NORTHWESTERN UNIVERSITY

Mathematical Models for Human-Robot Systems in Assistive Robotics: Perception, Inference, and Assistance

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Siddarth Jain

EVANSTON, ILLINOIS

September 2019

ABSTRACT

Mathematical Models for Human-Robot Systems in Assistive Robotics: Perception, Inference, and Assistance

Siddarth Jain

Assistive robotics focuses on human-robot systems that provide physical support and assistance to the elderly and people with motor-impairments. While assistive machines, such as the powered wheelchair, can significantly enhance the functional independence of individuals, many users are challenged by their direct operation, the way such systems are currently operated by the users. Moreover, as assistive machines become more capable, they often become more complex to control. This means paradoxically the more severe a person's motor-impairment, the more challenging it is for them to operate the very assistive machines meant to aid them.

An enduring goal is to address this discrepancy by incorporating robotics autonomy and intelligence into assistive machines to help ease the control burden on the user. Such human-robot systems emphasize proximate interaction, forming personal and collaborative relationships, and sharing control with human personnel. Thus, the design, sensing, control, and assessment of such systems becomes more sophisticated due to a *human-in-the-loop*. For autonomously providing assistance in shared autonomy, the robotics autonomy must be capable of: perceiving the potential goals of the human user, inferring user intentions, and sharing control with the user in a manner that is acceptable to the human and at the same time is efficient for task executions. These behaviors engage multiple disciplines including computer vision, machine learning, robotics, control systems, human psychology, and cognitive science.

In this dissertation, we focus on building mathematical models and algorithms for autonomous perception, inference, and assistance in human-robot systems for assistive robotics. Specifically we investigate, the computational perception of navigation goals involving wheelchair docking at table and desk structures, and manipulation goals for assistive robotic arms involving the detection of semantic grasp types on novel household objects. We investigate human intent recognition in shared autonomy by Bayesian filtering and modeling human actions in a probabilistic behavior model. For assistance personalization and adaptation, we investigate an intent-driven optimization that adapts the model to each individual user. Furthermore, we also investigate and present a novel application of body-machine interface in human-robot systems, which engage users in sustained physical activities with the aim to support partial recovery of movement skills. We validate all contributed algorithms and techniques in this dissertation on real hardware, using a wheelchair robot or a robotic arm platform. We conduct human subject experiments in a variety of shared-autonomy settings and report our findings. This dissertation contributes to and across multiple disciplines, providing a greater understanding of the computational and human requirements for successful human-robot systems in the assistive domain.

Keywords: Human-robot systems, assistive robotics, shared autonomy, human-in-the-loop, assistive teleoperation, human-robot interaction, robot perception, grasping, robot docking, intent inference, human intent recognition, probabilistic modeling, user personalization, human-robot interfaces

Acknowledgements

This dissertation marks an important milestone in my life, and I owe many people much gratitude for their support and encouragement. I wholeheartedly thank my Ph.D. advisor, Brenna Argall, for her invaluable supervision, supplying me with enough guidance to stay on the right track, and also for providing me enough freedom to pursue exciting research problems and applications. As a mentor, Brenna introduced me to the field of assistive robotics research, and I truly got inspired and became passionate about working on human-robot systems to benefit people in their daily lives. I am profoundly grateful to her for the continuous encouragement and support over the course of my doctoral studies.

I am also indebted to Brian Scassellati, Bryan Pardo and Michael Rubenstien for serving on my thesis committee and providing constructive feedback and suggestions. I want to pay special regards to Bryan Pardo, especially for always making time to meet with me and providing me with a unique perspective on facing challenges with positive attitude.

I feel extremely lucky to be part of the assistive & rehabilitation robotics laboratory (argallab) and the Shirley Ryan AbilityLab (formerly the Rehabilitation Institute of Chicago), during my Ph.D. journey. I thank all my lab-mates and colleagues for being a wonderful and supportive group. Thank you Ahmetcan, Alex, Chris, Deepak, Mahdieh, Matt and Michael. I will miss the lab lunches, hackathons, and social nights. I will also miss the annual lab dinner, thank you Brenna for the fondue dinner nights. I thank all my collaborators, and especially thanks to Jessica Pederson for her support during subject studies and helping me in subject recruitment for the studies. I am grateful to the EECS and TGS staff for

their prompt assistance in responding to my questions and requests. I would like to thank K Madhava Krishna, Katherine Barsness, and John Lizzi for providing me with wonderful summer opportunities to work on exciting research projects. I thank Stefanos Nikolaidis, Laura Herlant, Henny Admoni, Pete Trautman, Anca Dragan and Sidd Srinivasa for insightful discussions and help with the MICO. I would not be where I am today if it were not for the amazing teachers in my life, and I thank each one of my teachers for many various parts of my education.

This dissertation is a direct result of the unconditional love and encouragement of my parents, Mahavir and Sunita. I am grateful to my parents for their patience and unwavering support. Also, special thanks to Harsh and Prachi for their support over the years. I also thank my cousins, aunts, and uncles for bearing with me even when I was bad at keeping in touch. I thank Manish Joshi for being my ardent supporter ever since I started graduate school. I appreciate the support of my friends and especially, I would like to thank Mayank, Kat, and Karthik for being wonderful friends and providing me with much encouragement during my years of the graduate school.

I gratefully acknowledge the funding sources and sponsors of my research. I have been supported by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) and National Institute of Child and Human Development (NICHD) under award numbers R01EB019335 and R01EB024058. I am also grateful for the support provided by a grant from U.S. Office of Naval Research under the award number N00014-16-1-2247. The content presented in this dissertation is solely the responsibility of the authors and does not necessarily represent the official views of any sponsoring institution.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Recent technological advancements in robotics and automation have made it possible for robots to be utilized in a wide range of applications which can revolutionize many domains, spanning all manner of industry from manufacturing to healthcare. Although today robots can be helpful in a diverse range of areas, state-of-the-art technology is mostly suitable for robots to operate autonomously within controlled and isolated environments, for performing small and specific tasks.

An enduring goal is to move robots out of isolation and into human environments, forming personal and collaborative relationships, and operating side-by-side with human personnel. The field of Human-Robot Interaction (HRI) focuses on understanding, modeling, and assessing the collaboration between robotic systems and human partners. HRI has recently received considerable attention from the research community [**2**], pertaining to the increasing number of domains in which robots can be deployed, and the inevitable need to interact with humans in many of these domains. Examples of such influential areas include assistive robotics, robot-assisted search and rescue, and space exploration.

Assistive Robotics, a subfield of HRI, focuses on human-robot systems that seek to provide physical support to persons who could benefit from it, such as the elderly or people with motor-impairments. Assistive robotics emphasizes proximate interaction in HRI, where the humans and the robots are co-located. Although sometimes referred to by names other than robots, the types of robots/machines used in the assistive robotics domain vary widely in their physical appearance and applications. Some examples of such assistive machines include powered wheelchairs, assistive robotic arms, upper or lower limb prostheses, and exoskeletons (Figure 1.1).

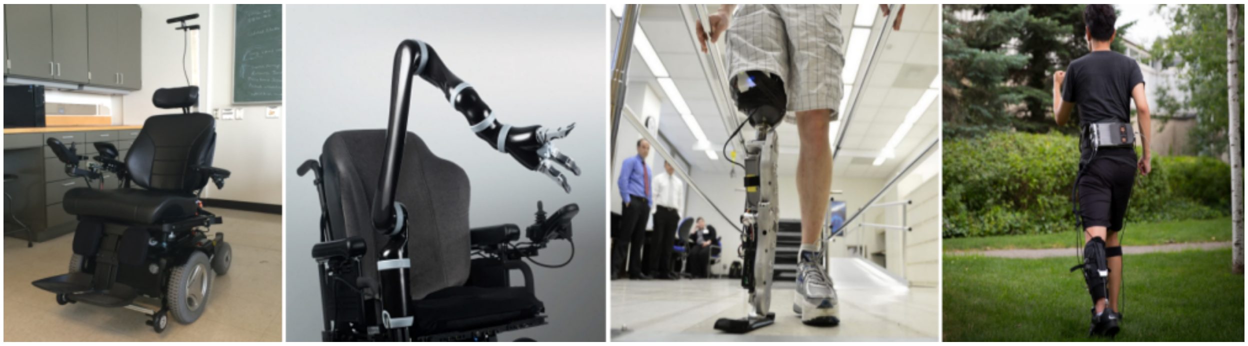


Figure 1.1. Examples of assistive human-robot systems. Powered wheelchair (Northwestern University), JACO robotic arm (Kinova, Canada), Prosthetic leg (Shirley Ryan AbilityLab, USA), Exosuit (Harvard University).

1.1. Motivation

Assistive robots can be used to perform physical tasks for people with motor-impairments, thereby improving the quality of life and promoting the functional independence of severely disabled individuals. The tasks usually are embedded within the context of normal human activities of daily living (ADLs) and would otherwise have to be performed by an attendant, for example a caregiver. Typically, the person with the disability controls the operation of the assistive machine, and thereby the design, control, sensing, and evaluation of such systems become more sophisticated due to a *human-in-the-loop*.

The most typical and ubiquitous example of an assistive machine is the *powered wheelchair*. There are more than 5.5 million people in the United States who benefit from the use of a wheelchair for their mobility needs [3]. For those with upper extremity disabilities and additionally for individuals who rely on a wheelchair to compensate for lower body impairment, *assistive robotic arms* can serve as helpers by aiding in activities like pick-and-place tasks, object retrieval, opening doors, pushing buttons, and even in personal hygiene and feeding. This population includes people with muscular dystrophy (MD), spinal cord injury (SCI), spinal muscular atrophy (SMA), multiple sclerosis (MS), amyotrophic lateral sclerosis (ALS), cerebral palsy (CP), rheumatoid arthritis, post-polio syndrome, and other severe motor paralysis [**4**].

While assistive robots can significantly enhance the functional independence of individuals, many users are challenged by their *direct teleoperation*, also referred as *direct control*; the manner in which powered wheelchairs and assistive robot manipulators like the JACO (Kinova Robotics, Canada) are currently operated by the users.

Definition 1.1.1. ***Direct Control*** *is characterized by fully manual control of robotic platform by the operator using an interface to activate and control different functionalities (e.g., the actuator velocities) of the robot.*

In direct control of assistive robots, the user inputs from a control interface are mapped directly to robot actions, putting the control burden entirely on the user. All the responsibility of performing tasks, platform safety, obstacle avoidance, or path planning to a goal is left to the human user. The user commands typically address the robot's translational and rotational velocities. For example, the typical operation of a powered wheelchair involves 2-D control (heading + speed) with the standard wheelchair joystick interface comprised of two axes that can be controlled simultaneously, one for forward/backward speed, and the other for heading direction.

Operation of assistive machines can be difficult, even when using traditional control interfaces such as a joystick. The operator must be able to accurately sense their environment and translate their desires into robotic system control commands. Moreover, as assistive machines become more capable, they often become more complex and exhibit higher degrees

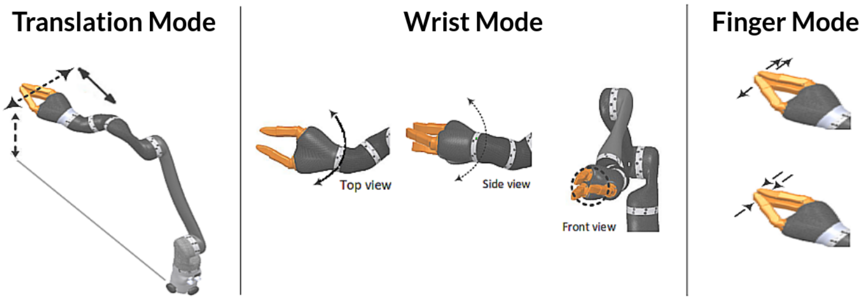


Figure 1.2. JACO arm teleoperation requires the user to switch between 3 control modes when using 3-axis joystick (Kinova JACO arm user guide).

of freedom (DoF). The requirement to control the many degrees of freedom makes their teleoperation even more challenging. For example, assistive robotic arms are controlled within 6-D (position + orientation of the end-effector) and thus the control requirements to operate them become much more complex. Traditional interfaces that seem intuitive to teleoperation of powered wheelchairs could easily become unintuitive and tedious for robotic arm teleoperation since they cover only a portion of the control space (6-D control problem).

Another significant factor that makes the teleoperation of assistive robotic systems challenging (both physically and cognitively) is *mode switching.* Typical teleoperation interfaces can cover only a portion of the control space and thus high-DoF systems such as robotic manipulators are customarily controlled via several movement modes, where each mode controls motion in some DoFs of the robotic system (Figure 1.2).

Definition 1.1.2. ***Mode Switching*** *is performed by the operator to manually select and cycle between appropriate control modes using the teleoperation interface—making direct teleoperation a multi-stepped process.*

Mode switching operations are slow, non-intuitive and require the user to divert their attention away from accomplishing the task, which results in sustained physical and even cognitive effort from the user [**5**]. For example, in the case of multi-axis (2-3 axis) joysticks,

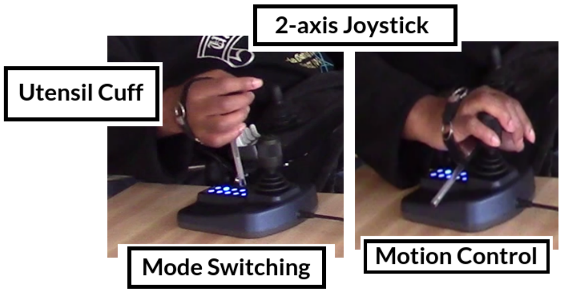


Figure 1.3. A C7 tetraplegic subject using a utensil cuff to press pushbuttons on a joystick in order to perform mode switching for robotic arm teleoperation.

a set of pushbuttons are used to switch between control modes and the directional nature of the joystick shaft allows for the arm motion in the selected mode. Operating such joysticks requires finer motor control in the user's fingers/thumb and thus can become limiting to the targeted population with severe motor-impairments (e.g., C7-C8 tetraplegic) (Figure 1.3).

The control problem is exacerbated by constraints arising from the user's physical impairment on the interface used to provide teleoperation commands. As the users age, or as their motor functions decline, their ability to use conventional joysticks can also be compromised. For example, users with quadriplegia experience challenges using conventional interfaces, and are required to operate more limited control interfaces such as the respiration-based Sip-N-Puff interface or the switch-based Head array. Paradoxically, the more severe a person's motor-impairment, the more limited the control interfaces available to them; limitations which can even impede their ability to operate the very assistive machine meant to aid them. Furthermore, these limited control interfaces can only issue control signals which are of low dimensionality and bandwidth—making the direct teleoperation and the requirements of mode switching even more exacting. For example, in the case of the Sip-N-Puff interface, the number of control modes increases in proportion to the robotic arm's controllable degrees of freedom—making the direct teleoperation intractable. Performing even simple manipulation tasks can involve several mode switches and the process can become very tedious and challenging.

1.2. Robotics Autonomy & Human-Robot Systems

Pertaining to the challenges in direct control of assistive robots, one research direction investigates and designs novel *human-machine interfaces*, including non-conventional control interfaces, that could enable the user to interact with assistive devices in a more efficient and satisfactory manner. We present a detailed review of such approaches in Chapter 2.3. Another significant research direction investigates the introduction of robotics autonomy to reduce the user's control burden for the operation of assistive machines.

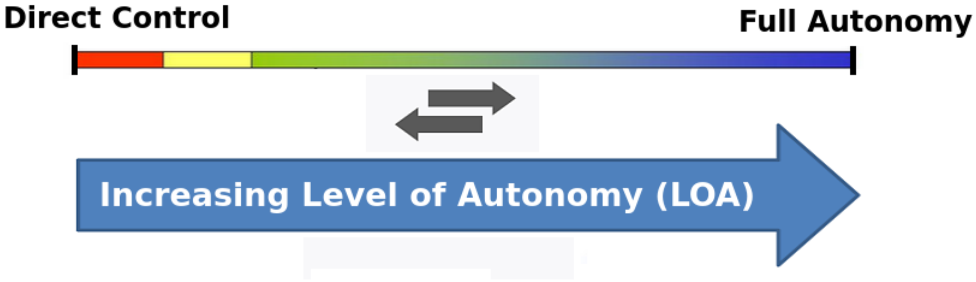


Figure 1.4. Continuum of levels of robotics autonomy with an emphasis on human-robot interaction in assistive robotics, wherein the level of autonomy can increase or decrease based on the user's ability and preferences.

Robotics autonomy has been discussed in the literature to describe many different aspects of robotics, from the robot's ability to self-govern much like a human, to the level of human intervention [**6**]. In this dissertation, we consider autonomy, as associated with robotic agents, to be a robotic system that can perform its own processes and operations. The robotics autonomy exists on a continuum from *no autonomy* to *full autonomy*, and the degree, scale or the *level of the autonomy* (LOA), generally refers to the amount of human intervention, as well as the function or control allocation between the human and the robot (Figure 1.4).

Definition 1.2.1. ***Level of Autonomy*** *is the extent to which a robot can sense the environment, plan based on that environment, and act upon that environment; with the objective of accomplishing or reaching some goal, and with little or no external control.*

Direct control, as discussed in assistive robotics, refers to the manual control of the assistive robot with no autonomy; wherein the robot offers no assistance, putting the control burden entirely on the user. On the other hand, in the case of full autonomy, the robotics autonomy provides maximum assistance; wherein the robotic system carries out all actions autonomously. While completely autonomous behaviors can be efficient, they take away all control from the user, and thereby can lead to a diminishing the sense of task accomplishment for the users. Studies report less user acceptance if the assistive robot is entirely autonomous—as users prefer to remain in control of the assistive device [**7**]; making full autonomy undesired in the assistive domain.

An attractive solution is *shared autonomy*, in which semi-autonomous behaviors can be achieved via control sharing; such that the robotics autonomy offloads some control burden of the user while both the robot and the user take the control decisions to accomplish tasks. Under shared autonomy, dynamic allocation of the level of autonomy could be achieved via fusion or arbitration of the inputs from manual control and automated controllers.

Such control sharing under shared autonomy has drawn considerable attention in recent research. Important considerations include how to modulate the balance between human control and autonomy control and what should be autonomy behavior, such that it is acceptable to humans while efficient for task executions. We present a detailed review of such approaches in Chapter 3.1.

1.3. Challenges & Contributions

Assistive human-robot systems emphasize proximate interaction, forming personal and collaborative relationships, and sharing control with human personnel. Thus, the design, sensing, control and assessment of such systems becomes more sophisticated due to a human-in-the-loop. Furthermore, unstructured human environments pose incredible challenges, and effective operation of human-robot systems in real-world environments requires a tightly integrated effort, combining perception, inference, planning and control.

While each aspect of human-robot systems has many unique requirements, most of the research and experimental work in the literature on shared autonomy focuses on control sharing with the human partner for providing robotic assistance. Other aspects are equally important for human-robot systems to be able to provide effective collaboration and assistance to the user. For instance, a feeding task under shared autonomy with a robotic arm, an important ADL task for assistive robotics [**8**], requires food perception, knowing what the user wants to eat and how to assist the user in performing food manipulation. The successful operation of such systems in real-world environments requires a substantial engineering effort.

In this dissertation, we present an exposition of what we believe to be some of the most important research questions and challenges facing those who seek to build complete systems for human-robot collaboration in the assistive domain:

- *Challenge 1: Robotic Perception for Human-Robot Systems*

  Perception is one of the most important challenges facing the field of robotics. Unstructured human environments present incredible challenges that require dealing with novel unknown objects, diverse variations, cluttered workspaces, and noisy sensor data. Although robots in human environments will almost always face uncertainty due to their limited view of a changing world, automated perception can reduce this uncertainty and enable more robust operation.

In order to assist in collaborative tasks, assistive robotic systems must be able to sense and perceive their working environment. For widespread adoption and use of assistive technology, it is also important for the perception to generalize and work without the use of landmarks, fiducial markers, or predefined object models. The key for human-robot systems is to develop perception models that are centered on the human user's intentions and preferences, that is to identify and detect the high-level potential targets or goals in the environment that might be of interest to the human collaborator. Perception can thus simplify human-robot interaction by potentially reducing the burden on robotics autonomy for inferring user intentions with the detection of a finite set of candidate goals. For robotic perception in human-robot systems, it is important to consider the following research questions:
How can the robot detect and model the environment of the human user?
How can we develop perception centered on user intentions and preferences?
How can the robot detect and identify high-level potential targets or user objectives?
How can we leverage perception to ease the burden of inferring user intentions?

- *Challenge 2: Implicit Human Intent Recognition in Human-Robot Systems*

  Effective human-robot collaboration in shared autonomy requires reasoning about the intentions of the human partner. To provide meaningful assistance and appropriately share control with a human, the autonomy must correctly predict, or infer, the intentions or goals of the human collaborator. Requiring explicit communication from the user could lead to ineffective collaboration and increased

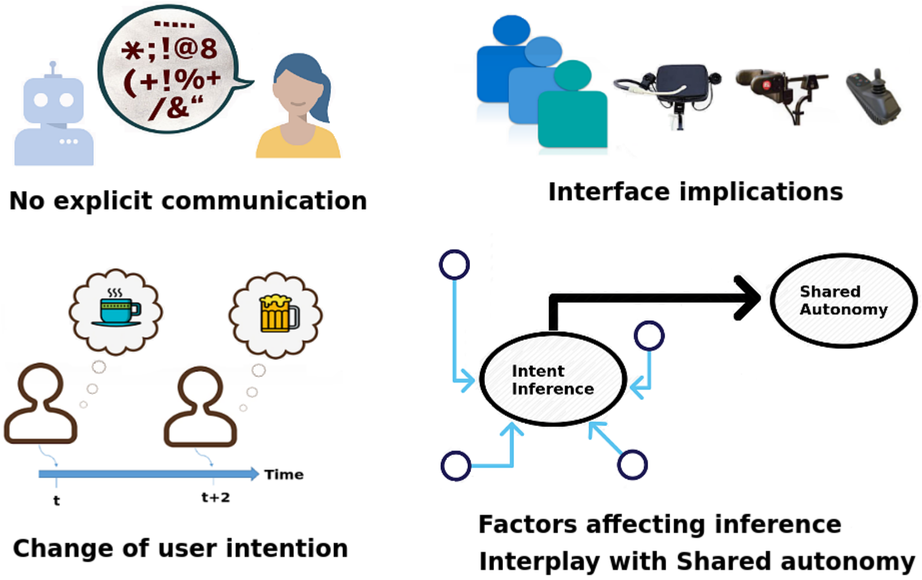


Figure 1.5. Illustration of important challenges facing human intent recognition.

cognitive load [**9**]. Thus, for seamless interaction in shared autonomy, it is vital for the robotics autonomy to perform implicit inference of user intentions. Additionally, inferring the intentions of the user presents challenges in the assistive domain as a consequence of the limitations of the interface and/or the human, due to the underlying motor-impairments. Furthermore, it is important to express uncertainty in the robot's prediction of the intended goal, because assisting towards the wrong goal could be worse than providing no assistance. For intent inference in human-robot systems, it is important to consider the following research questions:

How can the robot infer the objectives or goals of the human user?

How can the robot perform intent inference without explicit communication?

How can the robot express uncertainty in its predictions of user intentions?

What are the implications of limited control interfaces on intent inference?

How does change in user intentions affect intent inference?

What are the implications of intent inference on shared autonomy performance?

- *Challenge 3: Shared Autonomy & User Personalization in Human-Robot Systems*

  In the assistive domain, abandonment of assistive machines is a big concern [**10**]. Consideration of the social and physical ramifications of assistive technology is critical in predicting its adoption and feasibility of long-term integration with end-users. For successful control sharing under shared autonomy in human-robot systems, the following research questions are important:

  What should the behavior of autonomy be for providing assistance?

  How to perform control sharing with the user?

  What should the level of autonomy be?

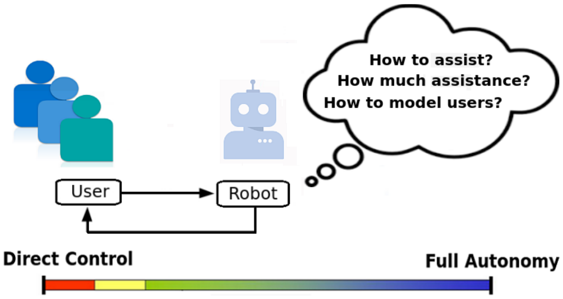


Figure 1.6. Illustration of important challenges facing shared autonomy and user personalization.

Furthermore, different people have different abilities to control assistive machines, depending on their physical capabilities and the type of interface they could use to operate the assistive machine. Moreover, users also differ in their preferences regarding the desired amount of assistance. Thus, user personalization is critical in human-robot systems for successful user adoption. Such requirements present the following additional challenges:

How can we model users or their actions under shared autonomy systems?

How can we personalize robotic assistance to individual users?

To address the aforementioned challenges, this dissertation contributes to the understanding and development of mathematical models and algorithms for perception, inference, and assistance in human-robot systems, and presents results of human-robot user studies with insights into our formulations.

In this dissertation, we:

- Develop and evaluate novel geometric and machine learning models for robotic perception of high-level user goals, with applications for assistive wheelchair navigation and shared-control manipulation of assistive robotic arms. For wheelchair docking behavior, our algorithm is first in the domain to address fully automated perception of docking sites. Our grasp detection framework generates multiple semantic grasps on objects to handle multiple ways in which a single object can be grasped by human users. Moreover, grasps are similar to what a human would generate when teleoperating the robot, which could improve human-robot cooperation.
- Develop a mathematical framework for implicit human intent recognition in shared autonomy, enabling seamless fusion of observations to probabilistically reason about user goals. Our approach in many scenarios outperforms existing solutions for intent inference and otherwise performs comparably. More importantly, we examine the intent inference implications under multiple control interfaces that typically are utilized within the assistive domain, a first within the field.
- Develop mathematical models and introduce methods for user personalization in human-robot systems, including probabilistic modeling of human actions for intent inference and intent-driven optimization for user personalization in shared autonomy. Our models account for the behavior of individual users, and the motivation for our approaches are prior studies in the field which show that users vary greatly

in their performance and preferences, thus implying a need for assistive systems to adapt to each individual user.

- Conduct and implement a variety of user studies involving subjects without motor-impairments (uninjured) and end users to evaluate: human inference, autonomy inference, intent inference implications under limited control interfaces, effect of inference on shared autonomy, implications of change of intent during task execution, and effect of user personalization on inference and assistance in shared autonomy. Our findings provide a greater understanding of the computational modeling and human requirements for successful human-robot systems in the assistive domain.
- Develop a novel application of body-machine interface (BMI) and shared autonomy for assistive teleoperation of robotic arms, which engage users in sustained physical activities for potential partial recovery of movement skills.

1.4. Dissertation Organization

This dissertation is organized into four parts:

Part I, Preliminaries: Overview, Motivations, Related work for the background material of techniques that are employed in this dissertation, and Research platforms

Chapter 1	Overview of assistive robotics; Motivations for the research and discussion of its purposeful characteristics; Research directions, challenges and dissertation contributions.
Chapter 2	Background on assistive robotics, human-machine interfaces, intent inference in human-robot systems, and robot perception for the assistive domain.
Chapter 3	Shared autonomy theory, background and categorization based on control sharing with the human; User personalization and adaptation in human-robot systems.
Chapter 4	Research platforms and codebase.

Part II, Mathematical Models, Algorithms & User Studies: High-level Goal Perception, Probabilistic Intent Inference and Assistance in Shared Autonomy

Chapter 5 Models and algorithms for high-level perception of human goals; Geometric detection of wheelchair docking locations; Estimation of surface geometry in point clouds; Machine learning and model fitting for grasp detection and assistive manipulation in shared autonomy.

Chapter 6 Probabilistic human intent recognition for shared autonomy; Goal-directed probabilistic modeling of user actions with adjustable rationality; Intent-driven optimization for user personalization; User studies on intent inference and assistance in shared autonomy.

Part III, Human-Machine Interfaces: Body-Machine Interface (BMI) and Shared Autonomy for Assistive Manipulation

Chapter 7 Novel application of body-machine interface; Assistive manipulation with shared autonomy; Pilot study and experiments.

Part IV, Conclusions: Concluding thoughts and open questions

Chapter 8 Final thoughts and a concluding summary of the dissertation.

CHAPTER 2

Background

2.1. Overview

In this chapter, we first cover relevant background information to familiarize the reader with the domain of assistive robotics (Section 2.2). We then provide an overview of common control interfaces that are utilized for robotic teleoperation in different application areas, and highlight their limitations and challenges relevant to the assistive domain (Section 2.3). In addition, we also present the prior work focusing on the development of novel human-machine interfaces. In the next section, we review research in the field of human-robot interaction focusing on intent inference and prediction of user goals (Section 2.4). Finally, we present related work in the area of robot perception and computer vision, targeting human-robot interaction and assistive systems (Section 2.5). For incorporating of robotics autonomy into human-robot systems, intent inference and perception are two key components of autonomy. Equally important is the control sharing with the human and personalization and adaptation to individual users, which we discuss in Chapter 3.

2.2. Assistive Robotics

Many people face difficulties with a multitude of tasks associated with upper or lower body functions—such as walking, climbing stairs, lifting and grasping objects. In 2014, a survey by the U.S. Census Bureau [**3**] reported that more than 24 million individuals need assistance performing tasks of daily living (ADL), such as getting in or out of bed or picking up and holding an object.

In addition to the effects of aging, the other specific cause of functional limitations are motor-impairments pertaining to a physical injury, neurological diseases, or congenital disorders. Paraplegia is a severe or complete loss of motor function in the lower extremities and lower portions of the trunk. When four limbs are affected, the condition is known as quadriplegia or tertetraplegia. The major cause of paraplegia and quadriplegia is spinal cord injury (SCI) [**11**], which can have wider-reaching effects on mobility. When individuals with quadriplegia have no ability to produce movements (outside of eye movement) or to speak (aphonia, loss of speech), the condition is known as locked-in syndrome (LIS) [**12**]. In individuals with LIS, consciousness is preserved and the cognitive function is usually unaffected but communication is only possible through eye movements or blinking. The etiology ranges from acute brainstem stroke (most frequent cause) to chronic causes including Amyotrophic lateral sclerosis (ALS) [**13**]. ALS is a progressive disease that causes the degeneration of motor nerve cells in both the upper and lower motor neurons, and eventually affects the ability to move, speak, eat, and even breathe. Another neuromuscular disorder that affects mobility is muscular dystrophy (MD) [**14**], which results in increasing weakening and breakdown of skeletal muscles over time. A congenital disorder that affect the ability to move and maintain balance is cerebral palsy (CP) [**15**], which is caused by abnormal development of the brain or damage to the developing brain.

The daily lives of people with motor impairments can be greatly enhanced with the application of robotic technology assisting physical tasks. Mobility difficulties associated with the lower body functional limitations are more common and there are more than 5.5 million people who use a wheelchair for mobility needs. Unsurprisingly, by far the most ubiquitous powered assistive machine is the powered wheelchair [**16**]. Accordingly, the development of smart wheelchairs—which incorporate robotics autonomy into powered wheelchairs for providing assistance with collision avoidance and safe autonomous navigation behaviors has received the most attention within the academic research community [**17**–**20**]

Impairments in upper extremities often have a detrimental effect on the ability to perform ADL [**21**]. In addition, wheelchair bound individuals often also face difficulties in terms of upper extremity activities. In a survey, the ability to reach adequately for objects was rated as the most important concern related to wheelchair use [**22**]. For physical task support and providing manipulation capabilities, assistive robotic arms can compensate for the loss of mobility in the upper extremities. These robotic arms can be controlled through a joystick (via direct control) and can be fixed either on a table as stationary workstation or can be mounted to the user's wheelchair (WMRA). Only few commercialized assistive robotic manipulators are currently available on the market [**8**]. The two major commercialized assistive robotic manipulators are the JACO arm (Kinova Robotics, Canada) and the MANUS ARM (Exact Dynamics, The Netherlands). These assistive robotic manipulators have seven degrees of freedom (7-DoF) including the gripper, and they are both WMRAs. The JACO arm has a three-fingered hand and it can be controlled by a 2- or 3-DoF joystick. The MANUS ARM has an updated version called iARM that has a two-finger gripper and it can be controlled by a keypad or a joystick. Direct control of these arms involve toggling modes (mode switching) using the control interface in order to operate a subset of robot's DoF at a time [**23**], which makes their teleoperation challenging—specially for those with impaired motor abilities.

Some assistive robots involve a combination of a fixed workstation manipulation robot and a mobile robotic platform. Such service robots are capable of fetch and carry task, which can serve as the basis of a wider range of domestic tasks [**24**], and are targeted for people with severe motor impairments. Earlier manipulation assistance relied on remote operation by a human user with either a joystick, keypad input or speech input. However, the limited capabilities of the target user populations has driven the research into increasing autonomous control for mobile manipulation tasks— for example, vision based semi-autonomous object grasping [**25**, **26**].

