Gaze-Enabled Grasping Assistance For Teleoperation of Robotic Manipulators

by

Kevin Joseph

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2025

© Kevin Joseph 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Shared autonomy in robot teleoperation can ease task completion and lower cognitive load for operators by combining human intent with the autonomous capabilities of robots. As many manipulation control tasks involve the grasping of objects as the first step, augmenting assistance at this stage has the potential to improve user experience and task performance. Existing grasping assistance systems rely on classic grasp planners that limit their capabilities, while some utilize expensive hardware to provide the assistance. Virtual reality systems have been used for input and feedback in teleoperation and have also been applied in grasping assistance systems. Some virtual reality systems feature head-mounted displays that have eye-tracking capabilities, and research has been conducted to leverage the eye-tracking information for teleoperation applications.

This work introduces a novel grasping assistance framework that leverages user intent signalled through eye gaze. Specifically, the system retrieves eye gaze direction from a virtual reality headset during teleoperation. This gaze information is then used to automatically identify the operator's desired object for grasping, suggest suitable grasp options, and allow the operator to select a preferred grasp using their gaze. Once selected, the grasp is automatically executed via a predefined grasping sequence, eliminating the need for one-to-one motion mapping.

The grasping assistance system is implemented using ROS2 to control a Kinova Gen 3 robotic manipulator, with a Meta Quest Pro virtual reality headset providing the user interface. Additionally, the teleoperation system offers visual feedback from cameras in the manipulator's workspace, displayed through the head-mounted display, and incorporates a collision avoidance system to prevent unintended impacts.

A user study was performed with 30 participants using the developed system to compare the usability, workload, and performance of the grasping-assisted teleoperation with pure teleoperation (motion mapping). The study asks the operator to perform a pick-and-place task featuring four different objects in a specified order within the allotted time. Each participant performed the task with both pure teleoperation and grasping assistance modes in random order, and then completed questionnaires attempting to measure the usability of each system and measure their experienced workload. Results show that grasping assistance significantly reduces users' workload, but also leads to lower performance metrics with respect to pure teleoperation. The performance loss could be attributed to the implemented grasp planner. Therefore, a gaze-enabled grasping assistance framework such as the one presented in this thesis has the potential to reduce the workload experienced by users and improve performance metrics over standard motion mapping-based direct teleoperation frameworks.

Acknowledgements

I would like to thank my supervisor, Dr. Yue Hu, for her support, guidance, and dedication to my growth throughout my Master's degree. I am very thankful for the amount of time and effort she invested to help me in any way that she can.

I would also like to extend my gratitude to my friends in the Active and Interactive Robotics lab for their consistent willingness to think through ideas with me or help test something new. Their various expertise and friendship were critical to my progress throughout my Master's degree.

Thank you to the Natural Sciences and Engineering Research Council of Canada (NSERC), reference RGPIN-2022-03857, and the National Research Council (NRC), reference AiP-302-1, for helping fund parts of my research. I would also like to thank the Waterloo RoboHub for their support with the equipment and the University of Waterloo Engineering Excellence MASc Fellowship for their funding.

Finally, I would like to thank my family and friends outside of Waterloo for their constant support. As always, their presence in my life provided a separation from academics that I required at times to succeed.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

List of Figures viii

List of Tables x

1 Introduction 1

1.1 Objective . . . 3

1.2 Contributions . . . 4

1.3 Thesis Organization . . . 4

2 Literature Review 6

2.1 Inputs and Visual Feedback in Teleoperation . . . 6

2.2 Grasping Assistance Methods . . . 10

2.3 Grasp Planning Algorithms . . . 11

2.4 Gaze in Teleoperation . . . 13

2.5 Usability and Workload Studies in Teleoperation . . . 13

2.6 Gaps in Existing Literature . . . 14

3 Teleoperation Background **16**
3.1 Forward Kinematics . . . 16
3.2 Velocity Kinematics . . . 18
3.3 Inverse Kinematics . . . 19
3.4 Contact-GraspNet . . . 20

4 Teleoperation System **22**
4.1 Pose Tracking . . . 22
4.2 Teleoperation System Implementation . . . 25
4.3 Alternate Application of Teleoperation System . . . 27
4.4 Tracking Performance . . . 28

5 Grasping Assistance System **31**
5.1 Grasp Generation . . . 31
5.2 Gaze to Grasp System Implementation . . . 33
5.3 Grasping Assistance Sequence Example . . . 36

6 User Study to Compare Gaze to Grasp to Pure Teleoperation **40**
6.1 Experiment Setup . . . 40
6.2 Experiment Measurements . . . 41
6.3 Experiment Procedure . . . 42
6.4 Participants . . . 43

7 User Study Results and Discussion **44**
7.1 System Usability Scale Results . . . 44
7.2 NASA Taskload Index Results . . . 46
7.3 Completion Time and Completion Score Results . . . 48
7.4 User Study Findings . . . 48
7.5 Performance of the system . . . 49

8 Conclusion **51**

8.1 Conclusions 51

8.2 Future Work 52

References **54**

APPENDICES **63**

A Research Ethics Approval Letter **64**

List of Figures

1.1 Example teleoperation pipeline featuring an input device to control a robot and a camera that provides visual feedback. The dotted line represents the variable distances that could separate the system. 2

2.1 The GELLO unilateral teleoperation systems used to control different robotic manipulators: (A) Two Universal Robots UR5s, (B) a uFactory X-Arm 7 and (C) Franka Emika Panda [67]. ©2024 IEEE 7

2.2 A human operator controls the iCub humanoid robot during a pickup task using body tracking. The VR headset is used for visual feedback, and the VR controllers are used to switch between different modes [48]. ©2019 IEEE 9

2.3 A depth image is passed into the Contact-GraspNet model to generate potential grasps on the right. The bottom left shows an object segmentation model used to find the segmentation masks for objects in the scene, which is used to filter the grasps based on their associated objects [60]. ©2021 IEEE 12

3.1 An example kinematic chain of a manipulator with seven joints (7-DOF) [56] 17

4.1 System diagram of the teleoperation system 25

4.2 Teleoperation view showing the main camera feed, with the manipulator camera in the top right corner 26

4.3 End effector pose in response to desired pose input from the user. 29

5.1 The world frame is shown in red, the camera frame is shown in blue. The xy plane is coincident with the origin of the world frame and is on top of the light brown table 32

5.2 System diagram of the grasp assistance systems 34

5.3 The green circle indicates gaze, the blue highlight shows the segmentation mask, and grasp results use a rainbow scale: red for lowest and violet for highest confidence 35

5.4 Image sequence showing the operator view through the HMD while using Gaze to Grasp. In Fig. 5.4c, gaze locations are highlighted to enhance clarity for this figure. 38

5.5 Image sequence showing an external view of the operator and robot that is temporally aligned with Fig. 5.4. 39

6.1 Experimentals setup with the user wearing the VR headset and using the controller to grasp objects in order 1 to 4 and depositing into the bin. . . 41

7.1 Comparison of SUS, NASA-TLX, completion time, and completion score between the two modes 45

7.2 Box plots of NASA-TLX sources of load ratings 47

List of Tables

7.1 Statistics for SUS, NASA-TLX, completion time and completion score for both modes of teleoperation 46

Chapter 1

Introduction

Robot teleoperation is the remote operation of a robot by a human operator[11]. The distances between the robot and the user can range from indirectly controlling the robot from within the same room to controlling it across the world. The degree of control can also vary between different teleoperation modes. The most standard form of teleoperation involves direct mapping, where all of the robot's actions are controlled by the human operator's commands, such as those traditionally seen in the DaVinci surgical system[19]. Teleoperation can also involve the operator providing high-level instructions for the robot to interpret and execute [15, 5]. This work focuses on the teleoperation of robotic manipulators, but some references to humanoid robots that utilize robotic manipulators for their arms are also included. Fig. 1.1 presents a sample teleoperation pipeline.

The DaVinci surgical system [19] shows a key strength of teleoperation, where the precision provided by a robotic system can be leveraged alongside the high-level planning capabilities of the human user to achieve greater results. Remote surgery through teleoperation is an active research topic that aims to improve the viability of teleoperation for surgery from within the same building to across thousands of kilometres [10]. Similar to how teleoperated surgery can improve access to health care, teleoperation can be used in hazardous environments such as space [55] or nuclear waste handling [2] to reduce human exposure to risk. Furthermore, teleoperation better maintains the human presence that a fully automated system cannot. This human presence is particularly beneficial for caregiving, and teleoperation has also shown promise in this domain [23].

