}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{K}_{t,i}\mathbf{x}_{t,i} + \mathbf{k}_{t,i}, \mathbf{C}_{t,i}), \tag{5.2}$$

$$p_i(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(f_{\mathbf{x}t,i}\mathbf{x}_t + f_{\mathbf{u}t,i}\mathbf{u}_t + f_{ct,i}, \mathbf{F}_{t,i}). \tag{5.3}$$

The linear Gaussian controllers and dynamics can be efficiently learned with a small number of samples. A different set of controller and dynamics parameters are fitted for each training target trajectory (in our case: the human's reaching motion). But a single global policy is supervised by all of the local controllers, making it generalizable to different test target trajectories.

Levine et al. [92] suggest modifying the constraint in Eq. 5.1 by multiplying with $p(\mathbf{x}_t)$ and applying it to expected action, to make the constraint tractable:

$$\min_{p,\theta} E_p[\sum_{t=1}^{T} l(\mathbf{x}_t, \mathbf{u}_t)] \text{ s.t. } E_{p(\mathbf{x}_t,\mathbf{u}_t)}[\mathbf{u}_t] = E_{p(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)}[\mathbf{u}_t] \ \forall t. \tag{5.4}$$

The GPS algorithm alternates between generating optimal trajectories for each local controller with iLQR and training a global policy supervised by the local controllers. The global policy is also used to improve the local controllers, such that the local controllers stay close to the global policy. GPS thus alternates minimization of θ and p as follows:

$$\theta \leftarrow \arg\min_{\theta} \sum_{t=1}^{T} E_{p(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)}[\mathbf{u}_t^T \lambda_{\mu t}] + \nu_t E_{p(\mathbf{x}_t)}[D_{KL}(p(\mathbf{u}_t|\mathbf{x}_t)||\pi_\theta(\mathbf{u}_t|\mathbf{x}_t))], \tag{5.5}$$

$$p \leftarrow \arg\min_{p} \sum_{t=1}^{T} E_{p(\mathbf{x}_t,\mathbf{u}_t)}[l(\mathbf{x}_t,\mathbf{u}_t) - \mathbf{u}_t^T \lambda_{\mu t}] + \nu_t E_{p(\mathbf{x}_t)}[D_{KL}(\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)||p(\mathbf{u}_t|\mathbf{x}_t))], \tag{5.6}$$

$$\lambda_{\mu t} \leftarrow \lambda_{\mu t} + \alpha\nu_t(E_{p(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)}[\mathbf{u}_t] - E_{p(\mathbf{x}_t)p(\mathbf{u_t}|\mathbf{x}_t)}[\mathbf{u}_t]), \tag{5.7}$$

where $\lambda_{\mu t}$ is the Lagrange multiplier on the expected action at time t, ν_t is the weight of the Kullback–Leibler divergence term that serves to keep $p(\mathbf{u}_t|\mathbf{x}_t)$ close to $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$. For a detailed description of the GPS algorithm, see [92].

5.3 Policy Search Formulation of Handovers

We formalize the reach phase of a handover task as a reinforcement learning problem. To do so, we specify the state/action space, as well as a cost/reward function in the form of a differentiable function over the system states and control inputs.

5.3.1 System State Representation

Any reinforcement learning method is sensitive to its state representation, and in this work, we explored three alternatives for the system state $\mathbf{x}_t$. The first one is the `FULL` state, which might be available in a laboratory setup supported by a motion tracking system. In this representation, the state consists of the robot joint angles θ_r, the robot joint velocities $\dot{\theta}_r$, the human arm joint angles θ_h, the human arm joint velocities $\dot{\theta}_h$, the positions and velocities of three points on the object $(\mathbf{p}_o, \dot{\mathbf{p}}_o)$, the human hand $(\mathbf{p}_h, \dot{\mathbf{p}}_h)$ and the robot gripper $(\mathbf{p}_r, \dot{\mathbf{p}}_r)$, and the

robot gripper's width $g_r \in [0, g_{open}]$ (0 for fully closed, g_{open} for fully open). The positions are measured in an inertial frame fixed to the base of the robot. A state is thus given by

$$\mathbf{x}_t = [\theta_r, \dot{\theta}_r, \theta_h, \dot{\theta}_h, \mathbf{p}_o, \mathbf{p}_h, \mathbf{p}_r, \dot{\mathbf{p}}_o, \dot{\mathbf{p}}_h, \dot{\mathbf{p}}_r, g_r]_t. \tag{5.8}$$

As the human's joint angles are difficult to measure for a robot outside a laboratory, we also explore a `REDUCED` state representation, which excludes the human arm joint angles and joint velocities:

$$\mathbf{x}_t = [\theta_r, \dot{\theta}_r, \mathbf{p}_o, \mathbf{p}_h, \mathbf{p}_r, \dot{\mathbf{p}}_o, \dot{\mathbf{p}}_h, \dot{\mathbf{p}}_r, g_r]_t. \tag{5.9}$$

Given the possible large variation in human position, we also explore a third option, which includes the human hand and the object poses in the robot end-effector frame instead of an inertial frame fixed to the base of the robot. This `RELATIVE` representation corresponds to the configuration in which a camera is attached to the robot end-effector:

$$\mathbf{x}_t = [\theta_r, \dot{\theta}_r, \mathbf{p}_o^r, \mathbf{p}_h^r, \dot{\mathbf{p}}_o^r, \dot{\mathbf{p}}_h^r, g_r]_t. \tag{5.10}$$

In all of the three alternatives, the robot's control input $\mathbf{u}_t = [\tau, f_g]_t$ consists of the robot joint torques τ and the force applied by the gripper's actuator f_g, constrained by $\mathbf{u_{min}} \leq \mathbf{u}_t \leq \mathbf{u_{max}}$.

We use the `REDUCED` state representation in the majority of the results below. We conclude with an exploratory comparison of the three state representations. We use torques as control inputs instead of velocities or positions to take into account the dynamics of the robot. This eliminates the need to tune low-level position/velocity controllers.

5.3.2 Cost Function

In the reach phase of handovers, the robot should move its gripper towards the human hand. We represent this behavior with a cost function given by

$$c_{reach} = ||\mathbf{p}_r - \mathbf{p}_h||^2 + \ln(||\mathbf{p}_r - \mathbf{p}_h||^2 + \alpha_{reach}). \tag{5.11}$$

The first term of this cost function penalizes robot positions far away from the human hand, while the second term encourages precise placement due to its concave shape, as described in [96]. Thus this cost function encourages the robot to reach toward the human hand quickly and precisely. The parameter α_{reach} determines the penalty in the vicinity of the target. Similar to [96], we set $\alpha_{reach} = 1e - 5$ in the evaluations described in the next section.

5.4 Evaluation of GPS-BADMM in a Simulation Environment

We evaluate the performance of the global policy learned with GPS for large variations in target locations, moving targets, and changes in robot dynamics. To do so, we train a collaborative robot to perform handovers over repeated trials in a simulation environment with different training regimens and test on different target trajectories. We measure the performance of the global policy in terms of the error between the end-effector's position and the human hand's position.

We build upon the BADMM-GPS implementation by Finn et al. [47]. The collaborative robot in the handover task is simulated in MuJoCo [167] (Multi-Joint dynamics with Contact). Fig. 5.1 shows the MuJoCo simulation environment

built for this study. The robot on the left is a Franka-Emika Panda with 7 degrees of freedom (DoFs), equipped with a two-fingered gripper. In the remaining text, we call this robot the "learner". The environment also includes a pseudo-robot arm with two DoFs and a cylindrical mass rigidly attached to its end-effector. In the remaining text, we call this robot the "trainer" or the "tester" depending on whether it is used to train the global policy or to test the learned global policy. This robot stands in for the human and "teaches" the learner to perform handover reaching motions in simulation. The neural network policy used one hidden layer with 84 units and ReLU (Rectified Linear Unit) activation function of the form $f(z) = max(z, 0)$. The policy training was done in Caffe deep learning framework using the Euclidean loss function.

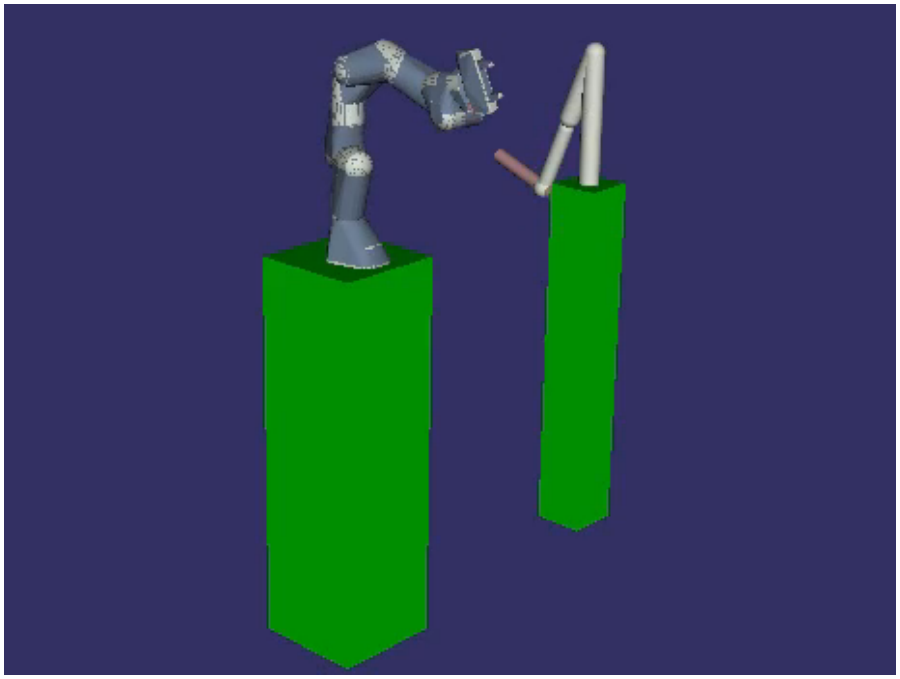
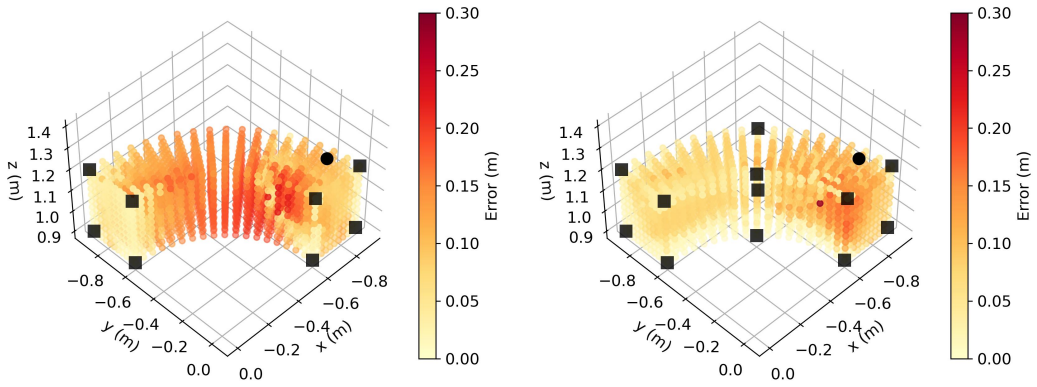
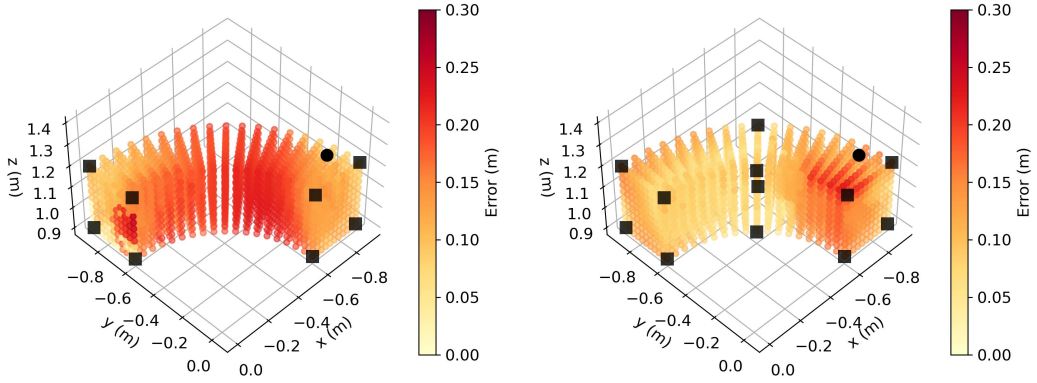


Figure 5.1: MuJoCo simulation environment. We train a Panda robot arm (left) to perform handover reaching motions by simulating the reach phase with another robot (right), standing in for the human.



(a) Static Trainer (8 Local Controllers) Static Tester

(b) Static Trainer (12 Local Controllers) Static Tester



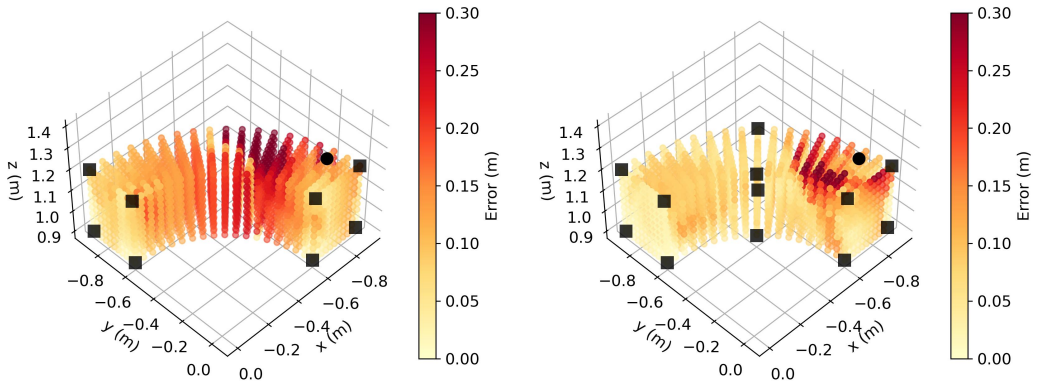
(c) Static Trainer (8 Local Controllers) Moving Tester

(d) Static Trainer (12 Local Controllers) Moving Tester

Figure 5.2: Global policy evaluation for different types of trainers and testers (continued on the next page). Error-values are clamped to the range [0.0, 0.3] for better visualization.

5.4.1 Simulation Results

The first research question that we investigate is the spatial generalizability of the learned global policy, i.e., how does the global policy perform for large spatial differences between training and test locations. To answer this question, we test the learned global policy at different locations of a static tester on a semi-hemispherical shell around the learner robot, which represents the workspace



(e) Moving Trainer (8 Local Controllers) Moving Tester

(f) Moving Trainer (12 Local Controllers) Moving Tester

Figure 5.2: Global policy evaluation for different types of trainers and testers (continued from the previous page). In the 'static' case, the trainer/tester stays in a fixed configuration. In the 'moving' case, the trainer/tester moves with a human-like trajectory and reaches the locations given by colored dots. Thus each point corresponds to the final position of the tester's gripper in a trial, and the black square markers correspond to the training target locations. The black round marker corresponds to the learner robot's gripper's starting position. The error between the learner's gripper position and the tester's gripper position is averaged over the last 0.5 seconds of each trial. We find that: 1) Error increases as the target location is shifted away from the training locations (all figures). 2) Increasing the number of training locations reduces the error (left column vs right column). 3) Error is higher if the trainer is static and the tester is moving (first row vs second row). 4) Error is more sensitive to location and performs worse in the worst case if the global policy is trained on a moving target (bottom row). Error-values are clamped to the range [0.0, 0.3] for better visualization.

of the robot. For each angle in $5\,\mathrm{deg}$ increments, we test on a grid of 11×11 targets, resulting in 2299 test locations. We initially train the global policy with eight local controllers for target locations at the corners of the workspace. Each trial runs for 2 seconds, both the learner and the trainer/tester start moving at the same time, and the global policy is improved over 12 trials. The test performance is measured as the mean error between the learner's gripper position and the tester's hand position over the last 0.5 seconds of each trial.

Fig. 5.2a shows the performance of the learned global policy, The training

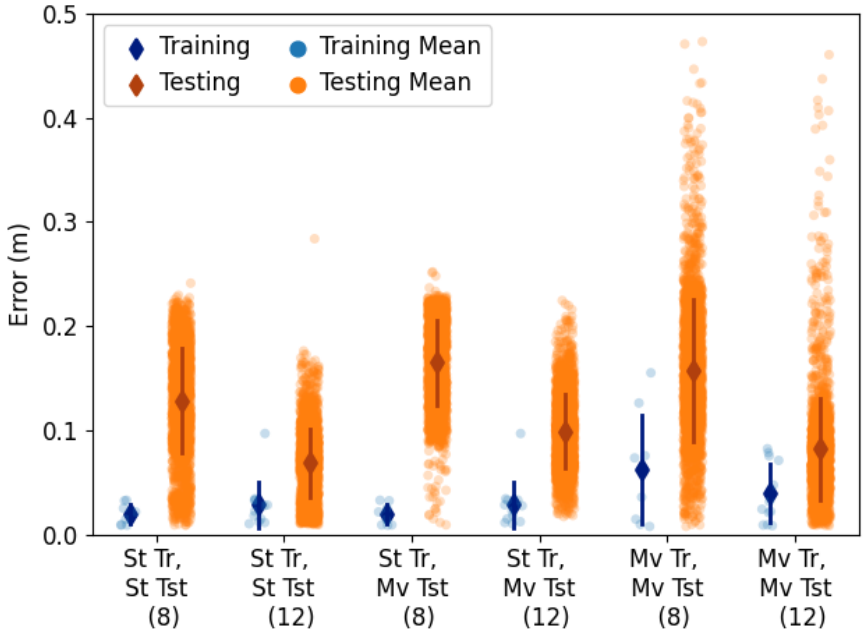


Figure 5.3: Distributions of training and testing performance for each target scenario. Each point is the mean error between the learner's gripper position and the tester's hand position over the last 0.5 seconds of a trial. Error bars show one standard deviation around the mean of each distribution.

locations are marked with black squares and the learner's gripper's initial position with a black circle.

Fig. 5.3 (left) shows the mean, range, and standard deviation of the error. The mean testing error (128 mm) is more than 6 times the mean training error (20 mm). The error increases up to 241 mm as the spatial distance between the training and the testing target locations increases. This issue can be somewhat addressed by adding four additional local controllers trained with target locations in the plane dividing the workspace (Fig. 5.2b). For a global policy trained with these 12 local controllers, the mean and standard deviation of the testing error is reduced to 69 ± 32mm. The worst case (outlier) testing error remains very high (284 mm).

Next, we investigate how GPS performs when the target is moving. First,

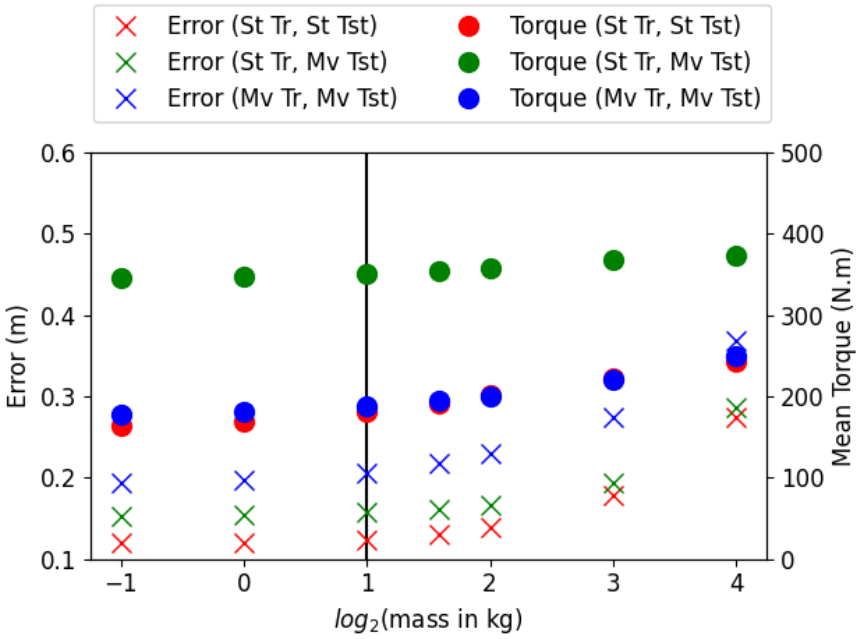


Figure 5.4: Mean error and torque for different robot end-effector masses (0.5–16kg). The vertical line represents the baseline mass of the robot end-effector used during training (2kg). Each error marker corresponds to the mean error of all testing locations on a semi-hemispherical shell around the robot as shown in Fig. 5.2. Torque markers show the mean of the norm of torques applied by the robot, averaged over the same testing locations. The error remains fairly constant over a wide range of the robot end-effector mass (up to 4kg), and the global policy produces total torques proportional to the changes in the robot mass. Torques are significantly higher for the 'static train, moving test' scenario, indicating highly inefficient trajectories.

we use the same global policy shown in Fig. 5.2a (static training) but instead of a static tester we simulate the tester to execute a human-like trajectory in joint space [142], given by

$$\theta_{h,i} = \frac{a(\theta_{f,i} - \theta_{0,i})}{b + e^{\frac{-ct}{t_f}}} + \theta_{0,i} \quad \forall i \in 1, 2, \tag{5.12}$$

where $a = 9.05e^{-4}$, $b = 8.908e^{-4}$ and $c = 12.87$ are empirical coefficients determined by Rasch et al. [142] from human arm motion data. $\theta_{0,i}$ and $\theta_{f,i}$ are the initial and final values of the i^{th} joint angle, respectively, t_f is the movement

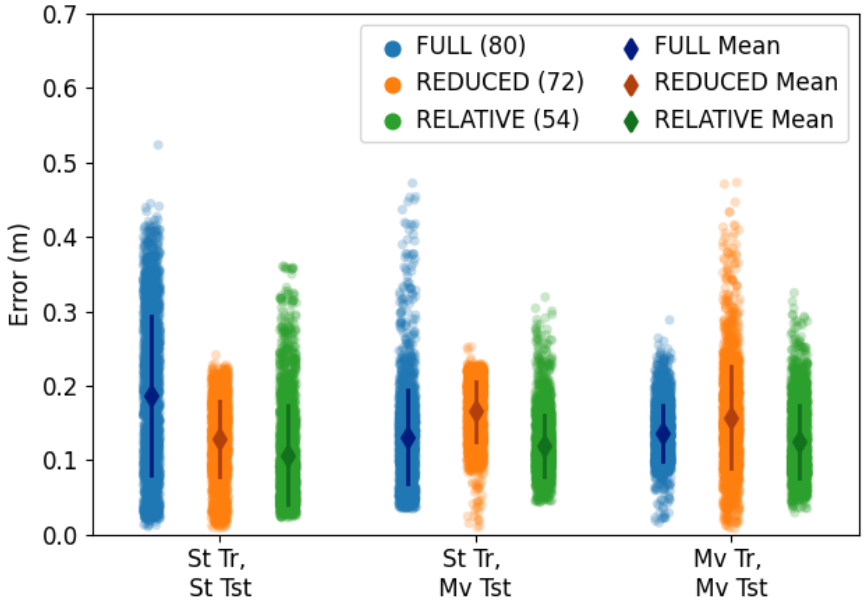


Figure 5.5: Distributions of testing performance for different state representations described in Section 5.3.1. In each case the global policy is trained with 8 local controllers. The number in parentheses represents the number of state variables in each representation. Each point is the mean error between the learner's gripper position and the tester's hand position over the last 0.5 seconds of a trial. Error bars show one standard deviation around the mean of each distribution.

duration, while $\theta_{h,1}$ and $\theta_{h,2}$ correspond to the shoulder and elbow joints, respectively. We set $\theta_{0,1} = \theta_{0,2} = 0$, $t_f = 1$, and use inverse kinematics to compute the final values of $\theta_{h,1}, \theta_{h,2}$ for a given Cartesian position of the tester's gripper.

We vary the tester's trajectories such that its gripper's final position is on the same semi-hemispherical shell around the learner robot as before. The global policy's performance is again measured as the mean error between the learner's gripper position and the tester's gripper position over the last $0.5s$ of each trial. Since we set $t_f = 1$ in Eq. 5.12, this error is calculated after the tester has reached the final position.

Fig. 5.2c shows the results for the global policy trained with 8 local con-

trollers; Fig. 5.3 (middle) shows the mean, range, and standard deviation of the error. The performance is worse with a mean testing error of 165 mm for a moving target, 28.9% higher than the mean testing error for a static target, but the range of error is comparable. Few target locations result in low errors. For the global policy trained with 12 local controllers (Fig. 5.2d), the mean testing error is 99 mm for a moving target, 43.5% higher than the mean testing error for a static target. The range, again, is comparable, with more target locations, as compared to the global policy trained with 8 local controllers, having low errors.

That said, the trajectories generated by these "Static Trainer, Moving Tester" trials are highly inefficient. Fig. 5.4 (center line) shows that the mean torque i.e. the L^2 norm of the robot's joint torques averaged over all test points and time-steps, is almost double over the trajectory.

A possible way to address this issue is to train the controller with a moving target, also executing a human-like trajectory in the joint space, as described in Eq 5.12. Fig. 5.2e and Fig. 5.2f show the performance of the global policy for various final positions of the tester's gripper, defined as in previous trials. Fig. 5.3 (right) shows error distributions.

For the global policy trained with a moving trainer and 8 local controllers (Fig. 5.2e), the mean testing error is 157mm and thus does not provide a meaningful improvement. Moreover, the variance over the target location is high, and the worst-case error is 473 mm, 87.7% higher than the maximum error for the static trainer condition (252 mm). In fact, the GPS process does not converge to a low training error, which is more than 3x that of the static training results. For the global policy trained with a moving trainer and 12 local controllers (Fig. 5.2f), the mean testing error is reduced to 82 mm, 17.2% lower than

the mean testing error for the static trainer condition. But the variance of the performance remains high, with a 461 mm worst-case performance. That said, an inspection of the generated trajectories and torques shows that this approach results in more efficient trajectories and torques similar to those achieved with static targets.

The third research question that we address is how the global policy performs under changes in robot end-effector mass. To investigate this question, we train the robot with a baseline end-effector mass of 2kg and evaluate the performance of the global policy for different robot end-effector masses, ranging from 0.5kg to 16kg. Fig. 5.4 shows the mean error between the learner's gripper position and the tester's gripper position for different robot end-effector masses. We find that the mean error across the same testing locations shown in Fig. 5.2 remains largely unaffected between $0.5 - 4$kg, but the error increases if the end-effector's mass is increased beyond this limit. Fig. 5.4 also shows the means of the norm of torques applied by the seven joints of the robot for different robot masses. We find that the mean increases with an increase in the robot end-effector's mass, except when the robot is trained on static targets but tested on moving targets, where the torques are always high. We also investigated the effect of changing the total mass of the robot and found that for a baseline mass of 18.5kg the error remained fairly constant up to 100kg.

In Section 5.3.1, we proposed different possible state representations. Fig. 5.5 shows the performance of the global policy trained with 8 local controllers, across all three state representations. For policies trained on static targets, the `REDUCED` state representation has the lowest variance (best generalization), but this does not hold for policies trained on moving targets. Overall, a global policy

trained with the lowest-dimensional `RELATIVE` state representation (54 dimensions) has a better average performance than the other state representations. This suggests that lower-dimensional state models may be more appropriate for GPS-trained handover controllers.

5.5 Implementation of GPS-BADMM on a Collaborative Robot

We train a collaborative robot, a Franka-Emika Panda research robot ("Panda robot" henceforth), to perform reach-to-handover motions with the GPS-BADMM algorithm[1]. This is the same robot that we used in the simulation in the previous section. The robot, shown in Fig. 5.7, is a 7 degrees-of-freedom (DoF) robot arm with torque sensors at each joint, allowing adjustable stiffness/compliance and torque control. We use an OptiTrack motion tracking system to monitor the positions of the human's hand and the robot's end effector.

The robot is controlled via a distributed Robot Operating System (ROS) network across three personal computers (PC) as shown in Fig. 5.6: One PC runs the motion tracking software and streams the motion tracking data on a local network. The second PC converts the motion tracking data into the relevant coordinate frames, computes the state representations described in Section 5.3.1, and runs the GPS training and testing procedures to generate controls for the robot. These are sent to a third PC, which is physically connected to the Panda robot. This third PC runs a real-time kernel as required by the robot manufacturer.