Another category of assistive robots involve systems that can be worn on the body as an implant or an accessory. There have been recent advances in the areas of wearable assistive robotics [**27**], including actuated robot prostheses [**28**], orthoses, and exoskeletons [**29**]. A prosthesis is a device that supplants a person's missing limb, and acts in series with a residual limb. Unlike prosthesis, an orthosis is an externally applied device for an intact limb, to provide support or improve function for a section of the body. An exoskeleton provides assistance or enhancement of existing physical capability to a person with intact limbs. Exoskeletons can be utilized to partly augment the muscular strength or providing additional support in activities of daily living such as walking [**30**] and hand grasping [**31**]. For scenarios in which an assistive device would be worn for extended periods of time, some long-term goals are to create soft wearable robots that could use compliant actuators and sensors, does not restrict movement, are light-weight and to reduce the metabolic cost of wearing them.

2.3. Human-Machine Interfaces

In this section, we will present a summary of human-machine interfaces that have been used to control assistive robot devices. Many different kinds of interfaces are used for robotic teleoperation and human-robot interaction. Some typical examples include hand-controllers such as multi-axis joysticks, touchpads, and haptic devices; multimodal/multisensor interfaces that provide multiple control modes or use fused data displays, such as virtual reality headsets (VR); and supervisory control interfaces which are designed for high-level command generation and monitoring.

In assistive robotics, dedicated user interfaces are developed for providing inputs depending on the impairments of the user and the control channels available to them. The vast majority of powered wheelchair users operate a hand-controlled joystick [**32**]. A standard wheelchair joystick provides proportional control with two axes (2-D control) that can be controlled simultaneously, one for forward/backward speed, and the other for heading direction of the wheelchair. An alternative is a touch pad interface that uses smart phone touchscreen technology for ultra sensitive and precise proportional driving capabilities. It involves making contact with a glass surface of the device for full driving capabilities including speeding, turning and veering; no force is required.

For people with limited hand functions and those who are unable to operate a hand-controlled joystick, some other interface options are head-array, chin interface, and sip-n-puff (SNP). These interfaces usually offer a very limited bandwidth and a few distinct commands, as compared to joysticks.

Head-array or chin control interface is an option for the individuals with good head movement ability. A head-array comprises a series of switches mounted in a headrest that are activated by the head movement. The switches are programmed to provide a full range of acceleration and deceleration, with turning and veering control. Chin control is usually considered as different to head control. The chin sits in a cup shaped joystick handle and is usually controlled by neck flexion, extension and rotation. Operation of a head-array or chin control interface requires fine-tuned adjustments that are adjusted to each user's strengths, capabilities and preferences.

Sip-N-puff (SNP) interface is primarily used by people who do not have the use of their hands. The interface involves breathing in (inhaling or sipping) and breathing out (exhaling or puffing) of a mouthpiece, a straw-like device, that is activated by changing the air pressure. SNP require specific amounts of air pressure to be "sipped" or "puffed" by the user and these amounts are typically denoted as a hard sip/hard puff or soft sip/soft puff. Typically, these air pressure levels are set through an initial calibration relative to the user and depend on their breathing abilities. A sequence of sips and puffs generates a digital signals that is in turn interpreted by the wheelchair controller for forward/backward speed, and the heading direction of the wheelchair. SNP interfaces are commonly used by quadriplegics with high-level spinal cord injury or people with ALS.

While aforementioned interfaces are sufficient for controlling a 2-DoF system such as a powered wheelchair, these interfaces do not all directly scale to controlling high DoF systems, such as robot arms which typically requires 6 or more DoF control. The commercially available assistive robot arms (MANUS ARM and JACO) are controlled via a 2- or 3-axis joystick, by cycling through which degree of freedoms are being controlled at a time. Such mode switching partitions the controllable degrees of freedom of the robot such that each control mode maps the input dimensions to a subset of the arm's controllable degrees of freedom and thus performing tasks involves control of the shaft (for the robot motion control) and the pushbuttons (to switch between control modes). Performing even simple manipulation tasks can involve multiple mode switches and the process can become extremely challenging with the more limited interfaces like a Sip-N-Puff or a head array that are available to individuals with severe motor impairments.

For users facing challenges with existing interfaces (people suffering from quadriplegia and locked-in syndrome) and to improve the control of high-DoF assistive robotic systems, novel interfaces are researched [**33**] for particular applications.

An Xbox Adaptive Controller (Microsoft, USA) is designed for people with disabilities to make it easier to play games on Xbox One. The interface design focus on connectivity and customization, with players able to build a setup that works for their capabilities and needs. Mouse-and-keyboard based input interfaces are developed for remote teleoperation of high-DoF robotic systems. For robotic manipulation, a 6-DoF ring-and-arrow marker based approach is generally used for independently controlling each axis of rotation and translation [**34**, **35**]. Such markers can be used to directly move an end-effector or set pose goals to which the robot can autonomously plan and move. Multiple control methods based on the ring-and-arrow marker design have been evaluated [**36**]. An alternative point-and-click interface approach enables more efficient grasp poses based on a single mouse click [**37**].

Novel interfaces that use non-conventional input methods are developed for the assistive domain. A Tongue Drive System (TDS) uses two magnetic sensors placed on the side of the operator's head and a magnetic tongue barbell, which allows individuals with severe motor impairments such as tetraplegia to control powered wheelchair using voluntary tongue motion [**38**]. An eye-control based model on electrooculographic (EOG) signal is used for wheelchair control [**39**]. Eye and gaze tracking have also been utilized for applications in human-robot interaction [**40**–**42**]. A different promising alternative are Brain Computer Interfaces (BCIs), which are based on the decoding of the electrical brain activity. BCIs could enable the use of the brain as a new communication channel for assistive robotics, given that they are well adapted for tetraplegic patients [**43**]. Invasive BCI methods provide the highest spatial and temporal resolution, since they can be placed closer to the area of interest, and have shown successful use for continuous high degree of freedom control of upper limb prosthetics [**44**]. However, these devices suffer from the electrode deterioration and carry the risks associated with the surgical intervention [**45**]. Alternative noninvasive methods that do not require surgery include Electroencephalography (EEG). EEG is the safest way of recording brain activity and uses electrodes placed directly on the scalp to measure weak electrical potentials. Non-invasive EEG-based BCI have been used for continuous mental control of a wheelchair [**46**] and a prosthetic arm [**47**].

Another challenge for people with motor impairments is the rehabilitation process, which aims to allow patients to keep their remaining motor function and possibly even recover some lost function. To facilitate the rehabilitation process, body-machine interface (BMI) targets to extract signals from body motions for operating external devices [**48**, **49**]. Thus, unlike BCIs, the body-machine interface engage users in sustained physical activities to support partial recovery of movement skills. However, the inclusion of sensory interfaces in BMI and the decoding of body signals into machine commands presents a greater challenge. In this dissertation, we present a novel application of body-machine interface and shared autonomy for assistive teleoperation of robotic arms (Chapter 7).

2.4. Intent Inference

Intent inference—also referred to as inference of the desired goal, target, action, or behavior—has been investigated under various settings [**50**]. For instance, intent inference include human activity recognition in the area of computer vision [**51**] using spatio-temporal representations, and task executions in surgical telemanipulation systems using Hidden Markov Models [**52**].

Probabilistic methods have been developed to infer unknown intentions from human movements. A latent variable model is proposed to infer the human intentions from an on-going movement, which is verified for target inference in robot table tennis and movement recognition for interactive humanoid robots [**53**]. A data-driven approach synthesizes anticipatory knowledge of human motions and subsequent action steps for the prediction of human reaching motion towards a target [**54**]. Predictive inverse optimal control methods also are utilized to estimate human motion trajectories and shown to work for anticipating the target locations and activity intentions of human hand motions [**55**]. Another important line of research explores inference of motion trajectories and intent expressive motions for improved communication of intent via legible motions [**56**] and intention-driven behaviors [**57**].

In shared autonomy, many systems simplify the intent inference problem by focusing on predefined tasks and behaviors or by assuming that the robot has access to the user's intent *a priori* [**58**, **59**]. There exist works that rely on explicit commands from the user to communicate intent via an interface, such as GUI [**60**]. There are however studies that suggest that effective human-robot collaboration should not rely on explicit communications [**9**]. A number of human-robot interaction approaches investigate the use of non-verbal communication including gestures, expressions and gaze. For example, user's gaze patterns are studied to understand human control behavior for shared manipulation [**61**] and to perform anticipatory actions [**42**]. The eye gaze data quality can depend on the calibration accuracy and projection method errors for eye-trackers [**42**]. Gestures, along with language referential expressions, are utilized for object inference in human-robot interactions [**62**].

Intent inference in shared autonomy is often represented as a confidence in the prediction of the intended goal, based on instantaneous observations [**63**–**65**]. Some shared-autonomy systems compute a belief over the space of possible goals. A memory-based inference approach utilizes the history of trajectory inputs and applies Laplace's method about the optimal trajectory between any two points to approximate the distribution over goals [**63**]. Memory-based inference is utilized in previous works involving shared autonomy and human-robot systems [**66**] [**63**] [**56**]. Another approach uses the Laplace's approximation [**63**] and formulates the problem as one of optimizing a Partially Observable Markov Decision Process (POMDP) over the user's goal to arbitrate control over a distribution of possible outcomes [**67**]. The approach considers user inputs for the prediction model and uses a hand-specified distance-based user cost function in order to achieve a closed-form value function computation.

For intent inference in the assistive domain, it is important we investigate how shared autonomy systems can take advantage of the indirect signals people implicitly provide when operating a robot through a direct control interface (e.g., a joystick), and what are the implications of the limited control interface limitations on intent inference. Note that these interfaces differ in the continuity and dimensionality of the issued control signals and, for all, the dimensionality of the control signal is considerably lower than the control space of the robot—making the intent inference problem more challenging.

In this dissertation, we present a framework for intent inference under shared autonomy (Chapter 6) that models the uncertainty over the user's goal within a Bayesian filtering framework which enables seamless fusion of multiple observations. We model the user's actions within a probabilistic behavior model that incorporates an adjustable rationality and, importantly, we introduce an intent-driven optimization that adapts the rationality index value to each individual user, and thus can account for their particular behavior including the implications of the limited control interface limitations. The motivation for our approach are prior studies which show that users vary greatly in their performance, preferences and desires [**63**, **67**, **68**], suggesting a need for assistive systems to adapt to each individual user.

2.5. Robot Perception for HRI & Assistive Robotics

Robot perception using computer vision for robotic systems in HRI and the assistive domain encompass multitude of tasks [**69**]. The aim of perception systems are to provide environment abstraction and modeling for executing autonomous behaviors, simplify the interaction between the user and the machine, as well as autonomous perception of candidate goals within the environment, which might be candidate locations of interest to visit, or objects of interest to manipulate.

In Table 2.1 a scheme of the relationships between assistive systems and the involved computer vision tasks is shown.

Table 2.1. Computer vision tasks involved in assistive robotic systems.

Assistive systems → CV tasks ↓	Smart Wheelchairs	Robotic Arms	Mobile Manipulators	Exoskeletons
Localization	x		x	
SLAM	x		x	
Object detection		x	x	
Object tracking		x	x	
Human activity recognition	x		x	x
Doorway detection	x			
Docking detection	x			
Incline detection	x			x
Stairs detection				x
Eye Gaze estimation	x	x	x	x
Head pose estimation	x	x	x	x
Gesture recognition	x	x	x	x
Grasp detection		x	x	x
OCR	x		x	

2.5.1. Perception for Mobility and Navigation

Computer vision techniques have been exploited in the development of intelligent wheelchairs. For development of command interfaces that does not require the function of hands, the wheelchair autonomy addresses problems like face detection and face movements [**70**], head pose estimation [**71**, **72**] and/or eye gaze recognition [**73**] for generating control signals for the wheelchair. Vision based autonomous localization frameworks have been also used to make efficient navigation for wheelchairs. Visual self-localization and mapping (SLAM) techniques have been used to calculate the wheelchair motion [**74**, **75**]. Vision techniques are also utilized to provide assistance for desired behaviors, involving spatially-constrained maneuvers [**76**], person following [**77**], sidewalk following and terrain classification [**78**], and assisted doorway traversal [**79**, **80**].

2.5.1.1. Perception for Robot Docking. Many smart wheelchair works recognize docking as a desired behavior or mention it as a goal for future work [**81**–**89**]. In this dissertation, we present a novel approach for the automated detection of wheelchair docking locations at table and desk structures (Section 5.2). In this section, we present an overview of existing computer vision approaches for robot docking.

Approaches that address robot docking often make use of fiducial markers [**90**, **91**]. Other approaches simplify the problem by focusing on customized docking structures. For example, a laser scanner and geometric constraints are used to find the corners of a rectangular container of known dimensions [**92**] and a T-shaped fiducial is used for docking at a wheelchair at a U-shaped bed [**83**]. Approaches also consider wheelchair docking onto a vehicle lift platform, using visual feedback control and fiducial for image processing [**84**]. A handful of works does not rely on environment modifications like fiducial or customization. One approach identify table structures using vision with an unclear docking location, and the algorithm selects a location somewhere in between the edges as a compromise [**85**]. There also are works that place the identification of docking locations entirely with the human [**86**–**88**].

There are approaches that use 3D data to find supporting planes in the scene; an approach similar to our work. The application of a probabilistic model built using Expectation-Maximization (EM) is suggested to find possible docking objects by segmenting LIDAR data for modeling plane-like clusters—which however requires knowing *a priori* the number and types of clusters and is subject to poor local minimum solutions [**93**]. A 3D semantic map based approach extract planes in the scene using RANSAC, which are matched to entries in a model library to identify objects [**89**]. Our approach uses the geometric structures of the objects, similar to [**92**]—however without the requirement of knowing the structure dimensions *a priori*. Our work address docking perception as well as the safe position and alignment determination, and is also free from the usage of landmarks or fiducial markers (Section 5.2). Furthermore, our algorithm was recently adapted in a dissertation work involving powered wheelchair parking assistance system and evaluation of wheelchair docking assistance under shared autonomy with a user study [**94**].

2.5.2. Perception for Robotic Manipulation

In assistive robotics, computer vision algorithms are involved in the innovative area of the robotic arm and prosthetic limb control; where autonomy systems use vision techniques to autonomously detect and recognize target objects, object pose, robot hand orientations and the grasp pose for manipulation, based on the vision data acquired by camera images and/or depth data (RGB-D).

2.5.2.1. Environment Modeling & Surface Representations. Unstructured human environments present incredible challenges for robotic manipulation. Robotic manipulation in human environments requires dealing with novel unknown objects, cluttered workspaces, and noisy sensor data. Large variations in objects, lighting, and clutter make household manipulation a very difficult problem. Furthermore, with humans moving around in the environment, and for operating near humans safety is an important consideration for any assistive robot.

Building consistent and detailed models of the environment is required for robotic operation in unstructured environments. Models of the environment can be build using 2-D or 3-D visual sensors and surface representations [**95**]. However, surface representations are unable to distinguish between free and unknown space, and require large memory. In mobile manipulation scenarios, for example, being able to differentiate free from unknown space is essential for safe navigation and manipulation.

Thus, environment models use two representations: a probabilistic occupancy grid representation [**96**] for unmodeled or unrecognized parts of the environment and a semantic representation for known parts of the environment. An octree-based representation (Octomap) of the environment is utilized to incorporate the unmodeled parts of the environment [**97**,**98**]. Octomap is an occupancy-grid like probabilistic representation of the environment that can account for unknown space and can maintain a persistent view of the environment by automatically incorporating new sensor data. Such environment models can serves as the primary source of input for collision checking and can also be used for motion planning and grasping. For known parts (e.g., from object detection), a database of prebuilt 3-D mesh models for common objects can be used to register the objects in the scene [**99**].

2.5.2.2. Object Detection & Tracking. Object detection forms the basis for solving complex or high level vision tasks such as segmentation, scene understanding, tracking, image captioning, and event or activity recognition [**100**, **101**]. The detection deals with identifying the presence of various individual objects in a scene using image or RGB-D data. The goal is to determine whether or not there are any instances of objects from the given categories (such as cup, glass, orange, apple etc.) in the scene and, if present, to return their spatial location e.g., via a bounding box [**102**, **103**]. A lot of research is being done in the area of object recognition and detection during the last two decades. Recently, deep learning techniques [**104**,**105**] have emerged as powerful methods for learning feature representations automatically from data and such techniques have provided significant improvement for object detection problems.

Object tracking in an image sequence involves continuously identifying its location when either the object or the camera are moving. Variety of approaches can provide the object's 2D image position in terms of its centroid and scale or of an affine transformation [**106**,

107]. However, these methods does not involve recovering the actual position in space. By contrast, 3D tracking can continuously recover all six degrees of freedom that define the camera position and orientation relative to the scene, or, equivalently, the 3D displacement of an object relative to the camera [**108**].

2.5.2.3. Grasp Detection. Grasp detection is essential for robotic manipulation of objects, in which vision techniques autonomously detect and recognize robot hand positions and orientations that can be used to grasp and manipulate an object of interest. The problem of robot grasping has been widely studied in the literature and is an active research field. One camp address grasping using analytical approaches [**109**,**110**] that leverage force-closure and form-closure to assure a stable grasp configuration. Such approaches assume the availability of contact points on the object, and rely on well-defined 3D models of the object, which however generally are not available in real-world scenarios. Another line of research is database-driven (for a survey, see [**111**]), which also assumes the availability of complete 3D models of the objects and uses a pre-computed grasp dataset [**112**–**114**]. Some systems use image data for grasp detection and generally require a manually-labeled grasp dataset to learn models for detection [**115**, **116**]. Recently, approaches make use of deep learning methods [**117**], which require large training datasets and generally are not suitable for real-time applications.

Other approaches approximate an object with shape primitives. Shape primitive fitting for grasp selection is used using known models of the objects [**118**], and also performed using manual assignment by hand [**119**]. Box primitives are suggested for enveloping 3D data points constellations for object grasping by parts, however such decomposition step is time-expensive [**120**]. Another approach uses principal component analysis (PCA) with bounding box computation on segmented point clusters for reactive grasping with tactile feedback sensors [**121**]. Methods furthermore rely on simulation software for the grasp generation [**118**, **120**], which does not always correlate well with physical robot grasp performance. There exist approaches that use point cloud data for grasp detection. Mesh generation for grasp detection in simulation is used with point cloud annotation and geometric features [**122**]. An approach for grasping household objects with a mobile manipulator is proposed which does not rely on object models and detects a single overhead grasp in point clouds of object [**26**].

Researchers also study human grasp postures using demonstrations [**123**], for human-like robotic grasping. We anticipate that in collaborative human-robot scenarios, detecting a single object grasp will be insufficient—as environment change, the human user's choice of grasp location also will vary. In this dissertation, we present an approach that strives to satisfy these aims by autonomously generating candidate grasps that are similar to what users would themselves generate (Section 5.3). Our approach also is unique in that it generates multiple semantic candidate grasp poses (top, side, pinch) on objects.

CHAPTER 3

Shared Autonomy

In recent research, there has been a significant effort to leverage robotics autonomy to make the direct control of robotic systems easier by distributing the control burden between the robotics autonomy and the user control of the robot. One instance of such control sharing arises in shared autonomy, where both the user and robotics autonomy act simultaneously to control the robot, in order to achieve the user's goal.

3.1. Control Sharing with the Human

Formally, given the existence of a set $\mathcal{F}$ of automated controllers $f(\cdot)$ generated by the robotics autonomy, where each controller produces the functioning or motion of an autonomous behavior for the robot; a vector of control signals $\mathbf{u}_r$ is generated by a function $f(\cdot)$,

$$\mathbf{u}_r \leftarrow f(\cdot)$$

while control signals $\mathbf{u}_h$ are generated by the human. In shared autonomy, these two signals are reasoned to generate a shared-control signal $\mathbf{u}_t$, that gets executed by the robot,

$$\mathbf{u}_t \leftarrow \phi(\mathbf{u}_r, \mathbf{u}_h).$$

The very nature of such control sharing, $\phi(\cdot)$, can be determined by different approaches—where the important considerations are: (i) how to modulate the balance between human control and autonomy control?, and (ii) what should be the autonomy behavior such that it is acceptable to the human and at the same time is efficient for task executions?

We next categorize paradigms of control sharing under shared autonomy as *(i) Series, (ii) Interleaved* and *(iii) Parallel*, and present a background review of relevant works under such categorization.

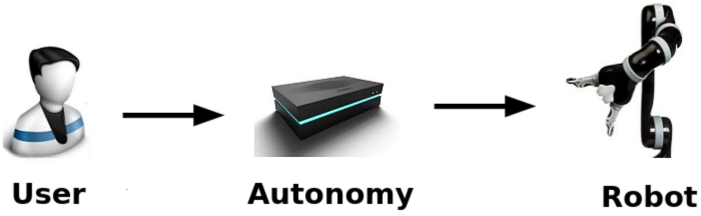


Figure 3.1. Series shared autonomy for control sharing.

Definition 3.1.1. ***Series shared autonomy*** *is a control sharing paradigm under which the autonomy inputs are generated and/or optimized based on the user inputs, which then are passed to the robot for execution (Figure 3.1).*

In its simplest and more common form, series shared autonomy launches a fully autonomous takeover by the robotics autonomy when some trigger is activated by the user—such as based on a high-level user command. In such approaches, the user indicate via some interface a high-level goal or a command, and the robotics autonomy is fully responsible for generating the lower-level motion control commands [**26**, **59**, **124**] that gets executed on the robot. Such methods generally present an additional interface requirement (e.g., a screen (GUI) or a pointer for goal selection). While these systems can be effective, studies suggest that users often prefer having more control [**59**].

A more sophisticated approach formulates the control sharing problem as one of optimizing a Partially Observable Markov Decision Process (POMDP) over the user's goal to arbitrate control over a distribution of possible outcomes [**67**]. Specifically, the user input $\mathbf{u}_h$ is provided to an autonomy policy π^r which generates actions $\mathbf{u}_r$ to minimize an expected sum of robot cost function $C^r(\mathbf{s}, \mathbf{u}_r, \mathbf{u}_r)$, where $\mathbf{s} = (\mathbf{x}, g)$ denotes the environment state $\mathbf{x}$ and user goal g. In such policy based method, the autonomy actions are optimized given the user actions, and both the actions are passed to the robot for execution. The approach does not rely on predicting a single user goal but instead optimizes for an assistance action that is helpful for many goals. However, computing the optimal solution for a POMDP with continuous states and actions is generally intractable and thus in practice [**67**], the solution is approximated through Hindsight Optimization [**125**–**127**].

Another variant of series shared autonomy probabilistically models the interaction between the user's intention and the autonomy behavior as a joint probability distribution. In such probabilistic shared control (PSC) [**128**], the control commands for the robot are generated by formulating shared control as the maximum a posteriori estimation (MAP) of the joint probability distribution (set to the autonomy commands that maximizes the joint probability distribution),

$$p(\mathbf{u}_h, \mathbf{u}_r | \mathbf{z}_{1:t}) = \psi(\mathbf{u}_h, \mathbf{u}_r) p(\mathbf{u}_h | \mathbf{z}_{1:t}) p(\mathbf{u}_r, \mathbf{z}_{1:t})$$

$$(\mathbf{u}_h^*, \mathbf{u}_r^*) = \arg \max_{(\mathbf{u}_h, \mathbf{u}_r)} p(\mathbf{u}_h, \mathbf{u}_r | \mathbf{z}_{1:t})$$

$$\mathbf{u}_t = \mathbf{u}_r^*$$

where, $\mathbf{z}_{1:t}$ represents the measurements of the human intentions and the robot's state up until time t, and $\psi(\mathbf{u}_h, \mathbf{u}_r)$ is coupling factor that represents the degree to which the human inputs and the autonomy inputs agree or are similar.

Lastly, series shared autonomy also encompass approaches which assume no prior knowledge of the dynamics of the system and the user's control policy, but instead learn an end-to-end mapping from environmental observations and user inputs to autonomy action values [**129**, **130**]. Such approaches are particularly suitable for complex human-machine systems (e.g., quadcopters). However, learning an effective model typically requires good training data, which can be challenging and burdensome to obtain for human users operating physical robots, particularly in the assistive domain.

Definition 3.1.2. ***Interleaved shared autonomy*** *is a control sharing paradigm involving control partitioning schemes, wherein certain functions of the robot are human operated while the remaining functions are controlled by the autonomy (Figure 3.2).*

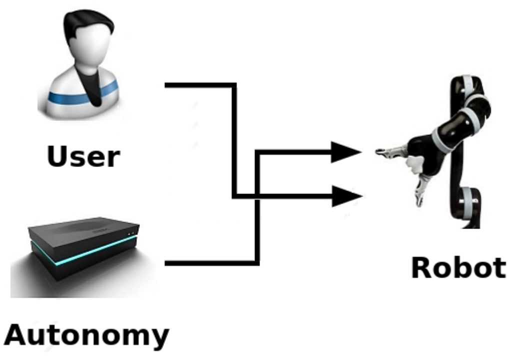


Figure 3.2. Interleaved shared autonomy with control partitioning.

Control partitioning in interleaved shared autonomy could be achieved in many ways. For example, the robot assumes full control of the orientation while the user is responsible for the translational motion [**131**]. Another work [**132**] place the control of end-effector position in z with the human and in x, y with the autonomy. The particulars of such control partitioning scheme might depend on the complexity of the task, robotic platform or preferences of the user. For assistive robotic arms involving mode switching, where users control subsets of the degrees-of-freedom of the robot in discrete modes, a related idea is for the autonomy to provide automatic mode switching assistance. Only recently shared control schemes for mode switching are gaining interest to make the teleoperation of assistive robotic arms easier [5,133], however their effectiveness have only been evaluated on a 2-D simulated robot [5].

Definition 3.1.3. ***Parallel shared autonomy*** *is a control sharing paradigm which simultaneously accepts both the manual user inputs and the autonomy inputs to mathematically reason about a shared-control signal that gets executed by the robot.*

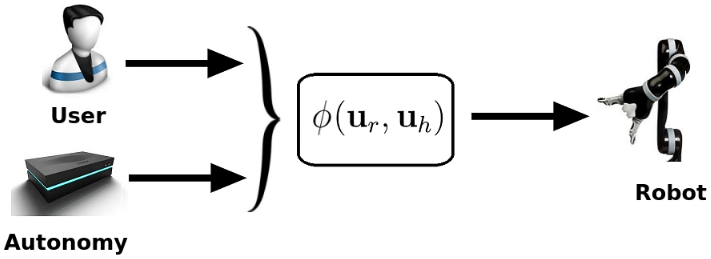


Figure 3.3. Parallel shared autonomy simultaneously accepts both the user and autonomy inputs to mathematically reason about control sharing.

Parallel shared autonomy is often implemented as a blending based method, that continuously combine the user control of the robot and the autonomy commands by some arbitration function $\phi(\cdot)$, and the arbitration determines the relative contribution of each (Figure 3.3). Such approaches have received considerable attention in recent research and blending is one of the most used shared control paradigm due to its computational efficiency, simplicity, and empirical effectiveness [18, 19, 64, 134–136].

Generally, linear blending is used to generate the control commands for the robot,

$$\mathbf{u}_t = (1 - \alpha) \cdot \mathbf{u}_h + \alpha \cdot \mathbf{u}_r$$

where $\alpha \in [0, 1]$ is a blending factor that dictates how much control lies with the human versus the autonomy. Thus, the level of autonomy is determined by α, and $\alpha = 0$ corresponds to no assistance and $\alpha = 1$ implies full autonomy. Assistance can be adjusted dynamically with arbitration and there exist works that study the nature of such arbitration [**64**, **134**]. For example, one common approach for arbitration is to consider the confidence in the robot's prediction of the user's goal.

3.2. User Personalization and Adaptation

Each person has a large set of qualities that make them unique. People vary significantly in their skill sets, culture, habits, behaviors etc. [**137**,**138**] and these qualities can affect both how people perceive robots, and also how robots perceive people. In addition to preexisting attitudes and expectations, a person's physical and cognitive abilities can greatly affect how a person interacts with a robot. There is great variation in people's physical abilities, and these can often change throughout the lifespan [**139**], including vision, hearing, mobility, strength and even controllable muscle movements. A user's cognitive abilities can also affect their interaction with a robot. These variations can be attributed to psychological or developmental disabilities, intelligence, language ability, problem solving skills or learning capability of individuals. These differences in physical and cognitive abilities across users greatly impact their technology expectations and likelihood of adoption [**140**]. User-centered design is a critical part of HRI that addresses the needs and preferences of human users, and existing literature have proposed various user-centered design frameworks [**141**–**144**].

Therefore, it is important to design robot systems that factor individuality since "one size does not fit all" and a robot behavior that is ideal for one person may not necessarily be ideal for another person. Moreover, such personalization has even more important implications in the assistive domain, where the user's abilities can potentially change from day to day [**145**]—either degrading due to stress, fatigue and degenerative conditions; or improving due successful rehabilitation and medication. For widespread adoption and incorporation of robots into assistive domain, there is a need for assistance personalization and adjustable autonomy based on individual preferences.

The idea of adjustable autonomy is recognized in different areas of robotics including socially assistive robotics [**140**, **146**, **147**], multi-robot systems [**148**, **149**] and space robotic systems [**150**]. In assistive robotics, customization of control sharing [**64**, **151**] and optimization techniques have been adopted to generate different strategies providing assistance under shared autonomy; for example, formulating the problem as a POMDP and inferring a distribution over goals [**67**] or concatenating energy-optimal motion primitives to create optimal trajectories [**152**]. Although existing approaches result in improved task performance, the assistance schemes are mixed in terms of user acceptance [**59**, **67**, **134**, **153**]. In particular, there are instances of assistance resulting in higher user dissatisfaction [**67**], and users preferring to be in control and more cautious [**152**]. In other studies users find the assistance at times to be uncooperative and tolerate a loss of control only for a significant improvement in performance [**68**].

Despite an improvement in task performance, existing works in shared autonomy were not able to guarantee high user satisfaction. The need for higher user satisfaction is crucial for the acceptance of robot autonomy by the end-users in the assistive domain. One way to achieve such behavior is through incorporation of a mathematical user model. Some approaches model human-robot systems as a cooperative two-player game [**154**–**156**], where the user and robot learn and adapt their strategies to each other. The user could even actively teach the shared autonomy system the assistance behavior they desire, and could even be aware of how the shared autonomy system adapts to their behavior, for example using cooperative inverse reinforcement learning [**157**]. Unfortunately, such models are computationally intractable in continuous domains and thus are not feasible in reality.

To address this gap, in this dissertation we introduce mathematical models for personalization to user for inference and assistance in shared autonomy. The motivation for our approach are prior studies which show that users vary greatly in their performance, preferences and desires [**59**, **67**, **134**, **153**], suggesting a need for assistive systems to adapt to each individual user. For improving intent inference and assistance, we model the user's actions within a probabilistic behavior model that incorporates an adjustable rationality and, importantly, we introduce an intent-driven optimization that adapts the rationality index value to each individual user, and thus can account for their particular behavior (Chapter 6).

CHAPTER 4

Research Platforms

The particular robot platforms used to validate the investigations of human-robot systems in this dissertation include, the smart wheelchair (Northwestern) and the MICO robotic arm (Kinova, Canada).

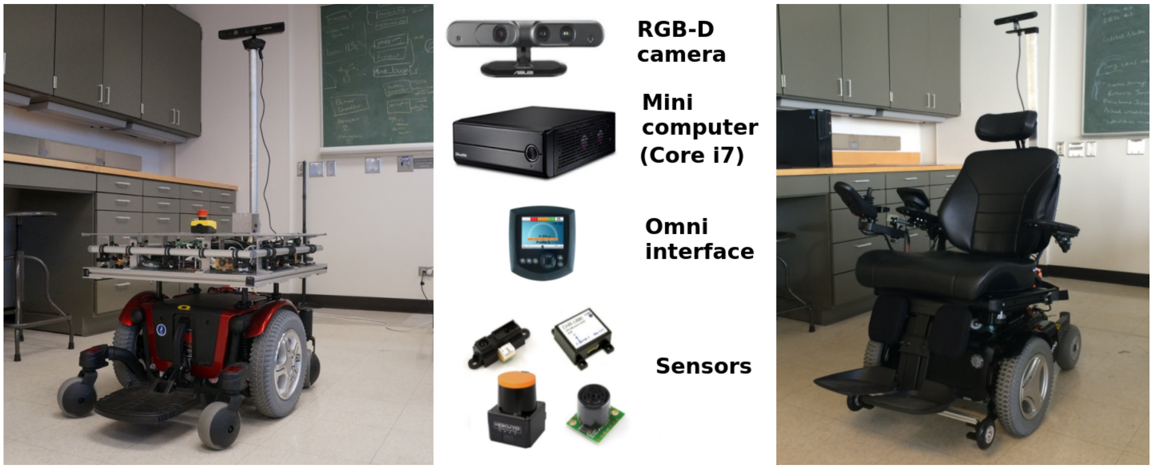


Figure 4.1. The smart wheelchair (Northwestern) research platform.