While teleoperation is a promising technology, its performance in task completion is still slower compared to human operators executing the task directly [61], representing a roadblock in deploying the technology in more application domains. Several factors con-

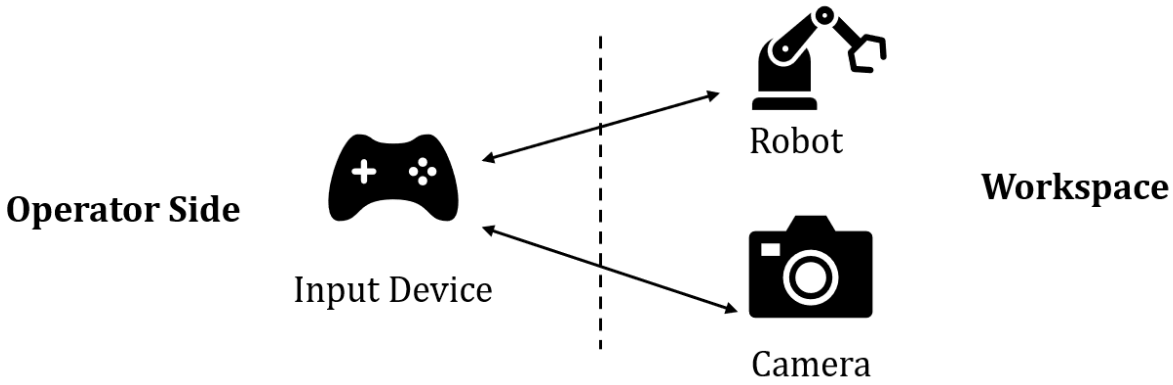


Figure 1.1: Example teleoperation pipeline featuring an input device to control a robot and a camera that provides visual feedback. The dotted line represents the variable distances that could separate the system.

tribute to this challenge, such as the physical limitation of robotic manipulators compared to human arms that constrain motion through joint limits and singularities [54]. When interacting with the environment, humans rely on their senses to get feedback about their own state and the state of the things they are interacting with, but this feedback is limited during teleoperation [25]. There is also additional workload placed on the user to constantly interpret and adjust their commands in response to the robot's environment while relying on limited sensory feedback [41].

In order to alleviate challenges with teleoperation and improve its capabilities, shared control architectures have been proposed to improve task performance during teleoperation and reduce the workload on the user. These architectures augment the inputs from the operator to achieve various goals. For example, shared control could be used to guide the operator to avoid undesirable collisions in the workspace, follow optimal paths with the robot, or maintain desired end-effector orientations. This guidance could be provided through various means, such as visual, haptic, auditory, or even forced on the operator by the automated system. Shared autonomy architectures elevate the guidance provided by shared control architectures to partially or fully automate certain teleoperated tasks to further reduce the workload on the operator.

Assisted teleoperation frameworks have been shown to be helpful in grasping tasks [5, 1, 72]. Though object grasping is an essential task in most teleoperated tasks, it can be significantly challenging and speed-limiting. Most robots use parallel grippers as an end-effector, which are functionally very different from the human hand that operators

are familiar with. To achieve a successful grasp, there is a requirement for precise gripper placement and orientation, which creates an added challenge. To aid in grasping an object during teleoperation, grasping assistance systems aim to make the grasping process intuitive for the user while improving the grasping performance [1, 72].

While current grasping assistance research shows promise, it still faces various challenges. Some systems rely on expensive hardware to provide their assistance, reducing the accessibility of those frameworks. Many grasping assistance systems rely on older grasp planners that limit the system's performance through poor grasping success rate, increased time to generate potential grasps, or limited variety of valid grasp options. This limitation results in the grasping assistance systems offering fewer viable grasps available to the user, which reduces the flexibility they have to complete tasks. Though using an improved grasp planner that creates high-quality grasps quickly and densely would enable providing more flexible grasp options to the user, it also increases the difficulty of designing a user interface that doesn't increase the workload.

1.1 Objective

In this thesis, the objective is to develop Gaze to Grasp, a teleoperated grasping assistance framework that helps reduce the mental load for operators while improving their performance on grasping tasks. Such a system should improve the user experience during teleoperation, improve their ability to use the system for extended periods of time through reduced workload, and enhance their grasping performance during operation.

Gaze to Grasp aims to improve the flexibility that operators have with grasping an object compared to existing grasping assistance frameworks. The primary way to improve this is by providing the operators with a selection of potential grasps that cover the majority of ways the object can be grasped. Flexibility can also be improved by increasing the variety of objects that the grasping assistance algorithm can support. While improving the variety of grasps available to the user could improve the functionality of the grasping assistance framework for the operator, it could also increase the mental load on the operator from a user interface perspective, but this could be alleviated through the use of eye gaze as one of the inputs to the system.

Although object grasping in robotics is implemented using a robotic manipulator, the manipulator could be part of a larger system that is on a mobile base. The framework aims to support grasping assistance even for manipulators that are part of a mobile robot. Full support for a mobile robot requires support for non-stationary cameras, which will be

necessary for grasp generation. This creates a challenge since many grasp planners can be sensitive to the orientation of the camera and often require a top-down view of the target object that is unattainable from a mobile robot, but addressing this challenge will create a more effective framework.

1.2 Contributions

Gaze to Grasp, a novel grasping assistance framework that leverages eye gaze as a source of input from the operator, was developed. It was tested by comparing it against a direct teleoperation framework with no assistance in a within-subjects experiment featuring 30 participants. The system is capable of generating dense potential grasps for a variety of objects, from which the user is able to easily select grasps to execute using their gaze. The contributions described in this thesis have been accepted for presentation at the 2025 IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). The specific contributions from the research presented in this thesis are as follows:

- A teleoperation framework that receives input from a VR controller and provides visual feedback of the workspace through the VR headset. It features smooth tracking capability, joint limit avoidance, and collision avoidance. The framework is implemented using a Kinova Gen 3 robot and a Meta Quest Pro VR system capable of eye tracking
- Development of Gaze to Grasp, a novel framework for teleoperation that utilizes a state-of-the-art grasp planner, and incorporates eye gaze data as an input to target objects for grasping and for executing a desired grasp
- A user study with 30 participants to evaluate the proposed framework that compares Gaze to Grasp's performance, usability, and workload relative to a pure teleoperation system that has only 1-to-1 motion mapping with no assistance

1.3 Thesis Organization

This section provides an overview of the structure of the thesis and a short summary of each of the chapters.

Chapter 2 explores related literature on the topics presented in this thesis to further explore the justification for decisions made and identify gaps in existing literature.

Chapter 3 provides the background necessary to understand the implementation of the pose tracking system used for teleoperation in this work.

Chapter 4 presents various components of the pure teleoperation (motion mapping) system and explains how they are integrated. Detailed implementation of the pose tracking, collision avoidance, and teleoperation system is outlined.

Chapter 5 reports the implementation of Gaze to Grasp by integrating it with the pure teleoperation system. The particulars of how gaze is utilized, how results from the grasp planner are applied, and how operators interact with the system are described.

Chapter 6 describes the experiment conducted to evaluate the Gaze to Grasp system in comparison to the pure teleoperation system, discusses the various experiment measurements taken, and provides insight into the participants of the study.

Chapter 7 then exposes the results from the experiment, discusses how the experiment answers the research questions, and evaluates the performance of the system.

Chapter 8 provides a summary of the thesis, discusses how the objectives for the research have been met, and then describes what future research directions could look like from this work.

Chapter 2

Literature Review

This chapter presents a review of grasping assistance systems and identifies gaps in existing literature. It begins by exploring input and visual feedback methods used in teleoperation to identify and utilize the benefits of VR systems. Grasping assistance research in teleoperation is then outlined, which results in the discussion of recent grasping assistance methods that utilize a VR system. Grasping assistance systems rely on a grasp planner, and the various requirements that an ideal grasp planner should possess are presented in 2.3. The discussion on grasp planners concludes that Contact GraspNet meets many of the presented criteria. While a good grasp planner will offer many potential grasps, it can also increase the usability challenge and workload on the operator. Therefore, gaze tracking is investigated as a possible option to alleviate these concerns, and the application of eye tracking in teleoperation research is explored. To evaluate whether the inclusion of gaze in a grasping assistance system is effective at improving usability and workload for the user, research on usability and workload in teleoperation applications is conducted. Finally, all the findings about gaps in existing literature related to Gaze to Grasp are summarized.