As described in the previous section, a `RELATIVE` state representation had a

[1]The code is available at `https://github.com/alapkshirsagar/real-franka-gps`

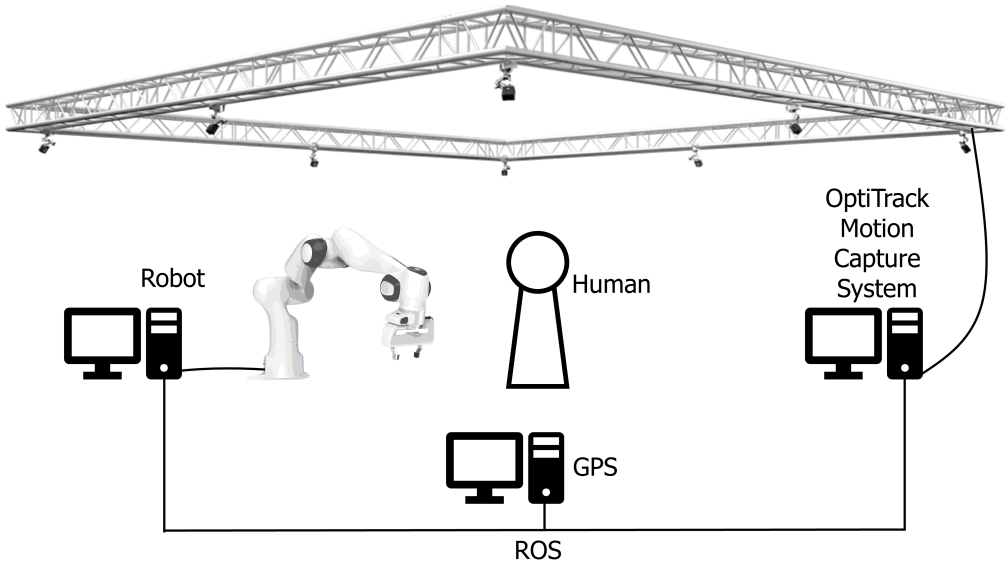


Figure 5.6: The system setup includes a Panda robot (left), an Optitrack motion capture system (top), and three PCs communicating through ROS: one collecting motion tracking data (right), one responsible for RL (bottom center), and one connected to the robot (left).

better overall performance. Here, we use a similar robot-centric state representation which consists of the robot joint angles θ_r, the robot joint velocities $\dot{\theta}_r$, the positions and velocities of the robot-end effector in the world frame attached to the base of the robot ($\mathbf{p}_r$,$\dot{\mathbf{p}}_r$), and the positions and velocities of the human hand in the robot end-effector frame ($\mathbf{p}_h^r$,$\dot{\mathbf{p}}_h^r$).

$$\mathbf{x}_t = [\theta_r, \dot{\theta}_r, \mathbf{p}_r, \mathbf{p}_h^r, \dot{\mathbf{p}}_r, \dot{\mathbf{p}}_h^r]_t. \tag{5.13}$$

The robot's control input $\mathbf{u}_t$ consists of the robot joint torques τ: $\mathbf{u}_t = \tau_t$. We do not operate the gripper attached to the robot arm, since the evaluation focuses on the reach-to-handover motion.

We build on the GPS-BADMM implementation of White [177], which was designed for a Kuka LWR robot [14] in the Gazebo simulation environment. White's code, in turn, was based on the GPS implementation of Finn [47], designed for a Willow Garage PR2 robot [53].

In order to adapt the implementation of White [177] to the physical Panda robot, we had to tune several parameters, including the initial gains, variances, and stiffness parameters for the local iLQR controllers, and the PID parameters used to command the robot back to its initial position, after each trial. Tables 5.1 and 5.2 compare the values used by previous implementations and the one presented in this paper. For more details about these adaptations, parameters we refer the reader to [45]. In general, we found that the high gains used in previous implementations resulted in the robot abruptly halting due to violations of joint trajectory or Cartesian trajectory requirements[2]. These requirements consist of safety limits on the robot's joints' positions, velocities, and torques, as well as on the robot end-effector's Cartesian position, velocity, acceleration, and jerk. Such inbuilt limits are common for robot arms designed for close-range human-robot interaction.

Table 5.1: Comparison of PID parameters of position controller for resetting the LWR and the Panda robots.

Joint	**LWR Values [177]**				**Panda Values**			
	P	I	D	I_{clamp}	P	I	D	I_{clamp}
1	2400	0	18	4	6	3	3	1
2	1200	0	20	4	6	3	3	1
3	1000	0	6	4	6	3	3	1
4	700	0	4	4	6	3	3	1
5	300	0	6	2	2.5	1	1	1
6	300	0	4	2	2.5	1	1	1
7	300	0	2	2	2.5	1	1	1

[2]Franka-Emika Panda Specifications: `https://frankaemika.github.io/docs/control_parameters.html`

Table 5.2: Initial controller parameter values of PR-2, LWR, and Panda robots.

Parameter	**PR-2 [47]**	**LWR [177]**	**Panda**
Joint 1 Gain	3.09	24	0.1
Joint 2 Gain	1.08	12	0.1
Joint 3 Gain	0.393	10	0.1
Joint 4 Gain	0.674	7	0.1
Joint 5 Gain	0.111	3	0.001
Joint 6 Gain	0.152	3	0.001
Joint 7 Gain	0.098	6	0.001
Initial Variance	1	30	0.5
Stiffness	0.5	60	1.0
Stiffness Velocity	0.25	0.25	0.5

5.6 Evaluation of GPS-BADMM on a Collaborative Robot

The goals of our evaluation of GPS on a physical collaborative robot are, first, to examine whether our previous findings, which were performed only in simulation, replicate, and, second, to investigate the challenges associated with the use of GPS in a physical HRI setup.

We do so in two stages: First, we test whether policies learned in simulation can be used in a physical setup ("Sim-to-Real Evaluation"). Then, we try to use GPS to train policies using the physical robot ("Real-to-Real-Evaluation").

5.6.1 Sim-to-Real Evaluation

We previously used GPS to train a robot reach-to-handover in a *MuJoCo* simulation environment. We trained a Panda robot for the task with a pseudo-robot arm with a cylindrical mass rigidly attached to its end-effector, substituting the human operator. In the first experiment of the present work, we ask whether the learned policy can be used on a physical Panda robot.

We find that the policy trained in simulation fails to generate correct trajectories on the physical Panda robot. This failure is due to two main reasons: differences in model parameters between the simulated and real robot and inbuilt safety constraints on the real robot.

While robot manufacturers provide some information about their hardware and software, some of it is proprietary or undocumented. As a result, simulation models almost always have to make assumptions about some important dynamics parameters. In this case, we used a MuJoCo model of the Panda robot created with the robot's meshes and specifications from *franka_ros*[3]. This model used the geometry provided by the manufacturer and any existing dynamics parameters. However, the manufacturer does not provide a damping factor for each joint, which was set arbitrarily, resulting in poor Sim-to-Real dynamics transfer.

Furthermore, the Panda robot, like most human-collaborative robots, has inbuilt safety constraints for joint positions, velocities, and torques (see also: Section 5.5). In the Sim-to-Real transfer, many of the torques generated by the GPS policy learned in simulation exceeded these constraints.

It is worth noting here that the GPS algorithm does not readily incorporate hard constraints. We tried to impose a torque reduction in the simulation training by adding a penalization term in the RL cost function for out-of-limit velocities and out-of-limit torques. The torques were reduced slightly but not enough to run the global policy on the real robot. We also clamped the torques before the torques were sent to the robot's joints in MuJoCo. In that case, the robot could not learn a handover trajectory at all and barely moved from its initial po-

[3]Franka-Emika Panda MuJoCo Model: `https://github.com/vikashplus/franka_sim`

sition. In summary, due to the architecture of GPS training, undisclosed robot parameters, and the fact that human-collaborative robots have hard safety constraints, it proved difficult to train in simulation and execute trajectories on a physical robot.

5.6.2 Real-to-Real Evaluation

In the second stage of evaluation, we used GPS to directly train a physical Panda robot to perform reaching motions towards a human's hand and tested the learned global policy for large variations in target locations and moving targets. To reiterate, we identified large target variations and moving targets as important characteristics of human-robot handovers that were difficult for GPS to generalize over.

In the remaining text, we denote the Panda robot as the "learner", and the human as the "trainer" or "tester" for the training and testing phase, respectively. Since it is not practical to have a human perform exactly the same reaching motion in all training iterations, we used recorded human hand motions and replayed them to the robot via ROS.

The first research question examined in our study is the spatial generalizability of the learned global policy, i.e., how does the global policy perform for large spatial differences between training and testing locations. To answer this question, we first tried to use the same training and testing locations as our previous evaluation in a simulation environment. However, we found that the Panda robot ran into joint and Cartesian limit violations for those training locations. Since Panda is a collaborative robot, these constraints are set in firmware

and cannot be overridden. Instead, we had to reduce the robot's workspace by trial-and-error. The final workspace used in our evaluation is shown in Fig. 5.7. The overall angular range around the robot was $45\,\mathrm{deg}$. For each angle in $5\,\mathrm{deg}$ increments, we tested on a grid of 3×3 targets, resulting in 90 test locations.

Replicating the conditions in the simulation study, we compared two scenarios of local controllers (initial conditions): one with 8 local controllers and another with 12 local controllers. The global policy was trained with these local controllers for 11 iterations. Both the learner and the trainer/tester commenced their movement in each trial simultaneously. The learner's movement lasted for 5 seconds, while trainer's/tester's movement lasted for 1 second. The test performance was measured as the mean error between the learner's gripper position and the tester's hand position at the last time step of each trial.

The performance of the learned global policy is shown in Fig. 5.8a. The black circle represents the learner's gripper's initial position, and the black squares represent the training locations. The colored circles indicate the error at the last time step for each handover location. Mean error and its standard deviation are shown in Fig. 5.9 (left). The mean testing error (41.7mm) is about twice as large as the mean training error (22.7mm).

If our previous studies replicate, this test error can be reduced by adding more local controllers in high error regions. In the current study, we similarly find that adding four additional local controllers in a vertical plane dividing the workspace (Fig. 5.8b), reduced the mean and standard deviation of the testing error. Trained on these 12 local controllers, the mean error of the global policy was reduced to 29 ± 2mm.

Next, we investigated how GPS performs when the target is moving. First, we used the same global policy shown in Fig. 5.8a (static training), but instead of a static tester, we used a moving target, i.e., a recorded human reaching motion. The final position of the motion was in a region similar to the one shown in Fig. 5.7.

Similar to our simulation studies, we find that the robot generated highly inefficient trajectories, and sometimes did not even execute these trajectories due to safety constraint violations. A possible way to address this issue is to train the controller with a moving target. To do so, we trained the robot with recorded human reaching motions and tested the policy on another set of recorded human reaching motions, as well as on real-time human handovers. Figs. 5.8c and 5.8d show the performance of the global policy for various final positions of the tester's gripper, defined as in previous trials. Fig. 5.9 (right) shows error distributions.

For the global policy trained with a moving trainer and 8 local controllers (Fig. 5.8c), the mean testing error is 124.3mm. Although the test errors are high as compared to the static tester scenario, the robot stays within the joint and Cartesian limits. Moreover, the variance over the target location is high, and the worst-case error is 791.1mm, 442% higher than the maximum error for the static tester condition (179mm). Surprisingly, the tester's motion for the worst-case error is close to one of the training motions. This can be attributed to the highly non-linear nature of the global policy. Interestingly, GPS does not converge to a low training error for the moving trainer scenario, averaging at 123.2mm which is 544% higher than the training error for a static trainer 22.7mm. Training the global policy with a moving trainer and 12 local controllers (Fig. 5.8d), reduces

the mean testing error to 37.9mm, which is comparable to a static trainer. The worst-case error also improves to 138.7 mm. Fig. 5.9 shows the distributions of training and testing performance for each target scenario.

5.7 Discussion

We evaluated the feasibility of GPS as a learning method for generating robot reaching motions for human-robot handovers. We evaluated GPS first only in simulation, and then on a physical collaborative robot. We used a variant of the GPS algorithm that does not require prior knowledge of the robot dynamics, and instead, learns locally linear dynamics models from the training data [92].

Previously, GPS was used for manipulation tasks in which the targets were static and the variations in target locations were small. To successfully complete a handover, however, the robot must cope with a dynamic environment including unpredictable human motion in a wide range of target locations and objects of different mass. Our evaluation in simulation thus contributes to the design of control policies for human-robot handover tasks by providing a detailed analysis of GPS in terms of three of these requirements: moving targets, large variations in the target location, and a changing end-effector mass.

When evaluating reaching toward static targets only, we find that the performance of the GPS-learned global policy does not generalize well to spatial variations in target locations, and its performance worsens significantly (Fig. 5.2a). The performance of the global policy can be improved by training it with more local controllers (Fig. 5.2a vs Fig. 5.2b). The additional local controllers should be trained with target locations distributed in the regions with high testing er-

rors.

When evaluating the global policy with a moving target which was simulated to mimic human reaching motions, the performance of the global policy decreases on average, but can still achieve reasonable error performance, especially in areas near the training locations (Fig. 5.2a vs Fig. 5.2c).

Similar to the static case, the generalizability of the performance of the global policy can be improved by training it with more local controllers (Fig. 5.2e vs Fig. 5.2f). However, a global policy trained with static targets results in highly inefficient trajectories for moving targets, which are not only high-torque but would be confusing to a human confronted with them. The obvious solution of training the global policy with moving targets is a double-edged sword. It is successful in reducing the mean error and results in more legible and low-torque efficient trajectories, but at the cost of a more high-variance (unreliable) global policy with significantly larger worst-case errors. Further research is required to strike the best balance of trajectory shape, efficiency, and reach error.

In a handover task, the robot end-effector's mass could be different in the training and testing scenarios due to different objects being handed over. We found that the trained global policy adapts well to a range of changes in the robot end-effector's mass. The robot is able to reach the target locations with similar accuracy even with large variations in the end-effector's mass, but only up to a limit as shown in Fig. 5.4. This adaptability could be because our cost function (Eq. 5.11) results in a global policy that is similar to a proportional visual servoing controller, which adapts to changes in robot mass by applying control inputs proportional to the error between the desired position and the current position. Another possible explanation for the invariance of the error

under changes in robot mass could be that changes in the robot's mass do not have a large effect on the robot's trajectory in state-space, and hence, on the performance of the global policy. Contrarily, shifting the target location in the Cartesian space away from the training locations also shifts the robot's trajectory away from the explored region of the robot's state-space, and thus worsens the global policy's performance.

In contrast to prior works on GPS, we also conducted an exploratory study of the effect of different state representations on the performance of GPS. We found that removing the human's joint angles and velocities from the state representation, and expressing the human hand's position and velocity in a reference frame attached to the robot gripper, improved the performance of the trained global policy. This suggests that a low dimensional state-space would be more suitable for GPS, even though it contains less information about the task dynamics.

Our attempts to transfer policies learned on a simulated robot to the physical setup proved infeasible. We found that the robot runs into joint and Cartesian limits and halts almost immediately. This can be attributed to the differences between the simulation model's dynamics and the real robot's dynamics. Tuning the simulation model dynamics parameters to match the real robot's parameters is often not possible owing to a large number of possibilities and undisclosed manufacturer parameter settings.

Though our evaluations in simulation found GPS to be robust to changes in the robot's dynamics within a certain range, our investigations on a real robot showed that GPS is not robust enough to directly transfer learning from simulation on a robot with unknown dynamics to a real robot. That said, a

higher-precision simulation model might allow for better Sim-to-Real transfer and automatically generating such a model could be a fruitful area of future research. Another hurdle for Sim-to-Real transfer of learned GPS policies lies in the prevalence of built-in safety limits found in human-collaborative robots, and the difficulty to impose such limits in the GPS training phase.

When GPS was used to directly train the physical robot, we found that the robot runs into these safety constraints during the training phase. To avoid these violations, we had to reduce the robot's workspace by trial-and-error. In this reduced workspace (Fig. 5.7), we found that most of our results from the simulation environment replicate. When the robot is trained to reach only static target locations, the global policy performance can be improved by adding local controllers in regions with highest test errors (Fig. 5.8a compared to Fig. 5.8b). When evaluating the global policy trained with static targets on a moving test target, the robot generates highly inefficient trajectories, sometimes resulting in halts due to joint limit violations. To overcome this issue, we trained the global policy with moving targets. Nevertheless, this solution is not free of drawbacks. It successfully reduces the mean error and results in more efficient and low-torque trajectories, but also results in a high-variance (and therefore unreliable) global policy with significantly larger worst-case errors. This issue can be addressed by adding local controllers to the training phase, improving the global policy performance (Fig. 5.8d). These findings also support our previous findings in simulation, namely that GPS works best with variable moving targets when the space of possible end-positions is covered densely with the training end-positions.

We hope that our findings can contribute to the understanding of the chal-

lenges and applicability of GPS in a real-world human-robot interaction context. GPS continues to be a promising approach, but to be useful for safety-critical scenarios around humans, it needs to take into account the safety constraints required in such contexts. In addition, while GPS-BADMM is theoretically agnostic to the robot's dynamics, there is a need for highly accurate dynamics models to allow training in simulation, which may often be required. For example, it would be infeasible to have a human repetitively provide training data around a physical robot, requiring good Sim-to-Real performance. These challenges provide a fertile ground for future research.

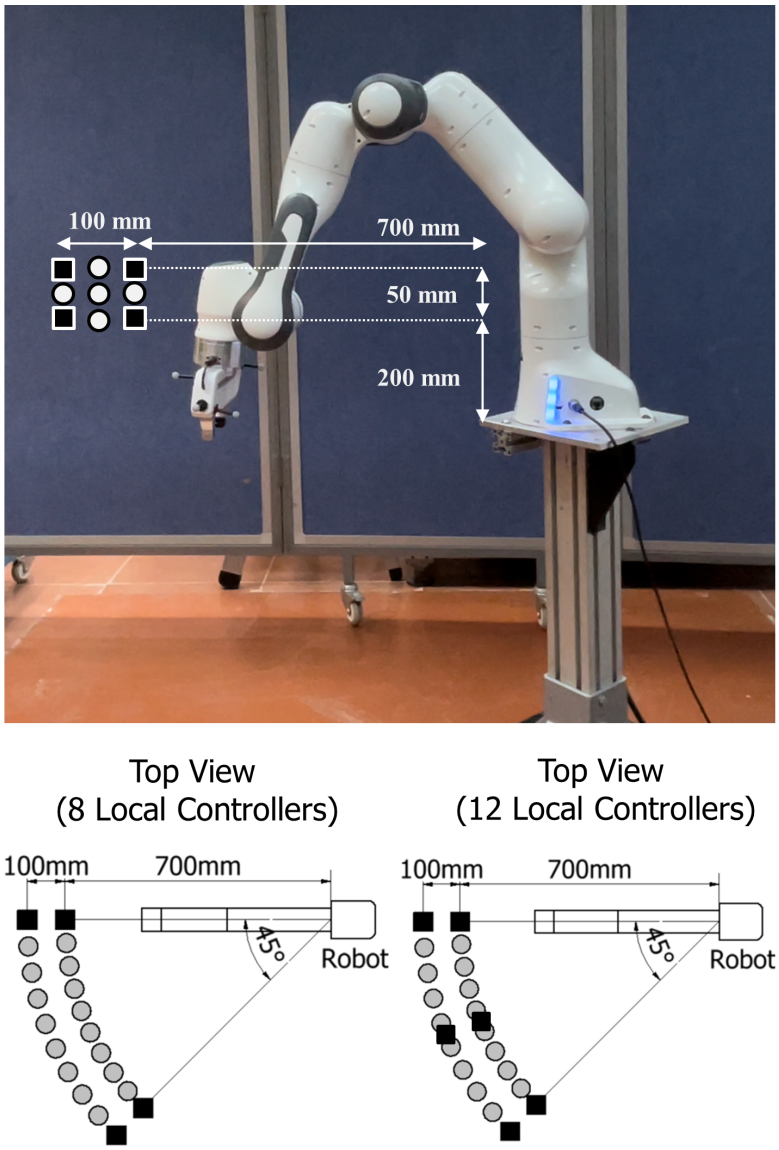
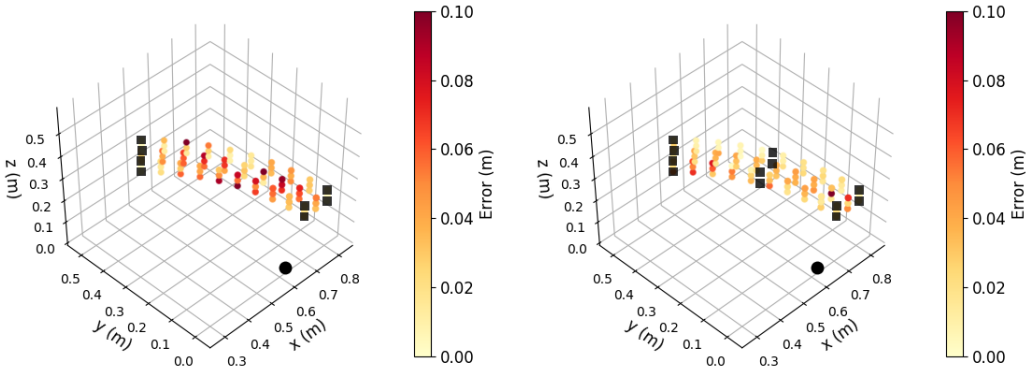
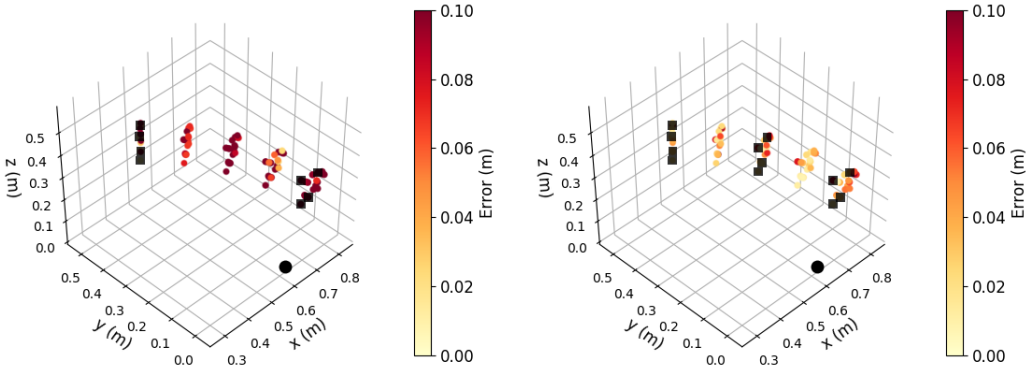


Figure 5.7: The training and testing locations for 8 and 12 local controllers. The squares represent the initial training locations, and the circles represent the testing locations. This region was selected by trial and error to ensure that the robot does not run into joint/Cartesian limits in the training/testing process. For 12 local controllers, additional 4 training locations are located in a vertical plane dividing the workspace.



(a) Static Trainer (8 Local Controllers) Static Tester

(b) Static Trainer (12 Local Controllers) Static Tester



(c) Moving Trainer (8 Local Controllers) Moving Tester

(d) Moving Trainer (12 Local Controllers) Moving Tester

Figure 5.8: Global policy evaluation for different types of trainers and testers. The black circle represents the learner's gripper's initial position, and the black squares represent the training locations. In the 'static' case, the trainer/tester stays in a fixed configuration. In the 'moving' case, the trainer/tester moves with a human-like trajectory (that was recorded in advance) and reaches the locations given by colored dots. Thus, each point corresponds to the final position of the tester's gripper in a trial. The error between the learner's gripper position and the tester's gripper position is calculated over the last time step of each trial. Error-values are clamped to the range [0.0, 0.1] for better visualization.

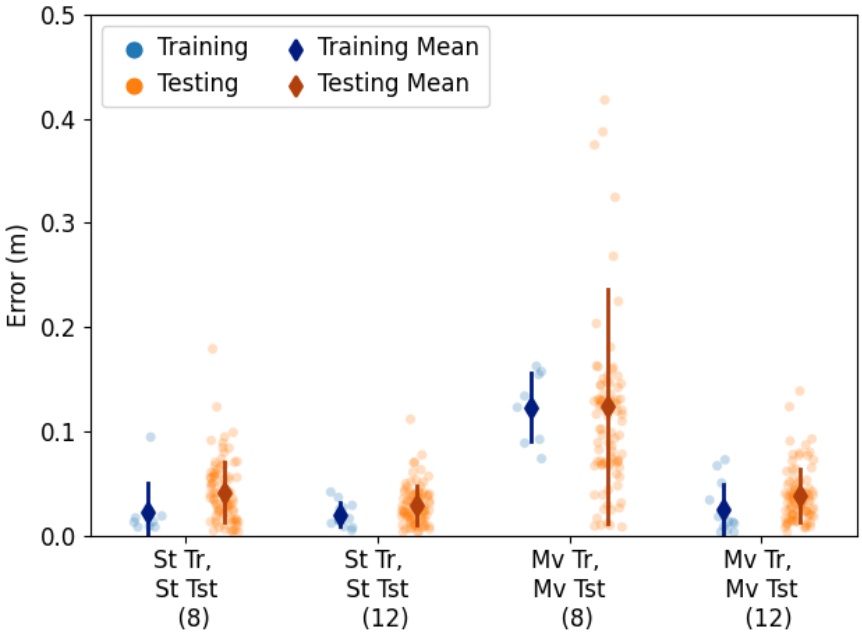


Figure 5.9: Distributions of training and testing performance for each target scenario. Each point is the error between the learner's gripper position and the tester's hand position at the last time step of a trial. Error bars show one standard deviation around the mean of each distribution.

CHAPTER 6

ROBOT GAZE IN HUMAN-TO-ROBOT HANDOVERS

6.1 Introduction

Our second research direction concerns robot gaze behaviors when receiving an object from a human. Past research has shown that the robot's head gaze behaviors affect the subjective experience and timing of robot-to-human object handovers [115, 188, 48, 83] (See Section: 2.3.5). However, all of these works only studied robot-to-human handovers, where the robot was the giver. Human-to-robot handovers, where the robot is the receiver, are equally important with many applications in various domains. Therefore, we investigate robot gaze behaviors in human-to-robot handovers.

First, to find candidates for robot gaze behaviors, we analyze gaze behaviors of human receivers in human-to-human handovers by annotating gaze locations in over 14000 frames of a public data-set of handovers [25]. Our categorized dataset of frame-by-frame gaze labels complements previous studies on giver's gaze behaviors in human-human handovers [115] and provides new insights into the receiver's gaze behaviors.

Second, we implement and compare four human-inspired robot head gaze behaviors during the reach phase of human-to-robot handovers. We conduct two human-to-robot handover studies, a video study, and an in-person study,

A substantial portion of this chapter is taken from [81, 46] ©2019 IEEE. Reprinted, with permission, from A. Kshirsagar, M. Lim, S. Christian and G. Hoffman, "Robot Gaze Behaviors in Human-to-Robot Handovers" IEEE Robotics and Automation Letters, 2020. Reprinted by permission from Springer Nature Customer Service Center GmbH, International Journal of Social Robotics, T. Faibish*, A. Kshirsagar*, G. Hoffman and Y. Edan, "Human Preferences for Robot Eye Gaze in Human-to-Robot Handovers", ©2022.

to compare people's experiences of handovers with these gaze behaviors. In the video study, participants compare videos of a handover from a human actor to a robot exhibiting these four gaze behaviors. In the in-person study, participants physically perform object handovers with the robot and rate their experiences of the handovers for each of the four gaze behaviors of the robot. We find that—for observers of a handover—a transition gaze, in which the robot initially looks at the giver's face and then at the giver's hand, is perceived as more anthropomorphic, likable, and communicative of timing compared to continuously looking at the giver's face or hand. We also find weaker evidence that—for participants in a handover—continuously looking at the giver's face or initially looking at the face and then at the giver's hand is perceived as more anthropomorphic and likable compared to continuously looking at the giver's hand. We do not find any evidence that a robot's head gaze behaviors have an effect on the reach-start time of a giver, defined as the difference between the starting of the robot's and the human's motions, in robot-initiated human-to-robot handovers.