4.1. Smart Wheelchair

The smart wheelchair (Northwestern) is built on a commercially available powered wheelchair, a Permobil C300 (Timra, Sweden), outfitted with additional components including a computer, electronics and sensors (Figure 4.1). To interface with the wheelchair electronics, the control commands are passed directly to the proprietary wheelchair motion controller. Hardware add-ons to the wheelchair base system include on-board computing (a mini-PC with Intel Core i7), electronics boards, override buttons, wheel encoders and a top-mounted RGB-D sensor (Asus Xtion). The platform is augmented with additional

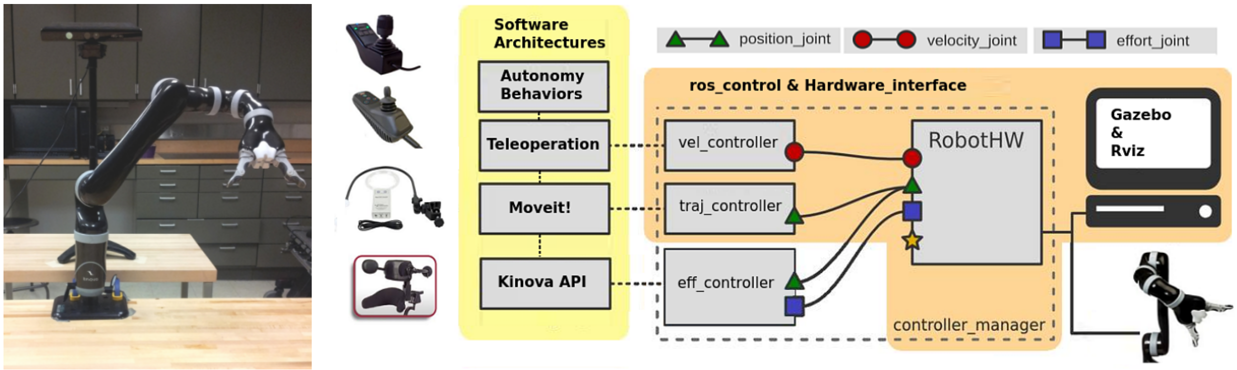


Figure 4.2. *Left:* The MICO robot platform (Kinova, Canada). *Right:* Our software architectures and ros control framework [**1**] integration for robotic arm operation under shared autonomy, with the control interfaces utilized in the assistive domain.

sensing including, a Hokuyo laser rangefinder, Sharp IR range finders, Maxbotix LV-EZ1 Ultra-sonic Range Finders and Ch Robotics UM6 IMU. The onboard computing system is directly powered by the wheelchair batteries. Control interfaces, through which the human provides input to drive the wheelchair, can directly be connected to the mini-PC computer via USB. The control framework [**65**] consists of a modular system of software components and the entire architecture is implemented within the Robot Operating System (ROS) [**34**]. The smart wheelchair (Northwestern) platform was used in the investigation of our algorithm for automated wheelchair docking (Chapter 5.2).

4.2. MICO Robotic Arm

The MICO robot (Kinova Robotics, Canada) is a 6-DoF manipulator with a 2 finger gripper (Figure 4.2, left). It is the research edition of the JACO arm (Kinova Robotics, Canada) which is used commercially within the assistive domain. The robot can be wheelchair-mounted or can also be mounted at a workstation. The arm can be teleoperated using the same interface that controls the powered wheelchair, typically a joystick. The control interface provided with the arm (3-axis joystick) has 3 DoFs and producing control commands requires either: tilting the stick forward or backward, tilting the stick left or right, and twisting the stick clockwise or counterclockwise. To control the robot arm's hand location and orientation, 6 DoFs are needed, and a seventh is required to open and close the gripper. Controlling 7 DoFs with a 3-axis joystick requires at least 3 control modes to divide the robot DoFs: translation mode, wrist mode, and finger mode. A control mode is switched or activated via buttons on the joystick, and an LED light pattern indicates the current mode.

Our developed software architecture (Figure 4.2, right) for the robot is implemented within the Robot Operating System (ROS) [**34**] and Moveit! [**35**], which enables joint and cartesian trajectory planning and execution on the robot. TRAC-IK [**158**] based implementation is utilized for inverse kinematics solutions. Robot controllers and robot hardware interface are implemented and managed under ros control framework for real-time performance and integration with the Kinova proprietary controllers. We furthermore implemented teleoperation functionality with limited control interfaces that typically are available to people with motor-impairments, including Sip-N-Puff and Head-Array. These control interface has 1 DoF and to control the robot arm's hand location, orientation, and gripper requires at least 7 control modes. The active mode is displayed via an Arduino based custom-built feedback display device using 7 LED light patterns (one for each mode). The MICO robotic arm and our control framework was used in the investigation of the models and algorithms developed in this dissertation.

CHAPTER 5

Algorithms for High-level Perception of Human Goals

5.1. Overview

In this chapter, we develop and evaluate novel algorithms for robotic perception of high-level user goals; with applications for assistive navigation of wheelchairs and shared-control manipulation of assistive robotic arms. The aim is to simplify the interaction between the user and the robot, by autonomous perception of high-level user goals in the environment—such as the candidate locations of interest to visit, or objects of interest. With such knowledge the robotics autonomy can efficiently generate assistance behaviors towards finite set of candidates in the environment. Furthermore, if the machine is able to autonomously perceive candidate user goals within the environment, this can reduce the burden on the user with respect to the requirements on the interface to indicate their intentions, and with a set of finite candidate goals, the process simplifies to inferring which element from this finite set—rather than needing to perform inference over the (infinite) set of all possible locations within the environment.

5.2. Automated Perception of Docking Locations

Powered wheelchairs provide a mobility solution for people unable to operate a manual wheelchair, for reasons of strength or impairment. However, operation of a powered wheelchair can still be a difficult, tedious or challenging task. In a survey of clinicians [**32**]

The work in this chapter was originally published as [**159–161**].

10% of patients who receive powered wheelchair training find it extremely difficult or impossible to use, and the number rises to 40% if they are asked about the maneuvering tasks associated with completing Activities of Daily Living (ADL). In order to provide assistance to this population, several "smart" wheelchairs have been proposed, that leverage robotics automation to assist in driving a powered wheelchair. Two spatially-constrained assistance behaviors that receive much attention within the literature are doorway-passage and docking. While multiple works address the autonomous perception required for doorway-passage [**162–164**], fully autonomous perception for docking assistance is minimally addressed.

In this dissertation, we develop automated perception algorithm to identify suitable docking surfaces in the environment, assess them for safety, and detect docking pose on the identified surfaces with accurate alignment or orientation information. The goal is to develop a complete docking assistance system for smart wheelchairs, which enables the robot to automatically perceive safe docking locations in the environment, with proper pose alignment.

5.2.1. Algorithm: Docking Perception

In this section, we present our algorithm for the automated perception of safe docking locations. Our algorithm first identifies candidate docking surfaces (Section 5.2.1.1), classifies them as rectangular or circular (Section 5.2.1.2) and extracts their edges/boundary (Section 5.2.1.3), and then scans along it for candidate docking locations which furthermore are assessed for safety (Section 5.2.2)). The input to our algorithm is a point cloud of the scene, where each point is represented by a tuple containing its 3D position in camera coordinates $\langle x, y, z \rangle$. The Kinect is mounted on the wheelchair 130 cm above the ground (Figure 5.1, left). In the camera coordinate frame the Y-axis is perpendicular to the ground, the Z-axis is back-to-front, and the X-axis is left-to-right. This information is then transformed onto the world coordinate frame and stored in the list χ_l of safe docking locations.

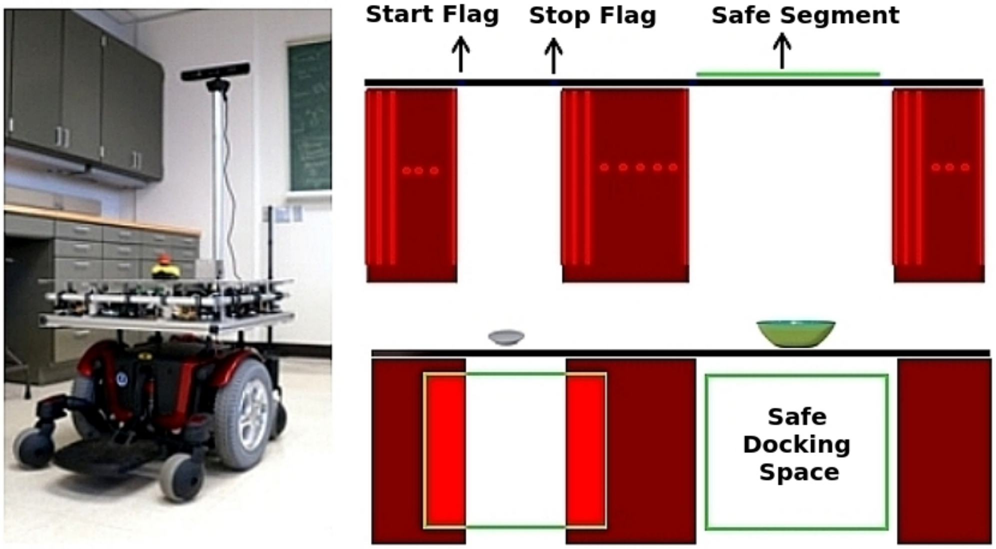


Figure 5.1. *Left:* Illustration of our smart wheelchair with Kinect. *Right:* The sequential search approach (top) and object anchoring approach for a restaurant/dinner table setting (bottom).

5.2.1.1. Candidate Docking Structure Extraction. The first challenge is to identify all the candidate docking structures, such as tables and desks, present in the scene. To identify such candidates, we use RANSAC to search for all the planar patches that satisfy the following conditions: (i) they are perpendicular to the Y-axis, $i.e$ the vertical axis of the camera frame; (ii) their height is at least τ above the ground plane; and (iii) their point cloud size is larger than η. In our implementation, we used $\tau = 28in$ and $\eta = 3000$. Each such planar patch is then clustered to form a set of planar regions P_c.

Once we have the candidate set P_c, the goal is to search each region $P_c^i \epsilon P_c$ for the set of locations χ_l that are safe for docking, where $\chi_l^i \epsilon \chi_l$ is a tuple containing the 3D position $\langle x_l^i, y_l^i, z_l^i \rangle$ of a safe docking location.

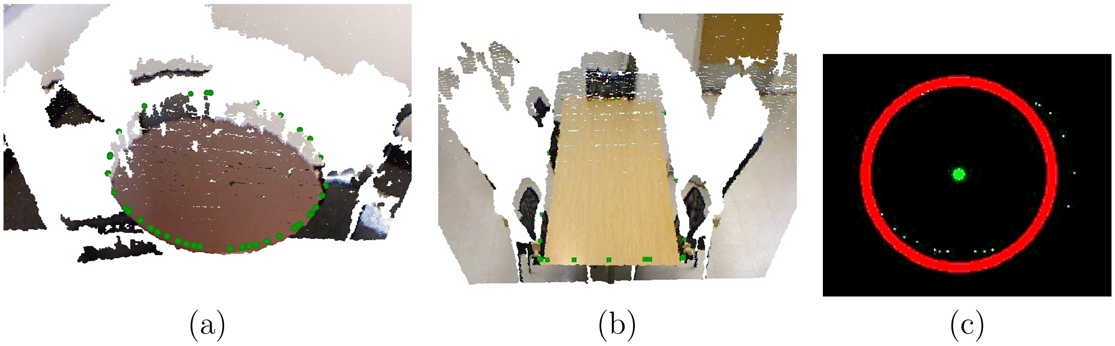


Figure 5.2. Example of the candidate structure shape classification by circle detection on the 2D-grid representation of the convex hull. Point Cloud of a scene with a round table *(a)* and a rectangular table *(b)*. Convex hull marked in green points. *(c)* Circle estimated from the circular table convex hull.

5.2.1.2. Candidate Structure Shape Classification. We restrict our search for candidate surface structures to shapes that are either rectangular or circular in geometry. We first reason about the shape (rectangular or circular) of the candidate structures in P_c, since the approach we follow to identify safe docking locations differs between the two.

To differentiate between the two shapes, we base our approach on the application of the Hough Transform technique to detect circle models. Instead of directly using the candidate structure point cloud data to run the Hough Transform, we extract the convex hull H_c^i for each $P_c^i \epsilon P_c$ and create a 2D-grid representation $G_{x,z}^i$ of P_c^i—using each pair $\langle x_c^{i,j}, z_c^{i,j} \rangle \epsilon H_c^i$ such that the grid size equals the area of the region P_c^i, and each grid cell that corresponds to a pair $\langle x_c^{i,j}, z_c^{i,j} \rangle$ is set to 1 with all others being set to 0. Note that quantization is performed in order to index the grid cells, and the y-coordinates are not considered here as we are interested in identifying candidate structures that are planar in geometry and perpendicular to the Y-axis. The 2D-grid representation $G_{x,z}^i$ simplifies running the Hough Transform and also reduces the time complexity. Based on the detection result of the Hough Transform we classify each $P_c^i \epsilon P_c$ to create a set of rectangular candidates R_c and a set of circular candidates C_c. An illustration of the circle detection on a round table structure is show in Figure 5.2.

5.2.1.3. Corner and Edge Extraction for Rectangular Structures. We detect each corner $R_c^i \epsilon R_c$ for two reasons: (i) to use as an anchor to initiate our search for safe docking locations, and (ii) to exclude corners from the set of locations χ_l (since it is unlikely that a user will want to dock at a corner location). We implemented and compared two approaches to identify corners.

The first approach is based on finding a rotated rectangle of the minimum area enclosing the input point set in R_c^i. Once we have the minimum area bounding rectangle for P_c^i, we compute the four corners using basic trigonometry.

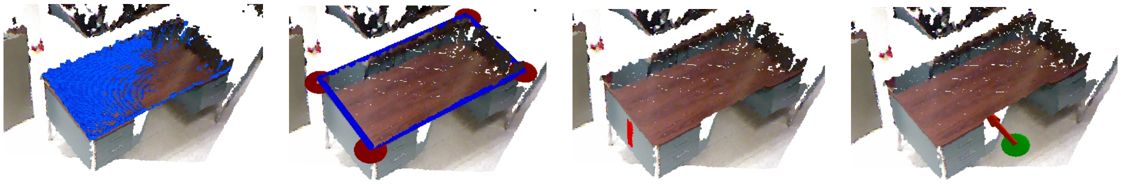
The second approach is based on the application of the Hough Transform to model lines in candidate region R_c^i and computing the line intersections to find the corner locations. We again create a 2D-grid representation $G_{x,z}^i$ as was done in Section 5.2.1.2, but this time we use the point cloud of R_c^i instead of using the convex hull. Furthermore, we apply morphological operations using a *3*x*3* square structuring element, $Mat(3,3)$, on $G_{x,z}^i$ to extract a smooth boundary for the candidate region R_c^i. Specifically, dilation $G_{x,z}^i \oplus Mat(3,3)$ is performed to blend the sparse regions in the point cloud data, followed by an erosion operation $G_{x,z}^i \ominus Mat(3,3)$. Boundary extraction is done by subtracting the result of the erosion operation from the dilation result, which gives the desired smooth boundary representation $B_{x,z}^i$ of the candidate structure. $B_{x,z}^i$ is then used to model lines by application of the Hough Transform.

A performance analysis of the two approaches for corner detection found the rectangle-fit approach to be superior to the Hough Transform approach (Tbl. 5.1, 10 runs for each of 3 viewing angles: 0°, 45° and -45° offset, so that the camera faces the edge and the

Table 5.1. Performance comparison for corner detection

Rectangle Fit Approach			
Offset	Avg.Time (ms)	True Corners	Spurious corners
0 degrees	0.20	40	0
45 degrees	0.21	40	0
-45 degrees	0.19	36	4

Hough Transform based Approach			
Offset	Avg.Time (ms)	True Corners	Spurious corners
0 degrees	17.07	36	3
45 degrees	18.23	32	3
-45 degrees	16.08	37	3



(a) (b) (c) (d)

Figure 5.3. Example of the sequential search approach. *(a)* Candidate surface extraction (blue region). *(b)* Corner locations (red markers) and extracted edges represented by line models (dark blue lines). *(c)* 3D box stripe (vertical red stripe) running along edges searching for free segments. *(d)* Identified docking location (green marker) and docking pose (red arrow).

two corners). Based on this evaluation, the rectangle-fit approach was integrated into our algorithm.

Once the four corners $\langle \kappa_c^{i,1}, \kappa_c^{i,2}, \kappa_c^{i,3}, \kappa_c^{i,4} \rangle$ for each P_c^i have been identified (Figure 5.3 (b)), line models $\langle L_c^{i,1}, L_c^{i,2}, L_c^{i,3}, L_c^{i,4} \rangle$ are computed between them. The edge points are computed according to (1), by progressively varying parameter λ. In this way the algorithm is able to identify the edges of the rectangular candidate structures present in the scene.

$$L_c^{i,j,k} = \kappa_c^{i,j} + \lambda^k \cdot (\kappa_c^{i,j} - \kappa_c^{i,l}) \tag{1}$$

5.2.1.4. Boundary Extraction for Circular Structures. In order to check the circular candidate structures C_c for the clearance needed for safe docking, we first extract the boundary of each $C_c^i \epsilon C_c$. We use RANSAC on the convex hull representation of C_c^i to find the model parameters of the circle. Having the center a and radius r for each C_c^i, the points $\langle Cb_c^{i,x,k}, Cb_c^{i,z,k} \rangle$ on the boundary are computed according to (2) and (3), by progressively varying parameter θ. The y-coordinate of each point, $Cb_c^{i,y,k}$, is then recovered from the point cloud data of C_c^i. In this way the algorithm is able to extract the boundary of circular candidate structures present in the scene.

$$Cb_c^{i,x,k} = a_c^{i,x} + r_c^{i,x} \cdot \sin(\theta^k) \tag{2}$$

$$Cb_c^{i,z,k} = a_c^{i,z} + r_c^{i,z} \cdot \cos(\theta^k) \tag{3}$$

5.2.2. Identification of Safe Locations

The next step is to scan for the safe docking locations for each of the candidate $P_c^i \epsilon P_c$. We propose two approaches that both make use of an oriented 3D box that we place or slide underneath the candidate structure, depending on the candidate edges or object clusters. We count the number of cloud points falling inside this 3D box and base our safety decision on this information. The size of the box is set based on table specifications in the Americans with Disabilities Act (ADA)[1] (26×30×19 *in* h×w×d).

5.2.2.1. Sequential Search—Identifying Safe Segments. For each candidate in P_c, we scan either along the four edges $\langle L_c^{i,1}, L_c^{i,2}, L_c^{i,3}, L_c^{i,4} \rangle$ or the boundary Cb_c^i. The 3D box is placed under the edge locations to find free segments $S_c^{i,j}$ that would be safe for the width of the wheelchair. Here, the 3D box slides along the four edges or the boundary of the

[1] Americans with Disabilities Act: `http://www.ada.gov/2010ADAstandards_index.htm`

structure candidate and is checked iteratively in sequential vertical strips (Figure 5.3 (c)). A safe docking location χ_c^i is marked as the midpoint of the box.

For alignment information for rectangular structures, we compute the docking pose orientation as the perpendicular direction to the corresponding edge that constitutes the safe segment (Figure 5.3 (d)). For circular structures, we calculate the direction of the vector pointing from the docking location to the center of the circle.

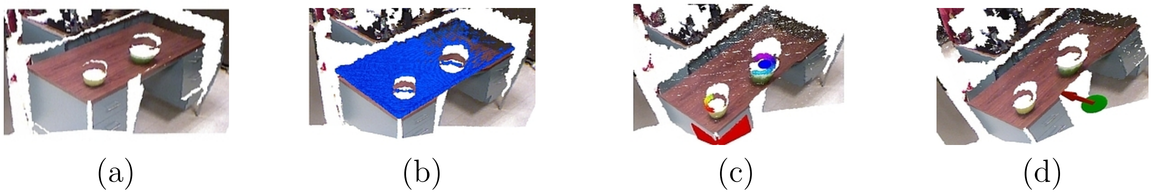


Figure 5.4. Example of the object anchoring approach. *(a)* Point Cloud of the scene with image overlay. *(b)* Candidate surface extraction (blue region). *(c)* Object Clusters (rainbow-colored circles), Hough Circle Detection (blue marker) and 3D box (red box). *(d)* Identified docking location (green marker) and docking pose (red arrow).

5.2.2.2. Restaurant/Dinner Table Setting—Object Anchoring. For each structure candidate in P_c (Section 5.2.1.1), we find object clusters O_c^i present on the candidate surface. To extract object clusters, we select the cloud points within a predefined threshold (2 cm $\leq \Gamma \leq$ 15 cm) above the surface and identify the points that fall within the convex hull of the candidate surface. A similar approach has been applied to extract door handles [**162**]. Next, object clusters are extracted by performing Euclidean clustering. We then base our candidate shape search on the identification of circle models in the object clusters O_c^i (Figure 5.4 (c))—since the objects considered are bowls, plates and mugs, we assume that the object clusters will have a circular shape. A 2D-grid representation $G_{x,z}^i$ for each of the object cluster O_c^i is created as described in Section 5.2.1.2, which is then used to detect a circle by the application of the Hough Transform. If a circle is detected, we place a 3D box underneath that cluster center to search for a safe docking space (Figure 5.4 (c)). For alignment information we identify the closest point on the edge or boundary (depending on the candidate structure shape) to the object of interest and use the method described in Section 5.2.2.1 to compute the orientation for docking alignment (Figure 5.4 (d)).

5.2.3. Evaluation and Results

A systematic approach to collecting data on docking structures was used to characterize the algorithm performance. The algorithm was validated for different configurations on four types of docking structures: a rectangular table, desk, workbench, and a round table (Figure 5.5). Experiments were carried out on an Intel Core i7 Quad-Core processor PC with 8 GB of RAM and running Ubuntu 12.04. Note that our approach is the first within the domain to address fully automated perception of safe docking locations on a variety of structures, and thus a comparison is not made with existing approaches.

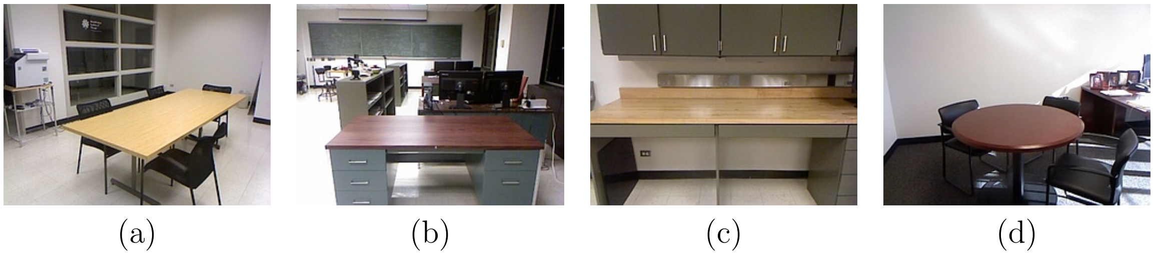


Figure 5.5. Docking structures used for evaluation. *(a)* Table *(b)* Desk *(c)* Workbench and *(d)* Round table.

To evaluate shape classification (Section 5.2.1.2) we tested each of the four docking structures for 10 runs of the algorithm. It was observed that they were correctly classified by the approach. Table 5.2 summarizes the results.

To test the sequential search and the object anchoring approach, different configurations that might be commonly encountered were evaluated. For rectangular structures each configuration was tested from three different views: 0°, 45° and -45° offset. Each combination

Table 5.2. Performance evaluation for shape classification

Docking Structure Type	Number	Classified as	
		Rectangular	Circular
Table	10	10	0
Desk	10	10	0
Workbench	10	10	0
Round table	10	0	10

of docking configuration and viewing angle was evaluated over 10 cycles. The point cloud stream was paused at each cycle of the algorithm to facilitate the online manual counting of correctly identified locations, falsely identified locations and missed locations.

The point cloud data underneath the candidate docking structures often is visible only for the sides closest to the sensor. Our algorithm considers locations to be safe until it has concrete data to say otherwise. Accordingly, docking locations identified on far sides of the tables are occasionally marked as safe when in fact they are not free. We consider this risk to be minimal however, since the wheelchair necessarily needs to drive near a location to dock at it, at which point the necessary sensor data becomes available. Such cases were not counted as false positives in the evaluation.

In each of the described cases, the algorithm performed quite well and succeeded in detecting the safe docking locations, with relatively few missed locations. Table 5.3 presents the performance data for each of the testing configurations, and the angles at which they were evaluated. The true positives were 305 (out of 350) and true negatives were 150 (out of 160), resulting in an overall accuracy of 89.21%. Figure 5.6 presents a number of examples of the algorithm at work on variety of docking structure configurations.

In order to achieve good run-time speed, the RANSAC algorithm was used for plane-fitting, which occasionally lead to somewhat strange surface selections and resulted in false positive identifications of docking locations. The effective working distance range of the algorithm was observed to be approximately 3 *m*. At distances greater than this, the

Table 5.3. Performance evaluation at varying offset angles on different docking structures and configurations

Structure & Configuration	**Offset Angle (°)**	**# Safe locations**	**True Positives**	**True Negatives**	**False Positives**	**False Negatives**
Table with 4 chairs	0 / 45 / -45	40 / 40 / 40	37 / 32 / 36	20 / 30 / 10	0 / 5 / 2	3 / 8 / 4
Desk	0 / 45 / -45	10 / 10 / 10	10 / 10 / 9	- / - / -	0 / 0 / 1	0 / 0 / 1
Desk - 2 objects	0 / 45 / -45	10 / 10 / 10	9 / 8 / 8	10 / 10 / 10	0 / 0 / 0	1 / 2 / 2
Workbench	0 / 45 / -45	20 / 20 / 20	19 / 17 / 18	- / - / -	0 / 2 / 0	1 / 3 / 2
Workbench - 2 objects	0 / 45 / -45	20 / 20 / 20	17 / 17 / 16	- / - / -	0 / 0 / 0	3 / 3 / 4
Workbench - 1 object, 1 chair	0 / 45 / -45	10 / 10 / 10	9 / 8 / 8	10 / 10 / 10	0 / 0 / 0	1 / 2 / 2
Circular table - 3 chairs	-	10	9	20	0	1
Circular table - 2 objects,3 chairs	-	10	8	10	0	2

horizontal planar segmentation became less reliable since the point cloud data returned by the Kinect decreased in accuracy. However, this operational range is useful and appropriate for the intended task of assisting wheelchair users, as the docking assistance is mostly required when the user is in close proximity to the docking structure. A series of videos were recorded demonstrating the performance of the algorithm.[2] Finally, we note that a quantitative comparison to other docking approaches is not provided, since fully autonomous perception for docking assistance is not addressed by any other works.

5.2.4. Summary

We have introduced a novel method for the autonomous detection of safe docking locations using 3D point cloud data, and without any visual fiducial or environment customization requirements. Through evaluations on differently shaped docking structures in varying configurations, our algorithm has demonstrated good performance and was shown to be effective in the identification of safe and oriented docking locations. Finding safe docking locations is important in the context of assistive wheelchairs, and also for autonomous mobile robots.

[2] The videos can be found at `https://www.argallab.northwestern.edu/research/robot-platforms/robotic-wheelchair/`

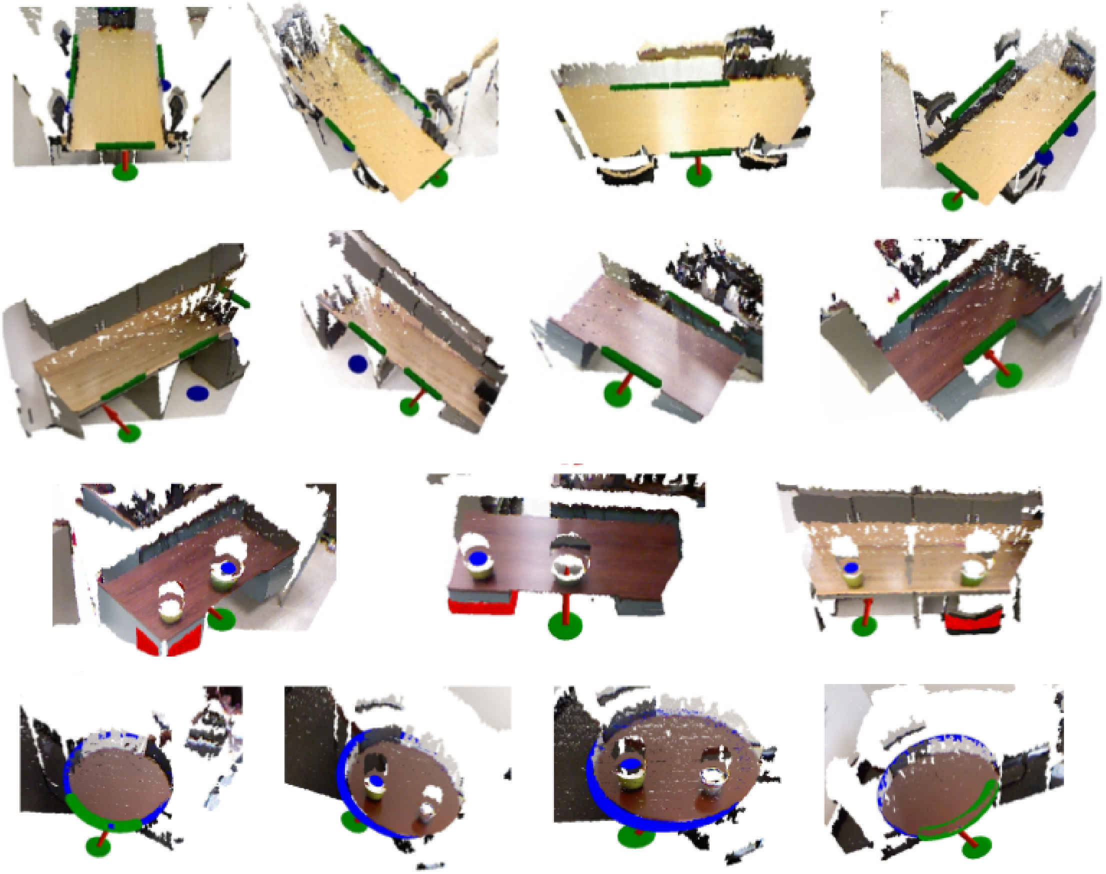


Figure 5.6. Examples of our algorithm working correctly on a variety of docking structures tested from varied viewpoints. The green marker indicates preferred docking location and the dark blue marker indicates all other docking locations, the red arrow marks the pose and blocked space is marked by red.

By detecting safe docking locations along with alignment information, custom trajectories can be planned and executed by a path planner to achieve the docking maneuver.

5.3. Grasp Detection for Assistive Robotic Manipulation

Robotic grasp detection is a challenging perception problem due to wide range of factors such as different object shapes and unlimited object poses. Over the past decades, several vision-based algorithms have been developed that compute grasps that are stable and suitable for use by an autonomous robotic system. However, such grasps may not be *predictable* to users in human-robot collaboration scenario. Our aim is to detect grasps that are suitable to be provided as high-level manipulation goals in shared-control systems. As part of this effort, in this dissertation we present a grasp detection algorithm that use real-time sensor data for grasping novel objects, with a focus on the two main components: (i) generates multiple grasps on objects to handle the multiple ways in which a single object can be grasped, and (ii) for the grasps to be similar to what a human would generate when teleoperating the robot under direct control.

While the use of deep convolutional neural networks (DCNN) has gained recent popularity in grasp detection, one of the most problematic requirements of deep learning lies in the requirement of a very large volume of training data [**165**]. Furthermore, although DCNN can be trained to find grasp location for an object in an image [**117**,**166**], it limits the grasp approach direction to the normal of the image plane and thus, impose the assumption that a good 2D grasp can be projected back to 3D space. Some approaches assume that the robot is supposed to grasp an object in a table-top scenario and thus limit the output of the DCNN to one overhead grasping pose for a given input image [**167**], for making the training process of the network easier. Detection of a single overhead grasp may not be suitable for shared-control operation in human-robot systems, where the human grasp choice could reflect the multiple ways in which a single object can be grasped. Furthermore, it is important to consider the depth information for robotic applications, as framing the perception problem using unimodal RGB image datasets can lead to failures in capturing the true semantic meaning of the world. In this dissertation, we explore a geometric approach for grasp detection that can use real-time sensor depth data and detect multiple semantic grasps on novel objects.

The set of everyday objects that a robot could encounter in its tasks is vast and diverse as household objects vary in their size, color, texture and shape. However, there exist certain regularities that can be exploited to inform and help robot perception in order to act on novel objects. In particular, the 3D surface geometries of objects contain characteristic information that generalizes over a large variety of objects. Geometrical analysis of a point's neighborhood can provide discriminative information about the local surface type. The identification of such local surface types in object point clouds can be utilized to reason about the shape and global surface characteristics of the object—without the need to recognize the specific object instance type or category. Such local and global surface geometry information can be used in manipulation planning for the robots. Thus, the addition of this semantic information to the 3D representation of object point clouds can enable the robot perception system to operate in a more general way—enabling the robots to perceive and manipulate unseen objects with more flexibility and reliability.

In the following sections, we first investigate methods for the estimation of local and global surface geometries in the 3D point clouds of household objects and then present our framework for grasp detection based on such geometric characteristics, that satisfies the two aforementioned main components for application in human-robot systems.

5.3.1. Surface Geometry Estimation

In this section, we discuss local surface geometry estimation to determine the overall shape characteristics of object clusters. These shape characteristics then will be used for determination of a global shape which best approximates the object geometry. For local surface geometry estimation in point clouds, we discuss methods based on Principal Curvatures (PC) [**168**] and Radius-based Surface Descriptor (RSD) [**169**].