2.1 Inputs and Visual Feedback in Teleoperation

One of the first teleoperation implementations used for robotic manipulators is the bilateral teleoperation system [11]. Bilateral teleoperation involves the control of a follower robot using a leader mechanism that typically closely resembles the follower robot. A key feature for this teleoperation arrangement is transparency, which requires the scaling and transmission of forces and velocities experienced at either side of the control to be transmitted between the leader and the follower [38]. It's still among the most common arrangements

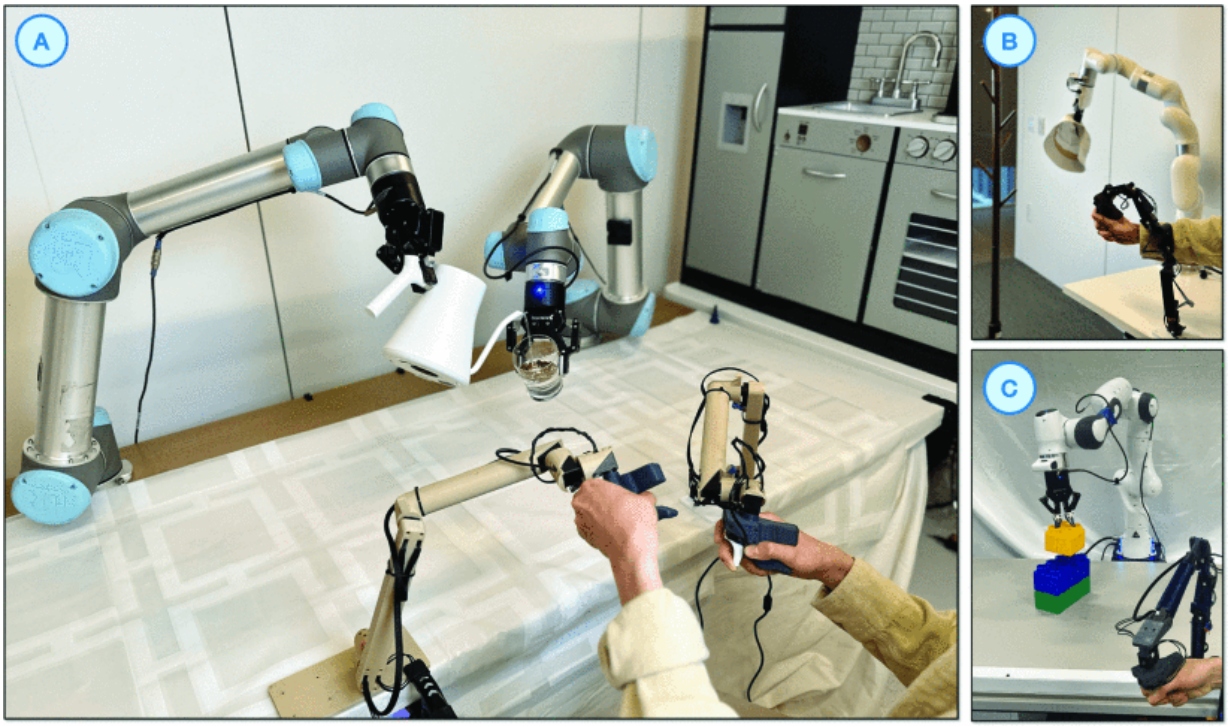


Figure 2.1: The GELLO unilateral teleoperation systems used to control different robotic manipulators: (A) Two Universal Robots UR5s, (B) a uFactory X-Arm 7 and (C) Franka Emika Panda [67]. ©2024 IEEE

for a teleoperation system since its control is not only intuitive for the operator to control, but is also performant [67]. The need for relatively expensive and custom hardware [67], combined with stability and time delay challenges [32], reduces bilateral teleoperation's accessibility and flexibility. Unilateral teleoperation systems, like in [67], aim to reduce the cost of a bilateral teleoperation system by eliminating the transparency feature, which results in the leader device being purely for input (Fig. 2.1).

Fig. 2.2 provides an example of teleoperation through body tracking. It is a modern teleoperation approach that relies on tracking the position of the operator's body through various means and retargeting those motions to make the robot follow [48, 36, 53]. Body pose tracking-based teleoperation frameworks aim to be an intuitive control method since body movement should be replicated by the robot as closely as possible. This input method is more common for teleoperation of humanoid robots [18] due to them having many degrees of freedom that can be controlled, which are similar to humans by design, and body pose

tracking can take advantage of this similarity [36]. The main drawback of body pose tracking is that it typically relies on expensive and uncomfortable motion capture suits and equipment to generate the inputs. Additionally, body tracking-based methods also lose the ability to provide haptic feedback of the forces experienced by the robot without additional equipment. Camera-based body pose trackers offer the potential for cheaper tracking, but remain an active area of research and suffer from accuracy limitations [31].

VR controllers as an input are the other major approach for teleoperation, thanks to their high tracking accuracy in 3D space[8]. This method, in most cases, limits its focus on control of the robotic manipulator's end-effector. The relative position and motion of the VR controllers can be scaled and converted to the desired position and movement of the end-effector. While this reduces direct control of the joints of the manipulator, end-effector control is often the essential part of most teleoperated tasks. Additionally, since only the end-effector's pose is usually controlled by the operator, the position of the joints can be optimized for secondary tasks such as joint limit avoidance [50], collision avoidance [58], or improved manipulability[72]. Furthermore, the VR controllers offer buttons and input that can be mapped to various functions such as UI interaction, end-effector activation, or control over other elements of the robotic system. While VR controllers aren't able to provide the same level of physical feedback of the robot state as bilateral teleoperation, the controllers feature haptic motors that can be used to mimic the physical feedback[26].

A key benefit of VR controller-based teleoperation systems is that the VR controllers naturally integrate with the VR head-mounted display (HMD). Visual feedback of the state of the robot and its workspace is necessary for the operator to perform complex tasks via teleoperation. While standard flat screens are commonly used for visual feedback due to their simplicity [1, 42], the loss of depth perception and reduced immersion is a significant disadvantage [59]. HMDs offer an immersive [59] alternative with their ability to present a different image to each of the user's eyes from a pair of cameras in a stereoscopic configuration [72], improving depth perception[59]. As consumer-grade VR systems have reduced in price, and software development kits for them have improved, VR headsets and systems are now frequently used for teleoperation [72, 68, 65, 18].

Speech-based teleoperation systems utilize spoken commands from an operator to guide a robot. While some systems, such as the one developed by Pulikottil et al. [51], rely exclusively on voice control, speech is also frequently used to augment other input methods, providing additional context about the operator's intent. For example, Kazi et al. [35] integrated speech with gesture inputs, and Chang et al. [15] combined speech with gaze tracking to control a robotic manipulator. This supervisory control approach, where the operator gives high-level instructions and the robot handles the execution, reduces the operator's cognitive load, as demonstrated by Chang et al. [15]. However, it often re-