Finally, we compare three human-inspired robot eye gaze patterns across all three phases of the handover process: reach, transfer, and retreat, to extend the above studies which only focused on the reach phase. Additionally, we investigate whether the object's size or fragility or the human's posture affects the human's preferences for the robot gaze. We conduct two sets of user studies (video and in-person), where a collaborative robot exhibits these gaze behaviors while receiving an object from a human. We find that, for both observers and participants in a handover, when the robot exhibits *Face-Hand-Face* gaze (gazing at the giver's face and then at the giver's hand during the reach phase and back at the giver's face during the retreat phase), participants consider the handover to be more likable, anthropomorphic, and communicative of timing

($p < 0.0001$). However, we do not find evidence of any effect of the object's size or fragility or the giver's posture on the gaze preference.

Note on contributions

The second set of studies, which investigated robot eye gaze for handovers of different object types, were carried out in collaboration with Tair Faibish (Master of Science, Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel). Both collaborators had an equal contribution. The study design and results are presented in detail in [45], and an abridged version is included in Sections 6.4.

6.2 Annotations of Gaze Behaviors in Human-Human Handovers

To examine human gaze behaviors in a handover, we performed frame-by-frame video analysis of a public dataset of human-human handovers [25]. The dataset consists of more than 1000 videos of object handovers with 18 volunteers, 10 objects, and several handover scenarios. The handover scenarios vary in terms of experiment type (volunteer-volunteer or volunteer-experimenter), the role of the volunteer (giver or receiver), and starting phase (with approach or without approach). We only considered the volunteer-volunteer handovers as these would be more natural. This gave us a total of 288 videos each for givers and receivers recorded at 8fps and a resolution of 1280×720 pixels. In total, we coded 14214 frames of handover videos. We used video analysis even though

the dataset contained motion capture data for participants' heads, as we found that gaze is often enacted only with the participant's eyes, without noticeable head movement.

We annotated each frame with the following discrete variables {G: Giver, R: Receiver}[1]:

1. **G's gaze:** R's face/R's hand/Own Hand/Other
2. **G's phase:** Reach/Transfer/Retreat
3. **R's gaze:** G's face/G's hand/Own Hand/Other
4. **R's phase:** Reach/Transfer/Retreat

Here, in the "reach" phase the persons move their hands toward the handover location, in the "transfer" phase both persons hold the object, and in the "retreat" phase the persons move their hands back to the rest position. In the videos, it was difficult to distinguish whether a person was looking at the other person's hand or the handover location. In such cases, we assumed that the individual in question was looking at the other person's hand. This choice was also motivated by the intuition that the handover location is not known to the person a-priori.

To validate our video-coding scheme, we had another coder annotate a part of the data consisting of 64 out of 288 videos, which makes for 22.2% of the total data. We found the inter-coder agreement to be 80.9%.

[1]The annotations are available at:
`https://github.com/alapkshirsagar/handover-gaze-annotations/`

6.3 Studies of Robot Gaze in the Reach Phase of Human-to-Robot Handovers

We are interested in designing gaze behaviors that a robot should exhibit when it reaches to receive an object from a human. We conduct two within-subject studies, a video study, and an in-person study, to investigate the likeability, anthropomorphism, and timing communication of four human-inspired robot gaze behaviors in human-to-robot handovers. Our studies seek to answer the research questions: Which human-inspired robot gaze behavior is preferred by an observer of a human-to-robot handover? Which human-inspired robot gaze behavior is preferred by the human giver in a human-to-robot handover?

6.3.1 Gaze Behaviors in Human-Human Reach

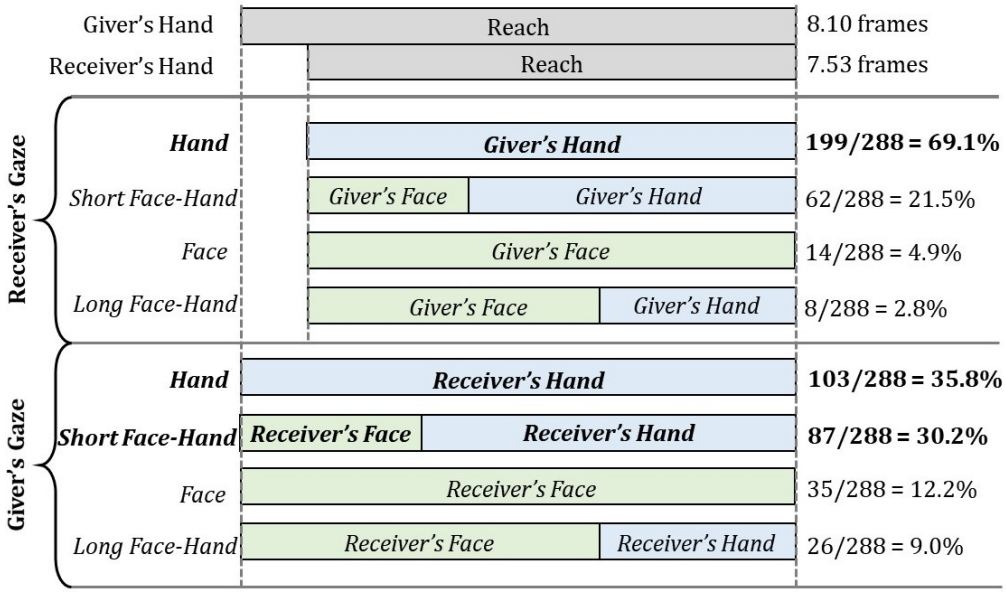


Figure 6.1: Analysis of gaze behaviors in the reach phase of human-human handovers: The most frequent gaze behavior in the reach phase of the handover is *Hand* gaze in which the person continuously looks at the other person's hand.

We analyzed the gaze annotations described in Section 6.2 to find the most frequent gaze behaviors in human-human handovers. We focused on the initial ("reach") phase of the handover, in which both agents extend their hands toward the handover location. Figure 6.1 shows the summary of video annotations for this phase.

We found the following gaze patterns to be the most frequent ones:

1. *Hand gaze*: Continuously look at the other person's hand.
2. *Face gaze*: Continuously look at the other person's face.
3. *Short Face-Hand gaze*: Initially look at the other person's face and then at their hand. The duration of the *Face* gaze is less than or equal to half the duration of the reach phase.
4. *Long Face-Hand gaze*: Initially look at the other person's face and then at their hand. The duration of *Face* gaze is more than half of the duration of the reach phase.

For both givers and receivers, gazing at the other participant's hand was the most frequent behavior. However, this was more pronounced for the receiver's gaze, with almost double the frequency (69.1% vs. 35.8%) for the *Hand* gaze. This initial finding emphasizes the need for a separate study of robot handover gaze in the robot-as-receiver role.

6.3.2 Video Study

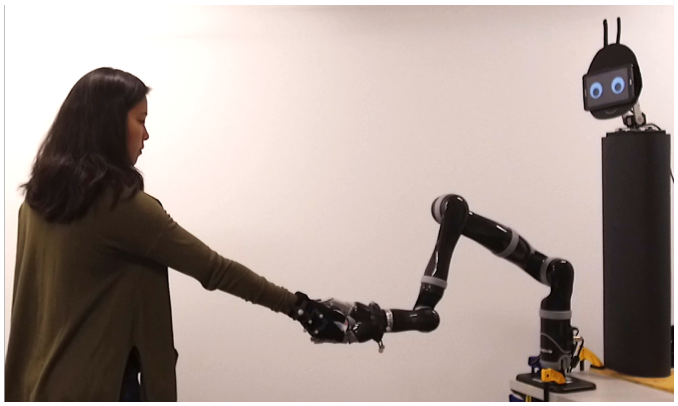
In the first study, participants watched and compared the robot gaze behaviors in videos of human-to-robot handovers. We recorded videos of an actor

handing over an object to a Kinova Jaco-2 robot arm, accompanied by a simple robot head. Fig. 6.2(a) shows a snapshot of a video recording. The robot arm had seven degrees of freedom and a three-fingered gripper. The robot head had four degrees of freedom and a 7-inch screen that displayed a static image of two eyes. We used an OptiTrack motion tracking system to track the positions of the human's hand, the robot gripper, and the object.

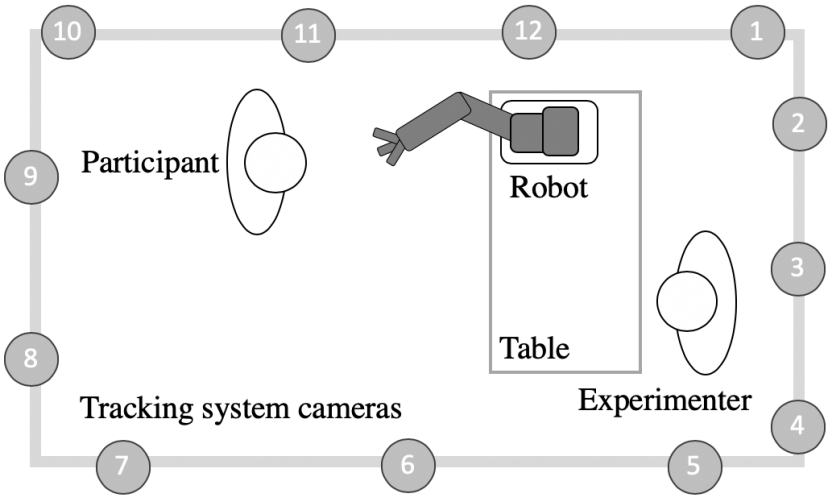
The robot arm was autonomous and programmed to reach a predefined position once the handover began. The robot grasped the object when the object was close enough. Finally, the robot retreated to its home position after the human released the object and started to retreat.

We programmed the head of the robot to exhibit the following four gaze behaviors:

1. *Face gaze*: Continuously look at the giver's face. The face gaze location was programmed manually to a fixed location.
2. *Hand gaze*: Continuously look at the giver's hand.
3. *Short Face-Hand transition gaze*: Initially look at the giver's face and transition to the giver's hand. The transition began as soon as the reach phase started. Thus the *Face* gaze was shorter than the *Hand* gaze.
4. *Long Face-Hand transition gaze*: Initially look at the giver's face and transition to the giver's hand. The transition began as soon as the robot's reach phase ended. Thus the *Face* gaze was longer than the *Hand* gaze.



(a)



(b)

Figure 6.2: The experiment setup consisted of a Kinova Jaco-2 robot arm, a robot head, and an OptiTrack motion tracking system with 12 cameras. (a) shows a video frame of an actor handing over an object to the robot, used in the video study. (b) shows a diagram of the setup for the in-person study.

Procedure

The study was conducted in a laboratory environment and the experimenter left the room after the participant started the study. Participants gave online consent, read the instructions, and then completed a practice session followed by 12 actual study sessions. In each session, they watched two handover videos, one after the other. The 12 sessions consisted of the six possible pairings of the

four gaze patterns and their reverse order. The instructions at the start of the experiment as well as the caption for each video stated that participants should pay close attention to the robot's head movement in the video. After every two videos, they were asked to answer the following questions:

1. Which handover did you like better? (1st or 2nd)
2. Which handover seemed more friendly? (1st or 2nd)
3. Which handover seemed more natural? (1st or 2nd)
4. Which handover seemed more humanlike? (1st or 2nd)
5. Which handover made it easier to tell when, exactly, the robot wanted the giver to give the object? (1st or 2nd)
6. Any other comments (optional)

This questionnaire is identical to the one in Zheng et al. [188]. Questions 1 and 2 measure the metric *likability* (Cronbach's $\alpha = 0.83$). Questions 3 and 4 measure the metric *anthropomorphism* (Cronbach's $\alpha = 0.91$). Question 5 measures the metric *timing communication*. Question 6 is to elicit open-ended responses.

Participants

A total of 24 participants participated in the experiment (13 Males, 11 Females). Participants were recruited through emails and posters. Each study session lasted for about 15 minutes. The participants were compensated with a $5 gift card for participating in the study. The order of the videos was randomized and counterbalanced among the 24 participants.

Preference Ranking Method

Participants evaluated the six pairings of four gaze conditions and their reverse order (for example, *Face* gaze vs. *Hand* gaze and *Hand* gaze vs. *Face* gaze). We used one-sample Wilcoxon signed-rank tests to check if participants exhibited any bias towards selecting first or the second handover. We did not find a bias [likeability: S = -156.5. p = 0.90, anthropomorphism: S = -610, p = 0.63, timing communication: S = 0, p = 1.00].

Table 6.1 shows the number of participants, out of the 24 participants, who chose the row condition over the column condition. For example, a rating of 5 for *anthropomorphism* in row "Face" and column "Hand" indicates that five participants chose *Face* gaze over *Hand* gaze in the pairwise comparisons. a_i represents the number of times row condition i "wins" against other conditions.[2] P_i is the probability that row condition i is preferred over other conditions. We used the iterative estimation algorithm proposed by Hunter [67] to compute the P_i values.

Hypotheses

Both studies tested the same single hypothesis across all three dependent measures, namely that there is a difference in the probability for preference as a consequence of the gaze type, i.e. $P_i \neq P_j \; \forall i \neq j$. We did not have a-priori hypotheses about the order of the conditions.

[2]The ratings for each row were obtained by averaging the ratings for both ordered pairwise comparisons. a_i shows the sum of ratings in each row. Since each subject compared condition i with $g-1$ conditions, where $g=4$ is the number of gaze behaviors, a_i represents the number of "wins" against other conditions in $n \times (g-1)$ comparisons, where n is the number of participants.

Quantitative Results

We used the Bradley-Terry method [16] to evaluate participants' rankings of the likeability, anthropomorphism and timing communication of gaze behaviors.[3] The results are shown in Figure 6.3. As all of the χ^2 values are large ($p < 0.0001$), we conclude that the gaze condition affects ratings. Participants prefer the *Face-Hand transition* gazes more than *Hand* and *Face* gazes. Also, *Face* gaze is the least preferred condition.

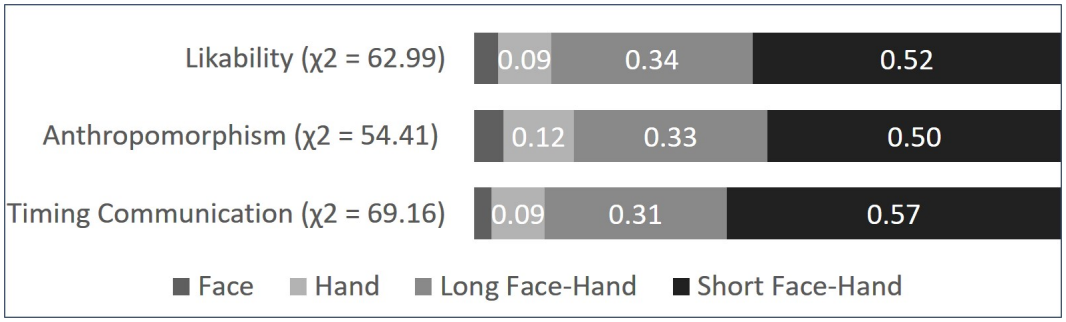


Figure 6.3: χ^2 values and win-probabilities of gaze conditions in the video study for the three dependent measures. (For *Face* gaze, the win-probabilities are 0.04, 0.05, and 0.03 for Likability, Anthropomorphism, and Timing Communication, respectively.)

Open-ended Reponses

Participants were asked to write optional comments after each session. 16 out of 24 participants gave at least one additional comment.

Eight participants made *Face* vs *Hand* gaze comparisons. Four said that they preferred *Hand* over *Face* gaze because it signalled attentiveness, while three said they preferred *Face* gaze because *Hand* gaze signalled shyness:

[3]Following the procedure used by Yamaoka et. al [181], for each metric, the value of $\chi_0^2 = ng(g-1)ln2 - 2Bln10$, where $B = n\sum_{i<j} log(P_i + P_j) - \sum_i a_i log P_i$, should be greater than the $\alpha = 0.005$ point of a χ^2 distribution with $(g-1)$ degrees of freedom.

P010: *"The aparent attentiveness in the second [Hand gaze] was reassurance [sic]."*

P023: *"I felt that in the 2nd one [Hand gaze] it was shy."*

Seven participants implied at least once that they preferred some head movement over no head movement.

P007: *"Some movement of the head is more friendly than no movement."*

P006: *"The lack of a head tilt in the second one seems much more artificial and less human-like."*

Four out of ten participants, who commented on comparing the two transition gazes, mentioned that they could not distinguish between them, while five of them said that they preferred *Short Face-Hand* transition gaze.

P007: *"Both [transition gazes] seemed identical."*

P006: *"The initiation of the head tilt in the second one [Short Face-Hand transition gaze] made it more apparent when the robot wanted to receive the object, first one tilted the head after it was already being handed the object."*

6.3.3 In-person Study

To study people's perceptions of robot gaze behaviors in handovers with a real robot, we ran a second study. A separate set of participants performed object

Table 6.1: Combined preferences of gaze behaviors in the video study. Larger a_i and P_i indicate a stronger preference to the row condition. L-FH = *Long Face-Hand transition* gaze, S-FH = *Short Face-Hand transition* gaze

		Face	Hand	L-FH	S-FH	a_i	P_i
Likability	Face	0	6.75	2.75	2.25	11.75	0.04
	Hand	17.25	0	3.5	4.75	25.5	0.09
	L-FH	21.25	20.5	0	8	49.75	0.34
	S-FH	21.75	19.25	16	0	57	0.52
Anthropomorphism	Face	0	5	3.5	3.25	11.75	0.05
	Hand	19	0	5	4.75	28.75	0.12
	L-FH	20.5	19	0	8.5	48	0.33
	S-FH	20.75	19.25	15.5	0	55.5	0.50
Timing Communication	Face	0	5.5	3	2	10.5	0.03
	Hand	18.5	0	3.5	4	26	0.09
	L-FH	21	20.5	0	7	48.5	0.31
	S-FH	22	20	17	0	59	0.57

handovers with the Kinova Jaco-2 arm, accompanied by the same simple robot head. The setup is shown in Figure 6.2 (b). The robot arm and the robot head were programmed in the same way as the video study described in Section 6.3.2.

Procedure

After entering the lab, participants electronically signed the consent form and answered a question about their familiarity with a collaborative robot such as the one shown. The mean familiarity with this type of robot was found to be low (M=1.29, SD = 0.55, on a scale of 1-5).

The experimenter then verbally described the experiment and participants completed a practice session. This was followed by 12 study sessions. In each session, the participants performed two handovers with the robot, sequentially. The object being handed over was a partially filled water bottle weighing about

200g. The 12 sessions consisted of the six possible pairings of the four gaze patterns and their reverse order. After every session, they were asked to answer the same choice questions as in the video study.

Participants

A total of 24 participants, different from the participants in the video study, participated in the experiments (6 Males, 18 Females). Participants were recruited through an online recruitment system. Each study session lasted for about 15 minutes. The participants were compensated with a $5 gift card. The order of the conditions was randomized and counterbalanced.

Metrics and Hypotheses

We used the same preference ranking method as in Study 1, described in Section 6.3.2.

Table 6.2 shows the number of participants, out of the 24 participants, who chose the row condition over the column condition. We had the same three hypotheses about overall differences in preferences caused by gaze manipulation.

In addition, in Study 2, we measured the human's *reach start time* in each handover, defined as the difference between the start of the human hand's reaching motion and the start of the robot arm's reaching motion. Previous studies on robot-to-human handovers found that the robot's gaze behavior affected the human's reach start time [115, 188]. We were interested to see if the robot's gaze behavior had a similar effect in a human-to-robot handover.

Table 6.2: Combined preferences of gaze behaviors in the in-person study. Larger a_i and P_i indicate a stronger preference to the row condition. L-FH = *Long Face-Hand transition* gaze, S-FH = *Short Face-Hand transition* gaze

		Face	Hand	L-FH	S-FH	a_i	P_i
Likability	Face	0	15	14.5	15.75	45.25	0.35
	Hand	9	0	7	7.75	23.75	0.14
	L-FH	9.5	17	0	13	39.5	0.28
	S-FH	8.25	16.25	11	0	35.5	0.23
Anthropomorphism	Face	0	14.25	13	15	42.25	0.31
	Hand	9.75	0	6.75	7.5	24	0.14
	L-FH	11	17.25	0	13.25	41.5	0.30
	S-FH	9	16.5	10.75	0	36.25	0.24
Timing Communication	Face	0	11.5	11.5	13.5	36.5	0.25
	Hand	12.5	0	7	8.5	28	0.17
	L-FH	12.5	17	0	13	42.5	0.32
	S-FH	10.5	15.5	11	0	37	0.25

Quantitative Results

We used one-sample Wilcoxon signed-rank tests to check if participants exhibited any bias towards selecting the first or the second handover. We did not find a bias [likeability: S = -1481, p = 0.25, anthropomorphism: S = -684, p = 0.59, timing communication: S = -289, p = 0.81]. We conducted a one-way ANOVA on the human's *reach start time* for four gaze conditions. The results indicated no detectable difference between the four conditions ($F = 0.429$, $p = 0.732$).

We used the Bradley-Terry method [16] to evaluate participants' rankings of the likeability, anthropomorphism, and timing communication of gaze behaviors. The results are shown in Figure 6.4. Compared to Study 1, we detected a weaker effect of condition on likability and anthropomorphism, with $p = 0.01$ and $p = 0.03$ respectively, and virtually no effect on timing communication ($p = 0.21$). Participants rated all dependent measures lower in the *Hand*

gaze condition than in other conditions.

Figure 6.4: χ^2 values and ratings of gaze conditions in the in-person study for the three dependent measures.

Open-ended Responses

20 out of 24 participants wrote at least one additional comment. In their comments, contrary to the comments in our video study, nine out of 14 participants said that they preferred the Face gaze over the Hand gaze, as it was more engaging, human-like, and friendly.

> P008 - *"With a task this simple, maintaining eye contact feels more engaged/natural than looking at the object."*

> P016 - *"In the second one [Face gaze], the robot looked me in the eye and seemed more humanlike."*

Six participants implied at least once that they preferred some head movement over no movement.

> P014 - *"No movement at all with 2nd one [Hand gaze] made it less humanlike."*

P023 - *"The ones where the head moves seem more natural and humanlike."*

Six out of the 12 participants who gave comments on the comparison between two transition gazes said that they could not distinguish between the two transition gazes, while four said that they preferred the *Long Face-Hand transition* gaze.

P008 - *"Both tries felt roughly the same to me."*

P016 - *"The robot head appeared to move more naturally in the second handover [Long Face-Hand transition gaze]."*

6.3.4 Discussion

Participants in a handover combine two types of gaze behavior: a task-oriented gaze toward the hand or the object, and a face-oriented gaze for social engagement. Prior studies with robots as givers showed that *Face* gaze is beneficial as it causes more positive evaluations of the robot. Unfortunately, there is not much prior literature to draw on with respect to receiver gaze in handovers, whether the receiver is a human or a robot. In our analysis of human-human handovers, most receivers used *Hand* gaze over other types of gazes, at almost double the rate as givers. This might indicate a stronger need for receivers to keep their gaze focused on the task, sacrificing the social benefits of eye contact. When the receiver is a robot, our studies paint a different picture: Task-oriented *Hand* gaze was consistently ranked as less preferred, especially when compared to a behavior that shifted from *Face* gaze to *Hand* gaze.

What about consistent face gazing? Comparing our video and in-person studies reveals different results for the preference of the *Face* gaze. When participants watched recorded videos of a human handing over an object to a robot, they preferred the *Hand* gaze over the *Face* gaze. Open-ended responses suggested that looking at the hand signaled that the robot was more attentive. In contrast, when the participants physically performed the object handover with the robot, the *Face* gaze dominated the *Hand* gaze. Open-ended responses suggested that looking at the face was a friendly gesture. This is in line with previous research that shows that people have different preferences when they are interacting with a robot in the first person versus when they are observing video interactions of a third person with the same robot [17]. However, these previous studies showed mostly a change in degree, not a categorical shift as found here.

In the video study, participants preferred the *Face-Hand transition* gaze behaviors to consistent *Hand* or *Face* gaze behaviors, citing concerns that the lack of movement was unnatural. Despite similar concerns in the in-person study, participants' preference ratings for *Face-Hand transition* gaze and *Face* gaze were similar. All of them were higher than the *Hand* gaze, indicating that the social benefits of the *Face* gaze—especially in terms of likability—overcome some of the concerns associated with lack of movement.

Combining the findings of the two studies, our recommendation to HRI designers is to implement a *Face-Hand transition* gaze when a robot is receiving objects from a human. There seems to be no preference for the timing of the transition. The least recommended behavior is a static *Hand* gaze, even though it is the most common behavior of a human receiver.

A human receiver may have to focus on the receiving task more than a robot

receiver who can be equipped with a more flexible sensor system. The robot should take advantage of this flexibility and incorporate the beneficial *Face* gaze behavior. If the robot does not have an actuated gaze, a *Face* gaze is preferred over a *Hand* gaze.

In previous studies [115, 188], researchers found that the robot's gaze behavior had an effect on the reach start time of the human in robot-to-human handovers. However, we did not find a similar effect of the robot's gaze behavior on the human's reach start time in human-to-robot handovers. A possible reason is that the repetitive handovers caused learning or fatigue, affecting the reaching behavior. However, the same prior works that found timing differences across gaze conditions also used repeated trials [115, 188]. This may suggest that, while the robot gaze could serve as a timing cue in robot-to-human handovers, it might not be an effective timing cue in human-to-robot handovers.

In contrast to prior works, which conducted in-person studies, we conducted a video study alongside an in-person study. Video studies offer more manipulation control and thus more internal validity, whereas in-person studies have more ecological validity. We used a different set of participants to prevent order effects. We did not have specific hypotheses that warranted a quantitative comparison of the data from the two studies, and thus presented the difference only qualitatively.

6.4 Studies of Robot Gaze in Human-to-Robot Handovers

Our studies in Section 6.3 were limited to a specific scenario where the giver stood in front of the robot and handed over a specific object (a plastic bottle) to

the robot. We conduct follow-up studies to expand and generalize the findings from our previous studies in three ways:

1. We investigate if the object's size (small/large), stiffness (fragile/non-fragile), and the giver's posture (standing/sitting) affect the human's preferences for robot eye gaze in human-to-robot handovers.
2. We consider gaze behavior in all three phases of a handover instead of just the reach phase.
3. We use robot eye gazes instead of head gazes, as eye gazes are more prominent than head gazes in natural human-human handovers

6.4.1 Hypotheses

We hypothesize that:

- **H1**: People prefer certain robot gaze behaviors over others in terms of likability, anthropomorphism, and timing communication.
- **H2**: Object size affects the user's ratings of the robot's gaze in a human-to-robot handover.
- **H3**: Object fragility affects the user's ratings of the robot's gaze in a human-to-robot handover.
- **H4**: The user's posture (standing and sitting) affects the user's ratings of the robot's gaze in a human-to-robot handover.
- **H5**: Observers of a handover and participants in a handover have different preference ratings of the robot's gaze in a human-to-robot handover.

H1 is motivated by our previous studies which found evidence for different user preference ratings for robot gaze behaviors. We do not have a-priori hypothesis about the preference order of gaze behaviors. **H2** and **H3** are based on the intuition that the object's size and fragility could affect the preferred gaze behavior of a receiver. For example, when receiving large or fragile objects, the robot could be expected to convey attentiveness by looking at the giver's hand, whereas, when receiving small or non-fragile objects, the robot could be better off looking at the giver's face to convey friendliness. **H4** is based on the intuition that a standing giver may have different preferred gaze behavior of a receiver than a sitting giver. For example, a standing person could like the robot gaze at their face as their eyes are at the same level, whereas a sitting person could feel uncomfortable with the robot gazing down at their face. **H5** results from our previous finding that observers of a handover and participants in a handover had different preference ratings of robot gaze behaviors in the reach phase. This research examines whether this holds for robot gaze behaviors in all three phases of a handover and handovers with different object types and giver postures.