RSD [**169**] describes the geometry of points in a local neighborhood by estimating their radial relationships. The radius estimation is performed by assuming each point pair to lie on a sphere. By exploiting the relation of the distance d between the points and the angle

α between the two point normals, the radius r is estimated,

$$\sqrt{2r}\sqrt{1 - cos(\alpha)} = d,$$

$$r \approx d/\alpha.$$

The radius approaches infinity for a planar surface and takes on increasingly lower values for surfaces with a higher curvature. The RSD feature for a point, based on its nearest neighbors, consists of a minimum radius and maximum curvature radius $[r_{min}, r_{max}]$ taken from all the point-neighbor spheres.

PC [**168**] describes a point's local surface geometry from the eigenvectors and eigenvalues of the principal surface curvatures on that point. For a query point $\boldsymbol{p}_q \in \mathbb{R}^3$, all normals $\boldsymbol{n}_j \in \mathbb{R}^3$, $j = 1...N$, of the set of its N neighborhood points are projected onto the tangent plane of the surface defined by the normal $\boldsymbol{n}_q$ at the query point. The centroid of the projected points $\bar{\boldsymbol{p}} \in \mathbb{R}^3$ and the covariance matrix C from all projections are computed as,

$$\bar{\boldsymbol{p}} = \frac{\sum_{j=1}^{N} \boldsymbol{p}_j}{N},$$

$$C = \frac{1}{N}\sum_{j=1}^{N}(\boldsymbol{p}_j - \bar{\boldsymbol{p}}) \cdot (\boldsymbol{p}_j - \bar{\boldsymbol{p}})^T$$

where

$$\boldsymbol{p}_j = (\boldsymbol{I} - \boldsymbol{n}_q \cdot \boldsymbol{n}_q^T) \cdot \boldsymbol{n}_j,$$

and $\boldsymbol{I}$ is a 3×3 identity matrix.

Principal Components Analysis (PCA) is then performed on the point normals of the surface patch in the tangent plane of the given point normal, to find the eigenvalues. The eigenvalues $\lambda_k \in \mathbb{R}$ and eigenvectors $\boldsymbol{v}_k \in \mathbb{R}^3$ are defined by the following relation and form an orthogonal frame that corresponds to the principal components of the set of neighborhood points:

$$C \cdot \boldsymbol{v}_k = \lambda_k \cdot \boldsymbol{v}_k,\ k \in \{0, 1, 2\}$$

where λ_3 corresponds to the maximum curvature and λ_2 to the minimum curvature denoted as pc_{max} and pc_{min}, respectively. The PC feature consists of these minimum and maximum curvatures $[pc_{max},\ pc_{min}]$ as well as the principal curvature direction $\boldsymbol{v}_3$, the normalized eigenvector of pc_{max} and pc_{min}.

Using the RSD and PC values, a geometric class label can be assigned to each point in the object cluster in order to classify local geometries. To determine the class label, thresholds on the feature values of the RSD descriptor are used to categorize local surfaces into classes *planar, cylindrical, spherical, edge, corner* or *noisy.* In particular, for each voxel with a width of 2.5 *cm*, we compute the minimum and maximum curvature radii $[r_{min}, r_{max}]$ and label the voxel surface by successive checks of the RSD radii in order to categorize it into one of the geometric classes. This is a simple and fast way to categorize local surfaces. Since RSD and PC are based on a similar geometrical approach (highest and lowest curvature), this technique is applicable to classify the PC values $[pc_{max}, pc_{min}]$.

5.3.2. Global Shape Classification

Many objects found on tables in typical household environments are cylindrical, spherical or box-like (that can be decomposed into planes), and thus can be well-represented by spherical, cylindrical and box-like geometries. To classify the shape primitive which best approximates the object geometry, we first present an empirical model based on local surface curvature and surface variation and then introduce a machine learning approach based on global feature representation for objects.

5.3.2.1. Empirical Model using Shape Variation. The Eigenanalysis (Principal Component Analysis) can be used to evaluate the local surface curvature around $\boldsymbol{p}_\star$. Specifically, eigenvalues λ_j of the covariance matrix C can be used as approximations of the local surface variation around $\boldsymbol{p}_\star$. The *surface variation* $\sigma_{\boldsymbol{p}_\star}$ at point $\boldsymbol{p}_\star$, calculated [**170**] as

$$\sigma_{\boldsymbol{p}_\star} = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}$$

provides an approximation to the change in curvature.[3] The surface variation $\sigma_{\boldsymbol{p}_i}$ is calculated for every point $\boldsymbol{p}_i \in O$, of the object cluster O.

We define a metric Δ that is used to determine the best 3D geometric primitive,

$$\Delta = \frac{M^*}{M}$$

M^*: Number of points $\boldsymbol{p}_i \in O$ for which $\sigma_{\boldsymbol{p}_i} \leq \delta$

M : Number of points $\boldsymbol{p}_i \in O$

where δ is an empirically-determined threshold (here $\delta = 0.01$). [4]

Higher values of Δ suggest that most of the object's points have low surface variation, and thus can be decomposed into planes and approximated by a box-like primitive. Small values of Δ suggest higher surface variation, and that the object can be approximated by a spherical or a cylindrical geometric primitive. The thresholds on Δ used for primitive classification (empirically determined) are presented in Table 5.4.

[3] Many alternative analyses of surface curvature require a mesh representation (rather than just sampled points). For example, the Gaussian and Mean curvatures of a smooth surface using a triangular mesh representation [**171**], which however do not provide a good enough representation here as there is strong correlation between estimated values. Other curvature formulations such as shape index are sensitive to noise and cannot be estimated from a set of sampled points directly.

[4] Geometric edges skew the interpretation of Δ. Edges are identified as points with extremely high curvature values by analyzing the distribution of curvatures $\sigma_{\boldsymbol{p}_i}$. Points in the upper quartile of this distribution are not included in Δ estimation.

Table 5.4. Classification for best fit primitive

Spherical	$0.0 \leq \Delta \leq 0.10$
Cylindrical	$0.10 < \Delta \leq 0.40$
Box-like	$0.40 < \Delta \leq 1.0$

5.3.2.2. Machine Learning using Global Descriptors. A global representation of object type can be formed from the local surface features (geometries). Specifically, for the RSD feature, once all voxels are annotated locally with a geometric class, the Global Radius-based Surface Descriptor (GRSD) [**169**] is computed, which produces a unique signature for a given object cluster based on the RSD local feature values. GRSD describes the transitions between different surface types (and free space) for an object. Note that in this work we count the transitions between surface types between the occupied voxels instead of along lines between occupied voxels. Each bin counts a transition between a pair of specific surface types, or between a specific surface type and free space. By using five local surface types (planar, cylindrical, spherical, edge and noisy/corner) and considering free space, the number of GRSD histogram bins is 21.

Following a similar approach, for the PC feature, we introduce a global histogram that counts transitions between the voxelized PC labels within an object cluster and also include bins for the number of voxels labeled as planar, cylindrical and spherical (empirically we found this to slightly increase the shape classification performance), making the bin size 24. We call this new assembled descriptor the Global Principal Curvature Shape Descriptor (GPCSD). Figure 5.7 shows the surface estimation on a cereal box object cluster.

We use machine learning to learn the mapping from global descriptors of objects to their primitive shape classes. For training dataset, we select 12 object categories from the large scale RGBD-Washington dataset [**172**]. For each category, there are four separate object instances in the dataset and each instance has more than 600 views of the object from

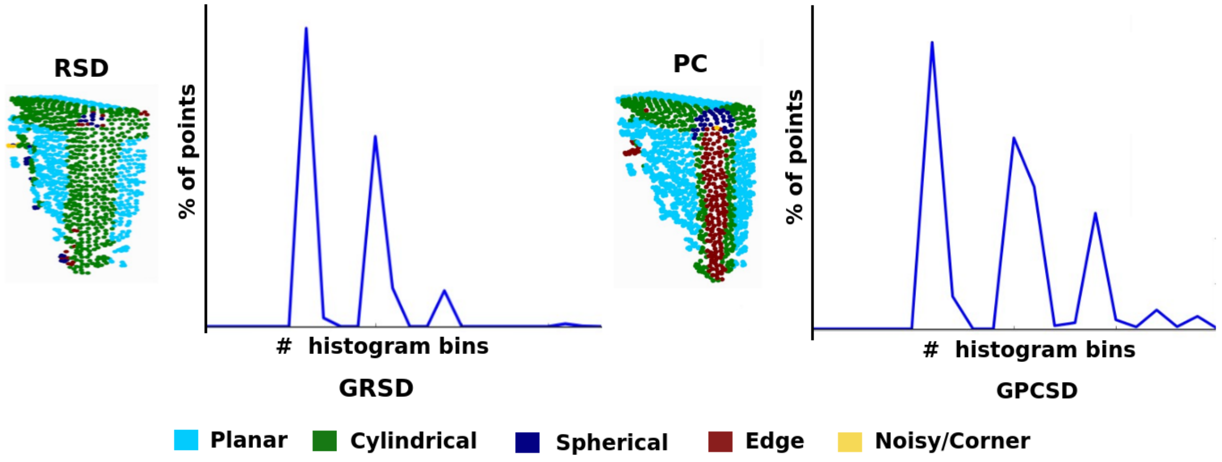


Figure 5.7. Surface geometry estimation for an example object cluster (cereal box). Local feature class labels shown as colors on the point cloud (*Left:* RSD, *Right:* PC). Global descriptor histogram bins shown as plots (*Left:* GRSD, *Right:* GPCSD).

different angles. Figure 5.8 shows an instance of each object for all categories along with the ground truth class labels. Note that RGB values from the dataset are only represented for illustration are not utilized in the descriptor computations. GRSD and GPCSD descriptors are computed for all instances of each object in the 12 categories.

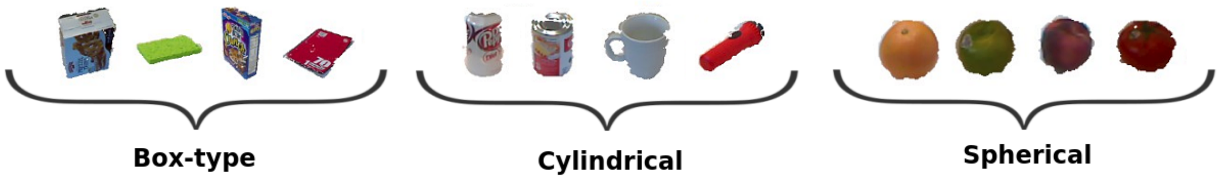


Figure 5.8. Test dataset showing instance of each object category used for evaluation.

An artificial neural network (ANN) is trained on three instances of each object category in the dataset, with one instance being randomly picked to leave out from each category in order to form the test set. The ANN structure has 21 input nodes for the GRSD descriptor (one node for each histogram bin) and 24 for the GPCSD descriptor, one hidden layer with

10 nodes and three output nodes for the class labels. (Note that multiple network structures were tested to prevent overfitting before arriving at this structure.)

5.3.3. Grasp Detection

We now present our algorithm for grasp detection that finds a combination of gripper orientations and positions relative to a given object using real-time sensor depth data. The sensory input to our framework is 3D point cloud data of the scene from a Kinect (Microsoft, USA) RGB-D sensor. The algorithm generates, for every observed object point cloud cluster O_j, a set of possible grasps $\mathcal{G}_{o_j}$. Each grasp $g^k_{o_j} \in \mathcal{G}_{o_j}$ is a $\mathbb{R}^{4\times4}$ transformation matrix (position and orientation of the robot gripper in base frame R_f) that can readily be used as a motion planning goal to grasp object O_j.[5]

The first step is to identify the parts of the input point cloud that are likely to belong to a single object (clusters). Note that any suitable segmentation approach can be used to identify the object clusters. In our implementation, two simplifying assumptions are made: the objects are sitting on a flat surface suitable for manipulation (such as a table), and the minimum distance between two objects in a scene is at least 3 *cm*. We compute a planar fit using Random Sample Consensus (RANSAC) to generate a model hypothesis for the flat surfaces and extract the dominant planar surface that provides support for the objects. We then find the individual object clusters O in the scene by performing Euclidean clustering.

Next, we model a simplified version of the object's geometry by automated categorization to 3D geometric primitives using global shape classification (Section 5.3.2). Such approximation does not need to match the object's geometry exactly, and need to only approximate the 2.5D depth data cloud well enough within some error bounds. We then generate model

[5] The remainder of this section describes operation on a single object $O_j \in \boldsymbol{O}$, and so for simplicity we drop the notation j.

hypotheses of the geometric model parameters for the identified shape primitives using RANSAC. The obtained models are refined using a linear least-squares fit for the plane and nonlinear Levenberg-Marquardt optimization for the cylindrical and spherical primitives.

The framework generates, for every observed object point cloud cluster O, a set of possible grasps $\mathcal{G}$, with the dual aims that grasps be similar to what the human user would generate when teleoperating the robot and to limit the huge number of possible grasps on the object.

We define strategies for detecting top, side and/or pinch grasps for each geometric primitive. Our algorithm generates multiple candidate grasps for each object—because we expect the user preference on where to grasp the object to change with environmental context. Since the aim also is for these grasps to be easily predicted by the human operators, we limit the full (infinite) scope of possible grasps to semantic groups (e.g., left side, right side, top) and generate a single grasp for each group. Our additional aim is for the generated grasps to be similar to what a human would generate during teleoperation of a robotic arm. A study involving grasping by human users [**123**] show that humans tend to grasp with wrist orientations that are orthogonal to the object's principal axis and its perpendiculars. Such grasps also generally are more stable than those that do not value orthogonality. Thus, we design our strategies to incorporate orthogonality by generating grasps along the object centroid, wherever possible, which also minimizes torque during grasp execution.

The model parameters for the spherical primitive consist of the sphere center $\boldsymbol{s_c}$ and radius s_r. We generate a top grasp pose and a side grasp pose (pinch is not feasible) by calculating the gripper position and orientation using the model parameters.

The model parameters for the cylindrical primitive consist of the axis vector $\vec{\boldsymbol{c}}_{\boldsymbol{a}}$, a point on the cylinder axis $\boldsymbol{c_p}$ and the cylinder's radius c_r. We project the inliers of the cylindrical

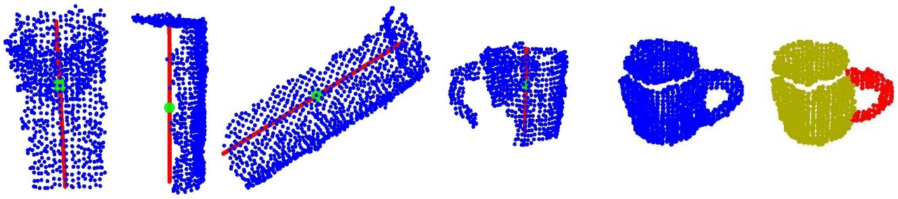


Figure 5.9. *Left:* Cylindrical objects center (green) and height (red line) computation. *Right:* Clusters of points on the cylindrical shape surface (yellow) are marked as protrusions (red).

model onto the axis vector and then compute the mean of the projected points to determine the centroid $\boldsymbol{c_c}$ of the cylinder. The height c_h of the cylindrical object is determined by computing the minimum and maximum bounds of the projected points along the axis $\vec{\boldsymbol{c}}_{\boldsymbol{a}}$. An illustration of this approach for different cylindrical objects is presented in Figure 5.9, left. Next, we look for possible spurious protrusions (including handles) from the object. Protrusions may hinder the grasp execution, and moreover in the case of handles we anticipate a potential handover to human partners. Therefore, objects are grasped on the opposite side of a protrusion. We employ a model-free approach to find protrusions rather than fitting specific geometric shapes—since the geometry varies for different kinds of objects, and we are grasping on the opposite side in any case. We examine the vicinity of the cylinder model's surface for clusters of points, and mark such clusters as protrusions (Figure 5.9, right). We also determine if the cylindrical object is in an upright or lateral position using the cylinder axis orientation. If the cylinder is in a *lateral position*, we compute a top grasp only. If it is *upright* and with *no protrusion*, we compute a side and top Grasp pose. For upright cylinders*with a protrusion*, the side Grasp is computed on the opposite side of the protrusion. To do this, we find direction vector $\vec{\boldsymbol{c}}_{\boldsymbol{pr}}$ along the protrusion cluster centroid and its projection on the cylinder axis. We then calculate unit vector $\hat{\boldsymbol{c}}_{\perp,pr}$, that is perpendicular to both $\vec{\boldsymbol{c}}_{\boldsymbol{pr}}$ and axis vector $\vec{\boldsymbol{c}}_{\boldsymbol{a}}$. For the case in which the cylinder the hollow (determined by filtering the number of points inside the surface), the algorithm generates a pinch grasp.

For box-like objects, we first model planar components. Specifically, the algorithm performs principal component analysis on the point cloud cluster O to obtain the eigenvectors $\vec{\boldsymbol{\nu}}_0$ and $\vec{\boldsymbol{\nu}}_1$, and the centroid $\boldsymbol{b_c}$. Next, the object is decomposed into planar components. We use RANSAC to extract planar models Π on the O iteratively until the size of O is greater than ρ (here ρ is set to 10% of the initial object cloud size). For each of the identified planes we compute its centroid $\boldsymbol{p_c}$ and the normal vector $\vec{\boldsymbol{n}}_{\boldsymbol{p}}$. Next, we find planes whose normals are in the direction of the major eigenvector $\vec{\boldsymbol{\nu}}_1$ to generate side grasps on the object and the planes whose normal is in the direction of $\hat{\boldsymbol{z}}$ to generate top grasp.

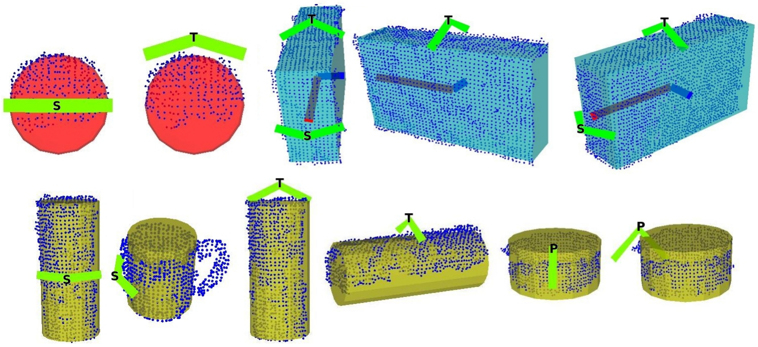


Figure 5.10. Grasp generation (top (T), side (S) and pinch (P)) for different shape primitives types, modeled from point clouds of object clusters.

The grasp generation strategies for shape primitives are summarized in Table 5.5 and Figure 5.10 shows the various grasps for different primitive types. Figure 5.11 summarizes the various steps involved in our algorithm.

5.3.4. Evaluation and Results

We performed three experiments to characterize the performance of our algorithm. We collected a set of 30 household objects that cover various categories, shapes, textures and sizes

Table 5.5. Semantic Grasp Generation using Global Shape Model Parameters

Global Shape Primitive	Model Parameters	Grasp
Spherical	center $\boldsymbol{s_c}$ and radius s_r	Side / Top
Cylindrical	center axis vector $\vec{\boldsymbol{c}}_{\boldsymbol{a}}$, a point on the cylinder axis $\boldsymbol{c_p}$ and the radius c_r.	Side / Top (upright) Top (lateral) Pinch (hollow)
Box-like	centroid $\boldsymbol{b_c}$, eigenvectors $\vec{\boldsymbol{\nu}}_0$ and $\vec{\boldsymbol{\nu}}_1$	Side / Top

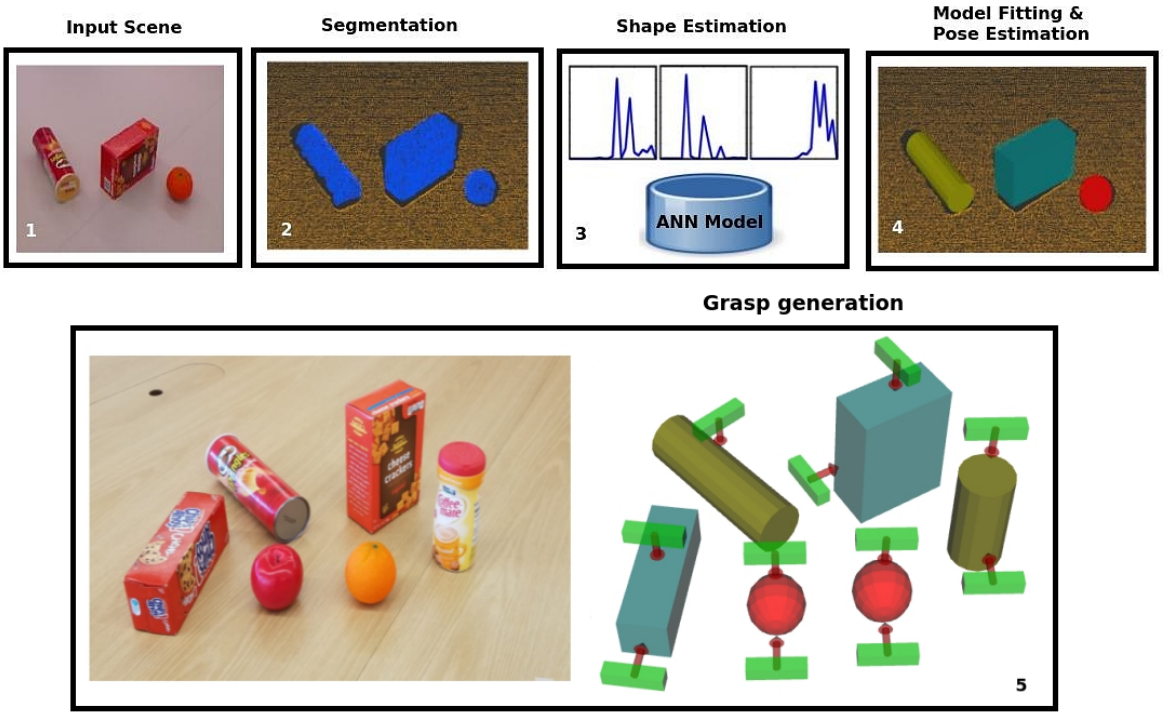


Figure 5.11. Grasp detection pipeline and sample grasps on a scene of objects. The predicted shape primitives are shown along with the grasps produced on the objects. The red arrow points along the approach direction, and the green box is positioned along the axis in which the gripper closes.

of objects (Figure 5.12), to test our approach.[6] More importantly, most object categories in the test set belong to the list of important objects identified for robotic retrieval tasks by motor-impaired patients [**173**]. Objects were presented in multiple *pose configurations*, consisting of 0°, 45°, -45°, 90° offsets. Experiments were carried out on an Intel Core i7 Quad-Core processor PC with 12 GB of RAM and running Ubuntu 12.04, the Robot Operating System (ROS Hydro) and Point Cloud Library (PCL 1.7). Grasp executions were carried out by the MICO robotic arm (6+1 DoF).

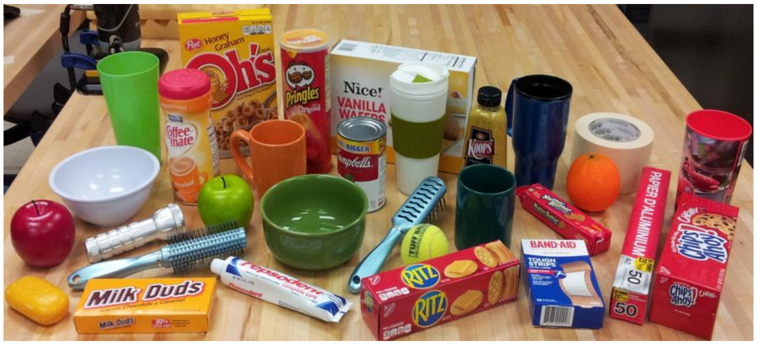


Figure 5.12. Test dataset of 30 household object.

Shape Primitive Selection: To evaluate shape primitive classification, we ran the algorithm using empirical model for global shape classification on each object in the test set, with all test pose configurations and lateral and upright positions when possible. Each pose configuration was evaluated for 10 runs of the algorithm. The algorithm selected the best shape primitive for the objects with very few errors. Table 5.6 summarizes the results.[7]

Additionally, we also tested the performance of the learned ANN models on the test dataset comprising of the RGBD-Washington dataset objects (Section 5.3.2.2). Both models perform well on the unseen test set objects and the classification accuracy of the learned

[6] Five of the test set objects—glass (green), mug (orange), tennis ball, coffeemate, plastic bowl—also were used in development of the algorithm (along with a box-like object not in the test set). The rest of the 25 objects in the test set were novel to the algorithm.

[7] For the purpose of grasp detection, long narrow box-like objects were represented equally well by cylindrical and box-like primitives.

Table 5.6. Evaluation of Best Shape Primitive Fit on the Test Objects
Key: #: Number of. %: Percent correct.

Object	#Pose	#Cylinder	#Sphere	#Box-like	%
Al. Foil box	40	10	0	30	75
Apple (green)	10	0	10	0	100
Apple (red)	10	0	10	0	100
Band-aid box	40	2	0	38	95
Cereals box	40	0	0	40	100
Ceramic bowl	10	10	0	0	100
Chocolate bar	40	30	0	10	100
Chocolate box	40	0	0	40	100
Coffeemate	50	50	0	0	100
Cookies box	40	2	0	38	95
Flashlight	40	40	0	0	100
Glass1	50	50	0	0	100
Glass2	50	50	0	0	100
Hairbrush1	40	40	0	0	100
Hairbrush2	40	40	0	0	100
Mug1	40	40	0	0	100
Mug2	40	40	0	0	100
Mustard	50	50	0	0	100
Orange	10	0	10	0	100
Plastic bowl	10	10	0	0	100
Pringles can	50	47	0	3	94
Ritz box	40	10	0	30	75
Soap	40	38	0	2	95
Soup can	50	49	0	1	98
Tape	10	10	0	0	100
Tennis ball	10	0	10	0	100
Toothpaste	40	30	0	10	75
Travel mug1	50	50	0	0	100
Travel mug2	40	38	2	0	95
Wafers box	40	0	0	40	100

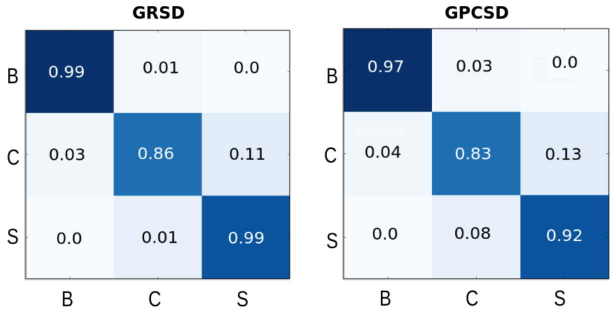


Figure 5.13. Normalized confusion matrices for GRSD and GPCSD classification using ANN models on the test set objects. Class labels are Box-shaped (B), Cylindrical (C) and Spherical (S).

GRSD model is 94.93% and that of the GPCSD model is 91.02%. The normalized confusion matrices are shown in Figure 5.13. The main performance difference between the two approaches appears to be the occasional mislabeling of spherical as cylindrical (1% for GRSD and 8% for GPCSD).

Grasp Similarity to Human Teleoperated Grasp: To evaluate the similarity between the algorithm generated grasps to those generated by a human when teleoperating the robot, a pilot study was performed in which three users (who were all lab members) teleoperated the MICO robot using a 3-axis joystick interface to grasp the test objects. The test objects were presented one by one on a table in front of the robot for the task in one of the above the pose configurations (selection was random and balanced). The users were not provided with any information about the algorithmic approach and were instructed to teleoperate the robot to grasp the object in a way they thought would be stable to lift the object from the table.

Before each trial the algorithm computed a set $\mathcal{G}$ of candidate grasps for the presented object O. As the user teleoperated the robot, the algorithm maintained a confidence score for each grasp $g \in \mathcal{G}$, based on: (1) the euclidean distance between the robot end-effector position and the grasp position, and (2) the end-effector orientation (roll, pitch and yaw) alignment with the grasp pose.

The most confident grasp g^* generated by the algorithm was then compared to the human-selected grasp g_h, defined as the final robot end-effector pose on the object with the robot gripper closed. For each trial we calculated the difference between poses g^* and g_h—as the difference in roll δR, pitch δP and yaw δY angles (in degrees), and the euclidean distance δT in position (in cm).

Table 5.7 provides results for each of the trials, and also reports whether the human selected a side (S), top (T) or pinch (P) grasp.

Table 5.7. Comparison of Algorithm Grasp Pose and User-Teleoperated Grasp Pose (see text for details)

Object	User 1					User 2					User 3				
	g	δR°	δP°	δY°	δTcm	g	δR°	δP°	δY°	δTcm	g	δR°	δP°	δY°	δTcm
Al. Foil box	T	11.45	-13.75	9.64	3.84	T	0.52	-6.88	-9.74	1.55	T	12.61	-7.45	13.18	5.19
Apple (green)	S	-11.28	9.74	-3.56	3.20	T	-1.72	4.01	-1.81	4.26	S	6.83	7.85	17.85	2.37
Apple (red)	S	-13.01	2.86	-9.83	2.66	S	9.12	1.83	2.55	5.04	T	3.08	10.31	24.21	6.23
Band-aid box	T	3.44	-5.16	6.21	1.92	T	-0.86	-1.88	-2.66	3.56	T	4.58	-7.96	8.03	4.58
Cereals box	S	-1.71	9.74	15.5	4.47	S	10.27	1.32	-3.59	5.52	S	2.48	10.89	5.24	9.06
Ceramic bowl	P	9.17	-2.29	40.83	10.21	P	1.72	0.92	-11.99	3.06	P	-1.72	3.70	4.67	5.22
Chocolate bar	T	4.01	-9.17	-6.63	1.17	S	-8.59	-2.86	0.34	3.71	T	-8.59	-3.15	4.09	8.53
Chocolate box	T	13.75	-5.16	-3.92	5.46	T	-3.44	-12.03	2.38	3.53	T	2.29	-4.41	12.09	4.67
Coffeemate	T	8.02	-8.50	1.27	5.21	T	9.17	-3.50	-1.14	4.20	S	-1.82	8.94	25.5	5.09
Mug1	S	5.68	10.88	21.1	4.00	T	0.00	-3.67	44.61	4.11	S	3.39	2.86	13.4	3.32
Mug2	S	0.47	6.88	12.43	3.26	S	3.97	7.45	7.48	1.62	S	-0.96	6.88	30.26	4.19
Cookies box	T	0.17	-5.73	-6.80	3.83	T	-6.30	-8.59	-0.61	5.61	T	-15.47	-8.02	-0.89	7.06
Flashlight	T	6.30	-4.58	3.63	4.84	T	10.89	-12.03	-3.97	4.66	T	0.52	-3.44	5.39	3.44
Glass1	S	3.97	9.74	-22.67	2.48	S	4.42	7.91	-7.54	3.76	S	3.79	12.61	33.06	4.90
Glass2	S	0.53	8.59	-12.81	5.73	S	5.11	6.88	23.01	4.74	S	2.43	6.44	21.51	3.71
Hairbrush1	T	-6.47	-2.29	-6.04	3.54	T	5.16	-8.59	-12.28	3.90	T	-10.31	-11.46	3.43	3.36
Hairbrush2	T	6.30	-13.75	-15.10	4.69	T	-11.46	-2.29	13.36	6.39	T	2.92	1.60	-8.51	4.52
Mustard	S	0.87	18.33	37.57	4.91	S	8.50	6.88	-3.07	3.58	S	7.40	6.53	23.70	2.59
Orange	S	3.39	4.58	2.78	5.28	T	26.67	-4.58	7.83	2.56	S	-3.71	12.60	22.32	6.04
Plastic bowl	P	22.92	-1.03	13.25	10.80	P	12.03	0.86	9.85	2.32	P	-2.80	-9.74	20.66	7.36
Pringles can	S	10.30	1.71	5.14	12.71	T	16.04	-11.46	1.75	3.22	T	3.21	-1.72	4.13	3.39
Ritz box	T	-4.01	1.95	-0.09	4.12	T	-10.31	-6.53	-6.56	6.17	T	-5.16	-6.30	3.35	4.62
Soap	T	13.18	1.15	7.36	1.34	T	-8.02	-9.74	-4.95	3.62	T	-15.47	-7.45	-9.13	3.54
Soup can	S	25.7	-9.74	20.20	7.38	S	4.54	4.76	-11.32	3.35	S	-2.68	-0.40	36.90	4.67
Tape	P	0.57	-2.86	-12.24	10.71	P	14.47	-7.45	-9.72	7.15	P	-1.49	-7.33	-6.39	4.27
Tennis ball	S	-1.80	9.16	10.64	2.66	S	-13.62	-4.58	-8.46	3.55	T	20.36	9.17	12.01	3.35
Toothpaste	T	6.30	-4.58	3.15	4.33	T	4.01	-5.73	2.78	3.04	T	10.89	-4.30	1.58	3.79
Travel mug1	S	1.10	5.16	23.3	5.93	S	3.97	8.02	5.13	3.77	S	-1.19	10.66	31.34	3.92
Travel mug2	S	-3.08	28.6	-21.2	3.06	T	4.54	4.24	25.24	4.97	S	3.97	9.74	20.16	4.82
Wafers box	T	8.59	5.39	-1.72	7.68	T	-6.88	5.16	2.61	5.20	T	9.91	-2.29	15.33	4.41
Mean		4.16	1.53	3.71	5.04		2.79	-1.73	1.65	4.05		0.97	1.17	12.94	4.74
S.E.		1.54	1.71	2.78	0.52		1.65	1.17	2.24	0.24		1.39	1.40	2.27	0.29

Across all users and objects (90 trials), the values for δR (2.64° ± 0.88), δY (6.10° ± 1.49) and δT (4.61 *cm* ± 0.21) were quite small. To help qualify *small*, Figure 5.14 provides a visualization of δR, δP and δY at 0°, 10° and 20°.