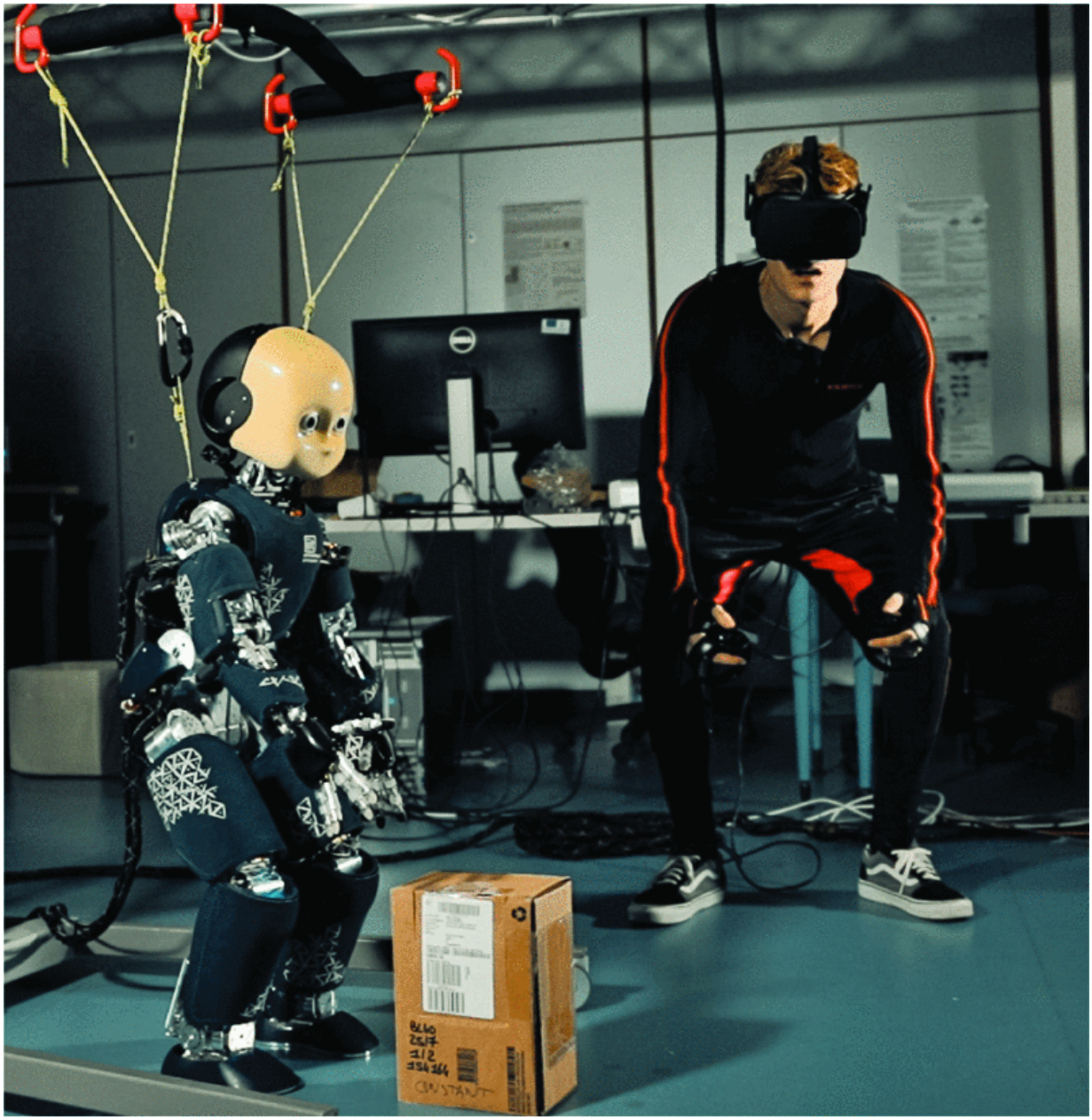


Figure 2.2: A human operator controls the iCub humanoid robot during a pickup task using body tracking. The VR headset is used for visual feedback, and the VR controllers are used to switch between different modes [48]. ©2019 IEEE

quires pre-programming specific behaviors, which can limit the system's generalizability compared to more direct input methods like bilateral controls, body tracking, or VR controllers. A promising new direction is the use of visual-language-action models, which aim to improve generalizability by using vision and text inputs to give high-level commands [62]. Speech-to-text systems can be integrated into this framework to enable spoken commands. Zhao et al. [70] took this a step further by developing a vision-language-action model that directly incorporates speech, and they demonstrated its ability to perform various manipulation tasks. While such systems with direct speech input hold great potential for the future of robotic teleoperation, they currently do not offer the same level of control and generalizability as the more traditional input methods discussed previously.

2.2 Grasping Assistance Methods

As discussed in the previous chapter, shared control frameworks assist the user during teleoperation. In [39], Leeper et al. proposed a "point and click" based visual interface for object grasping. In this framework, the user can select the object they would like to grasp, and then the system would provide a series of grasp candidates that the operator can select from. Upon selecting a grasp, the user has the option to edit its final pose as they wish, and then the robot will automatically execute the grasp. While related to shared control, this teleoperation approach is classified as supervisory control due to the human operator only being involved in the planning stage of the task.

Abi-Farraj et al. [2] developed a true shared control grasping assistance system that aimed to assist the operator in aligning the manipulator's gripper towards the target object during the pre-grasp phase using a haptic input device. The grasping assistance method with haptic assistance outperformed a pure teleoperation approach during their experimental evaluation. Though the shared control approach worked well, the grasping assistance provided to the user utilized the centre of mass of the object as the grasp target. This strategy limits the variety of possible grasps and may be an inaccurate grasping heuristic for more complex objects than those used in their experiment. Fine et al. [22] employed a similar method for grasping assistance, focusing on using an algorithmic approach to identify suitable grasps with various grippers. This approach still required prior knowledge of the objects in the scene and relied on haptic equipment similar to [2]. In [37], Langhi et al. presented a grasping assistance system designed to support dual-arm manipulation that also incorporates user intention estimation from operator VR controller input velocity. Although this system did not require prior knowledge of the objects, it is limited to detecting and providing grasping assistance for boxes.

In [1], Abi-Farraj et al. presented an improved version of their haptic-guided grasping assistance system, which utilizes a deep learning based grasp planner, discussed in the next section, to improve grasp variety and accuracy. While the system performed similarly to its predecessor, it applied to a larger variety of scenarios and objects. Although the system is very promising, it relies on a costly haptic device for input and haptic feedback, and requires a lengthy depth scan of the workspace before it can assist. To address some of these drawbacks, Zhu et al. [72] presented a VR-based shared control framework for grasping assistance. In addition to integrating grasping assistance with a cheaper and more immersive VR teleoperation system relative to the haptic device, they also eliminated the need for extensive scanning of the workspace through the use of point cloud template matching. The template matching relies on a template library of point clouds that represent the 3D shape of objects that will be in the scene, which is not applicable when the system encounters unknown objects. This limitation is evident in the paper, where similarly shaped water bottles are the only type of objects used for presenting the experiments. Finally, this system also limits the grasping flexibility provided to the operator, where they are only able to select which object they would like to grasp, and have no control over the type of grasp for the object.

2.3 Grasp Planning Algorithms

The grasp planner, also called a grasp generation algorithm, is a limiting factor for both [1] and [72], which required compensation from other means, such as extensive scans and point cloud template matching. Both of these works utilized the grasp pose detection (GPD) algorithm [63]. GPD is one of the earliest grasp generation algorithms capable of successfully generating 6-DOF grasp candidates. It utilized a hypothesis and test approach, which involved generating many possible grasps that may be feasible using heuristics, and then evaluating the feasibility of those grasps using a deep learning model. As deep learning techniques have continued to improve in various domains, they have also proven to be very effective for grasp generation [49]. A key categorization of grasp planners is their target mechanism, commonly either a parallel gripper or a dexterous hand [46]. Dexterous hand-based planners aim to create valid grasps for end-effectors that mimic the human arm, but due to the many degrees of freedom of such a hand and the lack of robustness of many of these mechanical hands, dexterous grasping is more niche and difficult compared to parallel gripper grasping [46]. As discussed in Platt et al.'s review paper [49] on learning-based approaches for grasp generation, another key categorization of grasp generation algorithms is based on degrees of freedom (DOF): 4-DOF algorithms generate top-down grasps (a 3D

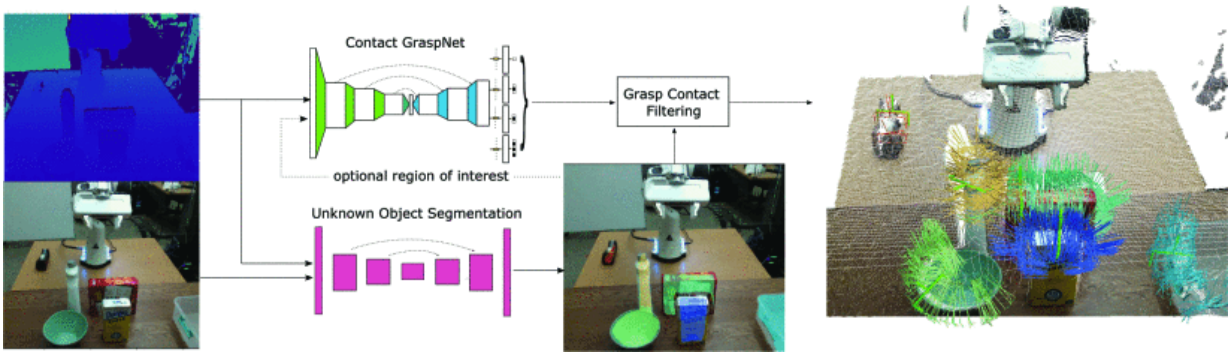


Figure 2.3: A depth image is passed into the Contact-GraspNet model to generate potential grasps on the right. The bottom left shows an object segmentation model used to find the segmentation masks for objects in the scene, which is used to filter the grasps based on their associated objects [60]. ©2021 IEEE

position and a rotation), while 6-DOF algorithms generate grasp poses defined using all six dimensions. Dex-Net [43] is a popular example of a 4-DOF grasp planner, and like all 4-DOF planners, it limits the variety of suitable grasps for an object. A grasp generation algorithm for a grasping assistance system needs to generate a variety of valid grasps.