6.4.2 Gaze Behaviors in Human-Human Handovers

To design human-inspired robot gaze behaviors for human-to-robot handovers, we extend our previous analysis of human gaze behaviors in the reach phase of a handover (Section 6.3.1) to the three phases of a handover: reach, transfer, and retreat. As shown in Fig. 6.5, we find that the most common gaze behaviors employed by people during handovers are:

1. *Hand-Face* gaze: The person continuously looks at the other person's hand during the reach and the transfer phases, and then looks at the other person's face during the retreat phase. The transition from hand to face happens slightly after the beginning of the retreat phase. More than 50% of receivers showed this behavior, whereas, only 25% of the givers in those videos exhibited this behavior.
2. *Face-Hand-Face* gaze: During the reach phase, the person initially looks at the other person's face and then at the other person's hand. They then continue looking at the other person's hand during the transfer phase. Finally, they look at the other person's face during the retreat phase. The transition from face to hand occurs halfway through the reach phase, while the transition from hand to face occurs halfway through the retreat phase. More than 40% of givers exhibited this gaze, whereas only 25% of receivers did.
3. *Hand* gaze: Continuously looks at the other person's hand. The least frequent gaze, only 17.4% of receivers and 15.9% of givers showed this behavior.

6.4.3 Overview of Studies

We conducted two within-subject studies: a video study and an in-person study. The video study aimed to investigate an observer's preferences of robot gaze behaviors, whereas the in-person study aimed to investigate a giver's preferences of robot gaze behaviors.

A total of 144 undergraduate industrial engineering students participated in the experiment (72 in each study) and were compensated with one bonus

point to their grade in a course for their participation. The average participation time was about 25 minutes. In the video study, there were 34 females and 38 males aged 23-29. In the in-person study, there were 36 females and 36 males aged 23-30. The study design was approved by the Human Subjects Research Committee at the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev.

The following three gaze behaviors were implemented on a Sawyer cobot based on insights from the human-human handover analyses:
i. *Hand-Face* gaze: The robot's eyes continuously looked in the direction of the giver's hand during the reach and transfer phases. After the robot started to retreat, the eyes transitioned to look at the giver's face. Both the hand gaze and the face gaze were programmed manually to fixed locations.
ii. *Face-Hand-Face* gaze: The robot's eyes looked at the giver's face during the reach phase, the giver's hand during the transfer phase, and the giver's face during the retreat phase.
iii. *Hand* gaze: The robot's eyes continuously looked in the direction of the giver's hand.

Given that the human gaze behavior was tied to the handover phase, as described above, we did not use fixed timings for the robot trajectory. Instead, the robot was programmed to use the sensor information to initiate the handovers and gaze behaviors depending on the phase of the handover. The robot arm was programmed to reach a predefined position once the giver started the handover which was detected using a range sensor. The robot's gripper was equipped with an infrared proximity sensor, and it grasped the object when the object was close enough. The robot retreated to its home position after grasping the

object. The robot was programmed in the Robot Operating System (ROS) environment with Rethink Robotics' Intera software development kit (SDK). The sensors were interfaced with the robot using an Arduino micro-controller.

Fig. 6.6(a) shows a snapshot of a video recording illustrating the experimental setup.[4]

6.4.4 Video Study of Human-to-Robot Handovers

Experiment Procedure and Evaluation

The study was conducted remotely, and each participant received links to the videos, the electronic consent form, and online questionnaires with study instructions. After signing the consent form and reading the instructions, they completed a practice session followed by 12 study sessions. Each session included one of the six pairing of the gaze patterns listed in Table 6.4, for a single condition out of the three listed in Table 6.3[5]. So that each participant watched all six pairs of gaze patterns twice, one for condition a and one for condition

[4]The videos are available at: https://youtu.be/9dD1YHG2Nco

[5]To represent objects of different fragility a plastic bottle and a glass bottle were used. In order to examine people's perceptions about the fragility of these objects, we conducted an online survey. This survey was conducted post-experiment. A total of 24 participants responded to the survey. The participants were undergraduate students from the Department of Industrial Engineering and Management at Ben-Gurion University, similar to the students who participated in our video and in-person experiments. The participants were told that this study deals with object handovers between a human and a robot.

The survey included 10 pictures of objects, made from different materials. The plastic bottle and the glass bottle used in our experiment were among these objects. Each picture was followed by a yes or no question: "Do you perceive this object to be fragile?". Results revealed that all of the 24 participants perceived the plastic bottle to be non-fragile. 23 out of 24 participants perceived the glass bottle to be fragile. Additionally, when asked the same question for three other different plastic and glass bottles, 24 participants denoted the plastic bottles as non-fragile and 23 denoted the glass bottles as fragile. Details about this survey are available in [45]. This supports our decision to choose plastic and glass bottles to represent objects of different fragility.

b. To reduce the recency effect, i.e. participants forgetting the previous conditions, counterbalanced pairwise comparisons were performed instead of three-way comparisons. All six pairwise comparisons were combined into a ranked ordered list of three gaze patterns [67]. In each session, they watched two handover videos, consecutively.The different objects and postures used in the experiment are shown in Fig. 6.7 and Fig. 6.6 respectively.

The instructions at the start of the experiment, as well as the caption for each video, stated that participants should pay close attention to the robot's eyes in the video. After every two videos, the participants were asked to fill out a questionnaire that collected subjective measures as detailed below. The questionnaire was identical to the one used in our previous study (Section 6.3) and in Zheng et al.'s study [188]. Questions 1 and 2 measure the metric *likability* (Cronbach's $\alpha = 0.83$). Questions 3 and 4 measure the metric *anthropomorphism* (Cronbach's $\alpha = 0.91$). Question 5 measures the metric *timing communication*.

1) Which handover did you like better? (1st or 2nd)
2) Which handover seemed more friendly? (1st or 2nd)
3) Which handover seemed more natural? (1st or 2nd)
4) Which handover seemed more humanlike? (1st or 2nd)
5) Which handover made it easier to tell when, exactly, the robot wanted the giver to give the object? (1st or 2nd)
6) Any other comments (optional)

Table 6.3: Study Conditions (24 participants per condition).

Condition 1: Object Size	a. Small Box b. Large Box
Condition 2: Object Fragility	a. Plastic Bottle b. Glass Bottle
Condition 3: User's Posture	a. Sitting b. Standing

Table 6.4: Six pairings of the three gaze patterns and their reverse order for each object or posture. Each participant experienced two versions (a/b of a single condition) of these pairings, for a total of 12 pairings.

First Handover	Second Handover
hand-face	face-hand-face
hand-face	hand
face-hand-face	hand
face-hand-face	hand-face
hand	hand-face
hand	face-hand-face

Experiment Design

The experiment was designed as a between-within experiment, using likability, anthropomorphism, and timing communication as the dependent variables. The participants were divided into three groups of 24 participants. Each group performed one of the three study conditions listed in Table 6.3. The order of the 12 sessions was randomized and counterbalanced among the subjects.

Analysis

The participants' ratings for the likability and anthropomorphism of the gaze behaviors were measured by averaging their responses to Questions 1-2 and 3-4 respectively. The one-sample Wilcoxon signed-rank test was used to check

if participants exhibited any bias towards selecting the first or the second handover. Similar to our previous studies (Section 6.3) and Zheng et. al's work [188], the Bradley-Terry model [16] was used to evaluate participants' rankings of the likeability, anthropomorphism, and timing communication of gaze behaviors. To evaluate the hypothesis **H1**, i.e. $P_i \neq P_j \forall i \neq j$, where P_i is the probability that one gaze condition is preferred over others, the χ^2 values for each metric were computed, as proposed by Yamaoka et. al [181]:

$$B = n \sum_{i<j} log(P_i + P_j) - \sum_i a_i log P_i, \tag{6.1}$$

$$\chi^2 = ng(g-1)ln2 - 2Bln10, \tag{6.2}$$

where, $g = 3$ is the number of gaze behaviors, n is the number of participants, a_i is the sum of ratings in each row of Tables B.1-B.5 (Appendix).

In order to examine **H2-H4**, we conducted two series of tests for each measured metric (likability, anthropomorphism and timing communication), and for each study scenario:

- Binary proportion difference tests for matched pairs [108], in which the difference between the proportion of participants who chose one gaze condition p_b over other p_c was evaluated in each study scenario. The distribution of differences $p_b - p_c$ is:

  $$p_b - p_c \sim \mathcal{N}(0, \sqrt{\frac{p_b + p_c - (p_b - p_c)^2}{n}}) \tag{6.3}$$

  where $n = 24$ is the number of participants in each scenario. The z-score is calculated according to the following formula:

  $$Z = \frac{(p_b - p_c)}{\sqrt{var(p_b - p_c)}} \tag{6.4}$$

A low Z-score means that the distribution of differences has zero mean with high probability.

- Equivalence tests based on McNemar's test for matched proportions [109, 116], in which the proportion of participants who changed their gaze preferences in each study scenario was compared within equivalence bounds of $\triangle = \pm 0.1$.

Quantitative Results

To test for order effects, we checked, but did not find any bias towards selecting the first or the second handover [like: z =-0.68, p = 0.50; friendly: z = 1.22, p = 0.22; natural: z =0.20, p = 0.84; humanlike: z = 1.36, p = 0.17; timing communication: z =1.23, p = 0.22].

Tables B.1 - B.3 (Appendix) and Fig. 6.8-6.10 show the robot gaze preferences of the participants in terms of likability, anthropomorphism and timing communication.

Gaze conditions differ significantly in ratings (all χ^2 values are large $(p < 0.0001)$), supporting **H1**. Participants prefer the *Face-Hand-Face* transition gazes over *Hand-Face* and *Hand* gazes. *Hand* gaze is the least preferred condition.

Based on the binary proportion difference test, we did not find evidence that the proportion of observers of a handover preferring one gaze condition over the other is affected by object size (Table B.7, Appendix), object fragility (Table B.8, Appendix) and user's posture (Table B.9, Appendix). Hypotheses **H2**, **H3** and **H4** are not supported (all p values are over 0.2).

However, based on the equivalence tests, we did not find evidence that the proportion of observers of a handover preferring one gaze condition over the other is equivalent for the two object sizes (Table B.7, Appendix), object fragilities (Table B.8, Appendix), or user's postures (Table B.9, Appendix). Thus, hypotheses **H2**, **H3** and **H4** can also not be rejected (all p values are over 0.15).

Open-ended Responses

All open-ended responses are presented in [45] with major insights detailed below.

10 out of 72 participants gave at least one additional comment. Four out of the eight participants, who made *Hand-Face* gaze vs. *Face-Hand-Face* gaze comparisons, preferred *Face-Hand-Face* gaze over *Hand-Face* gaze due to the extended eye contact by the robot.

> P059 - *"As much eye contact as possible."*
> P048 - *"I preferred handover 2 (Face-Hand-Face gaze) because the robot looked more at the human"*

Two participants mentioned that they could not distinguish between *Face-Hand-Face* gaze and *Hand-Face* gaze, while two participants commented on the advantages and disadvantages of the two gaze patterns.

> P041 - *"In handover 1 (Hand-Face gaze) you could tell that the robot was ready to receive the object. However, handover 2 (Face-Hand-Face gaze) felt more humanized because the robot looked at the giver's eyes right until the transfer was made."*

Four out of six participants, who commented on the comparison between *Hand-Face* gaze and *Hand* gaze, preferred *Hand-Face* gaze because of the eye movement.

> P008 - *"In my opinion, the change in eye movement creates a better human-robot interaction."*
> P009 - *"In the second handover (Hand-Face gaze) the eye movement, gave a good indication for the communication."*

Two participants mentioned that they could not distinguish between *Hand-Face* gaze and *Hand* gaze.

Six participants commented on *Face-Hand-Face* gaze vs. *Hand* gaze comparison. All of them said that they preferred *Face-Hand-Face* gaze over *Hand* gaze.

> P009 - *"At handover 2 (Face-Hand-Face gaze), the robot looked at the object precisely when it wanted to take it, so it was perceived more understandable."*
> P037 - *"In my opinion video 2 (Face-Hand-Face gaze) best simulated human-like behavior out of all the videos I have seen so far."*

6.4.5 In-person Study of Human-to-Robot Handovers

In the in-person study, another set of 72 participants was asked to perform object handovers with the Sawyer robot arm in a similar setup (Figure 6.6(c)). The robot arm and the robot eyes were programmed in the same way as the video study described in Section 6.3.2.

Experiment Procedure, Design and Evaluation

The experiment was conducted during the COVID-19 pandemic. Therefore, several precautions were taken. The participants were asked to wash their hands with soap when they entered and exited the lab. The equipment was sterilized before and after each participant, and the experiment room's door remained open at all times. Only one participant was allowed at a time inside the room. Both the participant and conductor of the experiment wore masks and kept at least 2 meters distance between them.

After entering the experiment room, participants signed the electronic consent form, and answered a question on a computer: *How familiar are you with a collaborative robot (such as the one shown)?* Participants ranked this question on a scale from 1 – *"Not at all familiar"* to 5 – *"Extremely familiar"*. The mean familiarity with this type of robot was found to be low (M=1.49, SD = 0.60, on a scale of 1-5).

The study instructions were given orally by the experimenter. Participants then completed a practice session followed by 12 randomly assigned study sessions. In each session, the participants performed two sequential handovers with the robot. The 12 sessions consisted of the same pairings of gaze behaviors as in the video experiment, followed by the same questionnaire questions. The only difference was in Question 5, which was "Which handover made it easier to tell when, exactly, the robot wanted you to give the object? (1st or 2nd)". The experimental design was also the same as the video study.

Analysis

The hypotheses **H1-H4** were evaluated using the same procedure as described in Section 6.4.4.

To evaluate hypothesis **H5**, we conducted two series of tests for each measured metric (likability, anthropomorphism, and timing communication), and for each study scenario. These tests are different from the tests for "matched pairs" which we performed for testing **H2-H4**, since for testing **H5** we need to compare two different participants' groups:

- Binary proportion difference tests for unmatched pairs [69], in which the difference between the proportion of participants who chose one gaze condition over the other in each study scenario for the in-person p_b and video p_c studies was evaluated. The distribution for the differences $p_b - p_c$ is:

$$p_b - p_c \sim \mathcal{N}(0,\ \sqrt{p_d(1-p_d)(\frac{1}{n_b} - \frac{1}{n_c})} \tag{6.5}$$

  where $n_b = 24$ and $n_c = 24$ are the number of participants in each scenario of the in-person study and video study respectively, and p_d is the pooled proportion calculated as follows:

$$p_d = \frac{X_b + X_c}{n_b + n_c} \tag{6.6}$$

  where X_b and X_c are the number of participants who preferred one gaze condition over the other (shown in Tables B.1 - B.6, Appendix) in the in-person and video study respectively. Then, the Z-score is calculated same as Equation 6.4.

- Equivalence tests for unmatched proportions [85], in which the proportion of participants who chose one gaze condition over the other in each study

scenario for the in-person p_b and video p_c studies was tested for equivalence within the bounds of $\triangle = \pm 0.1$.

Quantitative Results

There was no bias towards selecting the first or the second handover [like: z =-0.88, p = 0.38; friendly: z = -0.27, p = 0.79; natural: z =-0.48, p = 0.63; human-like: z = -1.16, p = 0.25; timing communication: z =0.34, p = 0.73]. Tables B.4-B.6 (Appendix) and Fig. 6.11- 6.13 show the robot gaze preferences of the participants in terms of likability, anthropomorphism, and timing communication. In all six experimental conditions, the gaze conditions differ significantly in ratings ($p < 0.0001$), supporting **H1**. As in the video study, participants preferred the *Face-Hand-Face transition* gazes over *Hand-Face* and *Hand* gazes. *Hand* gaze was the least preferred $(p < 0.0001)$.

Based on the binary proportion difference test, the proportion of participants in a handover preferring one gaze condition over other can not be claimed to be affected by object size (Table B.7, Appendix), object fragility (Table B.8, Appendix) and user's posture (Table B.9, Appendix), contradicting hypotheses **H2**, **H3** and **H4**. The proportion of participants in a handover preferring one gaze condition over other (Table B.10, B.11, B.12 Appendix) also cannot be claimed to be affected by the interaction modality (video or in-person), contradicting **H5**.

However, based on the equivalence tests, we did not find evidence that the proportion of participants in a handover preferring one gaze condition over the other is equivalent for the two object sizes (Table B.7, Appendix), object fragilities (Table B.8, Appendix), or user's postures (Table B.9, Appendix). Thus, hy-

potheses **H2**, **H3** and **H4** can also not be rejected (all p values are over 0.15). We also did not find evidence that the proportion of participants in a handover preferring one gaze condition over other (Table B.10, B.11, B.12 Appendix) is equivalent for the two interaction modalities (video or in-person). Thus hypothesis **H5** can also not be rejected.

Open-ended Responses

14 out of 72 participants gave additional comments.

Seven participants made *Hand-Face* gaze vs. *Face-Hand-Face* gaze comparisons. Two of these participants stated that they preferred *Face-Hand-Face* over *Hand-Face* gaze because they preferred longer eye contact from the robot.

> P020 - *"I preferred handover 1 (Face-Hand-Face gaze) because the robot stared at me before and after the handover, and I felt accompanied by it during the entire handover."*

Four participants mentioned that they could not distinguish between the two conditions, while one participant mentioned that *Face-Hand-Face* gaze pattern didn't feel natural.

Four out of the seven participants who commented on the comparison between *Hand-Face* gaze and *Hand* gaze, said that they preferred *Hand-Face* gaze.

> P014 - *"In the first handover (Hand-Face gaze) the robot looked straight at me after the handover and seemed to be more friendly."*
> P050 - *"In the first handover (Hand-Face gaze), the robot's eye movement*

was fully accompanied by the handover movement, and therefore it seemed more natural."

Three participants mentioned that they could not distinguish between *Hand-Face* gaze and *Hand* gaze.

Seven out of eight participants, who commented on the comparison between *Face-Hand-Face* gaze and *Hand* gaze gazes, said that they preferred *Face-Hand-Face* gaze over *Hand* gaze because of the longer eye contact by the robot.

P014 - *"In the first handover (Hand gaze), the robot focused only on the object, and in the second handover (Face-Hand-Face gaze) it focused on me too, so it felt more natural."*
P016 - *"I preferred the second handover (Face-Hand-Face gaze) mainly because the robot looked me in the eyes at the beginning and the end."*

6.4.6 Discussion

Prior works studying robot gaze in handovers did so either for a robot as the giver, or—in our prior studies on robot receiver gaze—for a small and rigid object, and one specific posture of the human. However, for a robot receiver, the object type or giver posture might influence preferences of robot gaze behavior. This raises the question of whether the findings in the prior work generalize over variations in the handover task. In this work, we investigated the effect of different object types and giver postures on preferred robot gaze behavior in a human-to-robot handover. We did not find evidence that the participants' gaze preference for a robot receiver in a handover is affected by small, large, fragile,

and non-fragile objects, standing or sitting postures, and the interaction modality i.e. video or in-person. However, in our study, the proportion of participants preferring one gaze condition over the other is not statistically equivalent. Thus we cannot completely reject the effect of these scenarios on gaze preferences. In addition, previously we studied the robot receiver's gaze behaviors only in the reach phase of human-to-robot handovers. The studies presented in this section **extend the empirical evidence by studying the gaze patterns for all three phases of the handover: reach, transfer, and retreat.**

As in our previous study (Section 6.3), results revealed that **the most preferred gaze behavior for a robot receiver was different from the observed most frequent behavior of a human receiver**. When a person receives an object from another person, the most frequent gaze behavior is a *Hand-Face* gaze, in which the receiver looks at the giver's hand throughout the reach and transfer phases, and then at the giver's face in the retreat phase. This indicates that receivers must keep their gaze focused on the task and thus sacrifice the social benefits of the face gaze. Our previous studies had revealed that a robot receiver can utilize the flexibility of its perception system to incorporate a face-oriented gaze for social engagement. This finding is reinforced by our current study as the participants preferred a *Face-Hand-Face* transition gaze behavior, in which, the robot initially looked at their face, then transitioned its gaze to their hand during the reach phase, continued to look at their hand during the transfer phase, and finally transitioned its gaze back to again look at their face during the retreat phase. Open-ended responses suggested that **people preferred the robot looking at their face at the beginning and the end of the handover, and the robot's eyes following the object during the transfer phase**. This gaze behavior complemented the robot's handover motion, and thus portrayed the robot

as more human-like, natural, and friendly. Another possible explanation is that the social aspects of a human receiver are implicit, whereas a robot has to establish its social agency for a better handover experience. Based on these findings, **we recommend to HRI designers to implement a *Face-Hand-Face* transition gaze when the robot receives an object from a human, regardless of human posture and characteristics of the object being handed over.**

There are several limitations of this study that could motivate future work. The results are limited by the sample size and the specific cultural and demographic makeup of its participants. Larger population samples of different age groups, backgrounds, and cultures should be investigated to help generalize the findings of our experiments. Moreover, as with any experimental study, there is a question of external validity. A handover that is part of a more complex collaborative or assistive task might elicit different expectations of the robot's gaze, a fact that should be considered by designers of HRI systems. To better understand these contextual requirements, additional realistic scenarios of assistive and collaborative tasks should be considered.

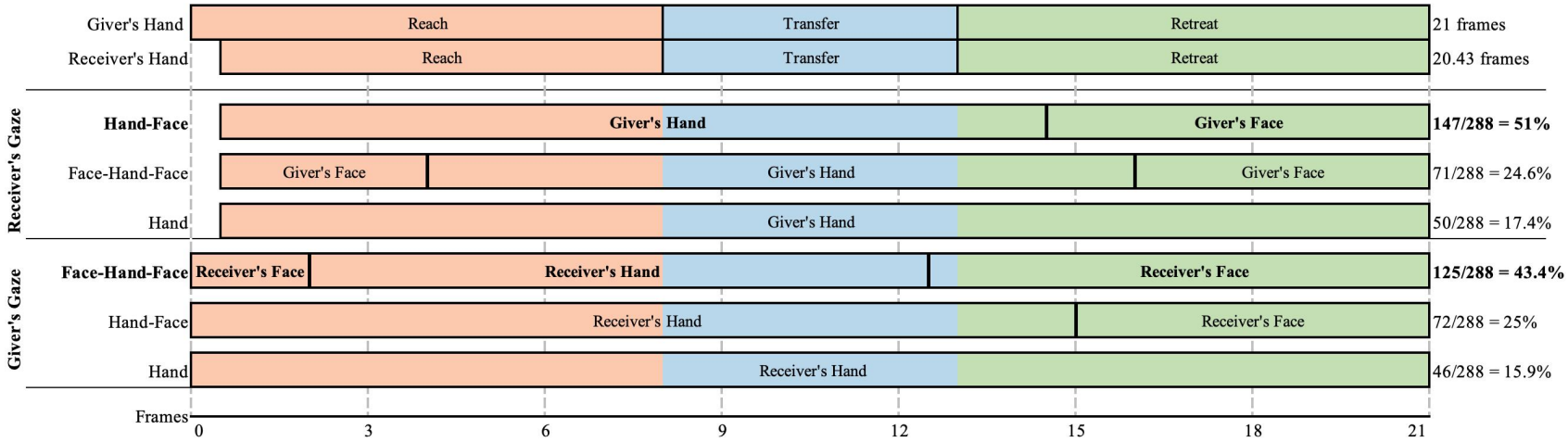
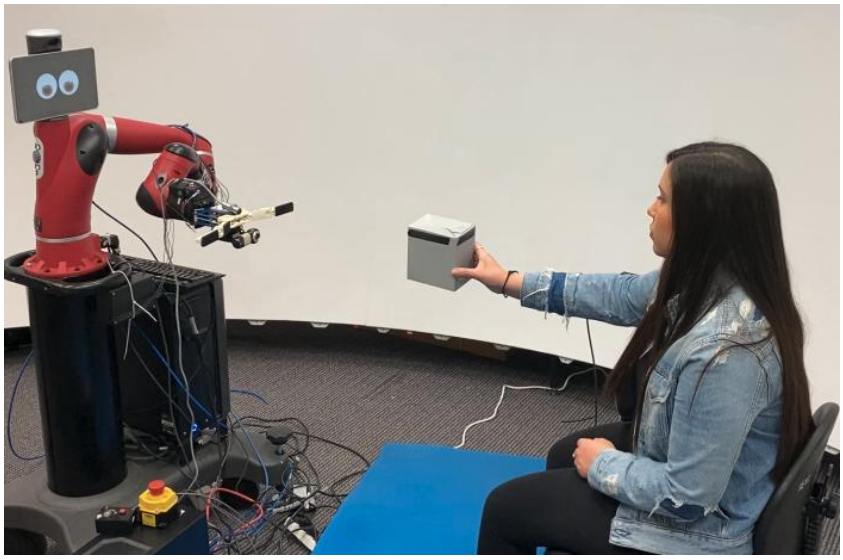
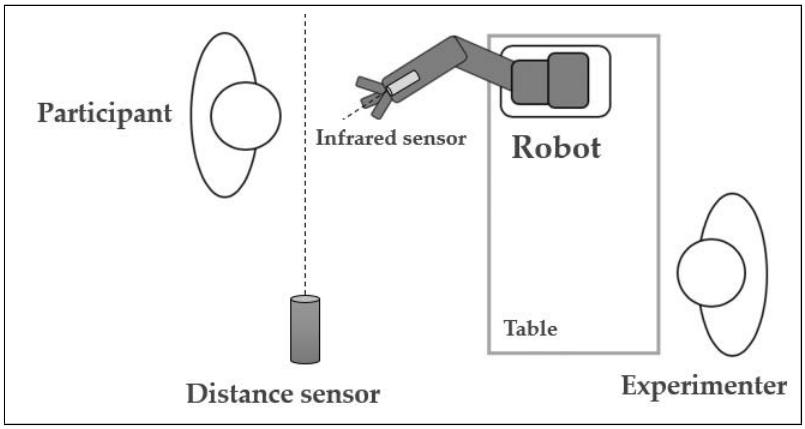


Figure 6.5: Analysis of gaze behaviors in the reach, transfer, and retreat phases of human-human handovers. Time flows left to right. Background colors (labeled on the top two rows) correspond to each phase of a handover: red: reach; blue: transfer; green: retreat. The bottom six rows show one handover behavior each, three for the receiver and three for the giver. Boundaries correspond to the average length of each phase. The prevalence of each behavior is noted at the right edge of the row. Givers and receivers have dissimilar frequently observed gaze behaviors.

(a)



(b)



(c)

Figure 6.6: Experimental Setup: Video frames of an actor handing over an object to the robot, used in the video study: (a) "Standing" posture (b) "Sitting" posture (c) Diagram of the setup for the in-person study.

(a)

(b)

Figure 6.7: The objects used in the experiments: (a) Object size (small box and large box), (b) Object fragility (plastic bottle and glass bottle)

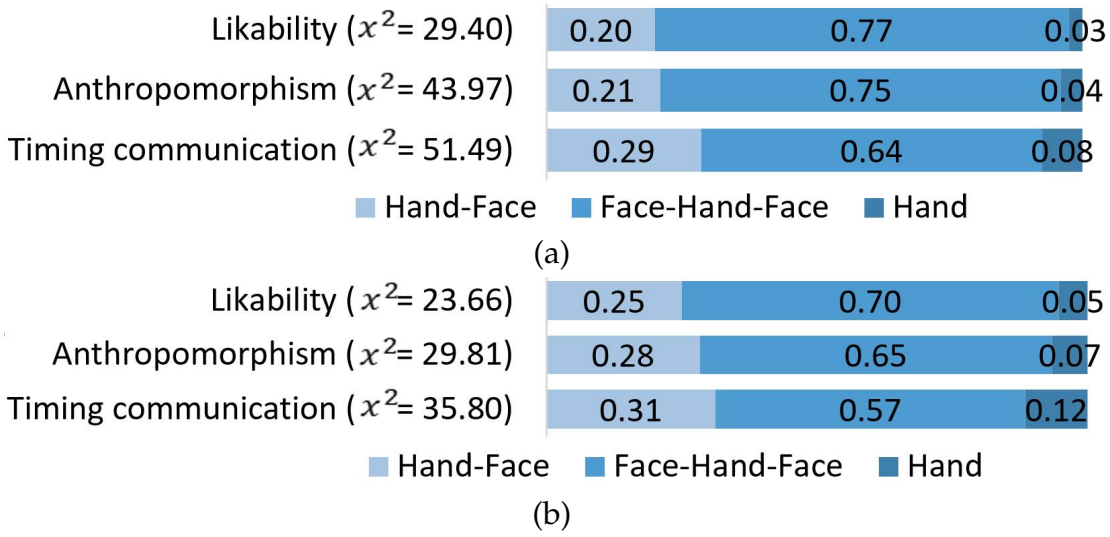


Figure 6.8: χ^2 values and win-probabilities of gaze conditions in the video study for the three dependent measures: (a) Small object , (b) Large object.