These results highlight the algorithm's capability to generates grasps that are similar in pose (both orientation and position) to what a human user generates when teleoperating the assistive robot. This similarity may help the human to predict the robot's motion during shared-control executions in collaborative tasks.

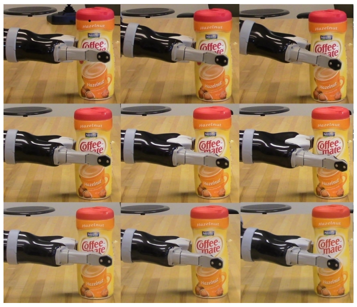


Figure 5.14. Visualization of δR (top row), δP (middle row) and δY (bottom row) for 0°, 10° and 20° (left to right).

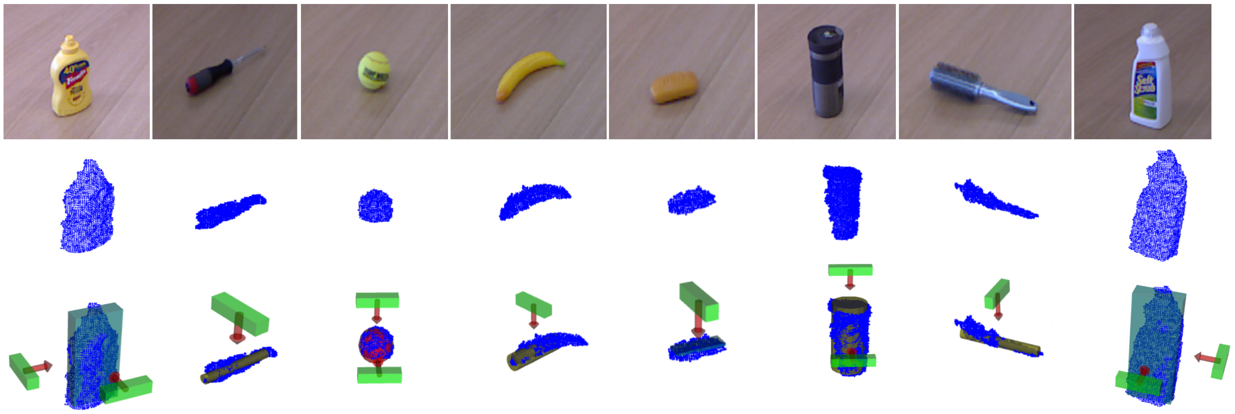


Figure 5.15. Pose estimation and grasp generation on a variety of novel objects. The segmented point clouds (middle row) are fit to primitive shapes (bottom row) and candidate grasps are generated. Grasp approach and orientation indicated by the red arrows and green boxes.

Grasp Robustness: Lastly, we evaluated our algorithm in experiments that autonomously grasped the test set household objects, again using the MICO robot. Each object from the test set was placed alone on a table within the reach of the robot in one of the pose configurations. The algorithm computed grasps $\mathcal{G}$ for the presented object, and each was executed using the manipulation planning framework [**174**] that allows the robot to plan in the presence of constraints on end-effector pose using the Task Space Regions (TSRs) representation. Figure 5.15 shows the pose estimation and grasp generation on a variety of novel objects. Success was defined by grasping the object, lifting it off the table and holding it in air stably for 60 seconds. There were 50 trials in total for the experiment and the robot achieved a success rate of 82%. Figure 5.16 shows the resulting grasps executed by the MICO robot on a sample of test objects.

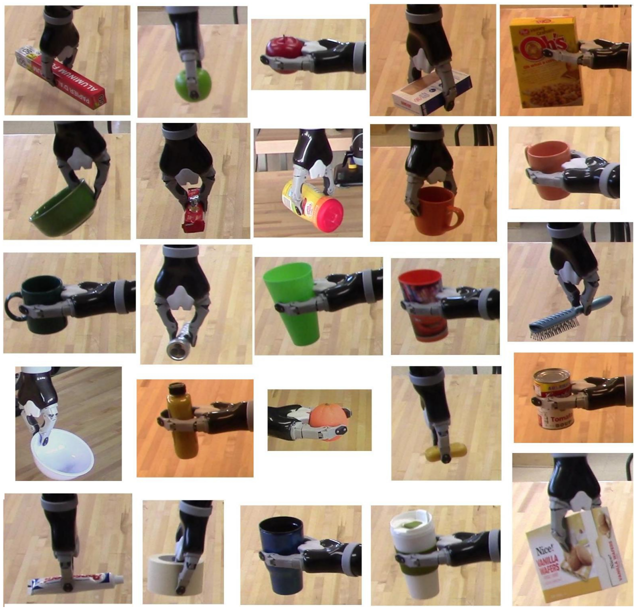


Figure 5.16. A sample of the grasps executed by the MICO robot.

5.3.5. Summary

We have presented an algorithm for detecting robotic grasps from point cloud depth data. Our method has several advantages over existing approaches for grasp detection as it does not require known models of the object or a predefined grasp dataset, and can generate multiple semantic grasps on objects. Furthermore, the algorithm is independent of an object recognition pipeline and can be used in real-world environments, without requiring expert intervention. The results have shown that the approach is capable of detecting human-like grasps for a wide range of unseen household objects, which could improve human-robot collaboration for assistive manipulation. The detected grasps also were used successfully for autonomous grasping by the MICO robotic arm.

CHAPTER 6

Human Intent Recognition and Assistance in Shared Autonomy

Human-robot teams are common in many application areas, such as domestic service, search and rescue, surgery and driving vehicles. In the area of assistive robotics, shared autonomy is utilized to aid the human in the operation of an assistive robot through the augmentation of human control of the robot with robotics autonomy [**58**, **59**, **64**, **66**]. One fundamental requirement for effective human-robot collaboration in shared autonomy is *human intent recognition*. In order to meaningfully assist the human collaborator the robot has to correctly infer the intended goal of the user from a number of potential task-relevant goals—known as the intent inference problem.

One approach to infer the user's intent could be to have the user communicate the intended goal explicitly, for example via verbal commands as observations. However, requiring explicit communication from the user could lead to ineffective collaboration and increased cognitive load [**9**]. Humans are very good at anticipating the intentions of others with non-verbal communication [**177**]. Research studies in experimental psychology show that humans demonstrate a strong and early inclination to interpret observed behaviors of others as goal-directed actions [**178**]. Evidence shows that even young human infants are inclined to interpret certain types of human behaviors as goal-directed actions. For example, a developmental proposition to the question of how the child develops a theory of mind posits that infants learn about the signal value of adult actions because these behaviors predict the locations of interesting objects and events [**179**]. Such a teleological interpretation of

The work in this chapter was originally published as [**175**, **176**].

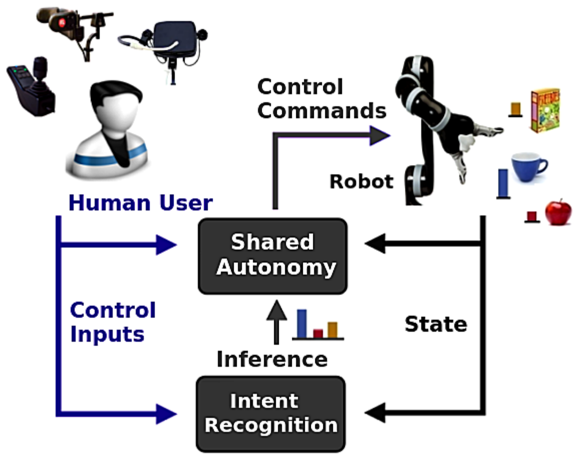


Figure 6.1. Human intent recognition in shared autonomy under limited control interfaces that typically are available to users in the assistive domain.

actions in terms of goals could inform us why the action has been performed, i.e., it provides a special type of explanation for the action.

Shared autonomy systems accept user actions as inputs into a robotic system via a control interface. The commercial control interfaces that are accessible to those with motor-impairments are limited in both the dimensionality and continuity of the control signals. Examples include 3- or 2-DoF joystick, and more limited interfaces (1-D control) such as Head Arrays and Sip-N-Puff control interfaces (Figure 6.1).

Directly leveraging the user's interaction with the control interface to perform intent recognition could pose critical challenges—as a consequence of the limitations of the interface and/or the human, due to the underlying motor-impairment. Taking inspiration from psychology and teleological interpretations of actions, we aim to explore how the robotics autonomy can implicitly observe and leverage such non-verbal human actions to perform human intent recognition. Specifically, we *model the user's interaction with the control interface* that they use to operate the robot as probabilistic goal-directed actions, in order to reason about the user's intention. Note that within the domain of humans operating assistive robots such as robotic arms or wheelchairs, control command information from the interface used to operate the robot is readily available and the operation of such an interface is already familiar to end users, and does not divert attention from the task execution.

We consider the 3-axis joystick and the Sip-N-Puff as the representative input interfaces, and study how challenging it is to decode the human user intentions in case of such limited interfaces. As discussed before, for modeling goal-directed actions we consider the user control inputs as representative of the action the user wants to take, we model such actions of the human agent in a probabilistic behavior model with adjustable rationality, and we use this model to inform human intent recognition. The adjustable rationality model allows to reason for a number of factors that might induce sub-optimality in the human actions (discussed further in Section 6.2).

In shared autonomy, often times the only cues available to the system comes from the input commands received from the human user (as discussed above) and, via sensing the robot's environment. In this dissertation, we present a formalism for intent inference under shared autonomy that models the uncertainty over the user's goal within a Bayesian filtering framework. Our approach incorporates a fusion of multiple observations and introduces a probabilistic modeling of user control inputs as goal-directed actions. Specifically, we model the user's actions within a probabilistic behavior model that incorporates an adjustable rationality and, importantly, we introduce an intent-driven optimization that adapts the rationality index value to each individual user, and thus can account for their particular behavior. The motivation for our approach are prior studies which show that users vary greatly in their performance, preferences and desires [**63**, **67**, **68**], suggesting a need for assistive systems to adapt to each individual user.

In the majority of the literature on shared-control systems that perform intent inference, the focus of the experimental work is on the control sharing and robot assistance [**64**–**66**, **180**], with the intent inference being mostly assessed only implicitly. Intent inference performance can directly affect different aspects in shared autonomy operation, however, thorough evaluations of intent inference are largely missing from the shared autonomy literature. In our work, we present a more extensive evaluation of intent inference than is typically seen within the literature.

6.1. Framework for Intent Inference under Shared Autonomy

We first mathematically define the intent inference problem and then present our framework. Our target domain is assistive robots endowed with shared autonomy that assists the user towards his/her intended goal.

6.1.1. Problem Formulation

Assuming the environment has a discrete set of accessible goals[1] $\boldsymbol{g}$, known at runtime to both user and robot, the intent inference problem is that the intended goal of the user is unknown to the robot, and so the robot must infer (predict) the most likely goal $g^* \in \boldsymbol{g}$ that the user is trying to reach. With knowledge of g^* the robotics autonomy can meaningfully assist the user in shared autonomy. Note that the user might change his/her intended goal during the execution, and thus the intent inference must update in real time so that the assistance provided can dynamically adjust.

[1] Potential goals could be detected by perception methods; such as recognizing objects clusters/grasps [**160**] or saliency maps [**181**] for manipulation and navigation goals such as doorways [**79**] and docking locations [**159**].

6.1.2. Intent Estimation

We formulate the intent inference problem for shared autonomy as Bayesian filtering in a Markov model, which allows us to model the uncertainty over the candidate goals as a probability distribution over the goals. Bayesian models have shown to be effective for inference in cognitive science [**182**] and human-robot interaction research [**62**]. We cast the intent inference problem as a classification task where the robot aims to infer the most likely goal class g^* from the set of possible goals $\boldsymbol{g}$, given a set of observations (features).

We represent the goal g_t as the query variable and the observed features Θ_0,...,Θ_t as the evidence variables, where Θ_t is a k-dimensional vector of k observations θ_t^i, $i = 1 : k$, and t represents the current time. For compactness, we use colon notation to write Θ_1,...,Θ_t as $\Theta_{0:t}$. The uncertainty over goals is then represented as the probability of each goal hypothesis. The goal probability conditioned on a single observation source θ^i over t timesteps can be represented by Bayes' rule as,

$$b_t(g) = P(g_t \mid \theta_{0:t}) \propto P(\theta_t \mid g_t, \theta_{0:t-1}) P(g_t \mid \theta_{0:t-1}) \tag{6.1}$$

where the superscript i has been dropped for notational simplicity, and the posterior probability $P(g_t \mid \theta_{0:t})$ at time t represents the belief $b_t(g)$ after taking the single observation source into account, where $b_t(g)$ is a single element of the posterior distribution b_t. Since the Hidden Markov Model allows for a conditional independence assumption between observations at the current and previous timesteps given the current goal estimate ($\theta_t \perp \theta_{0:t-1} \mid g_t$), we simplify $P(\theta_t \mid g_t, \theta_{0:t-1})$ to $P(\theta_t \mid g_t)$. Applying the law of total probability, the conditional goal probability becomes,

$$b_t(g) \propto P(\theta_t \mid g_t) \sum_{g_{t-1} \in \boldsymbol{g}} P(g_t, g_{t-1} \mid \theta_{0:t-1}) \tag{6.2}$$

ALGORITHM 1: Recursive Bayesian Intent Inference

```
Given Goals $\boldsymbol{g}$
Initialize $P(g_{t=0})\ \forall g \in \boldsymbol{g}$
Initialize $b_{t=0}(g) \leftarrow P(\Theta_{t=0} \mid g_{t=0})P(g_{t=0})\ \forall g \in \boldsymbol{g}$
Normalize $b_{t=0}$
while executing do
    Observe $\Theta_t$
    foreach $g \in \boldsymbol{g}$ do
        $b_t(g) \leftarrow \prod_{\theta_t \in \Theta_t} \mathrm{P}(\theta_t \mid g_t) \sum_{g_{t-1} \in \boldsymbol{g}} P(g_t \mid g_{t-1}) b_{t-1}(g_{t-1})$
    end
    Normalize $b_t$
    Update intent $g_t^* \leftarrow \arg\max_{g \in \boldsymbol{g}} b_t(g)$
end
```

which becomes,

$$b_t(g) \propto P(\theta_t \mid g_t) \sum_{g_{t-1} \in \boldsymbol{g}} P(g_t \mid g_{t-1}) b_{t-1}(g_{t-1}) \tag{6.3}$$

upon applying the definition of conditional probability to $P(g_t, g_{t-1}|\theta_{0:t-1})$ and the Markov assumption, $(g_t \perp \theta_{0:t-1} \mid g_{t-1})$. The computation of $b_t(g)$ thus is a recursive update, and so encodes memory of prior goal distributions. Furthermore, $P(g_t \mid g_{t-1})$ is the conditional transition distribution of changing to goal g_t at time t given that the goal was g_{t-1} at time $t-1$. The model thus encodes that the user's intent or goal can change over time.

We now take into consideration multiple observation sources $\theta^1, ..., \theta^k$ as k evidence variables, which can derive from any number of sources—for example, control commands or cues such as eye gaze. We assume the k observations sources to be conditionally independent of each other given a goal g, $(\theta^i \perp \theta^j \mid g)$, $\forall i \neq j$. Thus, Equation 6.3 becomes

$$\begin{aligned} b_t(g) = P(g_t \mid \Theta_{0:t}) \propto \prod_{\theta_t \in \Theta_t} P(\theta_t \mid g_t) \\ \sum_{g_{t-1} \in \boldsymbol{g}} P(g_t \mid g_{t-1}) b_{t-1}(g_{t-1}). \end{aligned} \tag{6.4}$$

We present our algorithm for inferring the probability distribution over goals in Algorithm 1. The posterior distribution at time t, denoted b_t, represents the belief after taking observations into account. The set of prior probabilities $P(g_{t=0})$, $\forall g \in \boldsymbol{g}$, initially represents the robot's belief over the goals. The beliefs then are continuously updated, by computing the posteriors $P(g_t \mid \Theta_{0:t})$, $\forall g \in \boldsymbol{g}$, as more observations become available.

Finally, to predict the most likely goal $g_t^* \in \boldsymbol{g}$, we select the goal class that is most probable according to the maximum *a posteriori* decision,

$$g_t^* = \arg\max_{g_t \in \boldsymbol{g}} P(g_t \mid \Theta_t). \tag{6.5}$$

6.1.3. Inference Uncertainty

Within the domain of shared-control assistive robotics, it is important to express uncertainty in the robot's prediction of the intended goal—because assisting towards the wrong goal could be worse than providing no assistance. We express prediction uncertainty as a *confidence* computed as the difference between the probability of the most probable and second most probable goals,

$$C(\boldsymbol{g}) = P(g^* \mid \Theta) - \arg\max_{g \in \boldsymbol{g} \setminus g^*} P(g \mid \Theta). \tag{6.6}$$

When the robot is uncertain about the intended goal of the user, a variety of behaviors might be implemented, for example to hold off on providing assistance or assist towards multiple goals simultaneously if possible.

6.1.4. Human Behavior Model as an Observation

We are interested in investigating how we can utilize for intent recognition the indirect signals people implicitly provide to operate the robot. Taking inspiration from psychology and teleological interpretations of actions [**178**], we aim to explore how the robotics autonomy can implicitly observe and leverage non-verbal human actions to perform human intent recognition. Specifically, we model the user's interaction with the control interface that they use to operate the robot as probabilistic goal-directed actions, in order to reason about the user's intention. We consider the user inputs as representative of the actions the user wants to take to reach a goal g. We model these actions as observations using Boltzmann-rationality, which has been shown to explain human behavior on various data sets [**183**].

We incorporate adjustable rationality in a probabilistic behavior model such that at any state $\boldsymbol{s}$ (robot configuration) the probability that action $\boldsymbol{u}_g$ is chosen by a rational human agent to reach goal g is given as,

$$P(\boldsymbol{u}_g \mid \boldsymbol{s}, g) \propto \exp(\beta \cdot Q_g(\boldsymbol{s}, \boldsymbol{u}_g)) \tag{6.7}$$

where $Q_g(\boldsymbol{s}, \boldsymbol{u}_g)$ denotes the Q-value when the intended goal is g. β is a rationality index (discussed further below) that controls how diffuse are the probabilities. We model $Q_g(\boldsymbol{s}, \boldsymbol{u}_g)$ as the cost of taking action $\boldsymbol{u}_g$ at configuration $\boldsymbol{s}$ and acting optimally from that point on to reach the goal g. We approximate optimal action selection with an autonomy policy, and compute this cost as the *agreement* between the user control $\boldsymbol{u}_h$ and the autonomy control $\boldsymbol{u}_r$ to reach the goal g. In our implementation, a policy based on potential fields [**184**] is employed, and agreement is measured in terms of the cosine similarity, computed as $\arccos\left((\boldsymbol{u}_h \cdot \boldsymbol{u}_r)/ \parallel \boldsymbol{u}_h \parallel \parallel \boldsymbol{u}_r \parallel\right)$, where $\cdot$ denotes the dot product and $\parallel \cdot \parallel$ denotes vector norm. A moving average filter with a two second time window is applied to the observations, in order to consider a brief history of observations and reduce any undesired oscillations due to noisy or corrective control signals.

6.2. Intent-driven Optimization for User Personalization

A perfectly rational model assumes that a human agent always acts optimally to reach his/her goal, when in reality a number of factors might induce sub-optimality—for example, limitations and challenges imposed by the lower degree of freedom (DoF) interfaces (e.g., 3-axis joystick, Sip-N-Puff, Head Array) available to control high-DoF robotic systems, cognitive impairments, physical impairments, and environment factors such as obstacles or distractions. Furthermore, not only users differ in their abilities but also in the manner in which they interact with the robot. Therefore, a critical detail is to include adjustable rationality in the human action model for user personalization and adaptation.

We introduce an intent-driven optimization of adjustable rationality to customize value of β for each user. Adjustable rationality is represented by the rationality index parameter β in the Boltzmann policy model (Equation 6.7). β reflects the robot's belief about the optimality of the human agent actions. By tuning β, the robot can account for human agents who may behave suboptimally in their actions, and thus can make better inference from their actions. In order to find the value of the rationality index for a human agent, we perform an optimization procedure on data gathered while the human teleoperates the robot to reach multiple goals in the environment. In particular, we find the value β^* that minimizes the average log-loss for goal inference across a dataset trajectories given the true labels Y for the intended goal and the probability estimates P for inference,

$$\beta^* = \arg\min_{\beta} \mathcal{L}_{log}(Y, P, \beta) \tag{6.8}$$

and the log-loss is computed as,

$$\mathcal{L}_{log}(Y, P, \beta) = -\frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N} y_{i,j} \log p_{i,j}(g_i \mid \Theta, \beta) \tag{6.9}$$

where M is the number of samples in the trajectory, $y_{i,j}$ is a binary indicator of whether or not the prediction j is the correct classification for goal instance i, and $p_{i,j}$ is the probability associated with the goal j at timestep i, in case of N goals. The value of β^* returned by the optimization procedure is then used to model the human actions using equation 6.7 and $P(\boldsymbol{u}_g \mid \boldsymbol{s}, g)$ are used as observations to make inference predictions.

6.3. Autonomy Inference Implementation

Our approach, Recursive Bayesian Intent Inference (RBII), allows for the seamless fusion of any number of observations to perform human intent recognition. In order to examine how the incorporation of multiple observation sources affects the intent inference and shared autonomy, we implement two different observation schemes (RBII-1 and RBII-2).

RBII-1: The first observation scheme considers a single modality, the *proximity* to a goal, as this feature is utilized most in existing shared autonomy work [**63**, **64**, **66**, **180**]. We compute proximity θ^d as the Euclidean distance between the current position of the robot end-effector $\boldsymbol{x}_r$ and the goal $\boldsymbol{x}_g$. For Algorithm 1, we model the likelihood using the principle of maximum entropy such that given the goal g, the class conditional probability decreases exponentially as the likelihood of g decreases, $P(\theta^d \mid g) \propto \exp(-\kappa \cdot \theta^d)$. κ is set to the mean of the range of values that θ^d can take.

RBII-2: In the second observation scheme we consider probabilistic fusion of observations, i.e., in addition to proximity to the goal, we model the actions of the human agent as probabilistic observations. Following a model of human action from cognitive science [**182**], we model the user as Boltzmann-rational in their actions to reach a goal g (discussed in Section 6.1.4).

Lastly, our approach encodes the possibility that the user's goal might change during task execution (Section 6.1.2). We compute the probability of changing goals in the case of n number of goals as,

$$P(g_t = g^i \mid g_{t-1} = g^j) = \begin{cases} 1 - \Delta & \text{if } i = j \\ \Delta/(n-1) & \text{otherwise.} \end{cases} \tag{6.10}$$

Note that when $\Delta = 0$, the model represents the case when the user exclusively pursues one goal during the execution. When $\Delta = (n-1)/n$, the model represents the possibility of choosing a new goal at random at each timestep. Our implementation initializes the probability distribution over goals to be uniform, and sets $\Delta = 0.1$.

6.4. Assistance under Shared Autonomy

Assistance under shared autonomy involves both the user and robotics autonomy acting simultaneously to control the robot, in order to achieve the user's goal (Chapter 3). In this section, we discuss implementation details of assistance under shared autonomy that utilizes the inferred intent. Blending based shared autonomy methods combine the user and autonomy control of the robot by some arbitration function that determines the relative contribution of each. Blending is one of the most used shared autonomy paradigms due to its computational efficiency and effectiveness [**63**, **64**, **135**]. Furthermore, it has been shown in subjective evaluations that users tend to prefer blending and find it easier to learn over some other approaches of shared autonomy, for example as compared to probabilistic shared control (PSC) [**185**] and a POMDP-based policy method [**67**]. We implement a blending-based paradigm to provide assistance,

$$\boldsymbol{u}_{blend} = \boldsymbol{u}_h \cdot (1 - \alpha) + \boldsymbol{u}_r \cdot \alpha, \tag{6.11}$$

where $\boldsymbol{u}_h$ denotes the user control command, $\boldsymbol{u}_r$ the autonomy control command generated under a potential field policy [**184**] and $\boldsymbol{u}_{blend}$ is the shared autonomy command sent to the robot. Note that $\boldsymbol{u}_r$ is available in all parts of the robot state space, for every goal $g \in \boldsymbol{g}$ such that g is treated as an attractor, and all the other goals $\boldsymbol{g} \setminus g$ as repeller.

$$\alpha = \begin{cases} 0 & C(\boldsymbol{g}) \leq \delta_1 \\ \frac{\delta_3}{(\delta_2 - \delta_1)} \cdot C(\boldsymbol{g}) & \delta_1 < C(\boldsymbol{g}) \leq \delta_2 \\ \delta_3 & C(\boldsymbol{g}) > \delta_2 \end{cases} \tag{6.12}$$

$\alpha \in [0, 1]$ is a blending factor which arbitrates how much control remains with the human user versus the autonomy. In our implementation, α is a piecewise linear function of the confidence in the intent prediction where $C(\boldsymbol{g})$ is defined as in Equation 6.6, the difference between the highest and second highest probable goals. δ_1 is a lower bound on $C(\boldsymbol{g})$ (set to 30%) below which assistance is not active, δ_2 is an upper bound on $C(\boldsymbol{g})$ (set to 90%), above which assistance is maximum. The upper bound on assistance α is given by δ_3 (set to 70%). Note that: (i) Different approaches to intent inference will generate different values for $C(\boldsymbol{g})$, and so the amount of assistance accordingly will differ. (ii) In particular, if $C(\boldsymbol{g})$ is lower, meaning that the inference is not very certain in its prediction, the amount of assistance also will be lower. (iii) If the inferred goal is wrong the robot will assist towards the wrong goal, with potentially serious implications.

6.5. Experiments on Intent Inference

Our experimental work aims to evaluate the performance of the intent inference algorithm, as well as its impact on shared autonomy. We furthermore are interested in how the

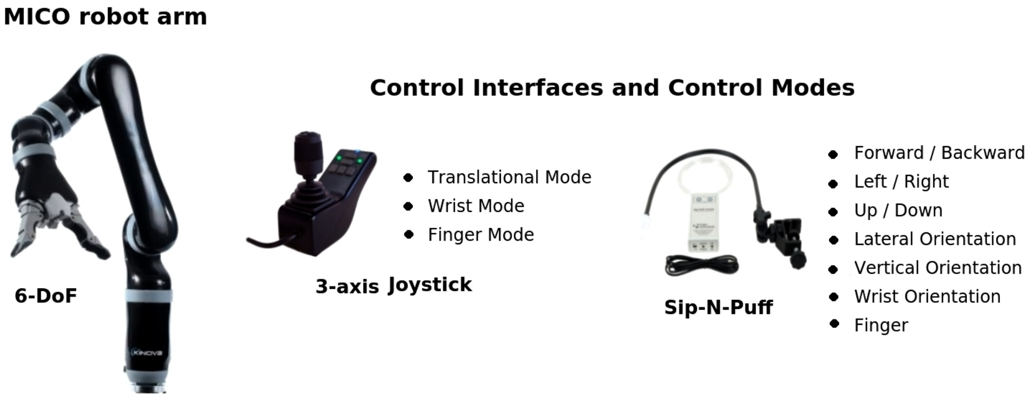


Figure 6.2. *Left:* MICO robotic arm. *Right:* The control interfaces used in the study and illustration of the respective control modes for the robotic arm teleoperation.

control interface used by the human affects the intent inference performance. This is particularly important in the domain of assistive robotics, as traditional joystick control interfaces are often not accessible to people with severe motor impairments, who instead use limited 1D control interfaces like Head Array or Sip-N-Puff to operate the robot. We performed two human subject studies, that aim (i) to characterize the complexity and variability of the intent inference problem, (ii) to compare the inference performance of our approach to existing approaches utilized in prior shared autonomy work and (iii) to evaluate the impact of inference on shared autonomy assistance.

Robot Platform: Our research platform for the designed experiments was the MICO robotic arm (Kinova Robotics, Canada), a 6-DoF manipulator with a 2 finger gripper (Figure 6.2, left). The MICO is the research edition of the JACO arm (Kinova Robotics, Canada) which is used commercially within the assistive domain.

Control Interfaces: The control interfaces used in the study were (i) a traditional 3-axis joystick (Kinova Robotics, Canada) that is typically been utilized for operating robotic arms and (ii) a limited 1-D Sip-N-Puff interface (Origin Instruments, United States) accessible to people with severe motor-impairments. Direct teleoperation using control interfaces, which are lower dimensional than the control space of the robotic arm, requires the user to switch between one of several control modes (mode switching). Modes partition the controllable degrees of freedom of the robot such that each control mode maps the input dimensions to a subset of the arm's controllable degrees of freedom. The control interfaces and defined control modes are shown in Figure 6.2, right. Teleoperation with the 3-axis joystick requires at minimum 3 control modes and the required number of control modes increases to 6 in the case of the Sip-N-Puff interface.

In addition to the two variants of our algorithm detailed in Section 6.3 (RBII-1 and RBII-2), for comparative purposes we also implement two approaches to intent inference utilized in previous shared autonomy works—Amnesic Inference [**63**–**65**] and Memory-based Inference [**56**, **63**, **66**, **67**].

Amnesic Inference: The Amnesic inference approach associates a confidence in the prediction of the user's goal as a hinge-loss function, where it is assumed that the closer a goal is, the more likely it is the intended goal,

$$c(g) = \max(0, 1 - \frac{d}{D}) \tag{6.13}$$

where d is the distance to the goal and D a threshold past which the confidence $c(g)$ is 0. It is possible to design richer confidence functions, but in practice this function often is used for its simplicity. The approach is termed as *amnesic prediction* [**63**], because it ignores all information except the instantaneous observations. In our implementation, d is the Euclidean distance $\| \boldsymbol{x}_g - \boldsymbol{x}_r \|$ between the current position of robot $\boldsymbol{x}_r$ and the goal $\boldsymbol{x}_g$, D is set to 1.0 m (maximum reach of the MICO robot arm).

Memory-based Inference: The Memory-based prediction [**63**] approach is a Bayesian formulation that takes into consideration the history of a trajectory to predict the most likely goal. Let $\xi_{x \to y}$ denote a trajectory starting at pose x and ending at y. Using the principle of maximum entropy [**186**], the probability of a trajectory for a specific goal g is given as $P(\xi \mid g) \propto \exp(-c_g(\xi))$; that is, the probability of the trajectory decreases exponentially with cost. It is assumed that the cost is additive along the trajectory.

$$P(\xi_{s \to x} \mid g) = \frac{\exp(-c_g(\xi_{s \to x})) \int_{\xi_{x \to g}} \exp(-c_g(\xi_{x \to g}))}{\int_{\xi_{s \to g}} \exp(-c_g(\xi_{s \to g}))}. \tag{6.14}$$

Such solution becomes too expensive to compute in high-dimensional spaces (e.g., for robotic manipulation), and so [**63**] estimates the most likely goal, by approximating the integral over trajectories using Laplace's method and first order approximation,

$$g^* = \arg\max_{g \in \boldsymbol{g}} \frac{\exp(-c_g(\xi_{s \to x}) - c_g(\xi^*_{x \to g}))}{\exp(-c_g(\xi^*_{s \to g}))} P(g). \tag{6.15}$$

In practice the cost c_g is often the Euclidean distance and the goal probabilities are initialized with a uniform prior [**66**, **180**], which we do in our implementation as well.

6.5.1. Human Inference Study

We first aim to ground the complexity of the intent inference problem through a study of human inference ability, by having a human observer interpret the motion of a robotic arm to infer its intended goal. Humans are very good at anticipating the intentions of others with non-verbal communication [**177**] and by observation of goal-directed actions [**178**].

In our experiments, the robot motion trajectories were pre-recorded demonstrations by a human expert operating a 3-axis joystick to reach a goal. Related work has addressed the topic of human interpretation of robot motion trajectories generated by autonomy policies [**56**] with the aim to generate intent inexpressive trajectories. We however are interested in robot motion generated by *human* control commands, given our target domain of assistive teleoperation. We study human inference under a variety of goal scenarios of varied complexity (Figure 6.3, left).

We furthermore are interested in how the *change of intent* can affect the inference. Our human demonstrator therefore provided two types of motion: (i) No change of intent – the robot motion maintained a single goal from start to end and (ii) Change of intent – the robot motion switched during the execution the goal it was reaching towards.