While it is difficult to compare the performance of various grasp generation algorithms for grasping assistance due to a lack of standardized performance metrics [49], it should provide a variety of valid grasps throughout the target object, be able to efficiently generate grasps close to real-time, support flexible camera positions, and be open-source for development flexibility. Although GPD is used for some state-of-the-art grasping assistance algorithms, it is surpassed by a more recent grasp planner such as Contact-GraspNet [60]. This learning based model takes a different approach than GPD to generate grasps, where it generates potential grasp poses directly from a point cloud of the scene without needing a dedicated sampling step. Another feature of this grasp planner is that it can take a segmentation mask in addition to a point cloud as input, which results in improved grasp suggestions and grasps that are only applicable to the object of interest. In addition to being open-source and supporting flexible camera poses, it generates dense grasp poses for a target object faster than other 6-DOF grasping algorithms based on their evaluation. Fig. 2.3 shows the contact graspnet pipeline and an example of grasps generated by the network.

2.4 Gaze in Teleoperation

Enabling users to select their preferred grasp for an object allows for choices based on the object's intended use. For instance, the optimal grasp for a knife varies depending on whether it's being passed to another person or used for cutting. While not yet applied to grasping assistance, eye gaze has shown promise in teleoperation for inferring and responding to user intent [7]. Aronson et al. [7] demonstrated that incorporating gaze as a supplementary input to joystick control enhanced robot arm teleoperation system performance.

However, Aronson et al. [7] also noted gaze's unreliability as a standalone input. More recent research has presented eye tracking inside VR headsets as a capable input method, provided it is designed around its limitations [34]. This discrepancy might stem from Aronson et. al.'s use of a potentially less effective screen-based eye-tracker compared to the head-mounted eye tracker employed in [34].

A potential concern with offering more grasp options is the increased cognitive workload on the user. Nevertheless, Chang et al. [15] utilized only gaze and speech inputs to teleoperate a robot manipulator, demonstrating how eye gaze, combined with instance segmentation, can accurately identify operator intent. Furthermore, users in that study reported reduced workload when using this teleoperation system. Therefore, VR headsets equipped with eye-tracking capabilities could offer a more economical and flexible input method for grasping assistance, without significantly increasing operator workload.

2.5 Usability and Workload Studies in Teleoperation

Teleoperation is a key area within human-robot interaction (HRI) that focuses on designing and evaluating how humans and robots interact [45]. To ensure consistent and comparable evaluations, it's crucial to use standardized usability and workload measurements. This not only validates the tools used but also allows for easier comparison of results across different studies. Ideally, we'd use measures from other grasping assistance research. However, even measures from general HRI or teleoperation studies are valuable, as the goals for evaluating usability and workload remain consistent across these fields.

The ISO 9241-11, an international standard that provides guidance on usability, describes usability as the degree to which something enables users to effectively and efficiently achieve their goals to their satisfaction when using it for its specified purpose [12]. In 2023, Apraiz et al. [4] presented a literature review on human factors in HRI, which identifies the

most commonly evaluated factors and how they were measured. They classified usability as a 'robot-related factor' since it typically pertains to the experience of interacting with a robot during HRI. The review paper identified two papers that discuss usability. Danielsson et al. [17] used the system usability scale (SUS) [14] to compare the usability between different versions of an augmented reality-based human-robot collaborative assembly system. In [3], Almeida et al. used the IBM Computer Usability Satisfaction Questionnaire [40] to compare the usability between a traditional computer monitor and an HMD during teleoperation. [66] is a more recent teleoperation study that utilized the SUS for usability evaluation, where teleoperation of a mobile robot and manipulator using a game pad is compared to a whole-body motion capture system. As shown by these papers, SUS is commonly used as a measure of usability in teleoperation [64]. All the evaluations here use usability measures to compare two different systems under similar conditions, and not as an objective measure of usability.

Workload is described as the amount of physical and mental resources a person must spend to accomplish a task by Hart et al. in [29]. Apraiz et al. [4] identified three papers on human factors in HRI that discussed workload. The three papers all used the NASA task load index (NASA TLX) to compare workload between different gesture interfaces [6], between human-human and human-robot teams [27], and between different types of user interfaces [47]. NASA TLX has also been used to compare workload in different grasping assistance tasks. Langi et al. [37] used NASA TLX to compare workload between a single manipulator with grasping assistance and a single manipulator without assistance, finding that grasping assistance reduced the workload. They also compared workload in different modes of assistance in the dual-arm manipulator case and again found that grasping assistance reduced the workload. Zhu et al. [72] also performed a workload comparison between their shared control grasping assistance mode and two different direct teleoperation modes. They also found that grasping assistance reduced the workload.

The NASA TLX is the established measure for workload within HRI and has been consistently employed for this purpose in various grasping assistance studies. While SUS is not universally adopted for measuring usability in HRI, it remains the most widely recognized and standard questionnaire for this assessment.

2.6 Gaps in Existing Literature

Existing grasping assistance frameworks, while demonstrating the benefits of shared control, are constrained by several factors: reliance on expensive hardware, reduced grasp density, extended grasp preparation times, and limited grasp flexibility for the user. Many

of these limitations stem from the performance of grasp planners available during their respective research periods. Currently, there is a notable absence of frameworks that leverage recent, more performant grasp generation algorithms, such as Contact-GraspNet. Furthermore, despite eye gaze proving valuable for teleoperation, its application to grasping assistance remains underexplored.

This research addresses these gaps by developing and testing Gaze to Grasp, a novel grasping assistance framework. Gaze to Grasp utilizes a gaze-enabled VR headset for both control and feedback, offers enhanced flexibility in grasp selection, integrates Contact-GraspNet for superior grasp planning, and investigates the potential of gaze in teleoperation. To evaluate its efficacy, a user study will be conducted using the NASA TLX and the SUS. This study will determine if Gaze to Grasp improves user usability and reduces workload compared to a direct teleoperation approach.

Chapter 3

Teleoperation Background

This chapter provides the necessary background to understand the controller used to move the end effector in response to input from the VR controller. First, a high-level explanation of forward kinematics is provided, which leads into velocity kinematics. From the explanations, the specific velocity controller applied in this research is explained, as well as its advantages and disadvantages. The last section provides some background on ContactGraspNet, the grasp generation algorithm that is used in Gaze to Grasp.

3.1 Forward Kinematics

A typical robotic manipulator consists of many links that are chained together using six or seven joints that can rotate (Fig. 3.1). These joints are actuated with motors that allow the links to move with respect to each other and critically change the pose of the end effector, the last link in the chain. The end effector is usually a tool that defines the purpose of the manipulator, such as a gripper, drill, or camera. Forward kinematics is the process of calculating the pose of the end effector x_e, which consists of both its position and orientation, from the joint positions $\theta \in \mathbb{R}^n$, where n is the number of joints in the manipulator. This problem requires the development of a function that uses the joint position θ variable in combination with fixed parameters that define the robot geometry to determine the position of the end effector in Cartesian space.