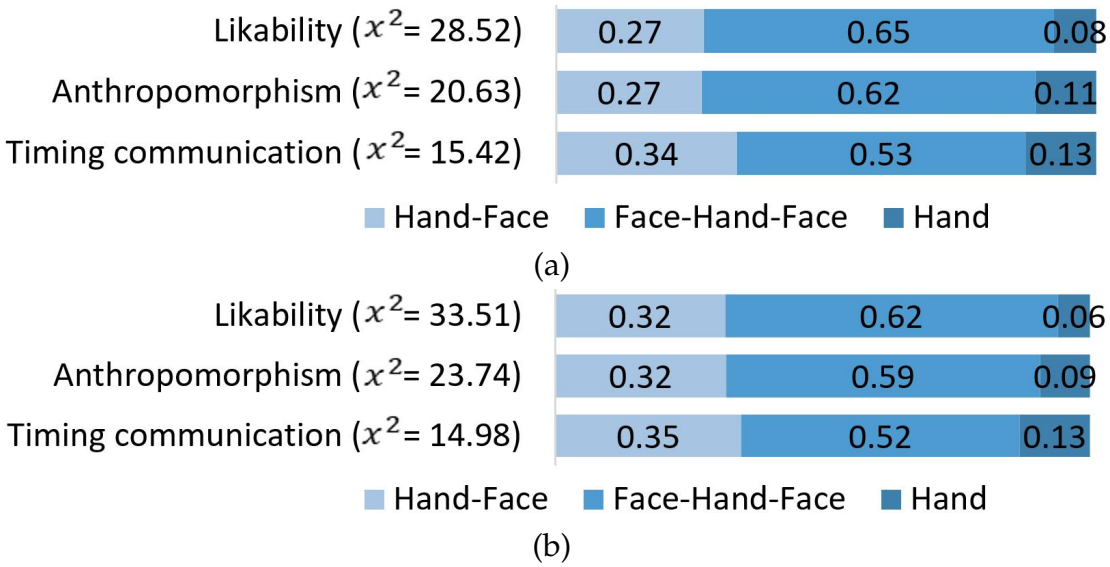


Figure 6.9: χ^2 values and win-probabilities of gaze conditions in the video study for the three dependent measures: (a) Non-Fragile object , (b) Fragile object.

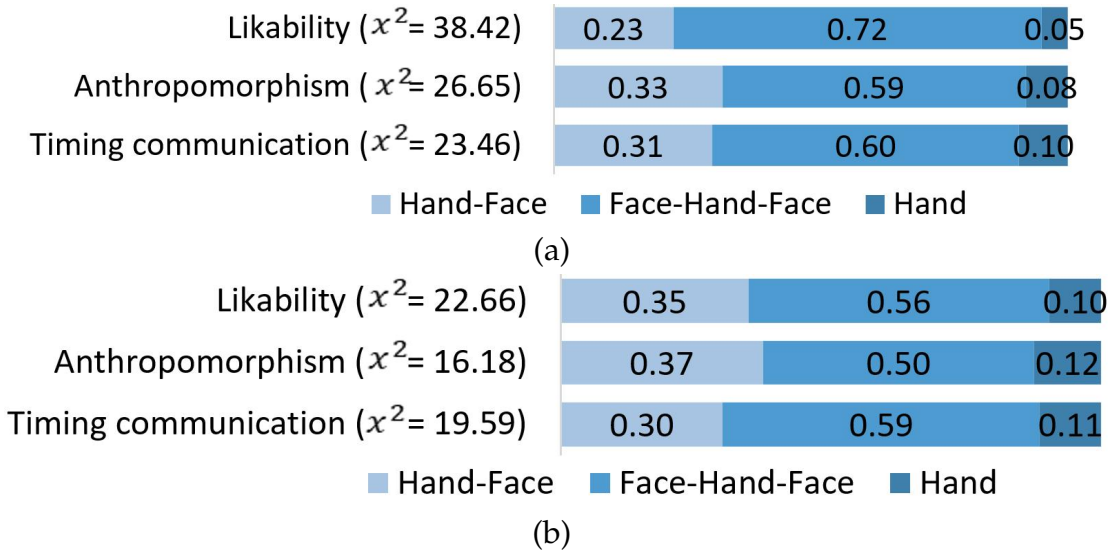


Figure 6.10: χ^2 values and win-probabilities of gaze conditions in the video study for the three dependent measures: (a) Standing , (b) Sitting.

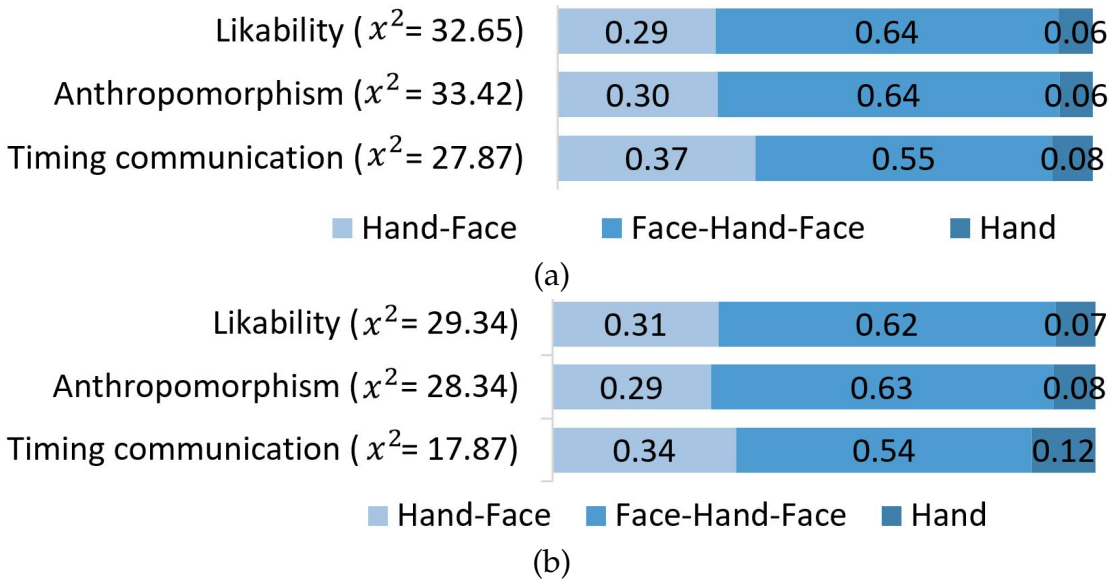


Figure 6.11: χ^2 values and win-probabilities of gaze conditions in the in-person study for the three dependent measures: (a) Small object , (b) Large object.

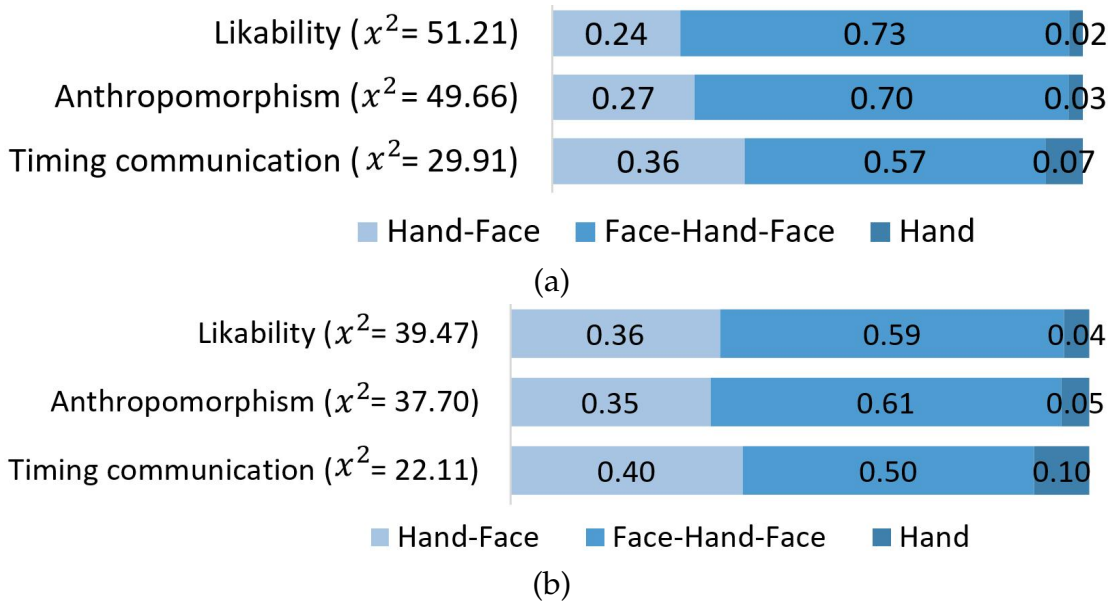


Figure 6.12: χ^2 values and win-probabilities of gaze conditions in the in-person study for the three dependent measures: (a) Non-Fragile object , (b) Fragile object.

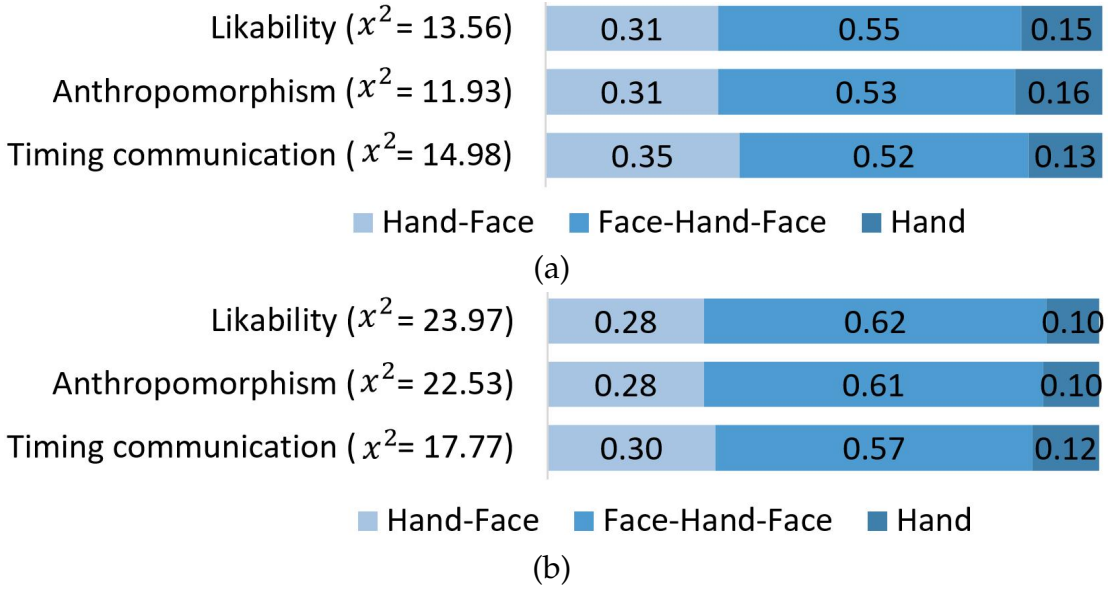


Figure 6.13: χ^2 values and win-probabilities of gaze conditions in the in-person study for the three dependent measures: (a) Standing , (b) Sitting.

CHAPTER 7

BIMANUAL HUMAN-HUMAN AND HUMAN-ROBOT HANDOVERS

7.1 Introduction

As described in Section 2.2 and Section 2.3, a lot of prior work has focused on single-handed human-human and human-robot object handovers. These studies have investigated different phases of handovers, such as approach [7, 90, 172, 75], reach [156, 66, 76, 19, 18, 125], and transfer [42, 123, 124, 57]. Many researchers have also studied the handover location and orientation in human-human handovers [7, 58, 71, 72, 34]. However, there is very little work on bimanual handovers, i.e. handovers of objects requiring two-handed grasps (Section 2.6), even though they are equally important and challenging. We seek to address this knowledge gap by investigating bimanual handovers.

First, we create a multi-sensor dataset of human-to-human bimanual handovers. Existing datasets of human-to-human handovers [25, 28, 183] have studied handovers of objects with single-handed grasps i.e. uni-manual handovers. To the best of our knowledge, there is no prior public dataset of bimanual handovers. Our dataset contains 240 recordings collected from 24 volunteers performing bimanual handovers with 10 objects. We also include 120 recordings from the same volunteers performing unimanual handovers with 5 of those objects. These can be used to compare the characteristics of unimanual and bimanual handovers of the same object. Each recording consists of the positions and orientations of 13 upper-body bones for each participant, the positions of 27 upper-body bone markers for each participant, the position/orientation of the object, and two RGB-D data streams. The recordings are annotated with the

three handover phases: reach, transfer, and retreat. The dataset also includes the height, weight, arm span, and waistline height of the participants.

Second, we create a multi-sensor dataset of human-human handovers in a shelving/unshelving task that requires multiple sequential handovers. Existing datasets of human-to-human handovers [25, 28, 183] have studied isolated handovers, whereas several real-life tasks require multiple sequential handovers. These handovers can be of different types: unimanual, bimanual, and even, self-handovers performed to adjust grasps. To the best of our knowledge, there is no public dataset of multiple sequential handovers. Our dataset contains 48 recordings of handovers with 30 objects (total of 1440 handovers) collected from 24 volunteers. Each recording consists of the positions and orientations of 13 upper-body bones for each participant, the positions of 27 upper-body bone markers for each participant, and two RGB-D data streams. The dataset also includes the height, weight, arm span, and waistline height of the participants.

Third, we explore the application of Bayesian Interaction Primitives (BIP) [22], an imitation learning technique, to train a humanoid robot to perform human-like bimanual reaching motions in handovers. BIPs have been used to train a robot for human-robot interaction tasks from human demonstrations. However, the previous evaluations did not involve tasks with coordination constraints on the robot's motion, whereas a bimanual reaching motion requires coordinated movement, both in space and in time, of both arms. Our evaluations reveal that the BIPs, in their existing formulation, do not work for generating a coordinated bi-manual reaching motion. We discuss some possible solutions to modify BIPs for bimanual human-robot handovers.

7.2 Multi-sensor Dataset of Bimanual Human-Human Handovers

We conducted a user study to create a multi-sensor dataset of bimanual human-to-human handovers (example shown in Fig 7.1). This section describes the study protocol and data collection process.

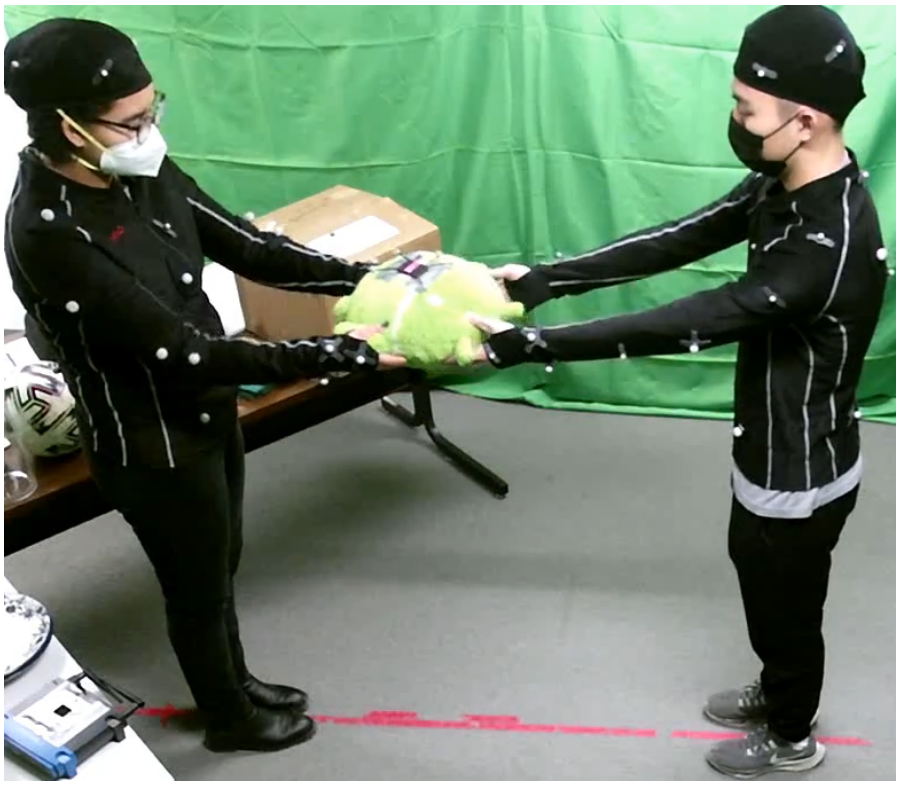


Figure 7.1: Example of a Bimanual Handover

7.2.1 Study setup

Our experiment setup, shown in Fig. 7.2, consisted of an OptiTrack motion tracking system with 12 Flex-13 cameras, and two Kinect v2 sensors. The area was surrounded by green screens to make it easier for other researchers to do

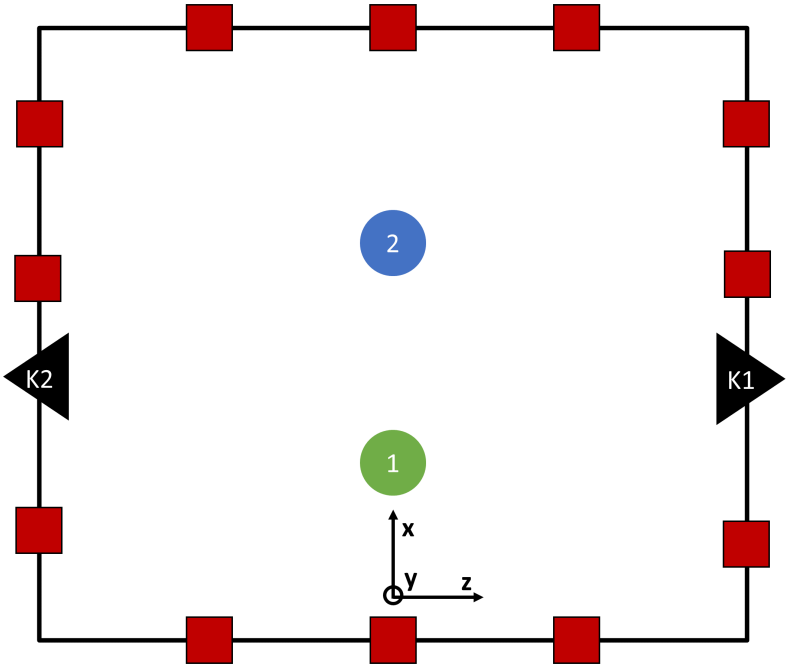


Figure 7.2: Schematic of the data collection setup of the bimanual handovers dataset. We used an OptiTrack motion tracking system with 12 cameras (red squares), and two Kinect v2 sensors (black triangles). The green and blue circles represent the positions of the participants. The reference frame in the figure shows the reference frame of the motion tracking system whose origin was on the floor and behind the participant shown with the green circle. The participants were instructed to stand along the x-axis of this reference frame, at an interpersonal distance that is comfortable to them.

post-processing on the video recordings in the future.

Motion Tracking Data

The OptiTrack cameras were connected to a desktop computer (CPU: Intel Core i5-7600 3.5GHz, GPU: 2GB AMD Radeon T R5 430 and 4 GB Intel HD Graphics 630, RAM: 8GB, running Windows 10 Pro) which recorded the skeleton tracking

data through OptiTrack's Motive software. To track the participants' motion in bimanual handovers, we used the "Conventional Upper Body" marker set consisting of 27 reflective markers placed at the positions shown in Fig. 7.3. Along with the 3D positions of these 27 markers, Motive also computes–from the marker positions–the 3D position and orientation of 13 upper-body bones shown in Fig. 7.3.

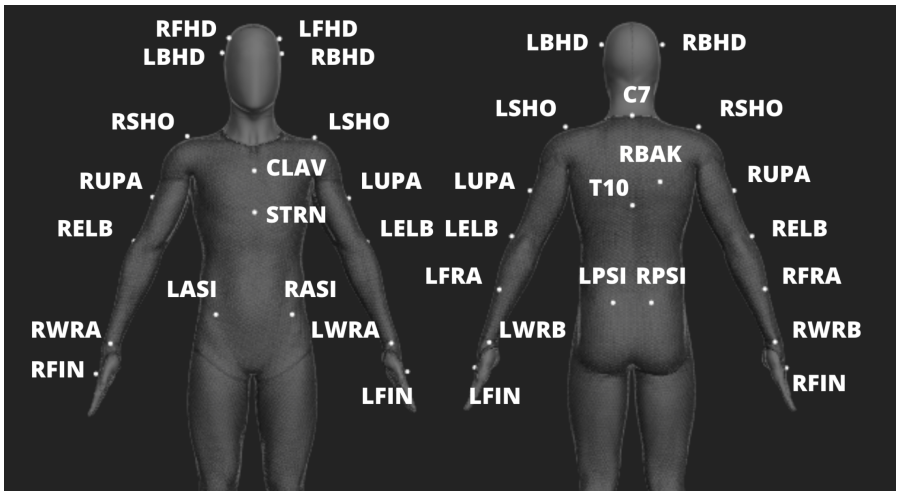


Figure 7.3: Names of bone markers in the "Conventional Upper Body" skeleton

We also recorded the position and orientation of the object being handed over with a rigid body marker set consisting of 6 reflective markers. The motion tracking data was recorded at 120Hz.

Kinect Sensor Data

The first Kinect v2 sensor (K1 in Fig. 7.2) was connected to a desktop computer (CPU: Intel Core i7-6700 3.4GHz, RAM: 16GB, GPU: 4GB NVIDIA Quadro P1000, running Windows 10 Pro) which recorded the data streams from the Kinect sensor in the Microsoft Kinect Studio software. The second Kinect v2 sensor (K2 in Fig. 7.2) was connected to a laptop (CPU: Intel Core i7-7700HQ

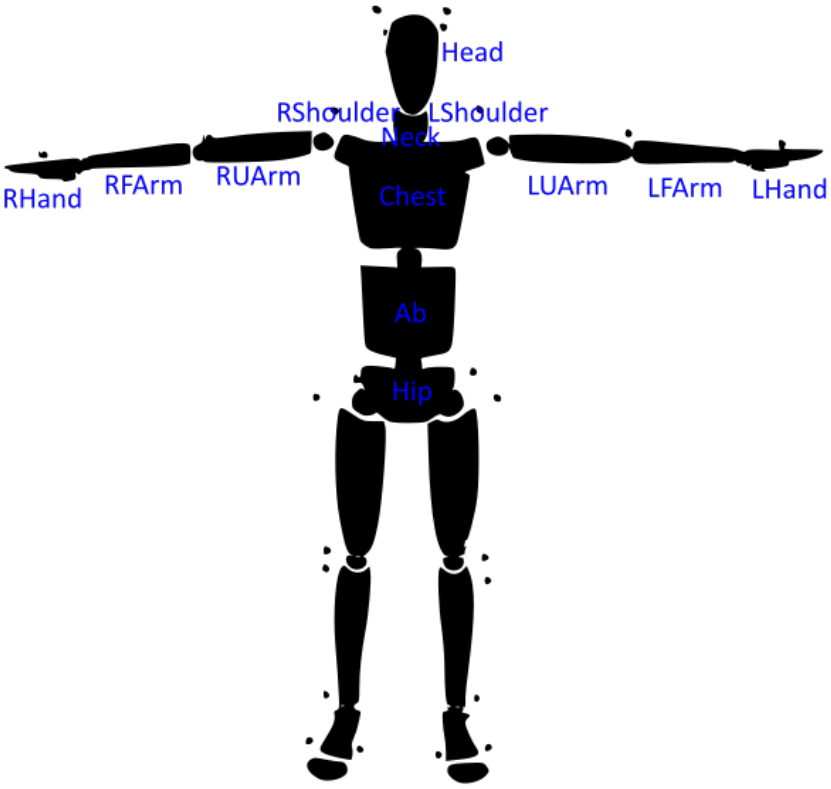


Figure 7.4: Names of bones in the "Conventional Upper Body" skeleton

2.8GHz, RAM: 16GB, GPU: 4GB NVIDIA GeForce GTX 1050 and 8GB Intel HD Graphics 630, running Windows 10 Home) which recorded the data streams from the Kinect sensor in the Microsoft Kinect Studio software. The Kinect V2 provides RGB-D streams in the following format: RGB - Resolution of 1920 x 1080, frame rate of 30 frames per second, which is the maximum allowed. If the light is low, the sensor changes to 15 frames per second to allow in more light for better exposure. D - Resolution of 512 x 424, frame rate of 30 frames per second.

7.2.2 Objects

We used 10 objects of different shapes, sizes, rigidity, and fragility, as shown in Table 7.1 and Figure 7.5, in the study.

Table 7.1: Characteristics of objects used in the bimanual handovers dataset. D: Diameter, H: Height

#	Name	Geometry	Size (mm)	Weight (kg)	Rigidity	Fragility
1	Flower Pot	Cylinder	153 D, 112 H	0.60	Hard	Fragile
2	Dishes	Cylinder	250 D, 80 H	1.61	Hard	Fragile
3	Laptop	Cuboid	342x240x25	1.36	Hard	Fragile
4	Plush Toy	Cylinder	380 D, 150 H	0.35	Deformable	Non-fragile
5	Wifi Router	Cuboid	200x173x45	0.38	Hard	Fragile
6	Basketball	Spherical	240 D	0.60	Hard	Non-fragile
7	Folded Clothes	Cuboid	420x300x70	1.04	Deformable	Non-fragile
8	Cardboard Box	Cuboid	400x390x280	1.16	Hard	Non-fragile
9	Cardboard Plank	Cuboid	560x355x15	0.21	Hard	Non-fragile
10	Backpack	Cuboid	460x395x35	1.30	Deformable	Non-fragile

Figure 7.5: Objects used in the bimanual handovers dataset

7.2.3 Study protocol

After entering the room containing the recording setup, participants signed the consent form and wore the motion capture suits. The experimenter calibrated the skeletons in the Motive software for each participant. Then the participants performed a shelving/unshelving task (for the sequential handovers dataset described in Section 7.3), followed by both unimanual (right-handed) and bimanual handovers of 5 objects shown in Fig. 7.5 (top row) and only bimanual handovers of 5 objects shown in Fig. 7.5 (bottom row). They performed these handovers standing face-to-face. They were instructed to stand along the x-axis of the reference frame of the motion tracking system (see Fig. 7.2), at an interpersonal distance that is comfortable to them. Finally, the experimenter measured the height, weight, arm span, and waistline height of the participants.

7.2.4 Participants

A total of 30 volunteers (15 pairs) participated in our study. Out of these, the first three pairs, six participants, were pilot trials. Each study session took approximately 30 minutes, and the participants were compensated with $5 for participation.

7.2.5 Data Processing and Annotations

We used the KinectXEFTools[1] library to extract the RGB videos in '.avi' format from the raw Kinect recordings which were in '.xef' format. We used the

[1] https://github.com/Isaac-W/KinectXEFTools

Kinect2Mat[2] library to extract the depth data in '.mat' format from the raw Kinect recordings. We split the data recordings of each study session into individual handovers manually. The data from the two Kinect sensors and the OptiTrack system was recorded on three separate computers. To help synchronize the three data streams during post-processing, we had asked participants to stand in a "T-pose" at the beginning of each study session. We used the timestamps of this event to synchronize the data recordings of the two Kinect sensors and the OptiTrack system. From the OptiTrack Motive software, we also exported the Quaternion orientations of 13 upper-body bones for each participant with respect to their parental segment, roughly representing the joint angles.

We annotated each frame of the data recordings with the three phases of a handover: reach, transfer, and retreat. The "reach" phase corresponds to the reaching motion of the giver and receiver before the receiver makes contact with the object. "Transfer" starts when both the giver and receiver touch the object for the first time, and ends when the giver releases the object. In the "retreat" phase, the giver and receiver retract their hands to the resting position. In our annotations, the "transfer" phase corresponds to the portion of the handover where the distance between the giver's and receiver's hands is within 20cm of the minimum distance, and their relative velocity is less than 15cm/s. These thresholds were obtained with trial-and-error.