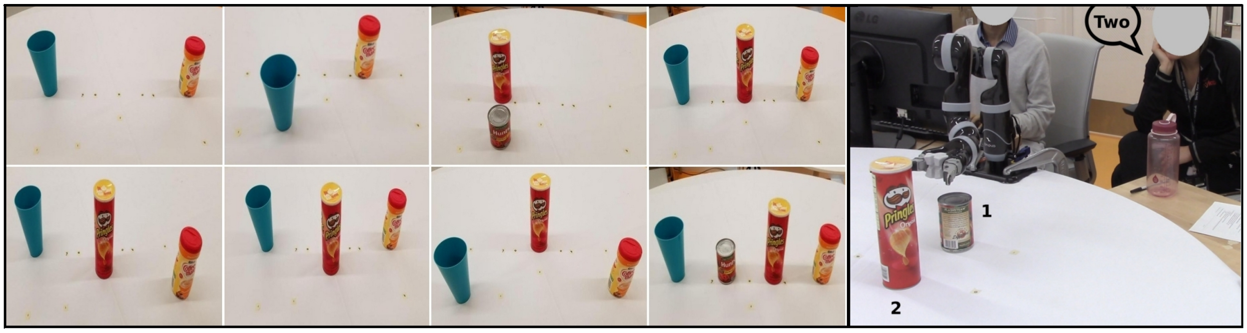


Figure 6.3. *Left*: Goal scenarios of varied complexity used in human inference study. *Right*: Example setup shows the robot executing a trajectory to reach a goal. The subject predicts the intended goal during robot motion.

Design: The study involved a variety of goal scenarios with varied complexity: in total, 8 different scenarios involving 2 to 4 potential goals (Figure 6.3, left). For each scenario, one change of intent demonstration was recorded (8 in total) as well as no change of intent trajectories for each goal in the scene (22 trajectories). We chose a within-subjects design and each subject observed the 30 recorded demonstrated trajectories replayed on the robot.

Subject Allocation: We recruited 12 subjects without motor-impairments from the local community (5 male, 7 female, aged 19-35). The subjects were novice users and had no prior experience operating a robotic arm. All participants gave their informed, signed consent to participate in the studies, which was approved by the Northwestern University Institutional Review Board.

Protocol: At the start of the experiment, the subjects were given a five minute training period in which they observed one pre-recorded human teleoperation trajectory towards a random goal in the environment, to get familiarized with the robot motion capabilities. The 30 teleoperated trajectories for the 8 goal scenarios then were executed (re-played) in counterbalanced order on the robot. For each trajectory motion, we tasked the human subjects to observe the motion of the robot and predict which object was the intended goal by verbally mentioning the "Object Name" (e.g.,"cup") or "Number" (e.g., "one"). The subjects were allowed to change their inference at any time during the robot motion. They were also given the option to express *uncertainty* about the intended goal, by saying "not sure" or "zero". Note that by default the inference is registered as uncertain ("zero") at the start of the trajectory motion. The subject inferences were recorded by the experimenter via a button press. Figure 6.3 (right) shows the experimental setup for the study.

6.5.2. Autonomy Inference Study

Our second study aimed to evaluate how well the autonomy could infer the intent of a user under *novice teleoperation* and how the inferred intent affects *assistance* in shared autonomy.

Design: We adopted a between-subjects experimental design with two conditions that differed in the control interface type used by the subjects to operate the robot: (i) a 3-axis joystick and (ii) a 1-D Sip-N-Puff interface. All other elements were same in the two conditions. Four intent inference approaches were evaluated: (i) Amnesic inference (ii) Memory-based inference (iii) RBII-1 and (iv) RBII-2. They each computed the inference

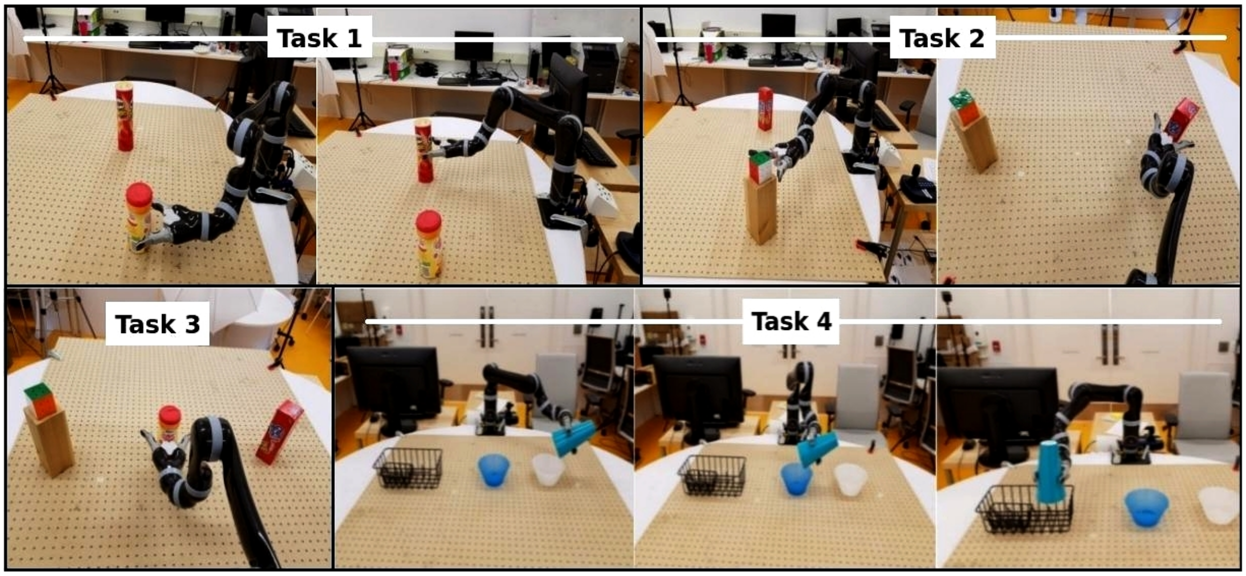


Figure 6.4. Task scenarios and associated grasp pose on goals used to evaluate intent inference under novice teleoperation (with and without shared autonomy). Note that for Task 3 the two goals on the extreme have the same grasp pose as shown in task 2.

online as the subject teleoperated the robot to complete tasks. Four different tasks of varied complexity were used to evaluate the intent inference performance (Figure 6.4). Fetching objects and manipulation for meal preparation have been identified as the top preferred ADL tasks [**8**]. The first three tasks involved object retrieval where the goals were the approach grasp pose $\in \mathbb{R}^6$ on the objects. The fourth task involved pouring and placing operations where the goals were tied to the manipulation operations, the pose of the initiation of pouring over the bowls and placing in the dish rack.

Subject Allocation: We recruited 24 subjects without motor-impairments from the local community (11 male, 13 female, aged 19-40) and 6 end-user subjects (5 male, 1 female, aged 18-47). Twelve of the subjects without motor-impairments (5 males, 2 females) and 3 of the end-user subjects (2 male, 1 female) used the 3-axis joystick interface, and the remaining subjects used the Sip-N-Puff interface. The end-user group included 4 subjects with spinal cord injury (SCI), 1 with muscular dystrophy (MD) and 1 with cerebral palsy (CP). All participants gave their informed, signed consent to participate in the studies, which was approved by the Northwestern University Institutional Review Board.

Protocol: At the start, the subjects were given a ten minute training period in which they got familiar using the control interface (a 3-axis joystick and a Sip-N-Puff) and the robot operation. Next, they teleoperated the robot to perform a training task (a similar setup as shown in Task 3 in Figure 6.4, but with different positions and orientations of the objects.) and the data of which was then used to perform optimization procedure to tune the rationality index parameter β for each subject (as discussed in Section 6.2). The optimized value β^* was used in the RBII-2 approach for intent inference.

There were two stages in the study protocol. In stage one, the subjects teleoperated the robot *without assistance*. The subjects performed the tasks shown in Figure 6.4, and they were instructed to (i) complete each goal in every task setup and (ii) for each task perform one additional trial in which they *change goal* during the task execution. The change of goal was recorded with a time stamp via a button press. All four approaches for intent inference computed the inference online as the user teleoperated the robot to complete tasks. In stage two, all the previous trials were performed again but now *with assistance* under shared autonomy. Furthermore, stage two trials were performed twice, once with the RBII-1 approach and once with the RBII-2 approach.

6.6. Analysis and Results

We first discuss performance measures and then present the analysis with the results. For each performance measure, one factor repeated measure ANOVA (Analysis of Variance) was performed to determine significant differences ($p < 0.05$) between the intent inference approaches. Once the significance was established, multiple post-hoc pairwise comparisons were performed by using Bonferroni Confidence interval adjustments. For all figures, the notation $*$ implies $p < 0.05$, $**$ implies $p < 0.01$, $***$ implies $p < 0.001$.

6.6.1. Performance Measures

Percentage of Correct Predictions: Percent correct predictions is a metric commonly employed in machine learning works. We compute percent correct as the percentage of time (success rate) the inference identified the correct intended goal of the user with confidence ($C(\boldsymbol{g}) > 30\%$).

Log-loss: Assessing the uncertainty of a prediction is an important indicator of performance which is not captured by the percent correct metric. The cross-entropy or log-loss considers prediction uncertainty by including the classification probability in its calculation. In the case of N goals, and given the true labels Y for the intended goal and the probability estimates P for inference, we calculate the average log-loss across a trajectory as,

$$\mathcal{L}_{log}(Y, P) = -\frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N} y_{i,j} \log p_{i,j} \tag{6.16}$$

where M is the number of samples in the trajectory, $y_{i,j}$ is a binary indicator of whether or not the prediction j is the correct classification for goal instance i, and $p_{i,j}$ is the probability associated with the goal j at timestep i. Note that a perfect inference model would have a log-loss of 0 and the log-loss increases as the predicted probability diverges from the intended goal. Note also that log-loss is unable to be computed for the Amnesic inference, as it is not a probabilistic method.

Task Completion Time: It is important to consider how quickly the user is able to complete a task with and without assistance. Our intuition is that the intent inference affects the shared autonomy assistance and thus will indirectly affect the task completion time. That is, better inference will result in correct, earlier, and stronger assistance.

Number of Control Mode Switches: Teleoperation of robotic arms (e.g., 6-D control) using traditional control interfaces, such as 3-axis joystick, require the user to switch between one of several control modes (mode switching) that are subsets of the full control space. Mode switches can become extremely challenging in the assistive domain, wherein the interfaces available to individuals with motor impairments are even more limited (e.g., 1-D Sip-N-Puff). Our intuition is that the intent inference will indirectly affect the number of mode switches, if earlier assistance results in fewer mode switches.

6.6.2. Human Inference

Figure 6.5 shows the performance of human intent inference and a comparison with autonomy intent inference on the demonstration trajectories. Percentage of time the predictions were (i) correct with confidence ($C(\boldsymbol{g}) > 30\%$), (ii) uncertain ($C(\boldsymbol{g}) < 30\%$) and (iii) incorrect with confidence ($C(\boldsymbol{g}) > 30\%$) are analyzed.

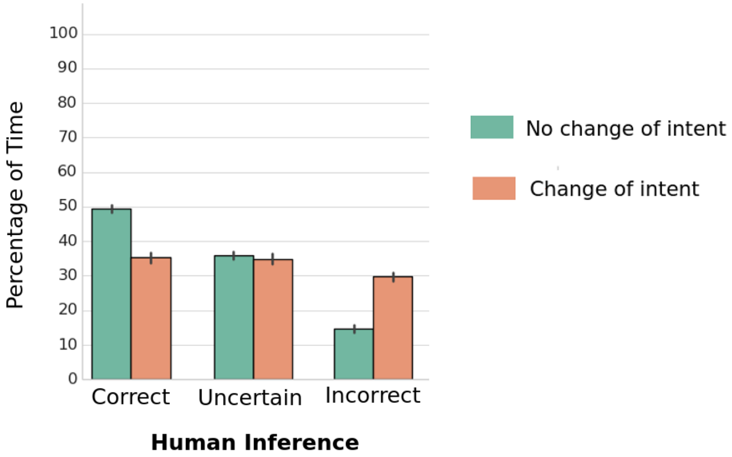


Figure 6.5. Human inference performance on the demonstration trajectories. Plot show mean and standard error.

The results show that inferring the intended goal of robotic arm motion is a challenging task, even for humans. The human subjects made fewer incorrect predictions as compared to correct predictions, but also were inclined to indicate more uncertainty. Also, the percentage of correct predictions were comparatively lower in the case of the change-of-intent trajectories in which the robot changed the intended goal during motion. Interestingly, the percentage of time inference was uncertain was unaffected by whether there was a change-of-intent.

6.6.3. Autonomy Inference

6.6.3.1. Novice teleoperation. We compared the predictive accuracy of our method against the Amnesic and Memory-based autonomy intent inference techniques. We first analyzed the inference performance with partially observed trajectories.

Figure 6.6 shows the average percentage time correct and confident predictions with partially observed trajectories. The Amnesic inference performed poorly and failed to get the prediction correct for a higher percentage of time as compared to the other methods. All other methods performed comparatively better, with RBII-2 performing best for the change-of-intent trajectories, the Memory-based approach performing best for trajectories without a change of intent, and RBII-2 consistently outperforming RBII-1. The Memory-based technique was not able to recover from the change of intent towards goals. The superior performance of RBII-2 on the change-of-intent trajectories is attributed to the incorporation of our more sophisticated probabilistic model of the goal-directed user actions in the RBII-2 likelihood model, which enabled the prediction to quickly recover to the intended goal.

The statistically significant trends were that all other techniques significantly outperformed the Amnesic inference technique in all scenarios ($p < 0.001$), except in the case of end users with the 3-axis joystick and change-of-intent trajectories. For end users with

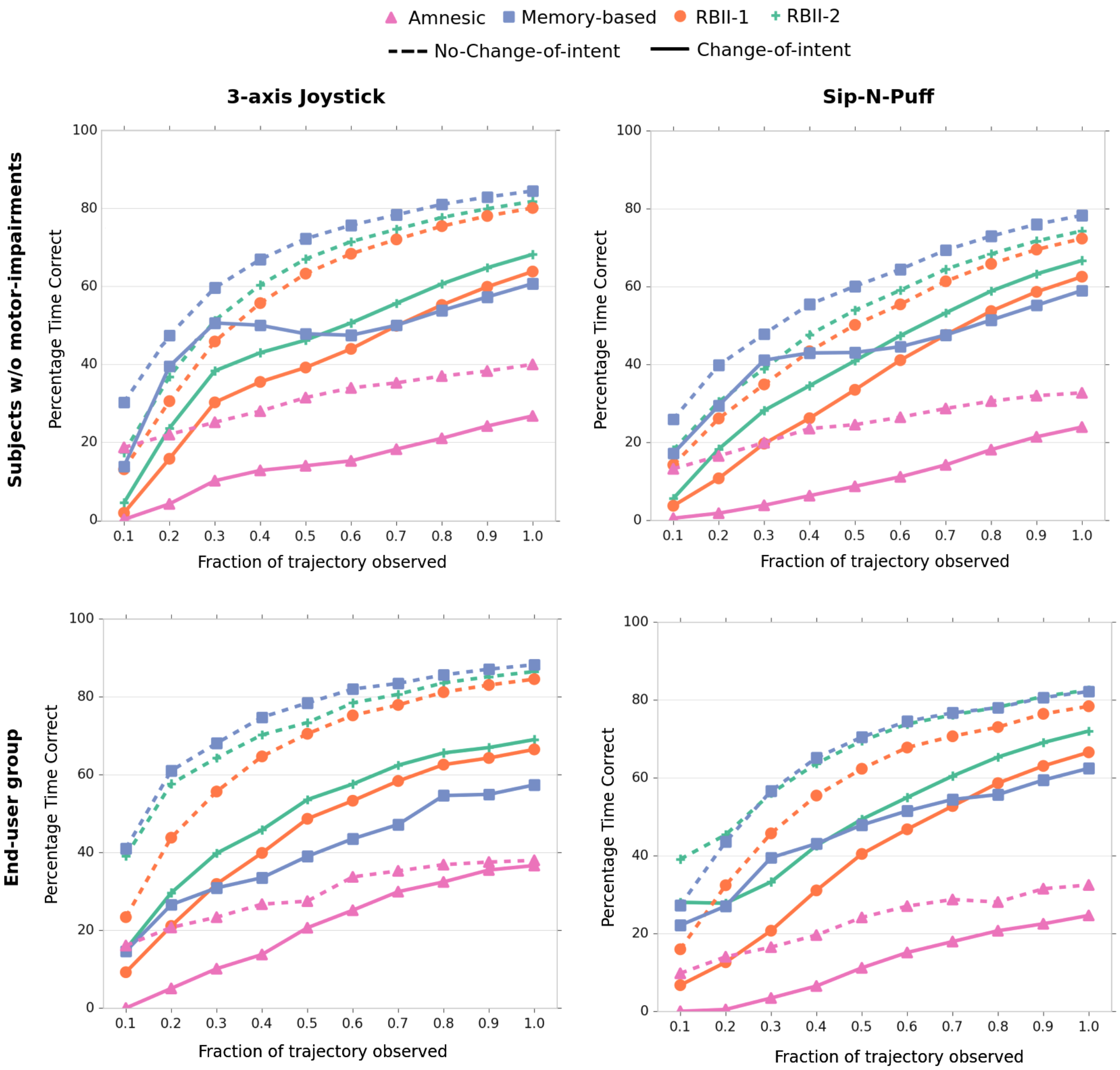


Figure 6.6. Performance comparison across four autonomy inference methods and two interfaces, during novice teleoperation for the subjects without motor-impairments and end-user subjects. The percentage of time the prediction is correct & confident is shown with partially observed trajectories.

the 3-axis joystick and change-of-intent trajectories, the Amnesic inference technique was outperformed significantly by the RBII-1 technique (p < 0.05) and the RBII-2 technique (p < 0.01). (The Memory-based technique performed better than the Amnesic technique but the results were not significant.)

An interesting observation is that the Memory-based method struggled to perform well in the case of end users operating the 3-axis interface, as compared to the subjects without motor-impairments using the same interface (higher percentage time correct predictions comparatively much earlier in the trajectory). These findings suggest that although the 3-axis joystick provides more degree of freedom control as compared to a Sip-N-Puff interface, it could also be more difficult to operate for people with motor-impairments (for example, because of limited finger and hand motion or hand tremor) and thus be more prone to noisy signals. Thus, there can be scenarios, when performing intent inference from a *lower-dimensional* interface is comparatively *less* challenging, as seen here in end-user data.

Figure 6.7 shows the performance of the autonomy intent inference methods during novice teleoperation (without autonomy assistance), measured in terms of average log-loss with partially observed trajectories. (Note that a perfect inference model would have a log-loss of 0.) The Memory-based technique performed comparatively better in the case of no-change-of-intent trajectories, in all scenarios. It was penalized for wrong predictions by the log-loss performance metric in two scenarios: when the user expressed change of intention during task execution and also when the tasks involved more than two goals. In the case of change-of-intent trajectories, the RBII-2 technique outperformed the other approaches. We saw one case where the RBII-2 technique was unable to outperform RBII-1: the end-user group operating the 3-axis joystick and without a change-of-intent. This further supports the finding that performing intent inference for the (arguably more informative) 3-axis joystick interface could be more challenging, for some user groups.

Figure 6.8 shows the mean log-loss across full trajectories. The statistically significant trends were that RBII-2 significantly outperformed the Memory-based approach for nearly all change-of-intent scenarios: subjects without motor-impairments operating both interfaces ($p < 0.01$) and end-user subjects operating the Sip-N-Puff interface ($p < 0.05$). (For

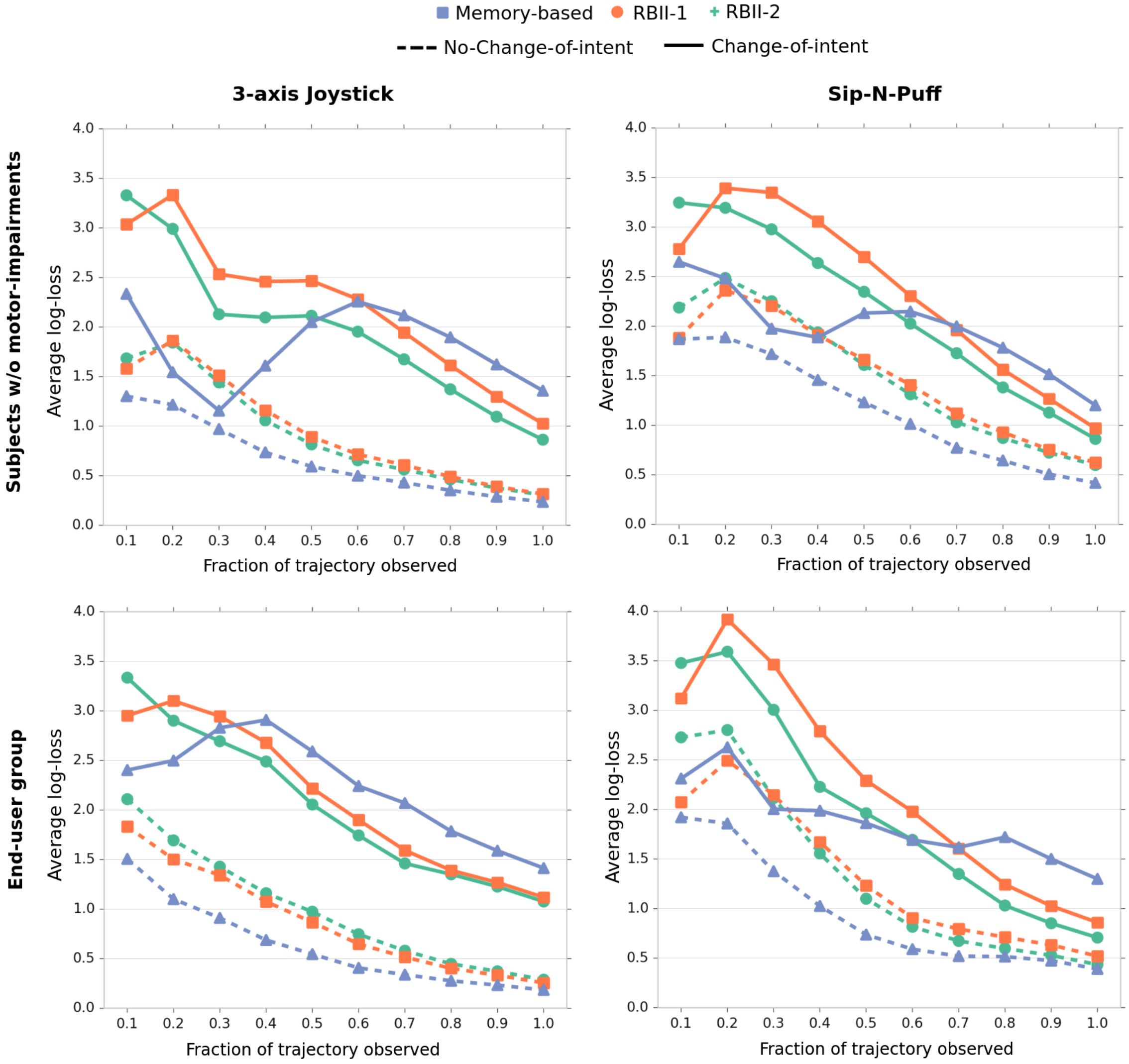


Figure 6.7. Performance comparison across the autonomy inference methods and two interfaces, during novice teleoperation for the subjects without motor-impairments and the end-user subjects. The average log-loss is shown with partially observed trajectories.

end users operating the joystick, the trend was observable but not significant.) For all approaches the log-loss was significantly ($p < 0.001$) higher with a change of intent (unsurprisingly) versus no change of intent.

6.6.3.2. Benefit of Adjustable Rationality Model of the Human Actions. We examined how the rationality index parameter affects performance. To test the benefit

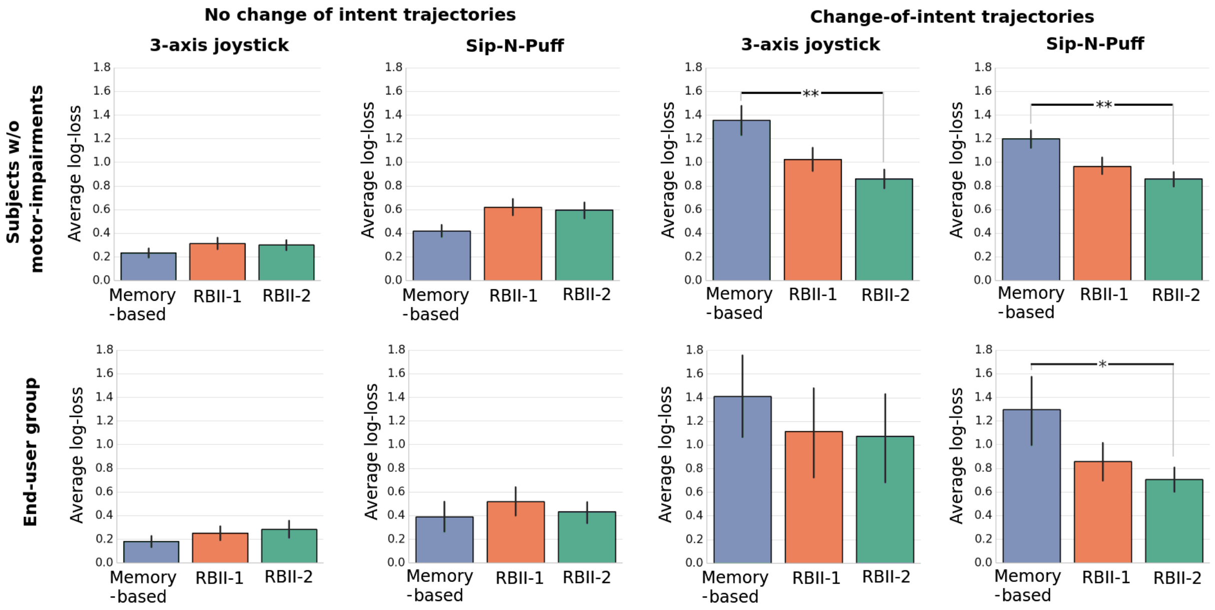


Figure 6.8. Average log-loss for the autonomy inference methods and two interfaces, during novice teleoperation. The average log-loss is shown, with mean and standard error.

of optimizing the rationality index for each subject, we computed the average log-loss using the optimized values of the rationality index β^* of that subject to compute the log-loss of the respective subject's trajectories versus when the optimized values of each other subject $\neg\beta^*$ was used to compute the average log-loss (i.e., β is not optimized for that subject). Figure 6.9 shows that the optimized values of the rationality index β^* resulted in significantly ($p < 0.001$) lower log-loss as compared to using the not-individualized rationality index values of other subjects $\neg\beta^*$. This indicates that optimizing the rationality index values for the human actions in the likelihood model results in better intent inference performance. Lastly, we also note that the optimized rationality index values β^* were comparatively higher for subjects using 3-axis joystick as compared to the subjects who used the Sip-N-Puff interface, indicating that the operators with the joystick were able to provide more rational actions.

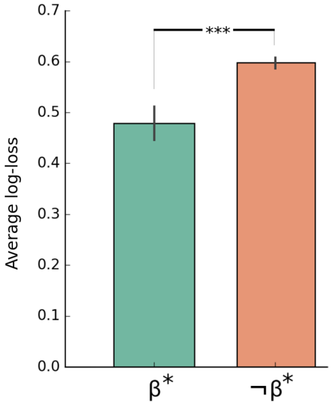


Figure 6.9. Benefit of optimized adjustable rationality for modeling user actions. Note that $\neg\beta^*$ refers to using the not-individualized rationality index values of other subjects.

6.6.4. Interplay of Intent Inference and Assistance in Shared Autonomy

We analyzed the implications of the underlying intent inference approach on shared autonomy performance (with assistance). For shared autonomy trials we focused on the RBII-1 and the RBII-2 techniques, which have been shown to perform well on novice teleoperation even with the change-of-intent trajectories.

Figure 6.10 shows the task completion times with and without assistance. The with-assistance trials involved shared-autonomy operation under the RBII-1 and RBII-2 inference techniques. The without-assistance trials imply direct teleoperation (Teleop). The task completion times were significantly higher ($p < 0.05$) with the Sip-N-Puff, in all cases as compared to the 3-axis joystick (not marked in figure to reduce visual clutter). Importantly, both variants of our approach RBII-1 and RBII-2, had significantly lower ($p < 0.001$) task completion times than Teleop, in all scenarios. Overall, better and faster inference during shared-autonomy operation with RBII-2 resulted in lower task completion times than RBII-1 in the case of both interfaces. For the subjects without motor-impairments, the task completion times in the case of RBII-2 were significantly lower ($p < 0.05$) than RBII-1 for both the 3-axis joystick and the Sip-N-Puff interface. Teleop had a significantly higher

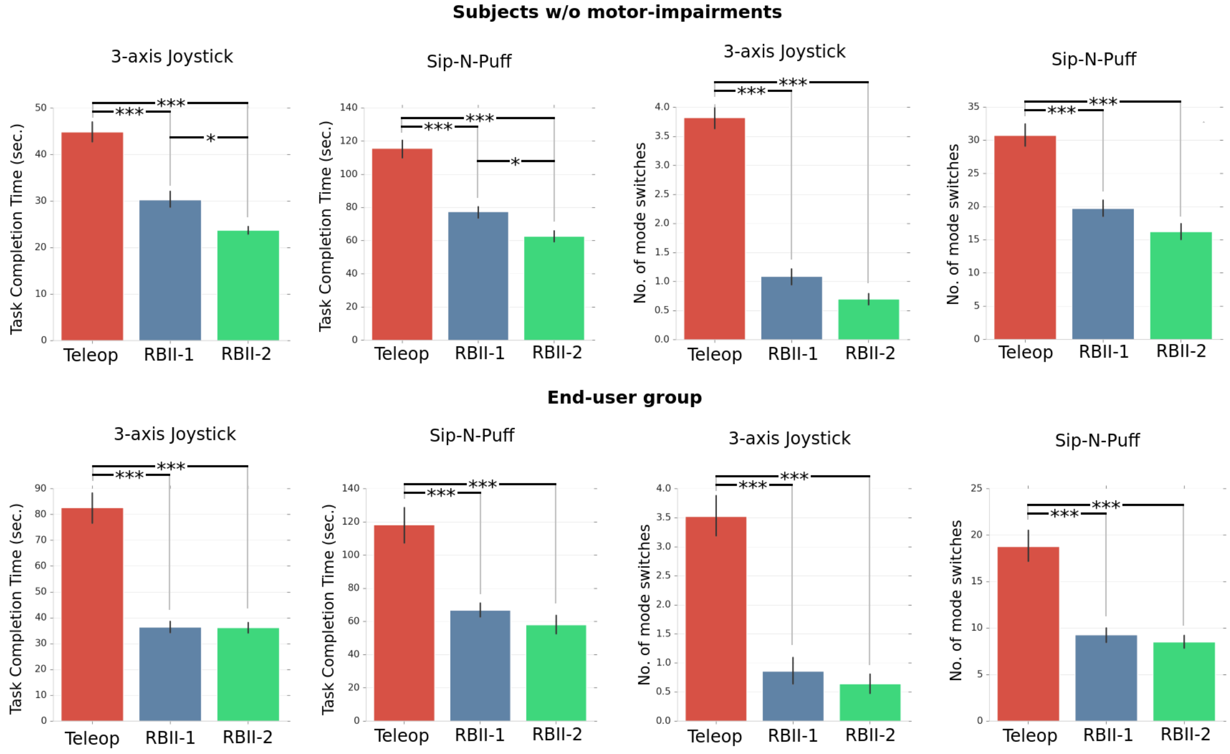


Figure 6.10. Comparison of task completion time and number of mode switches across two variants of our autonomy inference method under shared-control operation and teleoperation (without assistance), using two interfaces. Mean and standard error over all executions are shown.

number of mode switches ($p < 0.001$) than the RBII-1 and RBII-2 methods, in all scenarios. Overall, RBII-2 resulted in fewer average number of mode switches than RBII-1 indicating better assistance informed by the RBII-2 intent inference. Note that for the end user group, the RBII-2 method was not able to reduce the task completion time when compared to RBII-1 as much as in the case of the Sip-N-Puff interface. This further validates our consistent finding that performing intent inference in the case of the 3-axis joystick interface could be more challenging for some users, for example end users where the underlying impairment could result in producing more noisy control signals.

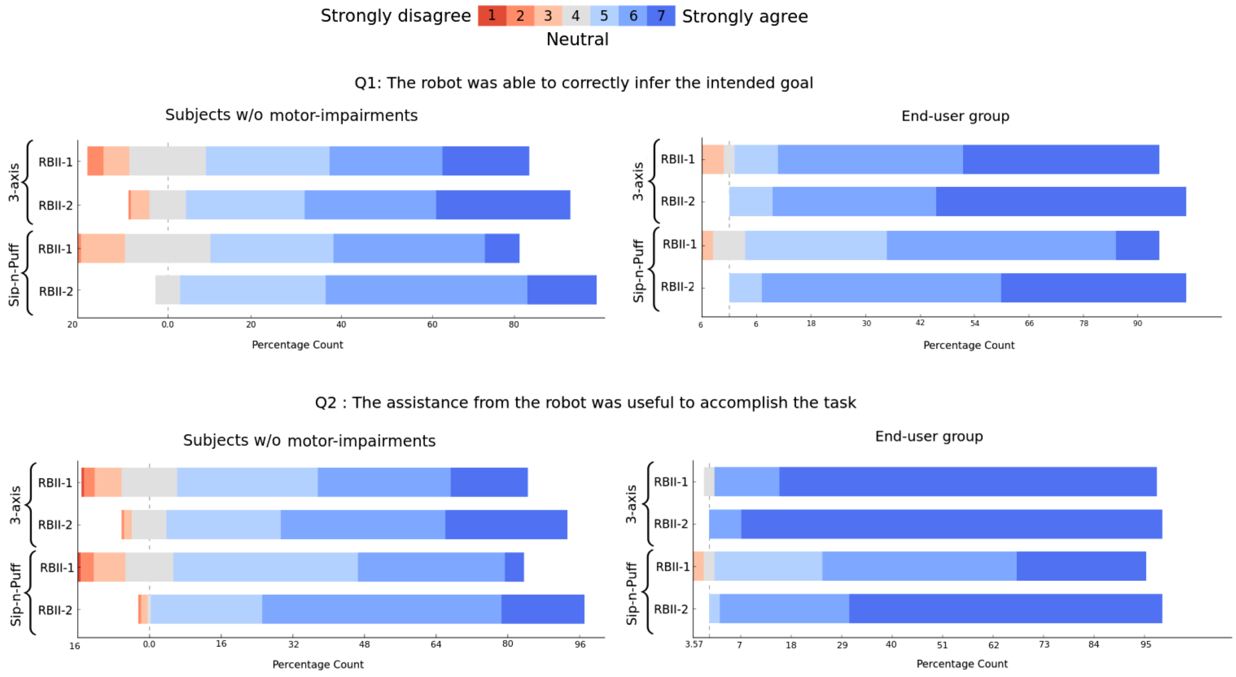


Figure 6.11. Subjective evaluation shows the percentage distribution of Likert scores (indicated in color codes).