Transformation matrices are a mathematical tool that can describe the rotational and positional relationship between links in one 4×4 matrix. Through combining these ma-

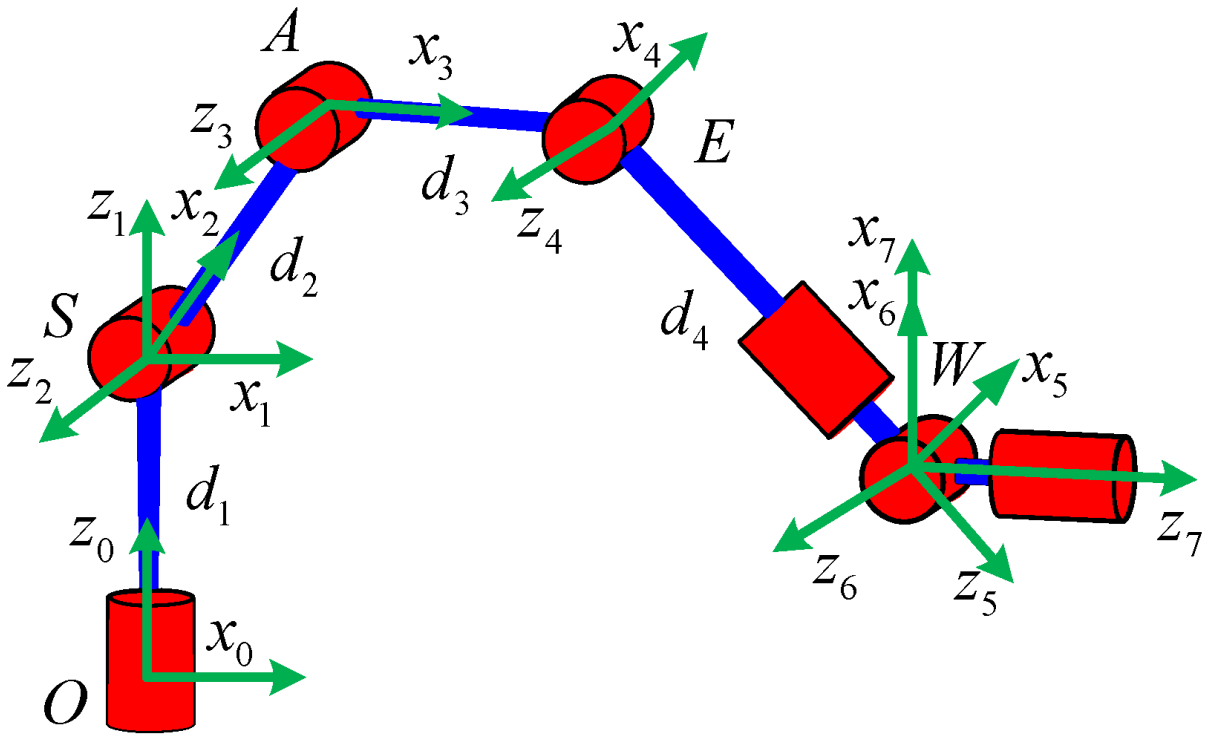


Figure 3.1: An example kinematic chain of a manipulator with seven joints (7-DOF) [56]

trices, the position of the effector can be described as follows:

$$T_n^0 = \prod_{i=1}^{n} T_i^{i-1}(\theta_i) \tag{3.1}$$

Where $H_n^0 \in \mathbb{R}^{4\times4}$ denotes the transformation matrix between the base link zero and the final link. This can also be thought of as the pose of the end effector x_e relative to the base link. $H_i^{i-1}(\theta_i) \in \mathbb{R}^{4\times4}$ represents the transformation to the current link, i, and from the link before, and as a function of θ_i, which represents the position of the current joint.

The Denavit-Hartenberg (DH) convention is typically used to find the necessary parameters that define the transformation between the links as a function of the joint angle θ. While the forward kinematics could be found manually through the use of the DH convention, in practice, it is a standard function provided by various kinematics libraries. These libraries typically rely on robot description files like the URDF (Unified Robot Description Format) to describe the collection of links that constitute the robot.

3.2 Velocity Kinematics

Similar to how the end effector pose x_e of the end effector is found using forward kinematics, the end effector's linear $v \in \mathbb{R}^3$and angular velocities $w \in \mathbb{R}^3$ can be found from its joint velocities $\dot{\theta}$. The Jacobian of the manipulator is the special matrix that relates the vector of joint velocities to the end effector velocity, like so:

$$\xi = J\dot{\theta} \tag{3.2}$$

$$\xi = \begin{bmatrix} v \\ w \end{bmatrix} \tag{3.3}$$

Where the end effector velocity (also known as a twist) $\xi \in \mathbb{R}^6$ is composed of the linear velocity component v and the angular velocity component w. The Jacobian $J \in \mathbb{R}^{6\times n}$ is a matrix dependent on the current joint position θ of the robot. Finally, $\dot{\theta} \in \mathbb{R}^n$ is the vector of current joint velocities of the robot.

Eqn. 3.2 can be modified to find the necessary joint velocities required to create a desired end effector velocity:

$$\dot{\theta} = J^{\dagger}\xi \tag{3.4}$$

$J^\dagger \in \mathbb{R}^{n\times 6}$ is the pseudoinverse of J , and this equation is valid as long as J is not in a singular configuration. It can be calculated using the following equation:

$$J^\dagger = J^T(JJ^T)^{-1} \tag{3.5}$$

3.3 Inverse Kinematics

In contrast to forward kinematics, the inverse kinematics problem aims to determine the specific joint positions θ required to achieve a desired end-effector pose x_d. The workspace of a robotic manipulator defines the total reachable volume for its end-effector. Within this, the dexterous workspace is a crucial subset, where the robot can reach any position with any desired orientation. Importantly, only manipulators with at least six degrees of freedom (6-DOF requiring six joints) can possess a dexterous workspace.

Many contemporary robotic manipulators, especially collaborative robots designed to work alongside humans, feature seven degrees of freedom (requiring seven joints), classifying them as redundant manipulators. This redundancy offers significant advantages, primarily providing a larger set of joint configurations that can achieve a desired pose within the dexterous workspace. While redundancy introduces some computational complexity and increased cost, it enables the integration of additional constraints and secondary objectives beyond simply reaching the target pose. Examples of such secondary goals include: minimizing the risk of collisions, maximizing the distance from joint limits, or reducing the average joint position change needed for a given movement.

Numerous approaches exist for solving inverse kinematics problems, each with its own benefits and drawbacks. A straightforward and commonly used method is the pseudoinverse approach given in Eqn. 3.4, but with an added scaling factor k_e, and desired twist ξ_d:

$$\dot{\theta} = k_e J^\dagger \xi_d \tag{3.6}$$

This technique treats inverse kinematics like a proportional control system, where the desired change in the end effector's position and orientation directly translates to a proportional linear and angular velocity. The iterative approach of the controller moves the end effector towards the target pose x_d after each step of the calculation using the joint velocity output $\dot{\theta}$. Given a desired end effector pose, and current end effector pose, the desired twist ξ_d in Eqn. 3.6 can be found using the logarithm map after calculating the relative transformation from the current pose to the desired pose.

A key property of the pseudoinverse controller is that it generates minimum joint velocities in response to desired twists over the duration that is required to reach the target pose [57]. A potential issue with this approach is the risk of generating large output velocities when the robot approaches a singular configuration (a state where the Jacobian matrix becomes non-invertible). However, this issue did not arise during controller testing. A disadvantage of this approach is that it doesn't account for joint limits, but this can be addressed using the gradient projection method [57]:

$$\dot{\theta} = J^{\dagger}\xi_d + (I - J^{\dagger}J)e_j \tag{3.7}$$

Where $\dot{\theta} \in \mathbb{R}^n$ is the calculated desired joint velocities, $J \in \mathbb{R}^{6\times n}$ is the Jacobian, $J^{\dagger} \in \mathbb{R}^{n\times 6}$ is the pseudoinverse of the Jacobian, $\xi_d \in \mathbb{R}^6$ is the desired twist, and $(I - J^{\dagger}J)e_j$ implements the gradient projection method. This method utilizes the null space projector, $I - J^{\dagger}J$, to achieve a secondary task described by $e_j \in \mathbb{R}^6$ without impacting the primary task. The null space describes the joint velocity vectors that result in no end effector velocity, and this can be leveraged in redundant manipulators to move the joints to more desirable locations without impacting the position of the end effector. The secondary task e_j can be the joint limit avoidance task as described in [50]:

$$e_{j_i} = \begin{cases} -\frac{\theta_i - \theta_i^u}{\theta_i^r} & \text{if } \theta_i > \theta_i^u \\ -\frac{\theta_i - \theta_i^l}{\theta_i^r} & \text{if } \theta_i < \theta_i^l \\ 0 & \text{in other case} \end{cases} \tag{3.8}$$

Where θ_i, θ_i^u, θ_i^l, θ_i^r, are the current position, upper bound, lower bound and range for joint i. The upper and lower bounds can be selected to determine at what joint position the secondary task will activate.