[2] `https://github.com/raysworld/Xef2Mat`

7.3 Multi-sensor Dataset of Sequential Handovers

We conducted a user study to create a multi-sensor dataset of sequential human-human handovers. This section describes the study protocol and data collection process.

7.3.1 Study setup

We used a similar setup as the one described in Section 7.2.1, with the addition of two tables and a shelf as shown in Fig. 7.7. Our study involved two types of tasks requiring sequential handovers: shelving and un-shelving. In the "shelving task", the giver picked up objects from the tables and handed them to the receiver who placed the objects on the shelf. In the "un-shelving task" the giver picked up objects from the shelf and handed them to the receiver who placed the objects on the tables.

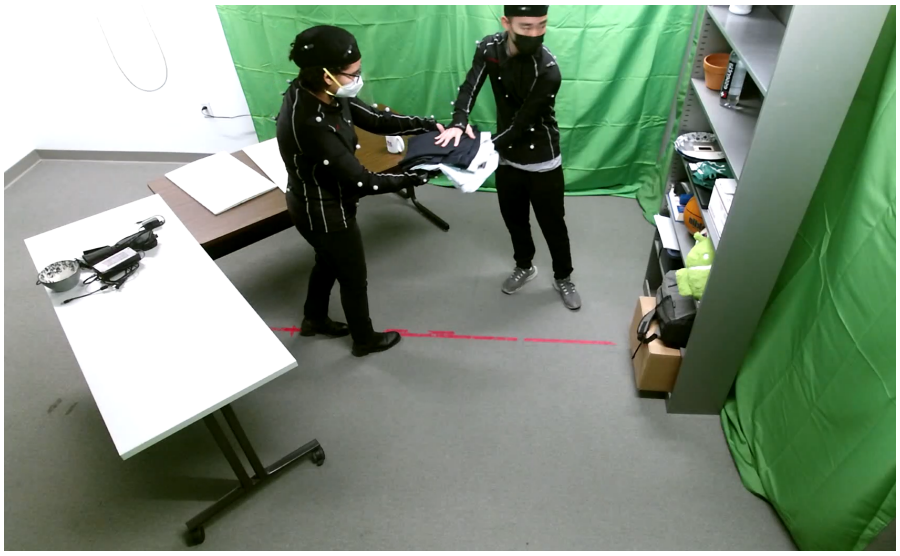


Figure 7.6: Example of a handover in the shelving task

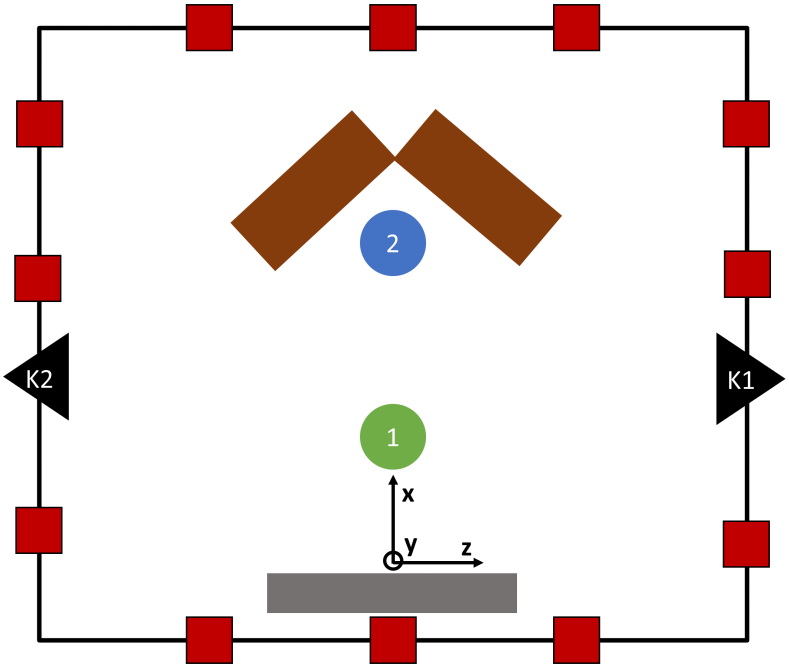


Figure 7.7: Schematic of the data collection setup of the multiple sequential handovers dataset. We used an OptiTrack motion tracking system with 12 cameras (red squares), and two Kinect v2 sensors (black triangles). The green and blue circles represent the positions of the participants. The brown rectangles represent two tables and the gray rectangle represents the shelf. The reference frame in the figure shows the reference frame of the motion tracking system whose origin was on the floor and in front of the shelf. The participants were instructed to stand along the x-axis of this reference frame, at an interpersonal distance that is comfortable to them.

7.3.2 Objects

We used 30 objects of different shapes, sizes, rigidity, and fragility, as shown in Table 7.2 and Figure 7.8, in the study.

Table 7.2: Characteristics of objects used in the sequential handovers dataset.

#	Name	Geometry	Size (mm)	Weight (kg)	Rigidity	Fragility
1	Flower Pot	Cylinder	153 D, 112 H	0.60	Hard	Fragile
2	Dishes	Cylinder	250 D, 80 H	1.61	Hard	Fragile
3	Laptop	Cuboid	342x240x25	1.36	Hard	Fragile
4	Plush Toy	Cylinder	380 D, 150 H	0.35	Deformable	Non-fragile
5	Wifi Router	Cuboid	200x173x45	0.38	Hard	Fragile
6	Basketball	Spherical	240 D	0.60	Hard	Non-fragile
7	Folded Clothes	Cuboid	420x300x70	1.04	Deformable	Non-fragile
8	Box (L)	Cuboid	400x390x280	1.16	Hard	Non-fragile
9	Cardboard Plank	Cuboid	560x355x15	0.21	Hard	Non-fragile
10	Backpack	Cuboid	460x395x35	1.30	Deformable	Non-fragile
11	Football	Spherical	220 D	0.41	Hard	Non-fragile
12	Box (M)	Cuboid	285x220x145	0.19	Hard	Non-fragile
13	Storage box	Cuboid	405x270x185	1.27	Hard	Non-fragile
14	Salad Bowl	Spherical	153 D, 75 H	0.52	Hard	Fragile
15	Umbrella	Cylinder	70 D, 410 H	0.30	Hard	Non-fragile
16	Textbook	Cuboid	260x210x60	1.28	Hard	Non-fragile
17	Laptop Charger	Cuboid	210x80x75*	0.61	Deformable	Non-fragile
18	Mobile Charger	Cuboid	120x45x30*	0.07	Deformable	Non-fragile
19	Wooden plank	Cuboid	610x450x10	0.88	Hard	Non-fragile
20	Spray Bottle	Cylinder	85 D, 295 H	1.04	Hard	Non-fragile
21	Water Bottle	Cylinder	80 D,270 H	1.04	Hard	Non-fragile
22	Mug	Cylinder	120 D, 85 H	0.48	Hard	Fragile
23	Stack of papers	Cuboid	279x215x10	0.17	Deformable	Non-fragile
24	Wrench	Cuboid	300x70x12	0.83	Hard	Non-fragile
25	Cloth bag	Cuboid	450x450x5	0.05	Deformable	Non-fragile
26	Glass Vase	Cylinder	127 D, 232 H	0.82	Hard	Fragile
27	Keyboard	Cuboid	450x155x30	0.54	Hard	Non-fragile
28	Clamp	Cuboid	300x155x13	0.26	Hard	Non-fragile
29	Tape	Cylinder	80 D, 46 H	0.02	Hard	Non-fragile
30	Paper roll	Cylinder	190 D, 200 H	1.75	Hard	Non-fragile

* in wrapped configuration, D: Diameter, H: Height

7.3.3 Study protocol

After entering the room containing the recording setup, participants signed the consent form and wore the motion capture suits. The experimenter calibrated the skeletons in the Motive software for each participant. Then the participants

Figure 7.8: Objects used in the sequential handovers dataset

performed the shelving/unshelving tasks in the following sequence: shelving task (P1 giver, P2 receiver), unshelving task (P2 giver, P1 receiver), shelving task (P2 giver, P1 receiver), unshelving task (P1 giver, P2 receiver). In each shelving task, the giver was instructed to stay close to the tables and the receiver was instructed to stay close to the shelf. In each unshelving task, the giver was instructed to stay close to the shelf and the receiver was instructed to stay close to the tables. The participants were instructed to hand over one object at a time, but they were not given any specific instructions about the type of handovers or the sequence of objects or the arrangement of objects on the shelf/on the table. After finishing the shelving/unshelving tasks, participants performed the second part of the study described in Section 7.2.3.

7.3.4 Data Processing and Annotations

We used the KinectXEFTools[3] library to extract the RGB videos in '.avi' format from the raw Kinect recordings which were in '.xef' format. We used the Kinect2Mat[4] library to extract the depth data in '.mat' format from the raw Kinect recordings. We split the data recordings of each study session into individual shelving/unshelving tasks manually. The data from the two Kinect sensors and the OptiTrack system was recorded on three separate computers. To help synchronize the three data streams during post-processing, we had asked participants to stand in a "T-pose" at the beginning of each shelving/unshelving task. We used the timestamps of this event to synchronize the data recordings of the two Kinect sensors and the OptiTrack system. From the OptiTrack Motive software, we also exported the Quaternion orientations of 13 upper-body bones for each participant with respect to their parental segment, roughly representing the joint angles.

7.4 Controller for Humanlike Reaching Motion in Bimanual Handovers

There is very little work on robot controllers for two-handed human-robot handovers [12, 159, 149, 59]. Also in these works, the robot's reaching motion is not human-like, which is required for humanoid robots. For single-handed handovers, several researchers have proposed robot controllers for human-like reaching motions [132, 133, 100, 101, 84, 147, 110, 157, 180, 186, 182] (See "Con-

[3] https://github.com/Isaac-W/KinectXEFTools
[4] https://github.com/raysworld/Xef2Mat

trollers Learned from Demonstrations or Feedback" in Section 2.5.2). But, to the best of our knowledge, there is no prior work on human-like two-handed handovers. We seek to address this gap by developing a robot controller for generating human-like robot reaching motions in bimanual handovers.

For generating human-like single-handed handovers, researchers have used motion primitives such as Dynamic Movement Primitives (DMPs) [132, 133], Probabilistic Movement Primitives (ProMPs) [100, 101], Triadic Interaction Meshes (IMs) [169], and Bayesian Interaction Primitives (BIPs) [22, 23, 24]. While DMPs and IMs can only utilize single demonstrations, IMs and BIPs use probabilistic methods to learn from multiple demonstrations and are thus more generalizable. BIPs are computationally more efficient than IMs [22] because IMs use computationally demanding Dynamic Time Warping (DTW) to estimate the current phase (i.e. time-step) of the interaction whereas BIPs couple the estimation of phase and state using techniques from the field of Simultaneous Localization and Mapping (SLAM). Therefore, we select BIPs to teach a humanoid robot to perform humanlike bimanual reaching motions in handovers.

7.4.1 Bayesian Interaction Primitives

BIPs are a human-robot interaction framework that focuses on extracting a probabilistic model of interaction dynamics as presented in a set of example demonstrations. Here we provide a conceptual overview of the BIPs and refer the reader to [21] for a detailed description and the mathematical formulation of BIPs. In the training phase of BIPs, demonstrations of the interaction, which consist of observations over time for each measured degree of freedom (DoF),

are decomposed into a linear combination of nonlinear basis functions. The coefficients of these basis functions, which represent the latent state of an interaction, are used to construct a prior in the form of a joint probability distribution of all DoFs. In the testing phase, this prior distribution is updated based on the sequence of observations from a subset of DoFs. This update step involves simultaneously inferring both the spatial state of the interaction (the latent state) and the temporal state (the phase). The posterior distribution can be used to infer future states of any DoF in the model, and also to generate control trajectories for the robot.

7.4.2 Learning Bimanual Handovers with BIPs

We use the IntPrims library[5] developed by Campbell et al. to evaluate BIPs for bimanual handovers. We focus on the reaching phase of a robot-to-human handover in which the robot needs to carry the object in both hands toward the human. Thus the robot needs to maintain a fixed distance between its hands, equal to the grip width of the object while generating a human-like reaching motion.

We use the human-human bimanual handovers data collected in our previous study 7.2 to train the BIPs. For the initial evaluation, we only consider the Cartesian trajectories of the giver's and receiver's left and right wrists, during 8 object handovers of one pair of participants. We also include the grip width of each object as a state variable. Fig. 7.9 shows the trained distributions. The black line indicates the mean of each degree of freedom and the shaded blue region represents one standard deviation.

[5] `https://github.com/ir-lab/intprim`

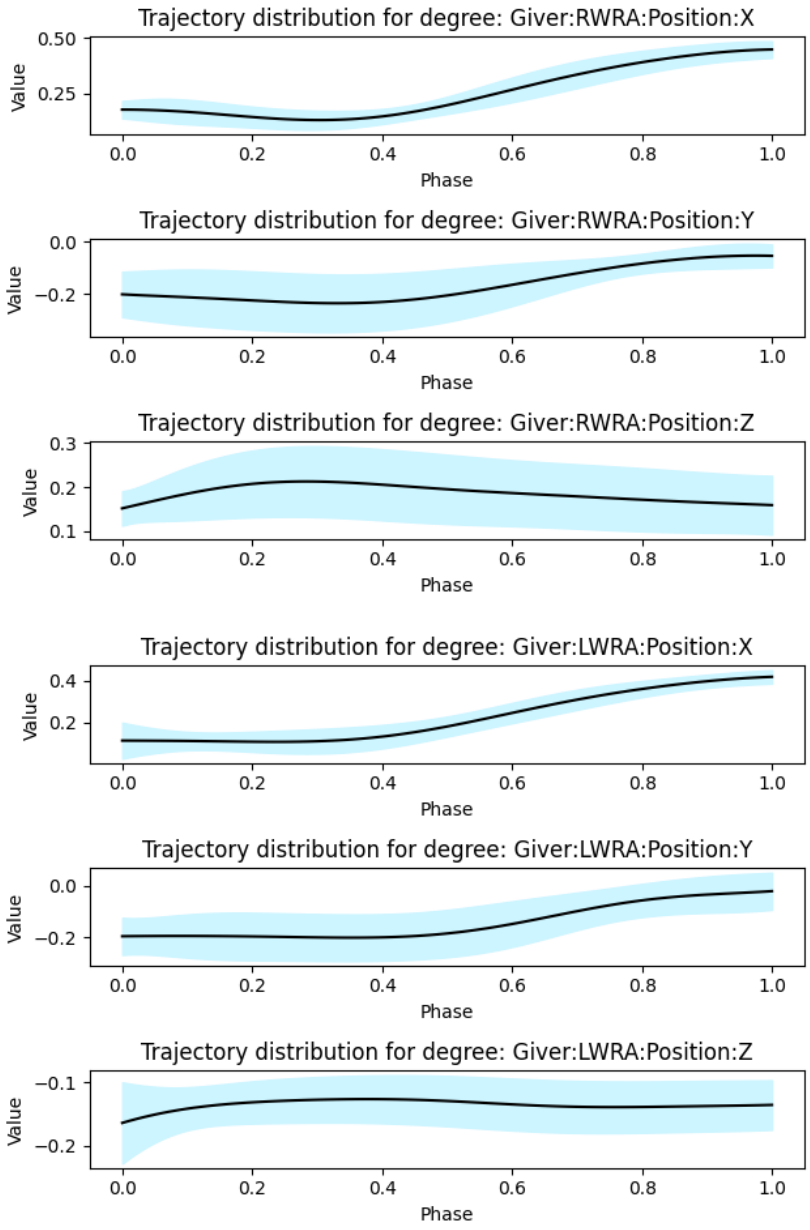


Figure 7.9: Distributions of training trajectories (continued on next page). RWRA: Right Wrist, LWRA: Left Wrist

We evaluate the trained BIPs on the reach-to-handover trajectories of a new object for the same pair of participants used for training the BIPs. Fig. 7.10 shows the distance between the left and right wrists of the giver: the green curve shows the mean distance in the training trajectories, the blue curve shows the distance in the observed test samples (10 timesteps), and the red dotted

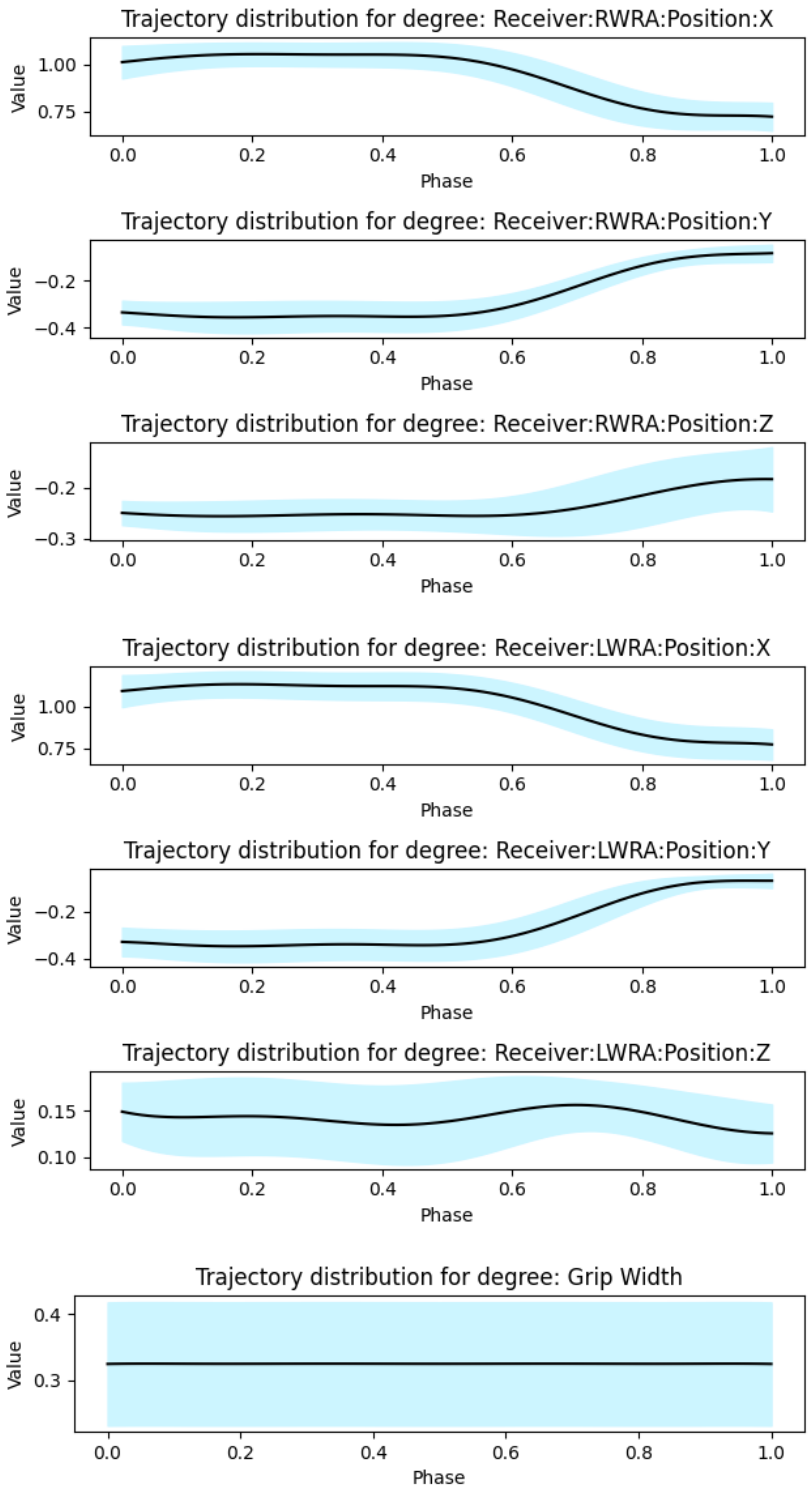


Figure 7.9: Distributions of training trajectories (continued from previous page). RWRA: Right Wrist, LWRA: Left Wrist

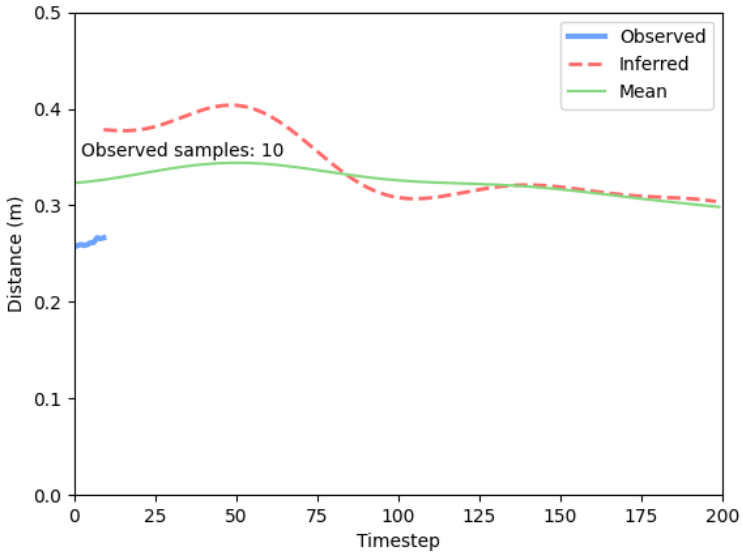


Figure 7.10: Distance between the left and right wrists of the giver

curve shows the distance in the predictions of BIPs. We find that the predictions of BIPs stay around the mean, whereas the observed distance is less than the mean. The observed distance corresponds to the actual grip width of the test object. Thus, if the predictions of BIPs are used to generate controls for a robot giver, the object will slip from the robot's hands.

7.5 Constraining Bayesian Interaction Primitives

The existing formulation of BIPs [22] does not involve adaptations to any joint or Cartesian space constraints, such as maintaining the object grip width in a bimanual handover. Here, we briefly discuss two possible solutions that can be explored to impose the bimanual grip width constraint in BIPs. The first solution involves updating the trained distributions of BIPs to satisfy the grip width constraint, and the second solution involves imposing this constraint dur-

ing the inference process. The first solution is similar to the method proposed by Frank et al. [52] for adapting ProMPs to joint/Cartesian constraints. They formulate adaptation as a constrained minimization problem, to minimize the Kullback–Leibler divergence between the adapted distribution and the distribution of the original primitive while adhering to the constraints. The second solution is to impose the constraint in the spatiotemporal filters that are used in BIPs for inference, for example, by utilizing constrained Ensemble Kalman methods [2].

CHAPTER 8

CONCLUSION

People frequently perform object handovers with each other, and robots will be expected to perform such handovers with people, both at work and home. Tasks such as collaborative assembly, surgical assistance, shelving in warehouses, and elderly care require a robot to receive objects from and/or give objects to a human. We presented three directions of research related to the collaborative task of object handovers. First, we developed robot controllers to address four previously unexplored scenarios of human-robot handovers. Second, we investigated a robot receiver's gaze in human-to-robot handovers. Third, we created two datasets of bimanual and sequential human-human handovers that can provide new insights into human behavior in handovers.

In the first research direction, we tried to tackle the challenge of designing robot controllers to address previously unexplored scenarios of human-robot handovers. First, we proposed and evaluated two robot controllers for single-handed handovers that allow users to specify intuitive controller parameters, and also provide feedback in case of failure. Second, we presented a robot controller for handovers that uses automated synthesis from formal specifications, a framework to unify different strategies for object handovers. Third, we evaluated a reinforcement learning technique called Guided Policy Search for training a robot to perform reach-to-handover motions, without requiring prior knowledge of the robot's dynamics. Finally, we evaluated an imitation learning technique called Bayesian Interaction Primitives for training a robot to perform human-like bimanual reach-to-handover motions, a desirable feature for humanoid robots. Each of our controllers has some unique features, as compared

to the features of the existing handover controllers, and developing a single controller that has all these features is still an open challenge.

Our second research direction investigated human preferences for a robot's gaze behaviors when it is receiving an object from a human. We conducted four studies to compare human-inspired robot gaze behaviors in human-to-robot handovers. We also contributed to the literature on non-verbal gestures in human-human interactions, by annotating a large public dataset of human-human handovers with the giver and receiver's gaze locations. Our findings could be useful for designing non-verbal behaviors of a collaborative robot when it takes on the role of a receiver in a handover. Our studies were limited to isolated handovers, a common limitation of prior studies as well, whereas, many real-world tasks require multiple sequential handovers. Future work can explore the design of robot gaze behaviors in more realistic collaborative or assistive scenarios.

Finally, we developed two datasets of human-human handovers, one for bimanual handovers and one for sequential handovers, two scenarios that have not been investigated in the existing datasets of human-human handovers. Our datasets could help in developing human motion models and robot controllers for bimanual and sequential object handovers.

Appendices

APPENDIX A

SUPPLEMENTARY MATERIALS FOR "TIMING-SPECIFIED CONTROLLERS WITH FEEDBACK FOR HUMAN-ROBOT HANDOVERS"

Tables A.1-A.4 show the Pearson's correlation scores for the User Experience Variables "Efficiency, "Perspicuity" and "Dependability", and NASA TLX Variables "Mental Demand", "Performance", "Effort" and "Frustration", for the PV and MCJ controllers in Study 1.

Table A.1: Person's correlation scores for the User Experience variables and NASA TLX variables for PV Controller in Study-1, Round-1

NASA TLX Variable	UE Variable	Pearson's r	p
Mental Demand	Efficiency	-0.243	0.196
Mental Demand	Perspicuity	-0.403	0.027
Mental Demand	Dependability	-0.371	0.043
Performance	Efficiency	0.395	0.031
Performance	Perspicuity	0.251	0.180
Performance	Dependability	0.527	0.003
Effort	Efficiency	-0.325	0.080
Effort	Perspicuity	-0.402	0.028
Effort	Dependability	-0.476	0.008
Frustration	Efficiency	-0.583	$< .001$
Frustration	Perspicuity	-0.598	$< .001$
Frustration	Dependability	-0.671	$< .001$

Table A.2: Person's correlation scores for the User Experience variables and NASA TLX variables for PV Controller in Study-1, Round-2

NASA TLX Variable	UE Variable	Pearson's r	p
Mental Demand	Efficiency	-0.384	0.036
Mental Demand	Perspicuity	-0.530	0.003
Mental Demand	Dependability	-0.486	0.006
Performance	Efficiency	0.445	0.014
Performance	Perspicuity	0.533	0.002
Performance	Dependability	0.604	$< .001$
Effort	Efficiency	-0.380	0.038
Effort	Perspicuity	-0.526	0.003
Effort	Dependability	-0.446	0.014
Frustration	Efficiency	-0.511	0.004
Frustration	Perspicuity	-0.512	0.004
Frustration	Dependability	-0.496	0.005

Table A.3: Person's correlation scores for the User Experience variables and NASA TLX variables for MCJ Controller in Study-1, Round-1

NASA TLX Variable	UE Variable	Pearson's r	p
Mental Demand	Efficiency	-0.347	0.060
Mental Demand	Perspicuity	-0.739	$<.001$
Mental Demand	Dependability	-0.444	0.014
Performance	Efficiency	0.447	0.013
Performance	Perspicuity	0.367	0.046
Performance	Dependability	0.461	0.010
Effort	Efficiency	-0.494	0.006
Effort	Perspicuity	-0.651	$<.001$
Effort	Dependability	-0.568	0.001
Frustration	Efficiency	-0.498	0.005
Frustration	Perspicuity	-0.560	0.001
Frustration	Dependability	-0.545	0.002

Table A.4: Person's correlation scores for the User Experience variables and NASA TLX variables for MCJ Controller in Study-1, Round-2

NASA TLX Variable	UE Variable	Pearson's r	p
Mental Demand	Efficiency	-0.464	0.010
Mental Demand	Perspicuity	-0.553	0.002
Mental Demand	Dependability	-0.286	0.126
Performance	Efficiency	0.046	0.810
Performance	Perspicuity	0.295	0.114
Performance	Dependability	0.527	0.003
Effort	Efficiency	-0.407	0.026
Effort	Perspicuity	-0.355	0.055
Effort	Dependability	-0.188	0.320
Frustration	Efficiency	-0.383	0.036
Frustration	Perspicuity	-0.212	0.262
Frustration	Dependability	-0.426	0.019

APPENDIX B

SUPPLEMENTARY MATERIALS FOR "ROBOT GAZE IN HUMAN-TO-ROBOT HANDOVERS"

Tables B.1 - B.6 show the robot gaze preferences of the participants in terms of Likability, Anthropomorphism and Timing Communication. The values in the first three columns indicate the number of "wins" of a row condition over a column condition i.e. the number of participants who preferred a row condition over a column condition. For example, in Table B.1 a Likability rating of 21 in the small object, "HF" row and "H" column shows that 21 participants liked the *Hand-Face* gaze over the *Hand* gaze. We obtained these ratings by averaging the participants' responses for both ordered pairwise comparisons, and thus some of these values are fractions. The values in the a_i column show the sum of the ratings for each row. The probability that a row condition is preferred over other conditions was calculated using an iterative estimation algorithm [67] and the probability values are shown in the P_i column.