6.6.5. Subjective Evaluation

Subjective evaluation was performed using an experiment survey after each trial to understand how subjects felt about the intent inference performance and the assistance under shared autonomy. This evaluation also provides insight as to whether users experienced performance differences during the robot operation under different intent inference schemes, both in terms of inference capability and usefulness of assistance to perform the tasks. The subjective evaluation results are summarized in Figure 6.11 and Table 6.1. Scores of 1 and 7 indicate that the responder strongly agrees or disagrees, respectively. There were noticeable difference in subjective ratings for both the intent inference capability and shared autonomy performance across the two intent inference techniques, with RBII-2 consistently rated better than RBII-1. The findings further indicate that the subjects were satisfied with the intent inference and shared autonomy performance, especially the end-user group.

Table 6.1. Subjective Evaluation - Autonomy Inference and Assistance

Question	Interface	Inference	Mean Score	
			Uninjured group	End-user group
The robot was able to correctly infer the intended goal.	3-axis	RBII-1	5.2 ± 1.3	6.1 ± 1.0
		RBII-2	5.7 ± 1.1	6.4 ± 0.6
	Sip-N-Puff	RBII-1	5.0 ± 1.1	5.5 ± 0.8
		RBII-2	5.7 ± 0.7	6.3 ± 0.6
The assistance from the robot was useful to accomplish the task.	3-axis	RBII-1	5.3 ± 1.2	6.7 ± 0.5
		RBII-2	5.7 ± 1.0	6.9 ± 0.2
	Sip-N-Puff	RBII-1	5.0 ± 1.1	5.9 ± 0.9
		RBII-2	5.8 ± 0.8	6.6 ± 0.5

6.7. Discussion

The human inference study results indicated that inferring the intended goal of a robot is a challenging task, even for humans. One important takeaway from our study is that humans tend both to make fewer incorrect predictions but also indicate more uncertainty. This has important implications for the assistive domain, since most often providing the wrong assistance is worse than providing no assistance. Thus, there is worth in knowing when the autonomy inference is uncertain and to what extent. The human subjects were also quickly able to switch their prediction in the case of change-of-intent, though with comparatively more incorrect predictions.

The autonomy intent inference methods were evaluated on novice teleoperation with and without shared autonomy assistance, and using two different control interfaces. The Amnesic inference failed to generate the correct prediction with sufficiently high confidence as compared to other techniques. This demonstrates the benefit of using a Bayesian approach for inference, under which taking proper account of the prior information to compute the posterior results in better performance. Memory-based inference that utilizes the information about the past trajectory history resulted in comparatively highly confident predictions much earlier in the execution. However, it also more often was wrong in its predictions with high probabilities and unable to recover when the user expressed a change-of-intent, particularly when the tasks involved more than two objects. Some limitations of the Memory-based method are recognized in an exploratory experiment [**63**]. RBII-1 has the same observation source for its likelihood model, but performed better than the Memory-based technique in the case of change-of-intent scenarios. Overall, RBII-2 outperformed other approaches in terms of faster correct predictions with higher probabilities. Interestingly, our approach was able to quickly respond in the case of change of intentions from the user, which thus also allowed it to quickly correct and recover the robot's belief from incorrect inferences. RBII-2 responded well to changing user goals, thus enabling the system to dynamically adjust its assistance to new predictions. In addition to contextual observations, probabilistic modeling and incorporating the human agent's behavior as goal-directed actions with intent-driven optimization of the adjustable rationality improved the overall performance.

We anticipated that it would be much more difficult to perform intent recognition in the case of the limited interface (e.g. Sip-N-Puff) which can only produce 1-D control signal, as compared to the 3-axis joystick. Surprisingly, however, the results demonstrated that with end users it frequently was the case that intent inference was more challenging under the axis operated joystick—perhaps because of signal noise that results from the underlying motor-impairments. Importantly, even with under limited interface operation our proposed technique was able to perform well to recognize human intent. Figure 6.12 shows the end-user subjects controlling the robot using the 3-axis joystick and the Sip-N-Puff interface, and successfully performing the study tasks with our proposed approach.

We have shown that with the probabilistic modeling of human actions as goal-directed observations the robotics autonomy can take advantage of control signals that the user implicitly provides during shared autonomy, even with lower dimensional interfaces controlling higher-DoF robot systems. The inclusion of adjustable rationality in our model accounts

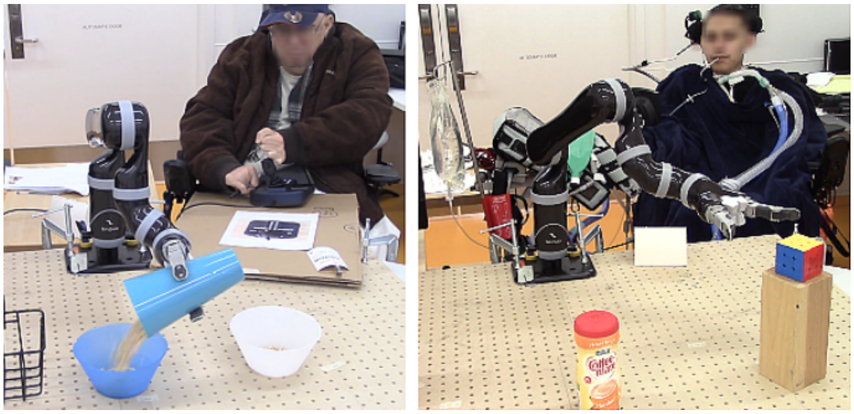


Figure 6.12. Example of study tasks performed by the end users, with the 3-axis joystick (left) and the Sip-N-Puff interface (right).

for suboptimal behavior in user actions, resulting in better inference performance. Notably, such suboptimal user behavior is inherent to the assistive domain. We also demonstrated the benefit of using an optimized adjustable rationality model for users, which produced better results. One weakness of our approach is that the data likelihood function under the Boltzmann-rationality model is not convex. The non-convexity makes finding optimal model parameters difficult and not efficient computationally. However, in practice finding optimal model parameters are only required once per subject.

Under direct teleoperation, the average task completion time as well as the average number of mode switches were significantly higher in the case of the Sip-N-Puff interface as compared to the 3-axis joystick. The inference predictions utilized to provide assistance in shared autonomy resulted in a significant reduction in the task completion time and the number of mode switches across both interfaces. Our results further verified that the underlying intent inference approach directly affects the assistance and the overall shared autonomy performance. The overall superior performance of RBII-2 as compared to RBII-1 under shared autonomy substantiates the benefit of the probabilistic modeling of human actions in addition to the contextual observations. Better and faster inference under shared autonomy with the RBII-2 technique was able to reduce the average task completion time and the average number of mode switches under shared autonomy. The subjective evaluation results further reinforce our findings. The subjects were satisfied with the intent inference and shared autonomy performance, favoring the RBII-2 technique—especially the end-user group who are the target population in assistive robotics domain.

We emphasize the importance of evaluating intent inference in shared autonomy operation. The implications of the control interface limitations on intent inference and the interplay of intent inference with shared autonomy are shown in the results. There is a particular benefit from probabilistic formulation of intent inference within the domain of assistive robotics—having an estimate of prediction uncertainty can be leveraged during shared-control operation, to determine whether and when to provide assistance, and by how much. For superior assistance under shared autonomy, the inference approach should provide correct predictions with high probabilities earlier in task executions. In future work, other observations such as spatial goal orientations and visual cues, distance metrics, and handling of continuous goal regions could further be explored. The effect of priors (initial goal probability distribution) and the goal transition probability are also interesting directions for future.

6.8. Summary

In this chapter, we mathematically defined the intent inference problem and presented a formulation that models the uncertainty over the user's goal in a recursive Bayesian intent inference algorithm to probabilistically reason about the intended goal of the user without explicit communication. The algorithm is able to fuse multiple observations to reason about the intended goal of the user. In user studies, we examined human inference on robot motion and furthermore evaluated and compared the performance of our algorithm to existing intent inference approaches—both with and without shared autonomy assistance, and using multiple control interfaces that are typically available to users in the assistive domain. Results of our study show that in addition to contextual observations, modeling and incorporating the human agent behavior as goal-directed actions with adjustable rationality improves intent recognition. We also demonstrate the benefit of optimizing adjustable rationality in the model of human actions for individual users. We furthermore demonstrated that the underlying intent inference approach directly affects assistance in shared autonomy, as do control interface limitations.

CHAPTER 7

Body-Machine Interface for Human-Robot Systems

An important line of research for human-robot systems in assistive robotics is the development of novel and dedicated user interfaces. The design of such interfaces take into consideration the impairments of the user and the control channels available to them. In this chapter, we present application of a novel body-machine interface (BMI) for human-robot systems. Body-machine interface (BMI), a non-conventional control interface can make use of an individual's residual motion capabilities for operation of human-robot systems in the assistive domain (Figure 7.1).

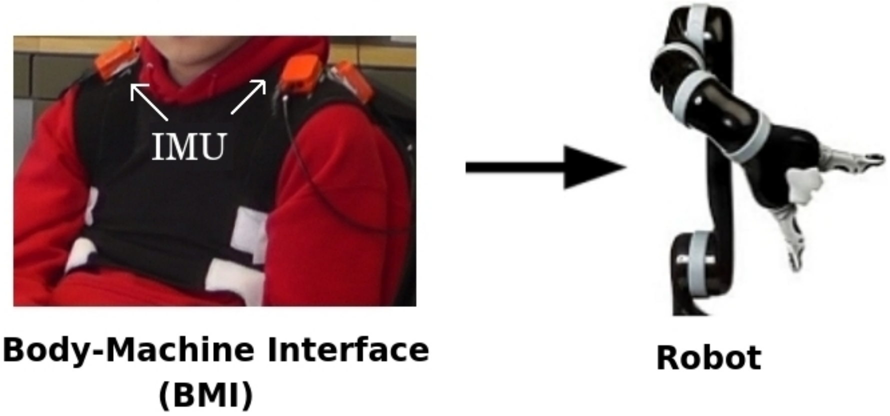


Figure 7.1. Our body-machine interface (BMI) make use of an individual's residual motion capabilities for the operation of robotic systems.

In particular, we develop a shared autonomy framework in which an individual's residual motion capabilities, captured through BMI, are used to generate control signals for assistive manipulation with a robotic arm. These low-dimensional controls are utilized in our shared-control framework, that shares control between the human user and robotics autonomy. We

The work in this chapter was originally published as [**58**].

present results from a pilot study and demonstrate 100% success rate on task performance. The novel application demonstrates the effectiveness of the proposed system for individuals with severe-motor impairments to control human-robot systems. An important advantage of BMI is that it encourages the continued use of muscular activity, as the participant's residual body movements are captured to provide control signals for an assistive device, and thus can facilitate rehabilitation process.

7.1. System Description

In this section, we present the system description for the body-machine interface and the proposed shared-control framework for assistive manipulation.

7.1.1. Body-Machine Interface and Control Signals

In a body-machine interface, body motions generate control signals to operate external devices. The BMI provides an effective pathway for control because even in people with severe impairments, some residual movements remain available. These movements are captured by multiple sensors, whose combined outputs define a signal space for controlling the external device. In the proposed BMI system, a high dimensional control signal captured from the participant's residual movements is mapped to a lower dimensional control vector. Importantly, these surviving degrees of freedom captured from the body are higher dimensional than the required control signal. This kinematic redundancy provides the BMI user with a unique opportunity to identify and coordinate a convenient subset of movements to achieve task objectives with a flexible and adaptable motor behavior [**187**]. This enables the user to effectively issue control signals for the robotic arm via a reorganization of their own high-dimensional upper body motions.

In the current BMI setup the user wears a vest that is equipped with four MTx (Xsens Technologies B.V., Netherlands) motion trackers in order to capture shoulder movements. An IMU is placed on the front and back of each shoulder as can be seen in Figure 7.1. The orientation of each sensor is computed by a sensor fusion algorithm through the combination of the output of 3-DoF embedded accelerometers, gyroscopes and magnetometers. For the purpose of this study we only use roll and pitch as input signals for the interface because the yaw signals, derived from the magnetometer, have a tendency to drift in the presence of electric motors and large metallic objects. The IMU signals are captured at the rate of 50Hz. With four IMUs the body space is defined by an eight dimensional vector of coordinates captured from the four sensors.

The available residual movements depend on the injury and therefore the interface is user-specific. To this end, we use a calibration phase to map the user's movements to control signals. During the calibration phase, the participants are asked to engage in free-style motions of the upper body for twenty seconds. The purpose of this activity is to characterize the space of IMU signals that each subject could comfortably span. The mapping matrix A is obtained by Principal Component Analysis (PCA). A linear transformation, $C = A \cdot h$, is defined to map the body movements onto the 2-D vector C, that controls the motion of the robot. PCA lends itself quite naturally to this task, since the principal eigenvectors represent the dimensions with largest variability in the data—and thus also the dimensions with the largest capacity for movement from the user. The first two principal eigenvectors of the calibration data are extracted to form a 2-D control space. For further details of the interface and calibration, see [**188**].

7.1.2. Control Framework for Assistive Manipulation

We are interested in a system that keeps the user in control and at the same time provides assistance in manipulation tasks. Using low-dimensional control signals from the BMI, our aim is to enable the simultaneous operation of all degrees of freedom of a high-DoF robotic arm. To address this challenge, we introduce robot autonomy to reduce the user's control burden. By contrast, under direct teleoperation the user would be responsible for individually controlling each joint of the robotic arm at each time step, or equivalently the position and orientation of the end-effector. (For our experimental platform, both are 6-D control problems.)

Our intended system will create a sequence of functionally relevant piecewise segments based on the semantics of actions performed during a typical execution of a given manipulation task—such as reaching, grasping, and pouring. As a first step, in this work the autonomous system plans piecewise trajectory segments for predefined manipulation task using autonomously perceived goals (Section 7.2.2). Next, the motor-impaired user influences the execution of these trajectories through (i) control of the speed (U) of the manipulator along each segment of the task, and (ii) dynamically switching (S) between trajectory segments in order to complete the desired task. The 1-D continuous valued signal U, controls the speed of the manipulator along the current trajectory. The 1-D binary signal S triggers a switch between motion segments. The threshold to generate the binary signal is set as twice the standard deviation of the second principal component, and is obtained during the calibration stage of the BMI interface. This approach allows for operation of a high-DoF arm with the limited control signals $\langle$U, S$\rangle$ available from the BMI interface. Users thus are able to inject their preference and situational awareness into the otherwise autonomous task execution.

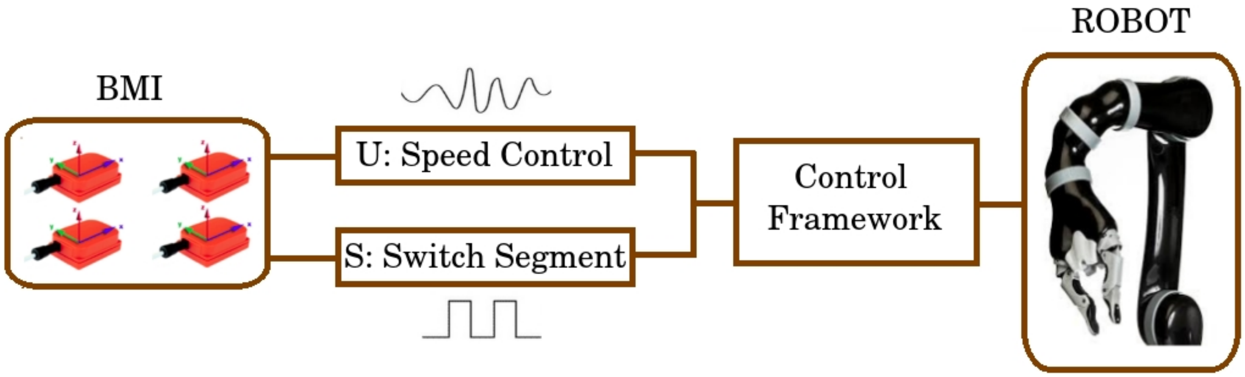


Figure 7.2. Schematic of the system pipeline.

The first step in the technical implementation of this framework is to autonomously generate trajectories from the robot's current configuration Q to the desired goal configuration. Any suitable motion planner can be used for this purpose. We used task-constrained motion planning [**174**] and the Constrained Bi-directional Rapidly exploring Random Tree (CBiRRT) [**189**] in our implementation. To achieve speed control along the trajectory, we calculate joint velocities

$$\boldsymbol{\nu} = \frac{\delta}{\tau} \cdot U$$

based on (i) the user's input signal, U $\in [0, 1]$, and (ii) the autonomy command, computed as the Euclidean distance δ between the current configuration Q of the robot and the next configuration waypoint along the path, divided by timestep τ. Here the command velocity $\nu \in \mathbb{R}$ is the set of joint velocities sent for execution on the robot manipulator. In order to progress along the trajectory, we update which waypoint is the current subgoal based on distance to current configuration Q, and continue to do this until we have achieved the final goal configuration.

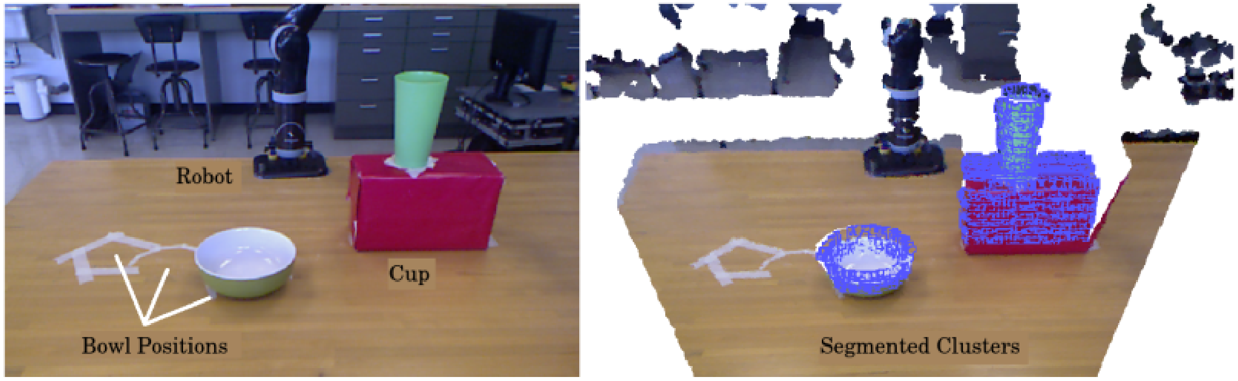


Figure 7.3. *Left*: Experimental set-up. *Right*: Segmented point cloud clusters (shown in blue).

7.2. System Implementation and Evaluation

To evaluate our proposed system, a user study was performed by subjects with and without high-level Spinal Cord Injury (SCI).

7.2.1. Task

The manipulation task of the user study consisted of using the robotic arm to pour the contents of a cup into a bowl. The task was a sequence of the following four motion segments: (i) reach for the cup, (ii) grasp it, (iii) carry it to the bowl and (iv) pour the contents of the cup into the bowl.

To assess more extensively the effect of the user's input, variability was introduced into the task by modulating the position of the bowl (three positions). Task success thus depended on the user appropriately triggering the transition between segments (iii) and (iv). If they did not switch in time, the assistive manipulator would continue along its trajectory, overshooting the bowl. The pouring task was explained to each participant, along with the effect of the control signals $\langle \mathrm{U}, \mathrm{S} \rangle$.

7.2.2. Autonomy

For the first and third segments we used the CBiRRT planner to generate a set of waypoints that define a path from the robot's current configuration to each subgoal position, where the final goal was defined to be past the three bowl positions (so that the final trajectory segment passed over all possible bowl locations). For the second and fourth segments, no planning was needed: segment (ii) involved simply closing the gripper, while segment (iv) involved rotating the wrist.

To compute the position of the cup, we implemented a tabletop segmentation and Euclidean clustering approach using the point cloud data obtained from the Kinect RGB-D sensor. This results in segmented clusters of the objects present in the scene (Figure 7.3).

7.2.3. User Input

The user provided 2-D input to the system using the BMI, as described in Section 7.1.2. The first signal allowed the user to control the speed of the arm along the various trajectories, and the second signal allowed the user to transition between segments (iii) and (iv). The transitions between other piecewise trajectories was performed autonomously, to simplify the task design, since these transitions were not modulated within the study design.

7.2.4. Execution

For each trial one of three bowl positions was randomly selected. The user began the execution by controlling the speed U during trajectory segment (i). As the robotic arm reached the cup, the autonomous system transitioned to segment (ii) and the user controlled the speed U at which the gripper was closed in order to grasp the cup. During segment (iii), the user again controlled the speed of the robotic arm along the path, until signal S

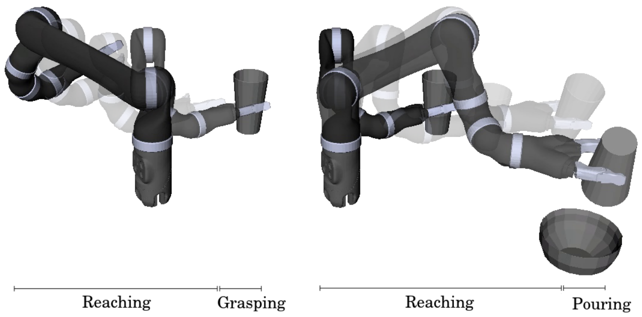


Figure 7.4. Illustration of the piecewise segments associated with the experimental task.

was issued by the user to switch to segment (iv). During segment (iv), the user speed U mapped to control the wrist rotation, and thereby poured the contents of the cup. Figure 7.4 represents an illustration of the experimental procedure.

7.2.5. Subjects

One SCI survivor (31 year old male, 13 years post-injury at the C5 level) and five uninjured control individuals (mean age: 28 ±3) participated in the user study. All participants gave their informed, signed consent to participate in this experiment, which was approved by Northwestern University's Institutional Review Board. After the calibration of the BMI each participant performed 24 reaching and pouring trials (8 trials per bowl position) in a randomized sequence. Note that continuous visual feedback of the control signals together with the switching threshold was provided on a computer screen that was positioned in front of the participants. Figure 7.5 shows the experimental setup and a user performing the task using the proposed system.

7.3. Experimental Results

All subjects were able to perform the task by reorganizing their shoulder movements. They learned to perform the task effectively after the very first trial and the performance

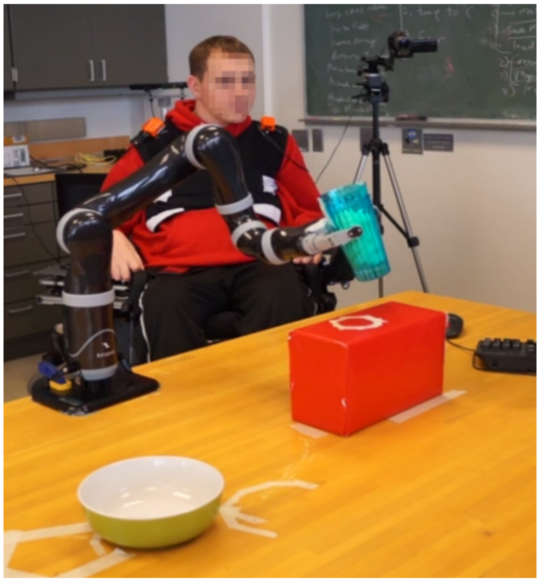


Figure 7.5. An SCI user controlling the robot with the BMI during the experimental task.

level stayed the same for the rest of the experiment. We furthermore observed similar performance between the SCI and non-injured subjects, across all measures. The end-effector position and task completion time were recorded for each of the trails.

Figure 7.6 shows the user control signals ⟨U, S⟩ and the end-effector position for a representative task trial. Note the use of signal U for the reaching, grasping and pouring segments, and the use of signal S to switch (around second 28) to pouring after reaching the bowl position.

Figure 7.7 shows the position of the robot end-effector at the end of each trial for the SCI participant and a representative control subject. It can be seen that the subjects were able to successfully switch the trajectory segment in order to perform the pouring task for each of the three bowl positions. More importantly, the performance of the SCI participant was comparable to other uninjured control individuals.

Figure 7.8 (left) represents the average time to completion for all participants. The time taken by the SCI participant for task completion was comparable that of the able-bodied

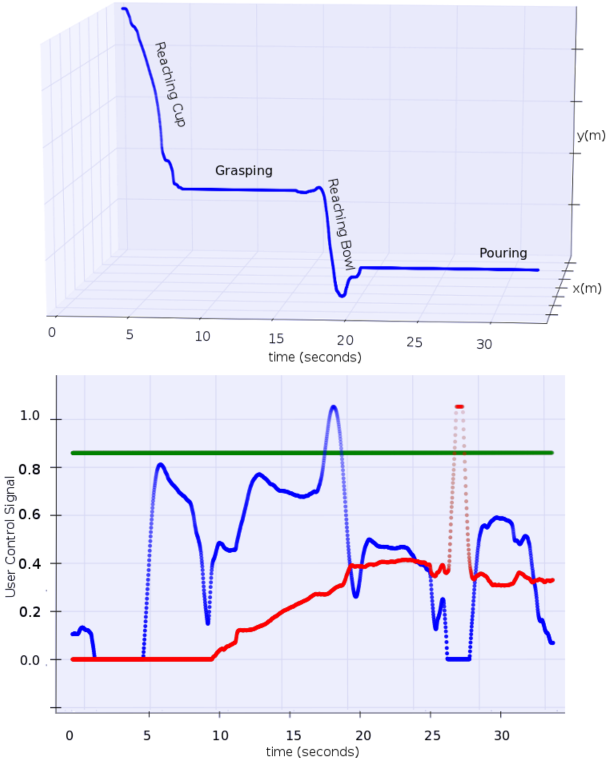


Figure 7.6. *Top*: Robot's end-effector in (x,y) space. *Bottom*: User's control signals U (blue) and S (red), and the threshold used to switch between segments (green).

individuals (C1-C5).Furthermore, to quantify movement smoothness we calculated jerk as

$$J = \left| \sum_{k=1}^{n} \dddot{x}(k) \right|$$

where $x(k)$ corresponds to discrete samples of the Euclidean norm of the robot end-effector position. Jerk is the third derivative of position, and a standard measure to quantify movement smoothness [**190**]. A second-order Butterworth filter with a cutoff frequency of 5Hz was used to smooth and attain the end-effector trajectory for each trial. Figure 7.8 (right) shows the average jerk index for all participants. Note that the SCI participant was as smooth as the uninjured subjects in controlling the arm movements. The above results demonstrate the effectiveness of the proposed framework and the BMI interface, as the

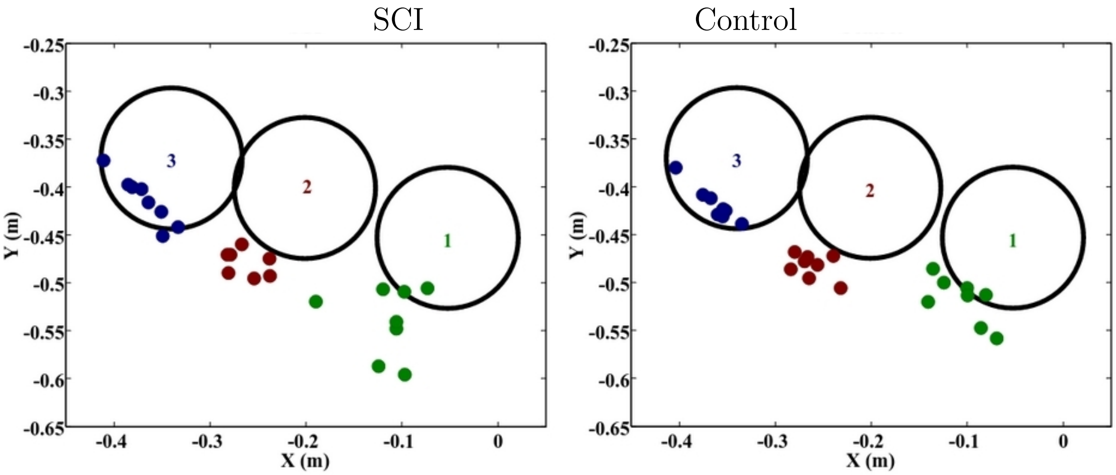


Figure 7.7. Position of the robot end-effector at the end of each trial for the SCI participant (left) and a representative control subject (right). Each color corresponds to one of the three positions of the bowl. Note that a successful pouring motion aligns the top of the cup over the bowl, which results in the robot end-effector position being offset (since the cup has non-negligible length).

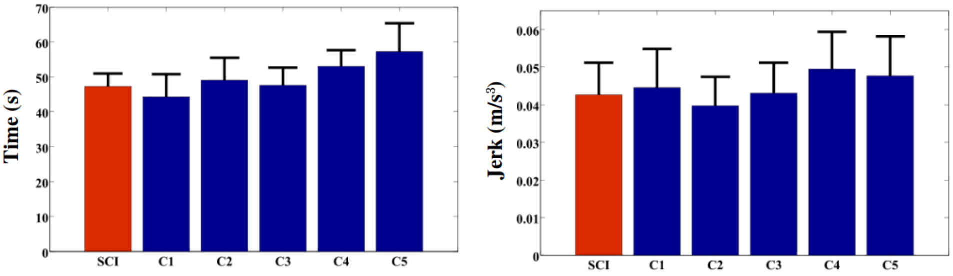


Figure 7.8. *Left*: Average time to completion for all participants. *Right*: Average movement smoothness for all participants. For both plots, error bars represent standard deviation.

performance of the SCI participant was comparable to the control individuals for the manipulation task. Future work can generalize the system to achieve assistive control on a variety of manipulation tasks and can explore mapping the BMI signal to alternate subsets of the control space, as well as the generation of higher dimensional BMI signals.

7.4. Summary

We have introduced a novel system for the control of assistive robotic manipulators, that makes use of both robotics autonomy and a body-machine interface. The aim of this work was a first evaluation of the BMI with the proposed control framework. The results of the user study indicate that individuals with severe motor impairments can effectively operate assistive robotic manipulators using the proposed system. Furthermore, the BMI engages the users in physical activity while they operate the manipulator, which may have potential rehabilitation benefits.

7.5. Declaration

The work presented in this chapter was previously published as [**58**]. S.Jain conceived of the presented control allocation and was lead in the study design and its execution. A. Farshchiansadegh contributed expertise related to the BMI. The software development, hardware integration, and study execution was collaboratively performed by S. Jain, A. Farshchiansadegh, and A. Broad, and F. Abdollahi assisted with the study sessions.

CHAPTER 8

Final Thoughts

This dissertation describes a body of work seeking to understand and improve upon the incorporation of robotics autonomy into human-robot systems for assistive robotics. The mathematical models presented in this dissertation address how robotics autonomy can: perceive high-level user goals; probabilistically recognize the intentions of the human user; model user actions for shared autonomy in assistive teleoperation; share control with the user; and finally improve human-robot team performance and increase user personalization with an intent-driven optimization of user actions in shared autonomy. We also presented a categorization of shared autonomy as series, interleaved, and parallel—based on the control sharing between the human and the robotics autonomy (Chapter 3). Such categorization can better inform the shared autonomy design decisions depending on the needs of control sharing. Pertaining to the challenges in direct control of assistive robots, another important research direction is to investigate and design novel human-machine interfaces. We presented a detailed review of such approaches (Section 2.3) and investigated novel application of the body-machine interface (BMI) for assistive manipulation with a robotic arm (Chapter 7). We next summarize our contributions with key findings of the research and studies presented in the dissertation.

Robotic Perception for Human-Robot Systems:

We developed perception algorithms for high-level perception of user goals using geometric modeling, computer vision, and machine learning (Chapter 5). We primarily worked with the detection of navigation goals involving wheelchair docking at table and desk structures, as well as manipulation goals involving detection of robotic grasps on novel household objects.

For wheelchair docking, our approach is the first in the domain to fully automate the perception of safe docking locations using point clouds. Our geometric approach can detect suitable docking pose at a variety of table and desk structures, and more importantly also assess them for safety according to the ADA specifications. The approach was evaluated on table, desk, and workbench docking structures in varying configurations, and proved effective in identifying safe docking locations with pose information. The algorithm provides a docking pose with accurate alignment information, which can readily be used by motion planners to plan and execute trajectories for assistance with docking maneuvers.