3.4 Contact-GraspNet

Contact-GraspNet is an efficient grasp planner capable of generating high-quality, dense grasp predictions [60]. Sundermeyer et al. [60] developed Contact-GraspNet to infer grasps from a single viewpoint, even in cluttered and partially occluded scenes. Beyond generating robust grasps, its design ensures that the predicted grasps densely cover various regions of the target objects, rather than being confined to a single, potentially optimal area. Furthermore, the generated grasps are post-filtered to prevent collisions with other objects in the scene. The model was trained using the ACRONYM dataset, which comprises nearly 9,000 distinct objects comprehensively labelled with simulated grasps [20].

Fig. 2.3 in Chapter 2 illustrates the inference pipeline of Contact-GraspNet. The network can generate scene grasps from a given point cloud or depth image. However, incorporating a segmentation mask enhances the quality of these predictions by focusing the grasp generation on specific object regions. Since Contact-GraspNet's inception, advancements in image and video segmentation algorithms, such as Segment Anything 2 (SAM2) [52], have demonstrated significant improvements in instance segmentation, which is expected to further refine grasp focusing across more diverse scenes. From its inputs, the model outputs potential grasp poses accompanied by confidence scores, reflecting the model's certainty for each predicted grasp. These grasps then need to be converted from the camera frame to the robot's frame for potential execution.

Chapter 4

Teleoperation System

Two teleoperation modalities are implemented in this work, both sharing underlying components. The two modalities are hereafter referred to as *pure teleoperation*, described in Fig. 4.1, which is a 1-to-1 Cartesian pose tracking from the VR controller to the robot; and *Gaze to Grasp*, which is our proposed new eye-gaze based framework that aims to meet the objectives outlined in chapter 1. Gaze to Grasp extends the pure teleoperation system, as shown in Fig. 5.2, and is described in the next chapter.

The teleoperation system uses the Meta Quest Pro VR headset for user input and feedback, and the robot being teleoperated is the Kinova Gen 3. This manipulator is a 7-DOF robot, characterized by four continuous joints without limits and three non-continuous joints. The overarching framework is implemented using the Robot Operating System 2 (ROS2). A Unity-based application, running directly on the Meta Quest Pro headset, facilitates communication with the ROS2 framework via the ROS TCP Connector package[1] developed by Unity. This section first details the pose tracking design, subsequently describes the comprehensive implementation of the pure teleoperation system, and concludes with insights into its tracking performance.

4.1 Pose Tracking

The relative position of the right controller determines the desired position of the end effector. Since only one Kinova Gen 3 is available during development, only the right VR controller is involved in the tracking, but this could be easily changed to use the left

[1] https://github.com/Unity-Technologies/ROS-TCP-Connector

controller instead. Inside Unity, the pose of both the controller and the HMD is available in a global frame that is defined during the device start-up. The default global frame for the Meta Quest Pro is inconsistent during every start-up, so the pose of the controller needs to be defined relative to a reliable frame. One natural option is the headset frame, but if the robot is tracking the right controller as expected, undesirable movement would occur when the operator observes the workspace using the HMD, even if the right controller remains stationary.

A stationary frame to define the relative pose of the controller is a good option if the operator remains stationary during operation, since only movement of the controller would yield any movement of the robot. Initially, the shoulder was chosen as the location of the stationary frame by offsetting the head pose as required to align the frame with the shoulder during startup. This proved to be too fatiguing during testing since it required the operator to raise their arm above their shoulder to prevent collision with the table, as shown in the workspace arrangement in Fig. 5.1. Shifting the stationary frame 20 cm further below the shoulder estimate proved to be a valid solution to reduce the fatigue experienced by operators.

As per [69], velocity control is used to track the relative pose of the right controller for smoother performance. The joint velocity desired from the robot actuators is generated using the pseudoinverse of the Jacobian method shown in 3.4. The Kinova Gen 3 has four continuous joints, and three non-continuous joints (joints 2, 4 and 6) for which the joint limits are avoided using the secondary task computed as follows:

$$e_{j_i} = \begin{cases} -\frac{k_j(\theta_i - \theta_i^u)^3}{\theta_i^r} & \text{if } \theta_i > \theta_i^u \\ -\frac{k_j(\theta_i - \theta_i^l)^3}{\theta_i^r} & \text{if } \theta_i < \theta_i^l \qquad \text{where } i \in \{2, 4, 6\} \\ 0 & \text{in other cases} \end{cases} \tag{4.1}$$

This is similar to Eqn. 3.8, but using a cubic function to increase the effect of the secondary task when closer to the joint limits, and decrease the effect when farther from the joint limits.

Although the Kinova Gen 3 supports a maximum joint speed of 1.2 rad/s for the joints, it is limited to 0.8 rad/s in this implementation to reduce stress on the joints and the speed of the robot's movement. Speed is reduced so that the operators have more time to adjust their inputs in response to the robot's movements. If the highest computed joint velocities $\dot{q}_{largest}$ exceed the joint maximum joint speed $\dot{q}_{max}$, all joint velocities must be uniformly

scaled down with:

$$k_s = \frac{\dot{\theta}_{max}}{\dot{\theta}_{largest}} \tag{4.2}$$

$$\dot{\theta}_{scaled} = k_s \dot{\theta} \tag{4.3}$$

This scaled joint velocity $\dot{\theta}_{scaled}$ can be used as input into the actuators to achieve a smooth tracking of the desired position of the user, but the system needs to account for robot collisions with itself and the environment.

The table seen in Fig. 5.1 is the most likely source of collision, so it is included as a collision geometry. Other collision geometries could easily be defined and manually added to the collision system if necessary. Collision checking is typically part of the path planning pipeline, but the developed inverse kinematics system is a local inverse kinematics system that only aims to move closer to the target pose at each time step. Therefore, collision checking will also have to be done at a local scale. The collision avoidance mechanism that is implemented for pose tracking leverages the collision checking capabilities of the MoveIt2 kinematics framework[2] and relies on the operator to respond to signs of impending collisions during operation.

The collision avoidance system determines if the current joint velocities would move the manipulator towards a collision with the environment or with itself under a constant velocity assumption. Specifically, the scaled joint velocity $\dot{\theta}_{scaled}$ is integrated at periodic time intervals to define sample joint positions up to a defined maximum collision check time t_{max}. These sample joint positions are tested for collision. If a sample position results in a collision, the sample time t_{col} associated with the joint position is used to scale down the joint velocities further. If the projected time at collision t_{col} is less than the minimum collision time t_{min}, the joint velocities are set to zero. This is maintained until the operator input changes and results in a more collision-free step. The collision scaling logic is summarized in the following equation:

$$k_{col} = \begin{cases} 1 & \text{if } t_{col} \geq t_{max} \\ 0 & \text{if } t_{col} \leq t_{min} \\ tanh(\alpha t_{col} - \beta) & \text{if } t_{min} < t_{col} < t_{max} \end{cases} \tag{4.4}$$

$$\dot{\theta}_{col,scaled} = k_{col} \dot{\theta} \tag{4.5}$$

[2]https://github.com/moveit/moveit2.