Tables B.7-B.9 show the results of binary proportion difference tests and equivalence tests for matched pairs which we used to evaluate H2-H4. We evaluated the user's preferred gaze behavior in terms of Likability, Anthropomorphism, and Timing Communication for different study conditions. The values in the "Z-score" column represent the test statistic. For example, in Table B.7, a Z-score of 0.00 and a P-value of 0.5 for Likability in *Hand-Face* vs. *Hand* gaze conditions means that the proportion of participants in the video study who liked *Hand-Face* over *Hand* condition for both small and large object is not statistically different. However, for the same scenario, a Z-score of -0.92 and a P-value of 0.18 for the Equivalence Test indicate that the proportions are not statistically

equivalent as well.

Table B.1: Combined preferences of gaze behaviors in the video study for the small and large object conditions. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, *L*: Likability, *A*: Anthropomorphism, *TC*: Timing Communication

		Small Object					Large Object				
		HF	FHF	H	a_i	P_i	HF	FHF	H	a_i	P_i
L	HF	0	5.25	21	26.25	0.2	0	5.75	21	26.75	0.25
	FHF	18.75	0	23.5	42.25	0.77	18.25	0	21	39.25	0.70
	H	3	0.5	0	3.5	0.03	3	3	0	6	0.05
A	HF	0	5	20.5	25.5	0.21	0	7.75	20.5	28.25	0.28
	FHF	19	0	22.5	41.5	0.75	16.25	0	20.75	37	0.65
	H	3.5	1.5	0	5	0.04	3.5	3.25	0	6.75	0.07
TC	HF	0	7.5	19	26.5	0.29	0	6	20	26	0.31
	FHF	16.5	0	21.5	38	0.63	18	0	17.5	35.5	0.57
	H	5	2.5	0	7.5	0.08	4	6.5	0	10.5	0.12

Table B.2: Combined preferences of gaze behaviors in the video study for the non-fragile object and fragile object conditions. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, *L*: Likability, *A*: Anthropomorphism, *TC*: Timing Communication

		Non-Fragile Object					Fragile Object				
		HF	FHF	H	a_i	P_i	HF	FHF	H	a_i	P_i
L	HF	0	6.25	19.5	25.75	0.27	0	7.25	21	28.25	0.32
	FHF	17.75	0	20.5	38.25	0.65	16.75	0	21	37.75	0.62
	H	4.5	3.5	0	8	0.08	3	3	0	6	0.06
A	HF	0	6	18.25	24.25	0.27	0	7.5	19.5	27	0.32
	FHF	18	0	19	37	0.62	16.5	0	19.75	36.25	0.59
	H	5.75	5	0	10.75	0.11	4.5	4.25	0	8.75	0.09
TC	HF	0	7	19.5	26.5	0.34	0	7.5	19.5	27	0.35
	FHF	17	0	17	34	0.53	16.5	0	17	33.5	0.52
	H	4.5	7	0	11.5	0.13	4.5	7	0	11.5	0.13

Table B.3: Combined preferences of gaze behaviors in the video study for the standing and sitting conditions. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, *L*: Likability, *A*: Anthropomorphism, *TC*: Timing Communication

		Standing					Sitting				
		HF	FHF	H	a_i	P_i	HF	FHF	H	a_i	P_i
L	HF	0	4.5	21	25.5	0.23	0	8.25	19.75	28	0.35
	FHF	19.5	0	21	40.5	0.72	15.75	0	19.5	35.25	0.55
	H	3	3	0	6	0.05	4.25	4.5	0	8.75	0.10
A	HF	0	7.25	20.5	27.75	0.33	0	9.25	19	28.25	0.37
	FHF	16.75	0	19.75	36.5	0.59	14.75	0	18.25	33	0.50
	H	3.5	4.25	0	7.75	0.08	5	5.75	0	10.75	0.13
TC	HF	0	6.5	20	26.5	0.31	0	7.5	18	25.5	0.30
	FHF	17.5	0	19	36.5	0.60	16.5	0	19.5	36	0.59
	H	4	5	0	9	0.09	5	4.5	0	9	0.11

Table B.4: Combined preferences of gaze behaviors in the in-person study for the small and large object conditions. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, *L*: Likability, *A*: Anthropomorphism, *TC*: Timing Communication

		Small Object					Large Object				
		HF	FHF	H	a_i	P_i	HF	FHF	H	a_i	P_i
L	HF	0	6.5	20.75	27.25	0.29	0	7.25	20	27.25	0.31
	FHF	17.5	0	20.75	38.25	0.64	16.75	0	20.75	37.5	0.62
	H	3.25	3.25	0	6.5	0.07	4	3.25	0	7.25	0.07
A	HF	0	6.5	21	27.5	0.3	0	7.5	19	26.5	0.29
	FHF	17.5	0	20.75	38.25	0.64	16.5	0	21.25	37.75	0.63
	H	3	3.25	0	6.25	0.06	5	2.75	0	7.75	0.08
TC	HF	0	7.5	22	29.5	0.37	0	8.5	18.5	27	0.34
	FHF	16.5	0	19	35.5	0.55	15.5	0	19	34.5	0.54
	H	2	5	0	7	0.08	5.5	5	0	10.5	0.12

Table B.5: Combined preferences of gaze behaviors in the in-person study for the non-fragile object and fragile object conditions. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, *L*: Likability, *A*: Anthropomorphism, *TC*: Timing Communication

		Non-Fragile Object					Fragile Object				
		HF	FHF	H	a_i	P_i	HF	FHF	H	a_i	P_i
L	HF	0	5.75	22	27.75	0.24	0	8	22.5	30.5	0.36
	FHF	18.25	0	23	41.25	0.73	16	0	21.25	37.25	0.59
	H	2	1	0	3	0.03	1.5	2.75	0	4.25	0.05
A	HF	0	6.25	22.25	28.5	0.27	0	8	21.75	29.75	0.34
	FHF	17.75	0	22.75	40.5	0.70	16	0	21.5	37.5	0.61
	H	1.75	1.25	0	3	0.03	2.25	2.5	0	4.75	0.05
TC	HF	0	8	21.5	29.5	0.36	0	9	21	30	0.40
	FHF	16	0	20	36	0.57	15	0	18.5	33.5	0.50
	H	2.5	4	0	6.5	0.07	3	5.5	0	8.5	0.10

Table B.6: Combined preferences of gaze behaviors in the in-person study for the standing and sitting conditions. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, *L*: Likability, *A*: Anthropomorphism, *TC*: Timing Communication

		Standing					Sitting				
		HF	FHF	H	a_i	P_i	HF	FHF	H	a_i	P_i
L	HF	0	7.5	17.5	25	0.31	0	6.75	18.75	25.5	0.28
	FHF	16.5	0	17.75	34.25	0.54	17.25	0	20	37.25	0.62
	H	6.5	6.25	0	12.75	0.15	5.25	4	0	9.25	0.10
A	HF	0	8.25	16.5	24.75	0.31	0	7.5	17.75	25.25	0.28
	FHF	15.75	0	18	33.75	0.53	16.5	0	20.5	37	0.62
	H	7.5	6	0	13.5	0.16	6.25	3.5	0	9.75	0.10
TC	HF	0	9	18	27	0.35	0	6.5	19	25.5	0.31
	FHF	15	0	18.5	33.5	0.52	17.5	0	18	35.5	0.57
	H	6	5.5	0	11.5	0.13	5	6	0	11	0.12

Table B.7: Results of binary proportion difference test and equivalence test for matched pairs comparing small object and large object user's preferences of robot gaze in handovers. Gaze condition in bold is the preferred choice in each pairwise comparison. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, L: Likability, A: Anthropomorphism, TC: Timing Communication, BPDT: Binary Proportion Difference Test, ET: Equivalence Test.

	Metrics	Gaze Conditions	BPDT		ET	
			Z score	P-value	Z score	P-value
Video	L	HF vs. **FHF**	-0.15	0.44	0.39	0.35
		HF vs. H	0.00	0.50	-0.92	0.18
		FHF vs. H	0.38	0.36	0.19	0.58
	A	HF vs. **FHF**	-0.78	0.22	0.31	0.38
		HF vs. H	0.00	0.50	0.80	0.21
		FHF vs. H	0.27	0.40	-0.32	0.38
	TC	HF vs. **FHF**	0.41	0.34	-0.75	0.23
		HF vs. H	-0.16	0.44	-0.25	0.60
		FHF vs. H	0.65	0.26	0.04	0.52
In-Person	L	HF vs. **FHF**	-0.20	0.42	0.51	0.30
		HF vs. H	0.12	0.45	0.45	0.33
		FHF vs. H	0.00	0.50	0.96	0.17
	A	HF vs. **FHF**	-0.27	0.40	0.59	0.28
		HF vs. H	0.32	0.38	0.06	0.47
		FHF vs. H	-0.08	0.47	0.27	0.39
	TC	HF vs. **FHF**	-0.25	0.40	0.27	0.39
		HF vs. H	0.55	0.29	0.36	0.36
		FHF vs. H	0.00	0.50	0.30	0.38

Table B.8: Results of binary proportion difference test and equivalence test for matched pairs comparing fragile object and non-fragile object user's preferences of robot gaze in handovers. Gaze condition in bold is the preferred choice in each pairwise comparison. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, L: Likability, A: Anthropomorphism, TC: Timing Communication, BPDT: Binary Proportion Difference Test, ET: Equivalence Test.

	Metrics	Gaze Conditions	BPDT		ET	
			Z score	P-value	Z score	P-value
Video	L	HF vs. **FHF**	-0.27	0.39	0.29	0.38
		HF vs. H	-0.24	0.41	-0.57	0.28
		FHF vs. H	-0.08	0.47	-0.92	0.18
	A	HF vs. **FHF**	-0.41	0.34	-0.67	0.25
		HF vs. H	-0.20	0.42	-0.34	0.37
		FHF vs. H	-0.12	0.45	0.83	0.20
	TC	HF vs. **FHF**	-0.13	0.45	-0.03	0.51
		HF vs. H	0.00	0.50	-0.33	0.37
		FHF vs. H	0.00	0.50	0.31	0.38
In-Person	L	HF vs. **FHF**	-0.61	0.27	-0.20	0.58
		HF vs. H	-0.07	0.47	0.41	0.66
		FHF vs. H	0.26	0.40	0.32	0.62
	A	HF vs. **FHF**	-0.47	0.32	0.43	0.33
		HF vs. H	0.08	0.47	0.56	0.71
		FHF vs. H	0.19	0.43	-0.23	0.41
	TC	HF vs. **FHF**	-0.24	0.41	-0.03	0.51
		HF vs. H	0.08	0.47	-0.14	0.44
		FHF vs. H	0.24	0.41	-0.66	0.25

Table B.9: Results of binary proportion difference test and equivalence test for matched pairs comparing sitting and standing user's preferences of robot gaze in handovers. Gaze condition in bold is the preferred choice in each pairwise comparison. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, L: Likability, A: Anthropomorphism, TC: Timing Communication, BPDT: Binary Proportion Difference Test, ET: Equivalence Test.

	Metrics	Gaze Conditions	BPDT		ET	
			Z score	P-value	Z score	P-value
Video	L	HF vs. **FHF**	-1.08	0.15	-0.63	0.26
		HF vs. H	0.20	0.42	-0.04	0.52
		FHF vs. H	0.24	0.41	-0.05	0.52
	A	HF vs. **FHF**	-0.49	0.31	0.71	0.24
		HF vs. H	0.24	0.41	-0.65	0.74
		FHF vs. H	0.24	0.41	-0.60	0.72
	TC	HF vs. **FHF**	-0.27	0.40	0.44	0.33
		HF vs. H	0.33	0.37	0.16	0.43
		FHF vs. H	-0.08	0.47	-0.21	0.58
In-Person	L	HF vs. **FHF**	0.20	0.42	-0.28	0.39
		HF vs. H	-0.21	0.42	-0.51	0.31
		FHF vs. H	-0.37	0.36	0.72	0.24
	A	HF vs. **FHF**	0.19	0.43	-0.20	0.42
		HF vs. H	-0.21	0.42	-0.39	0.35
		FHF vs. H	-0.40	0.35	0.06	0.48
	TC	HF vs. **FHF**	0.64	0.26	-0.45	0.33
		HF vs. H	-0.16	0.44	0.48	0.32
		FHF vs. H	0.08	0.47	-0.32	0.38

Table B.10: Results of binary proportion difference test and equivalence test for unmatched pairs comparing video and in-person user's preferences of robot gaze in handovers (for small and large object sizes). Gaze condition in bold is the preferred choice in each pairwise comparison. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, L: Likability, A: Anthropomorphism, TC: Timing Communication, BPDT: Binary Proportion Difference Test, ET: Equivalence Test.

Object	Metrics	Gaze Conditions	BPDT		ET	
			Statistic	P-value	Statistic	P-value
Small	L	HF vs. **FHF**	-0.42	0.67	0.39	0.35
		HF vs. H	0.11	0.91	-0.92	0.18
		FHF vs. H	1.48	0.14	0.19	0.58
	A	HF vs. **FHF**	-0.51	0.61	0.31	0.38
		HF vs. H	-0.21	0.83	0.80	0.21
		FHF vs. H	0.85	0.40	-0.32	0.38
	T	HF vs. **FHF**	0.00	1.00	-0.75	0.23
		HF vs. H	-1.23	0.22	-0.25	0.60
		FHF vs. H	0.99	0.32	0.04	0.52
Large	L	HF vs. **FHF**	-0.49	0.62	0.29	0.38
		HF vs. H	0.41	0.68	-0.57	0.28
		FHF vs. H	0.11	0.91	-0.92	0.18
	A	HF vs. **FHF**	0.08	0.94	-0.67	0.25
		HF vs. H	0.57	0.57	-0.34	0.37
		FHF vs. H	-0.22	0.83	0.83	0.20
	T	HF vs. **FHF**	-0.79	0.43	-0.03	0.51
		HF vs. H	0.54	0.59	-0.33	0.37
		FHF vs. H	-0.51	0.61	0.31	0.38

Table B.11: Results of binary proportion difference test and equivalence test for unmatched pairs comparing video and in-person user's preferences of robot gaze in handovers (for non-fragile and fragile object sizes). Gaze condition in bold is the preferred choice in each pairwise comparison. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, L: Likability, A: Anthropomorphism, TC: Timing Communication, BPDT: Binary Proportion Difference Test, ET: Equivalence Test.

Object	Metrics	Gaze Conditions	BPDT		ET	
			Statistic	P-value	Statistic	P-value
Non-Fragile	L	HF vs. **FHF**	0.17	0.87	-0.63	0.26
		HF vs. H	-1.05	0.29	-0.04	0.52
		FHF vs. H	-1.24	0.21	-0.05	0.52
	A	HF vs. **FHF**	-0.08	0.94	0.71	0.24
		HF vs. H	-1.59	0.11	-0.65	0.74
		FHF vs. H	-1.61	0.11	-0.60	0.72
	T	HF vs. **FHF**	-0.31	0.76	0.44	0.33
		HF vs. H	-0.82	0.41	0.16	0.43
		FHF vs. H	-1.03	0.30	-0.21	0.58
Fragile	L	HF vs. **FHF**	-0.23	0.82	0.51	0.30
		HF vs. H	-0.74	0.46	0.45	0.33
		FHF vs. H	-0.11	0.91	0.96	0.17
	A	HF vs. **FHF**	-0.15	0.88	0.59	0.28
		HF vs. H	-0.93	0.35	0.06	0.47
		FHF vs. H	-0.73	0.47	0.27	0.39
	T	HF vs. **FHF**	-0.46	0.65	0.27	0.39
		HF vs. H	-0.60	0.55	0.36	0.36
		FHF vs. H	-0.49	0.62	0.30	0.38

Table B.12: Results of binary proportion difference test and equivalence test for unmatched pairs comparing video and in-person user's preferences of robot gaze in handovers (for the giver's standing and sitting postures). Gaze condition in bold is the preferred choice in each pairwise comparison. HF: Hand-Face Gaze, FHF: Face-Hand-Face Gaze, H: Hand Gaze, L: Likability, A: Anthropomorphism, TC: Timing Communication, BPDT: Binary Proportion Difference Test, ET: Equivalence Test.

Posture	Metrics	Gaze Conditions	BPDT		ET	
			Statistic	P-value	Statistic	P-value
Standing	L	HF vs. **FHF**	-1.00	0.32	-0.20	0.58
		HF vs. H	1.27	0.20	0.41	0.66
		FHF vs. H	1.19	0.23	0.32	0.62
	A	HF vs. **FHF**	-0.31	0.76	0.43	0.33
		HF vs. H	1.37	0.17	0.56	0.71
		FHF vs. H	0.62	0.54	-0.23	0.41
	T	HF vs. **FHF**	-0.77	0.44	-0.03	0.51
		HF vs. H	0.71	0.48	-0.14	0.44
		FHF vs. H	0.17	0.87	-0.66	0.25
Sitting	L	HF vs. **FHF**	0.47	0.64	-0.28	0.39
		HF vs. H	0.36	0.72	-0.51	0.31
		FHF vs. H	-0.19	0.85	0.72	0.24
	A	HF vs. **FHF**	0.53	0.60	-0.20	0.42
		HF vs. H	0.43	0.67	-0.39	0.35
		FHF vs. H	-0.82	0.41	0.06	0.48
	T	HF vs. **FHF**	0.32	0.75	-0.45	0.33
		HF vs. H	-0.34	0.73	0.48	0.32
		FHF vs. H	0.52	0.60	-0.32	0.38

BIBLIOGRAPHY

[1] Henny Admoni, Anca Dragan, Siddhartha S. Srinivasa, and Brian Scassellati. Deliberate delays during robot-to-human handovers improve compliance with gaze communication. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 49–56, 2014.

[2] David J Albers, Paul-Adrien Blancquart, Matthew E Levine, Elnaz Esmaeilzadeh Seylabi, and Andrew Stuart. Ensemble kalman methods with constraints. *Inverse Problems*, 35(9):095007, 2019.

[3] Jacopo Aleotti, Vincenzo Micelli, and Stefano Caselli. Comfortable robot to human object hand-over. In *IEEE International Conference on Robot and Human Interactive Communication*, pages 771–776, 2012.

[4] Jacopo Aleotti, Vincenzo Micelli, and Stefano Caselli. An affordance sensitive system for robot to human object handover. *International Journal of Social Robotics*, 6(4):653–666, June 2014.

[5] Dejanira Araiza-Illan, David Western, Anthony Pipe, and Kerstin Eder. Coverage-Driven Verification. In *Haifa Verification Conference*, pages 69–84, 2015.

[6] Audi. New human-robot cooperation in audi's production processes. `https://www.automotiveworld.com/news-releases/new-human-robot-cooperation-audis-production-processes/`, 2015. [Online; accessed 05-July-2022].

[7] Patrizia Basili, Markus Huber, Thomas Brandt, Sandra Hirche, and Stefan Glasauer. Investigating human-human approach and hand-over. In *Human Centered Robot Systems. Cognitive Systems Monographs*, volume 6, pages 151–160. Springer, 2009.

[8] Mohamad Bdiwi, Alexey Kolker, Jozef Suchỳ, and Alexander Winkler. Automated assistance robot system for transferring model-free objects from/to human hand using vision/force control. In *International Conference on Social Robotics*, pages 40–53, 2013.

[9] Dmitry Berenson, Siddhartha S Srinivasa, Dave Ferguson, and James J Kuffner. Manipulation planning on constraint manifolds. In *IEEE International Conference on Robotics and Automation*, pages 625–632, 2009.

[10] Aaron Bestick, Ruzena Bajcsy, and Anca D. Dragan. Implicitly assisting humans to choose good grasps in robot to human handovers. In *Springer Proceedings in Advanced Robotics*, pages 341–354. Springer International Publishing, 2017.

[11] Aaron Bestick, Ravi Pandya, Ruzena Bajcsy, and Anca D. Dragan. Learning human ergonomic preferences for handovers. In *IEEE International Conference on Robotics and Automation*, May 2018.

[12] Aaron M. Bestick, Samuel A. Burden, Giorgia Willits, Nikhil Naikal, S. Shankar Sastry, and Ruzena Bajcsy. Personalized kinematics for human-robot collaborative manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2015.

[13] Matteo Bianchi, Giuseppe Averta, Edoardo Battaglia, Carlos Rosales, Manuel Bonilla, Alessandro Tondo, Mattia Poggiani, Gaspare Santaera, Simone Ciotti, Manuel G. Catalano, and Antonio Bicchi. Touch-based grasp primitives for soft hands: Applications to human-to-robot handover tasks and beyond. In *IEEE International Conference on Robotics and Automation*, May 2018.

[14] Rainer Bischoff, Johannes Kurth, Günter Schreiber, Ralf Koeppe, Alin Albu-Schäffer, Alexander Beyer, Oliver Eiberger, Sami Haddadin, Andreas Stemmer, Gerhard Grunwald, et al. The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing. In *41st International Symposium on Robotics and 6th German Conference on Robotics*, pages 1–8, 2010.

[15] BMW. Innovative human–robot cooperation in bmw group production. `https://www.press.bmwgroup.com/global/article/attachment/T0209722EN/301973`, 2013. [Online; accessed 05-July-2022].

[16] Ralph Bradley and Milton Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[17] Mason Bretan, Guy Hoffman, and Gil Weinberg. Emotionally expressive dynamic physical behaviors in robots. *International Journal of Human-Computer Studies*, 78:1–16, 2015.

[18] Maya Cakmak, Siddhartha S Srinivasa, Min Kyung Lee, Jodi Forlizzi, and Sara Kiesler. Human preferences for robot-human hand-over configura-

tions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1986–1993, 2011.

[19] Maya Cakmak, Siddhartha S Srinivasa, Min Kyung Lee, Sara Kiesler, and Jodi Forlizzi. Using spatial and temporal contrast for fluent robot-human hand-overs. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 489–496, 2011.

[20] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[21] Joseph Campbell. *Probabilistic Imitation Learning for Spatiotemporal Human-Robot Interaction*. PhD thesis, Arizona State University, 2021.

[22] Joseph Campbell and Heni Ben Amor. Bayesian interaction primitives: A slam approach to human-robot interaction. In *Conference on Robot Learning*, pages 379–387, 2017.

[23] Joseph Campbell, Arne Hitzmann, Simon Stepputtis, Shuhei Ikemoto, Koh Hosoda, and Heni Ben Amor. Learning interactive behaviors for musculoskeletal robots using bayesian interaction primitives. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5071–5078, 2019.

[24] Joseph Campbell, Simon Stepputtis, and Heni Ben Amor. Probabilistic multimodal modeling for human-robot interaction tasks. *Robotics: Science and Systems*, 2019.

[25] Alessandro Carfì, Francesco Foglino, Barbara Bruno, and Fulvio Mastrogiovanni. A multi-sensor dataset of human-human handover. *Data in Brief*, 22:109 – 117, 2019.

[26] Wesley P Chan, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Determining proper grasp configurations for handovers through observation of object movement patterns and inter-object interactions during usage. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1355–1360, 2014.

[27] Wesley P. Chan, Kotaro Nagahama, Hiroaki Yaguchi, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Implementation of a framework for learning handover grasp configurations through observation during human-robot object handovers. In *IEEE-RAS International Conference on Humanoid Robots*, November 2015.

[28] Wesley P. Chan, Matthew K. X. J. Pan, Elizabeth A. Croft, and Masayuki Inaba. An affordance and distance minimization based method for computing object orientations for robot human handovers. *International Journal of Social Robotics*, 12(1):143–162, April 2019.

[29] Wesley P Chan, Matthew KXJ Pan, Elizabeth A Croft, and Masayuki Inaba. Characterization of handover orientations used by humans for efficient robot to human handovers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6, 2015.

[30] Wesley P. Chan, Chris A.C. Parker, H.F. Machiel Van der Loos, and Elizabeth A. Croft. Grip forces and load forces in handovers: Implications for designing human-robot handover controllers. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 9–16, 2012.

[31] Wesley P Chan, Chris Ac Parker, Hf Machiel Van Der Loos, and Elizabeth A Croft. A human-inspired object handover controller. *The International Journal of Robotics Research*, 32(8):971–983, 2013.

[32] Yevgen Chebotar, Mrinal Kalakrishnan, Ali Yahya, Adrian Li, Stefan Schaal, and Sergey Levine. Path integral guided policy search. In *IEEE International Conference on Robotics and Automation*, pages 3381–3388, 2017.

[33] SB Chen, T Qiu, T Lin, L Wu, JS Tian, WX Lv, and Y Zhang. Intelligent technologies for robotic welding. In *Robotic Welding, Intelligence and Automation*, pages 123–143. Springer, 2004.

[34] F. Cini, V. Ortenzi, P. Corke, and M. Controzzi. On the choice of grasp type and location when handing over an object. *Science Robotics*, 4(27):eaau9757, February 2019.

[35] J Edward Colgate, J Edward, Michael A Peshkin, and Witaya Wannasuphoprasit. Cobots: Robots for collaboration with human operators, 1996.

[36] Marco Controzzi, Harmeet Singh, Francesca Cini, Torquato Cecchini, Alan Wing, and Christian Cipriani. Humans adjust their grip force when passing an object according to the observed speed of the partner's reaching out movement. *Experimental Brain Research*, 236(12):3363–3377, September 2018.

[37] Mohammad-Javad Davari, Michael Hegedus, Kamal Gupta, and Mehran Mehrandezh. Identifying multiple interaction events from tactile data

during robot-human object transfer. In *IEEE International Conference on Robot and Human Interactive Communication*, October 2019.

[38] Elena De Momi, Laurens Kranendonk, Marta Valenti, Nima Enayati, and Giancarlo Ferrigno. A neural network-based approach for trajectory planning in robot–human handover tasks. *Frontiers in Robotics and AI*, 3:34, 2016.

[39] Jinyu Du, Jian Fu, and Cong Li. Guided policy search methods: A review. In *Journal of Physics: Conference Series*, volume 1748/2, page 022039, 2021.

[40] Aaron Edsinger and Charles C Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. In *IEEE International Conference on Robot and Human Interactive Communication*, pages 1167–1172, 2007.

[41] A. Gómez Eguíluz, I. Rañó, S. A. Coleman, and T. M. McGinnity. Reliable robotic handovers through tactile sensing. *Autonomous Robots*, 43(7):1623–1637, January 2019.

[42] Satoshi Endo, Geoff Pegman, Mark Burgin, Tarek Toumi, and Alan M Wing. Haptics in between-person object transfer. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 103–111, 2012.

[43] Mustafa Suphi Erden, Kemal Leblebicioğlu, and Uğur Halici. Multi-agent system-based fuzzy controller design with genetic tuning for a mobile manipulator robot in the hand over task. *Journal of Intelligent and Robotic Systems*, 39(3):287–306, 2004.

[44] Marco Ewerton, Gerhard Neumann, Rudolf Lioutikov, Heni Ben Amor, Jan Peters, and Guilherme Maeda. Learning multiple collaborative tasks with a mixture of interaction primitives. In *IEEE International Conference on Robotics and Automation*, May 2015.

[45] Tair Faibish. Human-robot handovers: Human preferences and robot learning. Master's thesis, Department of Industrial Engineering and Management, Ben-Gurion university of the Negev, Beer Sheva, Israel, 2022.

[46] Tair Faibish*, Alap Kshirsagar*, Guy Hoffman, and Yael Edan. Human preferences for robot eye gaze in human-to-robot handovers. *International Journal of Social Robotics*, pages 1–18, 2022.