For grasp detection on novel objects, we investigated methods for estimating local and global surface geometries in point clouds and developed a framework for grasp detection based on such geometric characteristics. Our approach approximates object geometries and generates multiple semantic grasps on objects to address multiple ways in which human users can grasp a single object. A first set of experiments evaluated the perception model's grasp detection ability for grasp similarity between the autonomy generated grasps and those generated by a human when teleoperating the robot. Multiple grasps were generated by our approach on each object presented in the test set, according to the semantic groups (top, side, pinch). The results highlighted the resemblance between the detected grasp poses (both orientation and position) and what the human subjects generated when teleoperating the robot. This similarity may help the user to predict the robot's movement during shared autonomy operation, and thus can help improve human-robot collaboration in tasks.

Human Intent Recognition and Assistance in Human-Robot Systems:

For inferring human intentions in shared autonomy (Chapter 6), we first drew a connection between human actions in experimental psychology and robot inference of human actions, enabling us to probabilistically model user behavior as goal-directed actions. The idea being that such teleological interpretation of actions in terms of user goals can inform us as to why the action has been performed, thus providing a special type of explanation for the user's action. We performed probabilistic modeling of implicit observations, and in addition to a distance-based observation, we modeled the user's interaction with the control interface as goal-directed actions with adjustable rationality. For user personalization in shared autonomy, we furthermore introduced an intent-driven optimization that adapts the rationality index value to each individual user, and thus can account for a number of factors that might induce sub-optimality in actions, including the limitations of control interfaces. We presented a formalism for intent inference that models the uncertainty over the user's goal within a Bayesian filtering framework and enables the seamless fusion of observations. Our approach computes and maintains a probabilistic belief over the user's goals and expresses uncertainty in the robot prediction of the intended goal, which has important implications in assistive domain.

We performed a pilot study with the aim to ground the complexity of the intent inference problem through analysis of human inference ability, by having human observers interpret the motion of a robotic arm to infer its intended goal. Study results suggest that inferring the robot's intended goal is a challenging task, even for humans. One important takeaway from our study is that humans tend to make less incorrect predictions, but also indicate more uncertainty. Human subjects were also quickly able to switch their prediction in the case of change of intentions, though with comparatively more incorrect predictions. These findings have important implications for the assistive domain, since most often providing the wrong assistance is worse than providing no assistance.

We conducted a subject study to evaluate our human intent recognition framework and compare performance with existing methods. Importantly, we also examined control interface implications and performed the study using multiple control interfaces that are typically available to users in the assistive domain. The study furthermore provides insights into the interplay of intent inference and assistance in shared autonomy. The findings reveal that the inference of human intent was less accurate under more limited control interfaces, the mechanism of intent inference indeed impacts control sharing and assistance in shared autonomy, and probabilistic modeling of human actions and our intent-driven optimization resulted in better intent recognition with faster assistance during shared-control operation. Interestingly, our approach was able to quickly respond in the case of change of intentions from the user, which thus also allowed to quickly correct and recover the robot's belief from incorrect inferences. Surprisingly, the results indicate that for end-users, intent inference was often more challenging under the axis operated joystick—perhaps because of signal noise that resulted from the underlying motor-impairments. Even with the limited interface like Sip-N-Puff, our proposed technique performed well in recognizing human intent. The subjective evaluations further reinforce our findings, and especially the end users who are the target population in the assistive domain were satisfied with the intent inference and shared autonomy performance.

Body-Machine Interface for Human-Robot Systems:

We investigated novel application of body-machine interface (BMI) and presented a shared autonomy framework that utilizes the non-conventional interface for assistive manipulation with a robotic arm (Chapter 7). An important advantage of our approach using the BMI is that it encourages the continued use of muscular activity, as the participant's residual body movements are captured to provide control signals for controlling an assistive device, and thus can facilitate rehabilitation process. The novel application demonstrates the effectiveness of the proposed system for individuals with severe-motor impairments to control high-DoF human-robot systems.

In conclusion, this dissertation has contributed mathematical models and algorithms for perception, inference, and assistance in human-robot systems, focusing on the assistive domain. We validated all the contributed algorithms and techniques in this dissertation on real hardware, using a wheelchair robot or a robotic arm platform. Exciting future work directions and open questions for human-robot systems based on the ideas from this dissertation include, perception and manipulation of deformable objects (e.g. assistive feeding), intent inference over continuous goal regions as opposed to discrete goal configurations, bi-directional intent recognition, probabilistic control sharing in shared autonomy, and self-evaluation of shared autonomy during online collaboration with human users.

References

[1] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. R. Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, *et al.*, "ros_control: A generic and simple control framework for ros," *Journal of Open Source Software*, vol. 2, no. 20, p. 456, 2017.

[2] M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends® in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2008.

[3] D. M. Taylor, "Americans with disabilities: 2014 household economic studies current population reports," 2018.

[4] I. Laffont, N. Biard, G. Chalubert, L. Delahoche, B. Marhic, F. C. Boyer, and C. Leroux, "Evaluation of a graphic interface to control a robotic grasping arm: a multicenter study," *Archives of physical medicine and rehabilitation*, vol. 90, no. 10, pp. 1740–1748, 2009.

[5] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, "Assistive teleoperation of robot arms via automatic time-optimal mode switching," in *Proceeding of ACM/IEEE International Conference on Human Robot Interation (HRI)*, 2016.

[6] J. M. Beer, A. D. Fisk, and W. A. Rogers, "Toward a framework for levels of robot autonomy in human-robot interaction," *Journal of human-robot interaction*, vol. 3, no. 2, p. 74, 2014.

[7] E. A. Biddiss and T. T. Chau, "Upper limb prosthesis use and abandonment: a survey of the last 25 years," *Prosthetics and orthotics international*, vol. 31, no. 3, pp. 236–257, 2007.

[8] C.-S. Chung, H. Wang, and R. A. Cooper, "Functional assessment and performance evaluation for assistive robotic manipulators: Literature review," *The journal of spinal cord medicine*, vol. 36, no. 4, pp. 273–289, 2013.

[9] M. A. Goodrich and D. R. Olsen, "Seven principles of efficient human robot interaction," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2003.

[10] B. Phillips and H. Zhao, "Predictors of assistive technology abandonment," *Assistive technology*, vol. 5, no. 1, pp. 36–45, 1993.

[11] F. M. Maynard Jr, M. B. Bracken, G. Creasey, J. F. Ditunno Jr, W. H. Donovan, T. B. Ducker, S. L. Garber, R. J. Marino, S. L. Stover, C. H. Tator, *et al.*, "International standards for neurological and functional classification of spinal cord injury," *Spinal cord*, vol. 35, no. 5, p. 266, 1997.

[12] E. Smith and M. Delargy, "Locked-in syndrome," *BMJ*, vol. 330, no. 7488, pp. 406–409, 2005.

[13] D. Lulé, C. Zickler, S. Häcker, M.-A. Bruno, A. Demertzi, F. Pellas, S. Laureys, and A. Kübler, "Life can be worth living in locked-in syndrome," *Progress in brain research*, vol. 177, pp. 339–351, 2009.

[14] K. Boström and G. Ahlström, "Living with a chronic deteriorating disease: the trajectory with muscular dystrophy over ten years," *Disability and Rehabilitation*, vol. 26, no. 23, pp. 1388–1398, 2004.

[15] K. P. Murphy, G. E. Molnar, and K. Lankasky, "Medical and functional status of adults with cerebral palsy," *Developmental Medicine & Child Neurology*, vol. 37, no. 12, pp. 1075–1084, 1995.

[16] R. C. Simpson, E. F. LoPresti, and R. A. Cooper, "How many people would benefit from a smart wheelchair?" *Journal of Rehabilitation Research & Development*, vol. 45, no. 1, 2008.

[17] R. C. Simpson, "Smart wheelchairs: A literature review," *Journal of rehabilitation research and development*, vol. 42, no. 4, p. 423, 2005.

[18] T. Carlson and Y. Demiris, "Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 876–888, 2012.

[19] A. Erdogan and B. D. Argall, "The effect of robotic wheelchair control paradigm and interface on user performance, effort and preference: an experimental assessment," *Robotics and Autonomous Systems*, vol. 94, pp. 282–297, 2017.

[20] J. Leaman and H. M. La, "A comprehensive review of smart wheelchairs: past, present, and future," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 486–499, 2017.

[21] S. L. Garber and T. L. Gregorio, "Upper extremity assistive devices: Assessment of use by spinal cord–injured patients with quadriplegia," *American Journal of Occupational Therapy*, vol. 44, no. 2, pp. 126–131, 1990.

[22] P. Holliday, A. Mihailidis, R. Rolfson, and G. Fernie, "Understanding and measuring powered wheelchair mobility and manoeuvrability. part i. reach in confined spaces," *Disability and rehabilitation*, vol. 27, no. 16, pp. 939–949, 2005.

[23] V. Maheu, P. S. Archambault, J. Frappier, and F. Routhier, "Evaluation of the jaco robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities," in *Proceeding of IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2011.

[24] B. Graf, M. Hans, and R. D. Schraft, "Care-o-bot iidevelopment of a next generation robotic home assistant," *Autonomous robots*, vol. 16, no. 2, pp. 193–205, 2004.

[25] C. Martens, N. Ruchel, O. Lang, O. Ivlev, and A. Graser, "A friend for assisting handicapped people," *IEEE Robotics & Automation Magazine*, vol. 8, no. 1, pp. 57–65, 2001.

[26] A. Jain and C. C. Kemp, "El-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots*, vol. 28, no. 1, pp. 45–64, 2010.

[27] A. T. Asbeck, S. M. De Rossi, I. Galiana, Y. Ding, and C. J. Walsh, "Stronger, smarter, softer: next-generation wearable robots," *IEEE Robotics & Automation Magazine*, vol. 21, no. 4, pp. 22–33, 2014.

[28] F. Cordella, A. L. Ciancio, R. Sacchetti, A. Davalli, A. G. Cutti, E. Guglielmelli, and L. Zollo, "Literature review on needs of upper limb prosthesis users," *Frontiers in neuroscience*, vol. 10, p. 209, 2016.

[29] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," *Robotics and Autonomous Systems*, vol. 64, pp. 120–136, 2015.

[30] A. T. Asbeck, S. M. De Rossi, K. G. Holt, and C. J. Walsh, "A biologically inspired soft exosuit for walking assistance," *The International Journal of Robotics Research*, vol. 34, no. 6, pp. 744–762, 2015.

[31] H. In, B. B. Kang, M. Sin, and K.-J. Cho, "Exo-glove: A wearable robot for the hand with a soft tendon routing system," *IEEE Robotics & Automation Magazine*, vol. 22, no. 1, pp. 97–105, 2015.

[32] L. Fehr, W. Langbein, and S. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey." *Development*, vol. 37, no. 3, pp. 353–360, 2000.

[33] C. G. Pinheiro, E. L. Naves, P. Pino, E. Losson, A. O. Andrade, and G. Bourhis, "Alternative communication systems for people with severe motor disabilities: a survey," *Biomedical engineering online*, vol. 10, no. 1, p. 31, 2011.

[34] M. Quigley, J. Faust, T. Foote, and J. Leibs, "ROS: an open-source robot operating system," in *In ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[35] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[36] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction (HRI)*, 2012, pp. 1–8.

[37] D. Kent, C. Saldanha, and S. Chernova, "A comparison of remote robot teleoperation interfaces for general object manipulation," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2017.

[38] J. Kim, H. Park, J. Bruce, E. Sutton, D. Rowles, D. Pucci, J. Holbrook, J. Minocha, B. Nardone, D. West, *et al.*, "The tongue enables computer and wheelchair control for people with spinal cord injury," *Science translational medicine*, vol. 5, no. 213, pp. 213ra166–213ra166, 2013.

[39] R. Barea, L. Boquete, M. Mazo, and E. López, "System for assisted mobility using eye movements based on electrooculography," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 10, no. 4, pp. 209–218, 2002.

[40] H. Admoni and S. Srinivasa, "Predicting user intent through eye gaze for shared autonomy," in *AAAI Fall Symposium Series: Shared Autonomy in Research and Practice (AAAI Fall Symposium)*, 2016.

[41] K. Sakita, K. Ogawara, S. Murakami, K. Kawamura, and K. Ikeuchi, "Flexible cooperation between human and robot by interpreting human intention from gaze information," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[42] C.-M. Huang and B. Mutlu, "Anticipatory robot control for efficient human-robot collaboration," in *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2016.

[43] M. M. Jackson and R. Mappus, "Applications for brain-computer interfaces," in *Brain-Computer Interfaces*, 2010, pp. 89–103.

[44] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, "High-performance neuroprosthetic control by an individual with tetraplegia," *The Lancet*, vol. 381, no. 9866, pp. 557–564, 2013.

[45] J. Simeral, S.-P. Kim, M. Black, J. Donoghue, and L. Hochberg, "Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array," *Journal of neural engineering*, vol. 8, no. 2, p. 025027, 2011.

[46] F. Galán, M. Nuttin, E. Lew, P. W. Ferrez, G. Vanacker, J. Philips, and J. d. R. Millán, "A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots," *Clinical neurophysiology*, vol. 119, no. 9, pp. 2159–2169, 2008.

[47] A. B. Schwartz, X. T. Cui, D. J. Weber, and D. W. Moran, "Brain-controlled interfaces: movement restoration with neural prosthetics," *Neuron*, vol. 52, no. 1, pp. 205–220, 2006.

[48] F. A. Mussa-Ivaldi, M. Casadio, and R. Ranganathan, "The body–machine interface: a pathway for rehabilitation and assistance in people with movement disorders," *Expert review of medical devices*, vol. 10, no. 2, pp. 145–147, 2013.

[49] M. Casadio, R. Ranganathan, and F. A. Mussa-Ivaldi, "The body-machine interface: a new perspective on an old theme," *Journal of Motor behavior*, vol. 44, no. 6, pp. 419–433, 2012.

[50] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, p. 010804, 2018.

[51] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[52] D. Kragic, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Human-machine collaborative systems for microsurgical applications," *The International Journal of Robotics Research (IJRR)*, vol. 24, no. 9, pp. 731–741, 2005.

[53] Z. Wang, K. Mülling, M. P. Deisenroth, H. Ben Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human–robot interaction," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 841–858, 2013.

[54] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[55] M. Monfort, A. Liu, and B. D. Ziebart, "Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation." in *AAAI*, 2015.

[56] A. Dragan and S. Srinivasa, "Integrating human observer inferences into robot motion planning," *Autonomous Robots*, vol. 37, no. 4, pp. 351–368, 2014.

[57] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[58] S. Jain, A. Farshchiansadegh, A. Broad, F. Abdollahi, F. Mussa-Ivaldi, and B. Argall, "Assistive robotic manipulation through shared autonomy and a body-machine interface," in *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2015.

[59] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 1, pp. 2–14, 2012.

[60] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau, "Development and evaluation of a flexible interface for a wheelchair mounted robotic arm," in *Proceedings of the ACM/IEEE international conference on Human robot interaction (HRI)*, 2008.

[61] R. M. Aronson, S. Thiago, T. C. Kubler, E. Kasneci, S. Srinivasa, and H. Admoni, "Anticipatory robot control for efficient human-robot collaboration," in *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2018.

[62] D. Whitney, M. Eldon, J. Oberlin, and S. Tellex, "Interpreting multimodal referring expressions in real time," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[63] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[64] D. Gopinath, S. Jain, and B. D. Argall, "Human-in-the-loop optimization of shared autonomy in assistive robotics," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 247–254, 2017.

[65] B. D. Argall, "Modular and adaptive wheelchair automation," in *Experimental Robotics*, 2016, pp. 835–848.

[66] K. Muelling, A. Venkatraman, J.-S. Valois, J. E. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell, "Autonomy infused teleoperation with application to brain computer interface controlled manipulation," *Autonomous Robots*, pp. 1–22, 2017.

[67] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, p. 0278364918776060, 2018.

[68] E. You and K. Hauser, "Assisted teleoperation strategies for aggressively controlling a robot arm with 2D input," in *Proceedings of Robotics: Science and Systems*, 2011.

[69] M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella, "Computer vision for assistive technologies," *Computer Vision and Image Understanding*, vol. 154, pp. 1–15, 2017.

[70] L. M. Bergasa, M. Mazo, A. Gardel, R. Barea, and L. Boquete, "Commands generation by face movements applied to the guidance of a wheelchair for handicapped people," in *Proceedings 15th International Conference on Pattern Recognition (ICPR)*, vol. 4, 2000, pp. 660–663.

[71] P. Jia, H. H. Hu, T. Lu, and K. Yuan, "Head gesture recognition for hands-free control of an intelligent wheelchair," *Industrial Robot: An International Journal*, vol. 34, no. 1, pp. 60–68, 2007.

[72] E. J. Rechy-Ramirez, H. Hu, and K. McDonald-Maier, "Head movements based control of an intelligent wheelchair in an indoor environment," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012.

[73] Q. X. Nguyen and S. Jo, "Electric wheelchair control using head pose free eye-gaze tracker," *Electronics Letters*, vol. 48, no. 13, pp. 750–752, 2012.

[74] C. Gao, M. Sands, and J. R. Spletzer, "Towards autonomous wheelchair systems in urban environments," in *Field and service robotics*, 2010, pp. 13–23.

[75] S. Panzieri, F. Pascucci, and G. Ulivi, "Vision based navigation using kalman approach for slam," in *Proceedings of 11th International Conference on Advanced Robotics*, 2003.

[76] F. Pasteau, A. Krupa, and M. Babel, "Vision-based assistance for wheelchair navigation along corridors," in *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[77] A. Argyros, P. Georgiadis, P. Trahanias, and D. Tsakiris, "Semi-autonomous navigation of a robotic wheelchair," *Journal of Intelligent and Robotic Systems*, vol. 34, no. 3, pp. 315–329, 2002.

[78] C. Montella, M. Pollock, D. Schwesinger, and J. Spletzer, "Stochastic classification of urban terrain for smart wheelchair navigation," in *Proceedings of the IROS Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs*, 2012.

[79] M. Derry and B. Argall, "Automated doorway detection for assistive shared-control wheelchairs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[80] F. Pasteau, V. K. Narayanan, M. Babel, and F. Chaumette, "A visual servoing approach for autonomous corridor following and doorway passing in a wheelchair," *Robotics and Autonomous Systems*, vol. 75, pp. 28–40, 2016.

[81] F. Leishman, O. Horn, and G. Bourhis, "Multimodal laser-vision approach for the deictic control of a smart wheelchair," in *Ambient Assistive Health and Wellness Management in the Heart of the City*, 2009, pp. 98–107.

[82] D. Vanhooydonck, E. Demeester, A. Hüntemann, J. Philips, G. Vanacker, H. Van Brussel, and M. Nuttin, "Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model," *Robotics and Autonomous Systems*, vol. 58, no. 8, pp. 963–977, 2010.

[83] Y. Ren, W. Zou, H. Fan, A. Ye, K. Yuan, and Y. Ma, "A docking control method in narrow space for intelligent wheelchair," in *Proceedings of ICMA*, 2012.

[84] H. Sermeno-Villalta and J. Spletzer, "Vision-based control of a smart wheelchair for the automated transport and retrieval system (ATRS)," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[85] M. Nuttin, D. Vanhooydonck, E. Demeester, and H. Van Brussel, "Selection of suitable human-robot interaction techniques for intelligent wheelchairs," in *Proceedings of ROMAN*, 2002.

[86] R. C. Simpson, D. Poirot, and F. Baxter, "The Hephaestus smart wheelchair system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 118–122, 2002.

[87] G. Wasson, P. Sheth, C. Huang, and M. Alwan, "Intelligent mobility aids for the elderly," in *Eldercare Technology for Clinical Practitioners*, 2008, pp. 53–76.

[88] T. Carlson and J. d. R. Millán, “Brain-controlled wheelchairs: a robotic architecture,” *IEEE Robotics and Automation Magazine*, vol. 20, no. EPFL-ARTICLE-181698, pp. 65–73, 2013.

[89] Z. Wei, W. Chen, and J. Wang, “3D semantic map-based shared control for smart wheelchair,” in *Intelligent Robotics and Applications*, 2012, pp. 41–51.

[90] C. Weber, S. Wermter, and A. Zochios, “Robot docking with neural vision and reinforcement,” *Knowledge-Based Systems*, vol. 17, no. 2, pp. 165–172, 2004.

[91] T. Martínez-Marín and T. Duckett, “Fast reinforcement learning for vision-guided mobile robots,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[92] T. Bailey, E. Nebot, J. Rosenblatt, and H. Durrant-Whyte, “Behaviour-based docking using the DAMN arbiter,” in *Proceedings of ACRA*, 1999.

[93] M. Williamson, R. Murray-Smith, and V. Hansen, “Robot docking using Mixtures of Gaussians,” in *Advances in Neural Information Processing Systems 11*, 1999.

[94] G. Foley, “A powered wheelchair parking assistance reasoning co-pilot (parc) system for cognitively impaired older adults,” Ph.D. dissertation, 2017.

[95] M. Habbecke and L. Kobbelt, “A surface-growing approach to multi-view stereo reconstruction,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[96] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao, “Mobile manipulation in unstructured environments: Perception, planning, and execution,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 58–71, 2012.

[97] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems,” in *ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, 2010.

[98] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: An efficient probabilistic 3d mapping framework based on octrees,” *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[99] H.-Y. Lin and M. Subbarao, “Vision system for fast 3-d model reconstruction,” *Optical Engineering*, vol. 43, no. 7, pp. 1651–1665, 2004.

[100] D. K. Prasad, “Survey of the problem of object detection in real images,” *International Journal of Image Processing (IJIP)*, vol. 6, no. 6, p. 441, 2012.

[101] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *arXiv preprint arXiv:1809.02165*, 2018.

[102] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[103] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[104] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[105] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[106] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 564–575, 2003.

[107] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[108] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[109] A. Bicchi, "On the closure properties of robotic grasping," *The International Journal of Robotics Research*, vol. 14, no. 4, pp. 319–334, 1995.

[110] W. S. Howard and V. Kumar, "On the stability of grasped objects," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 904–917, 1996.

[111] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[112] P. Brook, M. Ciocarlie, and K. Hsiao, "Collaborative grasp planning with multiple object representations," in *Proceeding of IEEE International Conference onRobotics and Automation (ICRA)*, 2011.

[113] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, "Data-driven grasping with partial sensor data," in *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[114] S. El-Khoury and A. Sahbani, "Handling objects by their handles," in *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[115] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[116] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *Proceeding of IEEE International Conference onRobotics and Automation (ICRA)*, 2013.

[117] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[118] A. T. Miller, S. Knoop, H. Christensen, P. K. Allen, *et al.*, "Automatic grasp planning using shape primitives," in *Proceeding of IEEE International Conference onRobotics and Automation (ICRA)*, 2003.

[119] N. Yamanobe and K. Nagata, "Grasp planning for everyday objects based on primitive shape representation for parallel jaw grippers," in *Proceeding of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2010.

[120] K. Huebner and D. Kragic, "Selection of robot pre-grasps using box-based shape approximation," in *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[121] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, "Contact-reactive grasping of objects with partial shape information," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[122] R. B. Rusu, A. Holzbach, R. Diankov, G. Bradski, and M. Beetz, "Perception for mobile manipulation and grasping using active stereo," in *Proceeding of IEEE-RAS International Conference on Humanoid Robots*, 2009.

[123] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuok, "Human-guided grasp measures improve grasp robustness on physical robot," in *Proceeding of IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[124] K. M. Tsui, D.-J. Kim, A. Behal, D. Kontak, and H. A. Yanco, "I want that: Human-in-the-loop control of a wheelchair-mounted robotic arm," *Applied Bionics and Biomechanics*, vol. 8, no. 1, pp. 127–147, 2011.

[125] E. K. Chong, R. L. Givan, and H. S. Chang, "A framework for simulation-based network control via hindsight optimization," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, 2000.

[126] S. W. Yoon, A. Fern, R. Givan, and S. Kambhampati, "Probabilistic planning via determinization in hindsight," 2008.

[127] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Machine Learning Proceedings 1995*, 1995, pp. 362–370.

[128] P. Trautman, "Assistive planning in complex, dynamic environments: a probabilistic approach," in *Proceesings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2015.

[129] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," *arXiv preprint arXiv:1802.01744*, 2018.

[130] A. Broad, T. Murphey, and B. Argall, "Learning models for shared control of human-machine systems with unknown dynamics," *arXiv preprint arXiv:1808.08268*, 2018.

[131] T. Debus, J. Stoll, R. D. Howe, and P. Dupont, "Cooperative human and machine perception in teleoperated assembly," in *Experimental Robotics VII*, 2001.

[132] B. Driessen, T. T. Kate, F. Liefhebber, A. Versluis, and J. van Woerden, "Collaborative control of the manus manipulator," *Universal Access in the Information Society*, vol. 4, no. 2, pp. 165–173, 2005.

[133] S. Jain and B. Argall, "Robot learning to switch control modes for assistive teleoperation," in *RSS 2016 Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics*, 2016.

[134] A. Dragan and S. Srinivasa, "Formalizing assistive teleoperation," in *Proceedings of Robotics: Science and Systems*, 2012.

[135] K. Muelling, A. Venkatraman, J.-S. Valois, J. Downey, J. Weiss, S. Javdani, M. Hebert, A. Schwartz, J. Collinger, and A. Bagnell, "Autonomy infused teleoperation with application to bci manipulation," in *Proceedings of Robotics: Science and Systems*, 2015.

[136] J. E. Downey, J. M. Weiss, K. Muelling, A. Venkatraman, J.-S. Valois, M. Hebert, J. A. Bagnell, A. B. Schwartz, and J. L. Collinger, "Blending of brain-machine interface and vision-guided autonomous robotics improves neuroprosthetic arm performance during grasping," *Journal of neuroengineering and rehabilitation*, vol. 13, no. 1, p. 1, 2016.

[137] J. Henrich, S. J. Heine, and A. Norenzayan, "The weirdest people in the world?" *Behavioral and brain sciences*, vol. 33, no. 2-3, pp. 61–83, 2010.

[138] J. Henrich, S. J. Heine, and A. Norenzayan, "Beyond weird: Towards a broad-based behavioral science," *Behavioral and Brain Sciences*, vol. 33, no. 2-3, pp. 111–135, 2010.

[139] S. Waller, M. Bradley, I. Hosking, and P. J. Clarkson, "Making the case for inclusive design," *Applied ergonomics*, vol. 46, pp. 297–303, 2015.

[140] A. Tapus, C. Tapus, and M. J. Mataric, "User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy," *Intelligent Service Robotics*, vol. 1, no. 2, p. 169, 2008.

[141] C. Bartneck and J. Hu, "Rapid prototyping for interactive robots," in *Proceeding of the 8th conference on intelligent autonomous systems (IAS-8)*, 2004.

[142] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS quarterly*, pp. 319–340, 1989.

[143] Y. Fernaeus, S. Ljungblad, M. Jacobsson, and A. Taylor, "Where third wave hci meets hri: report from a workshop on user-centred design of robots," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2009.

[144] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS quarterly*, pp. 425–478, 2003.

[145] S. K. Kane, C. Jayant, J. O. Wobbrock, and R. E. Ladner, "Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities," in *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, 2009.

[146] E. C. Grigore, A. Roncone, O. Mangin, and B. Scassellati, "Preference-based assistance prediction for human-robot collaboration tasks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[147] D. Leyzberg, S. Spaulding, and B. Scassellati, "Personalizing robot tutors to individuals' learning differences," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction (HRI)*, 2014.

[148] M. A. Goodrich, D. R. Olsen, J. W. Crandall, and T. J. Palmer, "Experiments in adjustable autonomy," in *Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents*, 2001.

[149] D. F. Glas, S. Satake, F. Ferreri, T. Kanda, N. Hagita, and H. Ishiguro, "The network robot system: enabling social human-robot interaction in public spaces," *Journal of Human-Robot Interaction*, vol. 1, no. 2, pp. 5–32, 2013.

[150] G. Dorais, R. P. Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost, "Adjustable autonomy for human-centered autonomous systems," in *Working notes of the Sixteenth International Joint Conference on Artificial Intelligence Workshop on Adjustable Autonomy Systems*, 1999, pp. 16–35.

[151] S. Jain and B. Argall, "An approach for online user customization of shared autonomy for intelligent assistive devices," in *ICRA 2016 Workshop on Human-Robot Interfaces for Enhanced Physical Interactions*, 2016.

[152] M. Lawitzky, M. Kimmel, P. Ritzer, and S. Hirche, "Trajectory generation under the least action principle for physical human-robot cooperation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[153] E. You and K. Hauser, "Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input," in *Proceedings of Robotics: Science and Systems*, 2011.

[154] R. J. Aumann, "The core of a cooperative game without side payments," *Transactions of the American Mathematical Society*, vol. 98, no. 3, pp. 539–552, 1961.

[155] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, "Role adaptation of human and robot in collaborative tasks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5602–5607.

[156] S. Nikolaidis, S. Nath, A. D. Procaccia, and S. Srinivasa, "Game-theoretic modeling of human adaptation in human-robot collaboration," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2017.

[157] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," in *Advances in neural information processing systems*, 2016, pp. 3909–3917.

[158] P. Beeson and B. Ames, "Trac-ik: An open-source library for improved solving of generic inverse kinematics," in *Proceedings of the IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 928–935.

[159] S. Jain and B. Argall, "Automated perception of safe docking locations with alignment information for assistive wheelchairs," in *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[160] S. Jain and B. Argall, "Grasp detection for assistive robotic manipulation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[161] S. Jain and B. Argall, "Estimation of surface geometries in point clouds for the manipulation of novel household objects," in *RSS 2017 Workshop on Spatial-Semantic Representations in Robotics*, 2017.

[162] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz, "Laser-based perception for door and handle identification," in *Proceedings of ICAR*, 2009.

[163] F. Cheein, C. De La Cruz, T. Bastos, and R. Carelli, "Slam-based cross-a-door solution approach for a robotic wheelchair," *International Journal of Advanced Robotic Systems*, vol. 7, no. 2, pp. 155–164, 2010.

[164] M. Derry and B. Argall, "Automated doorway detection for assistive shared-control wheelchairs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[165] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[166] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[167] P. Schmidt, N. Vahrenkamp, M. Wächter, and T. Asfour, "Grasping of unknown objects using deep convolutional neural networks based on depth images," in *Proceeding of IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[168] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[169] Z.-C. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2D–3D categorization and classification for multimodal perception systems," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1378–1402, 2011.

[170] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proceedings of the Conference on Visualization*, 2002.

[171] N. Dyn, K. Hormann, S.-J. Kim, and D. Levin, "Optimizing 3d triangulations using discrete curvature analysis," *Mathematical methods for curves and surfaces*, vol. 28, no. 5, pp. 135–146, 2001.

[172] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[173] Y. S. Choi, T. Deyle, T. Chen, J. D. Glass, and C. C. Kemp, "A list of household objects for robotic retrieval prioritized by people with als," in *Proceeding of IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2009.

[174] D. Berenson, S. S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, 2011.

[175] S. Jain and B. Argall, “Recursive bayesian human intent recognition in shared-control robotics,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[176] S. Jain and B. Argall, “Probabilistic human intent recognition for shared autonomy in assistive robotics,” *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–23, 2019.

[177] A. I. Goldman, “In defense of the simulation theory,” *Mind & language*, vol. 7, no. 1-2, pp. 104–119, 1992.

[178] G. Csibra and G. Gergely, “‘obsessed with goals’: Functions and mechanisms of teleological interpretation of actions in humans,” *Acta psychologica*, vol. 124, no. 1, pp. 60–78, 2007.

[179] C. Moore and V. Corkum, “Social understanding at the end of the first year of life,” *Developmental Review*, vol. 14, no. 4, pp. 349–372, 1994.

[180] S. Javdani, S. Srinivasa, and J. A. D. Bagnell, “Shared autonomy via hindsight optimization,” in *Proceedings of Robotics: Science and Systems*, July 2015.

[181] A. Ciptadi, T. Hermans, and J. M. Rehg, “An in depth view of saliency.” in *BMVC*, 2013.

[182] C. L. Baker, R. Saxe, and J. B. Tenenbaum, “Action understanding as inverse planning,” *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.

[183] A. Hula, P. R. Montague, and P. Dayan, “Monte carlo planning method estimates planning horizons during interactive social exchange,” *PLoS computational biology*, vol. 11, no. 6, p. e1004254, 2015.

[184] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous robot vehicles*, 1986, pp. 396–404.

[185] C. Ezeh, P. Trautman, L. Devigne, V. Bureau, M. Babel, and T. Carlson, “Probabilistic vs linear blending approaches to shared control for wheelchair driving,” in *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2017.

[186] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” in *AAAI*, 2008, p. 14331438.

[187] N. A. Bernstein, “The co-ordination and regulation of movements,” 1967.

[188] A. Farshchiansadegh, F. Abdollahi, D. Chen, M.-H. Lee, J. Pedersen, C. Pierella, E. J. Roth, I. S. Gonzalez, E. B. Thorp, and F. A. Mussa-Ivaldi, “A body machine interface

based on inertial sensors," in *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014.

[189] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[190] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.