[47] Chelseaa Finn, Marvin Zhang, Justin Fu, William Montgomery, Xin Yu Tan, Zoe McCarthy, Bradly Stadie, Emily Scharff, and Sergey. Levine. Guided policy search code implementation. `https://rll.berkeley.edu/gps/`. [Online; accessed 05-July-2022].

[48] Kerstin Fischer, Lars Christian Jensen, Franziska Kirstein, Sebastian Stabinger, Özgür Erkent, Dadhichi Shukla, and Justus Piater. The effects of social gaze in human-robot collaborative assembly. In *International Conference on Social Robotics*, pages 204–213, 2015.

[49] Adam Fishman, Chris Paxton, Wei Yang, Nathan Ratliff, and Dieter Fox. Trajectory optimization for coordinated human-robot collaboration. *arXiv preprint arXiv:1910.04339*, 2019.

[50] J Randall Flanagan and Roland S Johansson. Action plans used in action observation. *Nature*, 424(6950):769, 2003.

[51] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985.

[52] Felix Frank, Alexandros Paraschos, Patrick van der Smagt, and Botond Cseke. Constrained probabilistic movement primitives for robot trajectory adaptation. *IEEE Transactions on Robotics*, 2021.

[53] Willow Garage. PR2 user manual. `https://www.clearpathrobotics.com/assets/downloads/pr2/pr2_manual_r321.pdf`, 2012. [Online; accessed 05-July-2022].

[54] Bill Gates. A Robot in Every Home. *Scientific American*, 296(1):58–65, Jan 2007.

[55] Elena Corina Grigore, Kerstin Eder, Anthony G. Pipe, Chris Melhuish, and Ute Leonards. Joint action understanding improves robot-to-human object handover. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, November 2013.

[56] Martin Hägele, Klas Nilsson, and Norberto J Pires. *Industrial Robotics*. Springer, 2008.

[57] Zhao Han and Holly Yanco. The effects of proactive release behaviors

during human-robot handovers. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 440–448, 2019.

[58] Clint Hansen, Paula Arambel, Khalil Ben Mansour, Véronique Perdereau, and Frédéric Marin. Human–human handover tasks and how distance and object mass matter. *Perceptual and Motor Skills*, 124(1):182–199, December 2016.

[59] Wei He, Jiashu Li, Zichen Yan, and Fei Chen. Bidirectional human-robot bimanual handover of big planar object with vertical posture. *IEEE Transactions on Automation Science and Engineering*, 2021.

[60] Wuwei He, Daniel Sidobre, and Ran ZHAO. A Reactive Trajectory Controller for Object Manipulation in Human Robot Interaction. In *International Conference on Informatics in Control, Automation and Robotics*, July 2013.

[61] Edmond SL Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. In *ACM SIGGRAPH 2010 papers*, pages 1–8. ACM, 2010.

[62] Guy Hoffman. Evaluating fluency in human-robot collaboration. In *Robotics: Science and Systems Workshop on Human Robot Collaboration*, 2013.

[63] Chien-Ming Huang, Maya Cakmak, and Bilge Mutlu. Adaptive coordination strategies for human-robot handovers. In *Robotics: Science and Systems*, 2015.

[64] M Huber, A Knoll, T Brandt, and S Glasauer. Handing over a cube: spatial features of physical joint-action. *Annals of the New York Academy of Sciences*, 1164:380–382, 2009.

[65] Markus Huber, Claus Lenz, Cornelia Wendt, Berthold Farber, Alois Knoll, and Stefan Glasauer. Increasing efficiency in robot-supported assemblies through predictive mechanisms: An experimental evaluation. In *IEEE International Conference on Robot and Human Interactive Communication*, August 2013.

[66] Markus Huber, Markus Rickert, Alois Knoll, Thomas Brandt, and Stefan Glasauer. Human-robot interaction in handing-over tasks. In *IEEE International Conference on Robot and Human Interactive Communication*, pages 107–112, 2008.

[67] David Hunter. MM algorithms for generalized bradley-terry models. *The Annals of Statistics*, 32(1):384–406, February 2003.

[68] IFR. Executive summary world robotics 2019 industrial robots. `https://ifr.org/downloads/press2018/Executive%20Summary%20WR%202019%20Industrial%20Robots.pdf`, 2019. [Online; accessed 05-July-2022].

[69] Barbara Illowsky and Susan Dean. *Introductory statistics*. OpenStax, Rice University, Houston, Texas, USA, 2018.

[70] Florian Jentsch. *Human-robot interactions in future military operations*. CRC Press, 2016.

[71] Saki Kato, Natsuki Yamanobe, Gentiane Venture, and Gowrishankar Ganesh. Humans can predict where their partner would make a handover. In *Companion of the ACM/IEEE International Conference on Human-Robot Interaction*, 2018.

[72] Saki Kato, Natsuki Yamanobe, Gentiane Venture, Eiichi Yoshida, and Gowrishankar Ganesh. The where of handovers by humans: Effect of partner characteristics, distance and visual feedback. *PloS one*, 14(6), 2019.

[73] Philipp Kellmeyer, Oliver Mueller, Ronit Feingold-Polak, and Shelly Levy-Tzedek. Social robots in rehabilitation: A question of trust. *Science Robotics*, 3:eaat1587, 2018.

[74] Jinsul Kim, Jihwan Park, Yong K Hwang, and Manjai Lee. Advanced grasp planning for handover operation between human and robot. In *International Conference on Autonomous Robots and Agents*, pages 13–15, 2004.

[75] Kheng Lee Koay, Emrah Akin Sisbot, Dag Sverre Syrdal, Mick L Walters, Kerstin Dautenhahn, and Rachid Alami. Exploratory study of a robot approaching a person in the context of handing over an object. In *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, pages 18–24, 2007.

[76] Ansgar Koene, Anthony Remazeilles, Miguel Prada, Ainara Garzo, Mildred Puerto, Satoshi Endo, and Alan M Wing. Relative importance of spatial and temporal precision for user satisfaction in human-robot object handover interactions. In *International Symposium on New Frontiers in Human-Robot Interaction*, 2014.

[77] Jelizaveta Konstantinova, Senka Krivic, Agostino Stilli, Justus Piater, and Kaspar Althoefer. Autonomous object handover using wrist tactile information. In *Annual Conference Towards Autonomous Robotic Systems*, pages 450–463, 2017.

[78] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. Synthesis for Robots: Guarantees and Feedback for robot behavior. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:211–236, 2018.

[79] Alap Kshirsagar, Guy Hoffman, and Armin Biess. Evaluating guided policy search for human-robot handovers. *IEEE Robotics and Automation Letters*, 6(2):3933–3940, 2021.

[80] Alap Kshirsagar, Hadas Kress-Gazit, and Guy Hoffman. Specifying and synthesizing human-robot handovers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5930–5936, 2019.

[81] Alap Kshirsagar, Melanie Lim, Shemar Christian, and Guy Hoffman. Robot gaze behaviors in human-to-robot handovers. *IEEE Robotics and Automation Letters*, 5(4):6552–6558, 2020.

[82] Alap Kshirsagar*, Rahul Kumar Ravi*, Hadas Kress-Gazit, and Guy Hoffman. Timing-specified controllers with feedback for human-robot handovers. In *IEEE International Conference on Robot and Human Interactive Communication*, August 2022.

[83] Barbara Kühnlenz, Zhi-Qiao Wang, and Kolja Kühnlenz. Impact of continuous eye contact of a humanoid robot on user experience and interactions with professional user background. In *IEEE International Conference on Robot and Human Interactive Communication*, pages 1037–1042, 2017.

[84] Andras Kupcsik, David Hsu, and Wee Sun Lee. Learning dynamic robot-to-human object handover from human feedback. *Robotics Research*, 1:161, 2017.

[85] Daniël Lakens, Anne M Scheel, and Peder M Isager. Equivalence testing for psychological research: A tutorial. *Advances in Methods and Practices in Psychological Science*, 1(2):259–269, 2018.

[86] Jens Lambrecht and Sebastian Nimpsch. Human prediction for the natural instruction of handovers in human robot collaboration. In *IEEE International Conference on Robot and Human Interactive Communication*, October 2019.

[87] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business and Information Systems Engineering*, 6(4):239–242, 2014.

[88] Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and evaluation of a user experience questionnaire. In *Lecture Notes in Computer Science*, pages 63–76. Springer Berlin Heidelberg, 2008.

[89] Daniel Leal and Yimesker Yihun. Progress in human-robot collaboration for object handover. In *IEEE International Symposium on Measurement and Control in Robotics*, September 2019.

[90] Min Kyung Lee, Jodi Forlizzi, Sara Kiesler, Maya Cakmak, and Siddhartha Srinivasa. Predictability or adaptivity? designing robot handoffs modeled from trained dogs and people. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 179–180, 2011.

[91] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*, pages 1071–1079, 2014.

[92] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[93] Sergey Levine and Vladlen Koltun. Guided policy search. In *International Conference on Machine Learning*, pages 1–9, 2013.

[94] Sergey Levine and Vladlen Koltun. Variational policy search via trajectory optimization. In *Advances in Neural Information Processing Systems*, pages 207–215, 2013.

[95] Sergey Levine and Vladlen Koltun. Learning complex neural network policies with trajectory optimization. In *International Conference on Machine Learning*, page II–829–II–837, 2014.

[96] Sergey Levine, Nolan Wagener, and Pieter Abbeel. Learning contact-rich manipulation skills with guided policy search. In *IEEE International Conference on Robotics and Automation*, May 2015.

[97] Wenchao Li, Dorsa Sadigh, S Shankar Sastry, and Sanjit A Seshia. Synthesis for human-in-the-loop control systems. In *International Conference*

on Tools and Algorithms for the Construction and Analysis of Systems, pages 470–484, 2014.

[98] Lars Lindemann and Dimos V Dimarogonas. Control barrier functions for signal temporal logic tasks. *IEEE Control Systems Letters*, 3(1):96–101, 2018.

[99] Ross G Macdonald and Benjamin W Tatler. Gaze in a real-world social interaction: A dual eye-tracking study. *Quarterly Journal of Experimental Psychology*, 71(10):2162–2173, 2018.

[100] Guilherme Maeda, Marco Ewerton, Rudolf Lioutikov, Heni Ben Amor, Jan Peters, and Gerhard Neumann. Learning interaction for collaborative tasks with probabilistic movement primitives. In *IEEE-RAS International Conference on Humanoid Robots*, November 2014.

[101] Guilherme Maeda, Gerhard Neumann, Marco Ewerton, Rudolf Lioutikov, Oliver Kroemer, and Jan Peters. Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Autonomous Robots*, 41(3):593–612, March 2016.

[102] Jim Mainprice, Mamoun Gharbi, Thierry Siméon, and Rachid Alami. Sharing effort in planning human-robot handover tasks. In *IEEE International Conference on Robot and Human Interactive Communication*, pages 764–770, 2012.

[103] Jim Mainprice, Emrah Akin Sisbot, Thierry Simeon, and Rachid Alami. Planning safe and legible hand-over motions for human-robot interaction. In *IARP/IEEE-RAS/EURON Workshop on Technical Challenges for Dependable Robots in Human Environments*, 2010.

[104] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004.

[105] E. Martinson, A. Huaman Quispe, and K. Oguchi. Towards understanding user preferences in robot-human handovers: How do we decide? In *IEEE International Conference on Robot and Human Interactive Communication*, August 2017.

[106] Naresh Marturi, Marek Kopicki, Alireza Rastegarpanah, Vijaykumar Rajasekaran, Maxime Adjigble, Rustam Stolkin, Aleš Leonardis, and

Yasemin Bekiroglu. Dynamic grasp and trajectory planning for moving objects. *Autonomous Robots*, 43(5):1241–1256, August 2018.

[107] Andrea H Mason and Christine L MacKenzie. Grip forces when passing an object to a partner. *Experimental Brain Research*, 163(2):173–187, 2005.

[108] Warren L May and William D Johnson. The validity and power of tests for equality of two correlated proportions. *Statistics in Medicine*, 16(10):1081–1096, 1997.

[109] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[110] José R Medina, Felix Duvallet, Murali Karnam, and Aude Billard. A human-inspired controller for fluid human-robot handovers. In *IEEE-RAS International Conference on Humanoid Robots*, pages 324–331, 2016.

[111] Matteo Melchiorre, Leonardo Sabatino Scimmi, Stefano Mauro, and Stefano Pastorelli. Influence of human limb motion speed in a collaborative hand-over task. In *International Conference on Informatics in Control, Automation and Robotics*, 2018.

[112] Vincenzo Micelli, Kyle Strabala, and Siddhartha Srinivasa. Perception and control challenges for effective human-robot handoffs. In *Robotics: Science and Systems Workshop on RGB-D Cameras*, 2011.

[113] G Michalos, S Makris, N Papakostas, D Mourtzis, and G Chryssolouris. Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach. *CIRP Journal of Manufacturing Science and Technology*, 2(2):81–91, 2010.

[114] William H Montgomery and Sergey Levine. Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems*, pages 4008–4016, 2016.

[115] AJung Moon, Daniel M. Troniak, Brian Gleeson, Matthew K.X.J. Pan, Minhua Zheng, Benjamin A. Blumer, Karon MacLean, and Elizabeth A. Croft. Meet me where i'm gazing: How shared attention gaze affects human-robot handover timing. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 334–341, 2014.

[116] Toshihiko Morikawa, Takashi Yanagawa, Akira Endou, and Isao

Yoshimura. Equivalence tests for pair-matched binary data. *Bulletin of Informatics and Cybernetics*, 28(1):31–46, 1996.

[117] Amir M Naghsh, Jeremi Gancet, Andry Tanoto, and Chris Roast. Analysis and design of human-robot swarm interaction in firefighting. In *IEEE International Conference on Robot and Human Interactive Communication*, pages 255–260, 2008.

[118] Heramb Nemlekar, Dharini Dutia, and Zhi Li. Object transfer point estimation for fluent human-robot handovers. In *IEEE International Conference on Robotics and Automation*, May 2019.

[119] Paramin Neranon. Robot-to-human object handover using a behavioural control strategy. In *IEEE International Conference on Smart Instrumentation, Measurement and Application*, November 2018.

[120] Paramin Neranon. Implicit force control approach for safe physical robot-to-human object handover. *Indonesian Journal of Electrical Engineering and Computer Science*, 17(2):615, February 2020.

[121] Research Nester. Military robots market - global industry demand analysis & opportunity assessment 2016–2024. `https://www.researchnester.com/reports/military-robots-market/1221`, 2019. [Online; accessed 05-July-2022].

[122] Valerio Ortenzi, Akansel Cosgun, Tommaso Pardi, Wesley P Chan, Elizabeth Croft, and Dana Kulić. Object handovers: a review for robotics. *IEEE Transactions on Robotics*, 2021.

[123] Matthew K. X. J. Pan, Elizabeth A. Croft, and Gunter Niemeyer. Exploration of geometry and forces occurring within human-to-robot handovers. In *IEEE Haptics Symposium*, March 2018.

[124] Matthew KXJ Pan, Elizabeth A Croft, and Günter Niemeyer. Evaluating social perception of human-to-robot handovers using the robot social attributes scale (rosas). In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 443–451, 2018.

[125] Matthew KXJ Pan, Espen Knoop, Moritz Bächer, and Günter Niemeyer. Fast handovers with a robot character: Small sensorimotor delays improve perceived qualities. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[126] Matthew KXJ Pan, Vidar Skjervøy, Wesley P Chan, Masayuki Inaba, and Elizabeth A Croft. Automated detection of handovers using kinematic features. *The International Journal of Robotics Research*, 36(5-7):721–738, February 2017.

[127] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[128] Sina Parastegari, Bahareh Abbasi, Ehsan Noohi, and Miloš Zefran. Modeling human reaching phase in human-human object handover with application in robot-human handover. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3597–3602, 2017.

[129] Sina Parastegari, Ehsan Noohi, Bahareh Abbasi, and Milos Zefran. A fail-safe object handover controller. In *IEEE International Conference on Robotics and Automation*, May 2016.

[130] Sina Parastegari, Ehsan Noohi, Bahareh Abbasi, and Milos Zefran. Failure recovery in robot–human object handover. *IEEE Transactions on Robotics*, 34(3):660–673, June 2018.

[131] Luka Peternel, Wansoo Kim, Jan Babic, and Arash Ajoudani. Towards ergonomic control of human-robot co-manipulation and handover. In *IEEE-RAS International Conference on Humanoid Robotics*, November 2017.

[132] Miguel Prada, Anthony Remazeilles, Ansgar Koene, and Satoshi Endo. Dynamic movement primitives for human-robot interaction: comparison with human behavioral observation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1168–1175, 2013.

[133] Miguel Prada, Anthony Remazeilles, Ansgar Koene, and Satoshi Endo. Implementation and experimental validation of Dynamic Movement Primitives for object handover. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2146–2153, 2014.

[134] Efi Psomopoulou and Zoe Doulgeri. A human inspired stable object load transfer for robots in hand-over tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 491–496, 2015.

[135] Yoonseok Pyo, Kouhei Nakashima, Tokuo Tsuji, Ryo Kurazume, and Ken'ichi Morooka. Motion planning for fetch-and-give task using wagon and service robot. In *IEEE International Conference on Advanced Intelligent Mechatronics*, July 2015.

[136] Ana C. Huaman Quispe, Eric Martinson, and Kentaro Oguchi. Learning user preferences for robot-human handovers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2017.

[137] SM Mizanoor Rahman, Yue Wang, Ian D Walker, Laine Mears, Richard Pak, and Sekou Remy. Trust-based compliant robot-human handovers of payloads in collaborative assembly in flexible manufacturing. In *IEEE International Conference on Automation Science and Engineering*, pages 355–360, 2016.

[138] Vasumathi Raman and Alexandre Donzè. BluSTL on Github. `https://github.com/BluSTL/BluSTL`. [Online; accessed 05-July-2022].

[139] Vasumathi Raman, Alexandre Donzé, Mehdi Maasoumy, Richard M Murray, Alberto Sangiovanni-Vincentelli, and Sanjit A Seshia. Model predictive control with signal temporal logic specifications. In *IEEE Annual Conference on Decision and Control*, pages 81–87, 2014.

[140] Vasumathi Raman, Alexandre Donzé, Dorsa Sadigh, Richard M Murray, and Sanjit A Seshia. Reactive synthesis from signal temporal logic specifications. In *International Conference on Hybrid Systems: Computation and Control*, pages 239–248, 2015.

[141] Robin Rasch, Sven Wachsmuth, and Matthias Kcînig. Understanding movements of hand-over between two persons to improve humanoid robot systems. In *IEEE-RAS International Conference on Humanoid Robotics*, pages 856–861, 2017.

[142] Robin Rasch, Sven Wachsmuth, and Matthias Konig. A joint motion model for human-like robot-human handover. In *IEEE-RAS International Conference on Humanoid Robots*, November 2018.

[143] Robin Rasch, Sven Wachsmuth, and Matthias Konig. Combining cartesian trajectories with joint constraints for human-like robot-human handover. In *IEEE-RAS International Conference on Humanoid Robots*, October 2019.

[144] Robin Rasch, Sven Wachsmuth, and Matthias König. An evaluation of robot-to-human handover configurations for commercial robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7588–7595, 2019.

[145] Rahul Kumar Ravi. Timing controller for human-robot object handover: Implementation and evaluation. Master's thesis, Cornell University, 2021.

[146] Christopher Reardon and Jonathan Fink. Air-ground robot team surveillance of complex 3d environments. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 320–327, 2016.

[147] Francesco Riccio, Roberto Capobianco, and Daniele Nardi. Learning human-robot handovers through π-STAM: Policy improvement with spatio-temporal affordance maps. In *IEEE-RAS International Conference on Humanoid Robots*, November 2016.

[148] Someshwar Roy and Yael Edan. Investigating joint-action in short-cycle repetitive handover tasks: The role of giver versus receiver and its implications for human-robot collaborative system design. *International Journal of Social Robotics*, March 2018.

[149] Seyed Sina Mirrazavi Salehian, Nadia Figueroa, and Aude Billard. A unified framework for coordinated multi-arm motion planning. *The International Journal of Robotics Research*, 37(10):1205–1232, April 2018.

[150] Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

[151] Leonardo Sabatino Scimmi, Matteo Melchiorre, Stefano Mauro, and Stefano Pastorelli. Experimental real-time setup for vision driven hand-over with a collaborative robot. In *International Conference on Control, Automation and Diagnosis*, July 2019.

[152] Sara Sheikholeslami, Gilwoo Lee, Justin W Hart, Siddhartha Srinivasa, and Elizabeth A Croft. A study of reaching motions for collaborative human-robot interaction. In *International Symposium on Experimental Robotics*, pages 584–594, 2018.

[153] Chao Shi, Satoru Satake, Takayuki Kanda, and Hiroshi Ishiguro. A robot that distributes flyers to pedestrians in a shopping mall. *International Journal of Social Robotics*, 10(4):421–437, November 2017.

[154] Chao Shi, Masahiro Shiomi, Christian Smith, Takayuki Kanda, and Hiroshi Ishiguro. A model of distributional handing interaction for a mobile robot. In *Robotics: Science and Systems*, pages 24–28, 2013.

[155] S. Shibata, B.M. Sahbi, K. Tanaka, and A. Shimizu. An analysis of the process of handing over an object and its application to robot motions. In

IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, 1997.

[156] Satoru Shibata, Kanya Tanaka, and Akira Shimizu. Experimental analysis of handing over. In *IEEE International Workshop on Robot and Human Communication*, pages 53–58, 1995.

[157] Antonis Sidiropoulos, Efi Psomopoulou, and Zoe Doulgeri. A human inspired handover policy using gaussian mixture models and haptic cues. *Autonomous Robots*, 43(6):1327–1342, 2019.

[158] Janneke Simmering, Sebastian Meyer zu Borgsen, Sven Wachsmuth, and Ayoub Al-Hamadi. Combining static and dynamic predictions of transfer points for human initiated handovers. In *Social Robotics*, pages 676–686. Springer International Publishing, 2019.

[159] Seyed Sina Mirrazavi Salehian, Nadia Figueroa, and Aude Billard. Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In *Robotics: Science and Systems*, 2016.

[160] Harmeet Singh, Marco Controzzi, Christian Cipriani, Gaetano Di Caterina, Lykourgos Petropoulakis, and John Soraghan. Online prediction of robot to human handover events using vibrations. In *European Signal Processing Conference*, September 2018.

[161] Emrah Akin Sisbot and Rachid Alami. A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(5):1045–1057, 2012.

[162] Kyle Strabala, Min Kyung Lee, Anca Dragan, Jodi Forlizzi, and Siddhartha S Srinivasa. Learning the communication of intent prior to physical collaboration. In *IEEE International Conference on Robot and Human Interactive Communication*, pages 968–973, 2012.

[163] Kyle Strabala, Min Kyung Lee, Anca Dragan, Jodi Forlizzi, Siddhartha S Srinivasa, Maya Cakmak, and Vincenzo Micelli. Toward seamless human-robot handovers. *Journal of Human-Robot Interaction*, 2(1):112–132, 2013.

[164] Halit Bener Suay and Emrah Akin Sisbot. A position generation algorithm utilizing a biomechanical model for robot-human object handover. In *IEEE International Conference on Robotics and Automation*, May 2015.

[165] Huajin Tang, Boon Hwa Tan, and Rui Yan. Robot-to-human handover

with obstacle avoidance via continuous time recurrent neural network. In *IEEE Congress on Evolutionary Computation*, pages 1204–1211, 2016.

[166] Kuo-Hao Tang, Chia-Feng Ho, Jan Mehlich, and Shih-Ting Chen. Assessment of handover prediction models in estimation of cycle times for manual assembly tasks in a human–robot collaborative environment. *Applied Sciences*, 10(2):556, January 2020.

[167] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2012.

[168] Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248–266, 2018.

[169] David Vogt, Simon Stepputtis, Bernhard Jung, and Heni Ben Amor. One-shot learning of human–robot handovers with triadic interaction meshes. *Autonomous Robots*, 42(5):1053–1065, 2018.

[170] Jules Waldhart, Mamoun Gharbi, and Rachid Alami. Planning handovers involving humans and robots in constrained environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6473–6478, 2015.

[171] Ian D. Walker, Laine Mears, Rahman S. M. Mizanoor, Richard Pak, Sekou Remy, and Yue Wang. Robot-human handovers based on trust. In *International Conference on Mathematics and Computers in Sciences and in Industry*, August 2015.

[172] M. L. Walters, K. Dautenhahn, S. N. Woods, K. L. Koay, R. Te Boekhorst, and D. Lee. Exploratory studies on social spaces between humans and a mechanical-looking robot. *Connection Science*, 18(4):429–439, December 2006.

[173] Chen Wang, Liang Peng, Zeng-Guang Hou, Lincong Luo, Sheng Chen, and Weiqun Wang. Experimental validation of minimum-jerk principle in physical human-robot interaction. In *International Conference on Neural Information Processing*, pages 499–509, 2018.

[174] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[175] Weitian Wang, Rui Li, Zachary Max Diekel, Yi Chen, Zhujun Zhang, and Yunyi Jia. Controlling object hand-over in human–robot collaboration via natural wearable sensing. *IEEE Transactions on Human-Machine Systems*, 49(1):59–71, February 2019.

[176] Matt Webster, David Western, Dejanira Araiza-Illan, Clare Dixon, Kerstin Eder, Michael Fisher, and Anthony G Pipe. An assurance-based approach to verification and validation of human–robot teams. *arXiv preprint arXiv:1608.07403*, 2016.

[177] Jack White. Guided policy search for a lightweight industrial robot arm. Master's thesis, Luleå University of Technology and Aalto University/Erasmus+, 2018.

[178] Dominik Widmann and Yiannis Karayiannidis. Human motion prediction in human-robot handovers based on dynamic movement primitives. In *European Control Conference*, June 2018.

[179] Min Wu, Bertram Taetz, Ernesto Dickel Saraiva, Gabriele Bleser, and Steven Liu. On-line motion prediction and adaptive control in human-robot handover tasks. In *2019 IEEE International Conference on Advanced Robotics and its Social Impacts*, October 2019.

[180] Katsu Yamane, Marcel Revfi, and Tamim Asfour. Synthesizing object receiving motions of humanoid robots with human motion database. In *IEEE International Conference on Robotics and Automation*, pages 1629–1636, 2013.

[181] Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. How contingent should a communication robot be? In *ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, 2006.

[182] Wei Yang, Chris Paxton, Maya Cakmak, and Dieter Fox. Human grasp classification for reactive human-to-robot handovers. *arXiv preprint arXiv:2003.06000*, 2020.

[183] Ruolin Ye, Wenqiang Xu, Zhendong Xue, Tutian Tang, Yanfeng Wang, and Cewu Lu. H2o: A benchmark for visual human-human object handover analysis. In *IEEE/CVF International Conference on Computer Vision*, pages 15762–15771, 2021.

[184] Andy Zelenak, Clinton Peterson, Jack Thompson, and Mitch Pryor. The

advantages of velocity control for reactive robot motion. In *ASME Dynamic Systems and Control Conference*, pages V003T43A003–V003T43A003, 2015.

[185] Tianhao Zhang, Gregory Kahn, Sergey Levine, and Pieter Abbeel. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In *IEEE International Conference on Robotics and Automation*, pages 528–535, 2016.

[186] Xuan Zhao, Sakmongkon Chumkamon, Shuanda Duan, Juan Rojas, and Jia Pan. Collaborative human-robot motion generation using LSTM-RNN. In *IEEE-RAS International Conference on Humanoid Robots*, November 2018.

[187] Minhua Zheng, A Jung Moon, Brian Gleeson, Daniel M. Troniak, Matthew K. X. J. Pan, Benjamin A. Blumer, Max Q. H. Meng, and Elizabeth A. Croft. Human behavioural responses to robot head gaze during robot-to-human handovers. In *IEEE International Conference on Robotics and Biomimetics*, December 2014.

[188] Minhua Zheng, AJung Moon, Elizabeth A Croft, and Max Q-H Meng. Impacts of robot head gaze on robot-to-human handovers. *International Journal of Social Robotics*, 7(5):783–798, 2015.

[189] Sebastian Meyer zu Borgsen, Jasmin Bernotat, and Sven Wachsmuth. Hand in hand with robots: differences between experienced and naive users in human-robot handover scenarios. In *International Conference on Social Robotics*, pages 587–596, 2017.