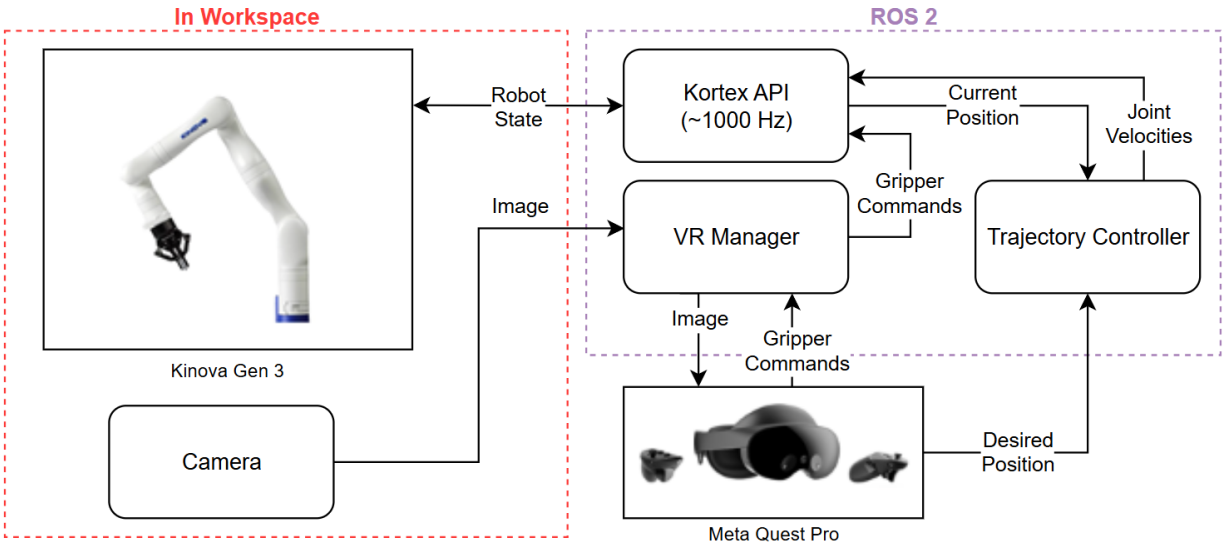


Figure 4.1: System diagram of the teleoperation system

Where $t_{col}, t_{max}, t_{min}$ are the projected times for collision, maximum collision check time, and minimum collision time. The hyperbolic tangent is used for its bounded outputs and to gradually reduce the effects of collision scaling when further from a collision point. For reference, $t_{max}, t_{min}, \alpha, \beta$ are set to 1 second, 0.25 seconds, 2.5 and 0.3, respectively, and 40 samples are distributed evenly between 0 and 1 second. The values are selected to provide the operator enough time to respond to potential collisions while maintaining smooth control of the manipulator.

4.2 Teleoperation System Implementation

The pseudoinverse-based tracking method discussed above is implemented using the Robot Operating System 2 (ROS2) inside the trajectory controller as shown in Fig. 4.1. The Meta Quest Pro's right controller pose is sent to the trajectory controller, which calculates the desired robot position x_d from the user. Using x_d, the current robot end effector position x calculated using the MoveIt 2 kinematics framework from the current robot state, the trajectory controller applies Eqs. 3.7, 4.3 and 4.5 to calculate the desired joint velocities.

While maintaining a constant 1000 Hz connection with the Kinova Gen 3 manipulator, the Kortex API[3] node executes the joint velocity requests from the trajectory controller.

[3]https://github.com/Kinovarobotics/Kinova-kortex2_Gen3_G3L

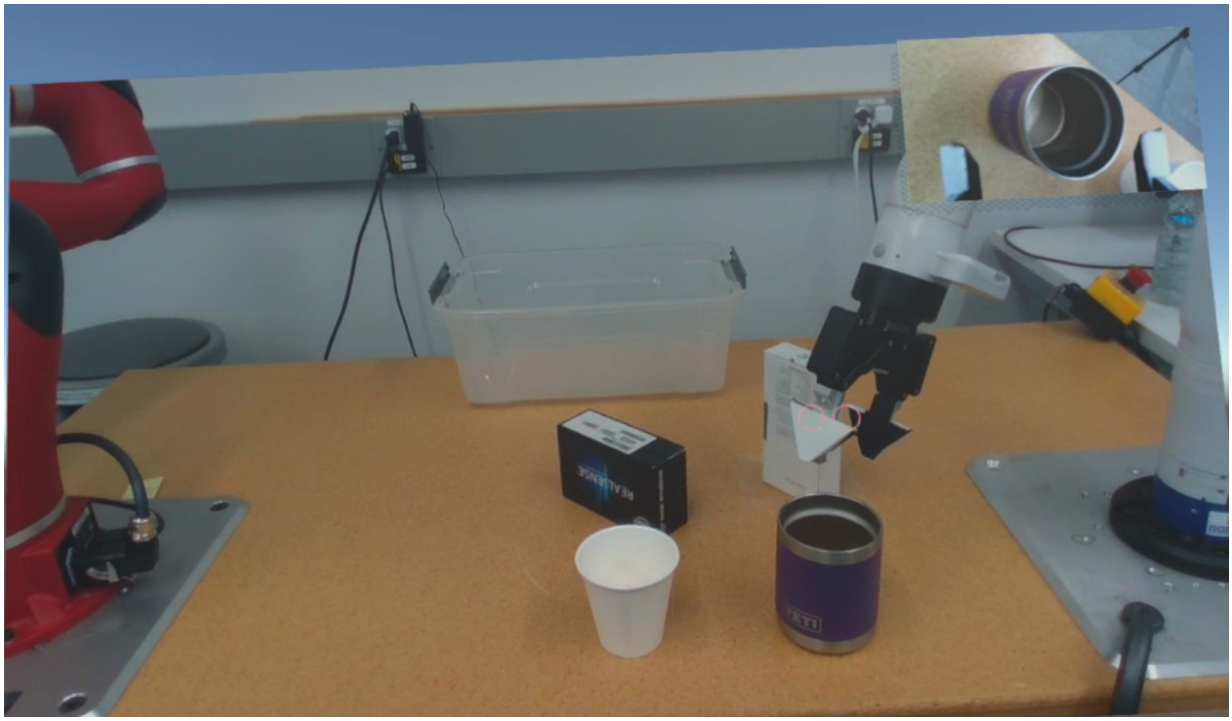


Figure 4.2: Teleoperation view showing the main camera feed, with the manipulator camera in the top right corner

Though Kinova provides a ROS2 interface for the Kinova Gen 3[4], it allows limited control and feedback compared to the Kortex API. Therefore, Kortex API is used to create a custom node that enables more control over the manipulator.

In addition to executing joint velocity requests, the Kortex API node also controls the gripper, so that the operator can issue commands to the gripper through the VR controllers. The VR Manager controls two images from the scene: one image overlooking the scene from the main camera and an image from the camera on the Kinova Gen 3 arm. They are provided to the user for visual feedback, as shown in Fig. 4.2. Presenting only the image from the main camera may severely hinder the operator's performance due to the lack of depth perception. Including the secondary view from the manipulator camera proved to be advantageous during testing. Although including another camera overlooking the scene as stereo vision feedback could provide some depth perception [59], it would compromise the Quest Pro's critical eye-tracking functionality for our grasping assistance system. The main camera is positioned for an unobstructed view of the workspace and clear depth images.

4.3 Alternate Application of Teleoperation System [5]

The pure teleoperation system is capable of easily changing the input device, the robot being controlled, or the trajectory controller being used. This functionality is enabled by the ROS2 framework and the node structure used to define the various components required to create the pure teleoperation system. Each node in the system communicates with other nodes using the ROS2 topics and services. Therefore, if the outputs from a new replacement node maintain the expected ROS2 topic and service structure, it can easily integrate into the system.

As an example, the pure teleoperation system was modified to enable another study that examined how long-distance teleoperation affected user perception. Research partners from the Tokyo University of Science in Japan required a VR teleoperation system to conduct their long-distance teleoperation study. The study aimed to compare how user perceptions associated with long-distance teleoperation compare to local teleoperation when the robot is in physical proximity to the user. The study features the NEXTAGE Fillie robot from

[4]https://github.com/Kinovarobotics/ros2_kortex

[5]The contents of this section refer to a study that has been submitted to the International Conference on Humanoid Robots 2025. S. Capy, T. M. Kwok, K. Joseph, Y. Kawasumi, K. Nagashima, T. Sasaki, Y. Hu, and E. Yoshida, "From Canada to Japan: How 10,000 km Affect User Perception in Robot Teleoperation"

Kawada Robotics being controlled through long-distance teleoperation, and the Kinova Gen 3 being controlled through local teleoperation.

Before developing the system for local teleoperation, the researchers from Japan focused on developing the more challenging long-distance portion of the study. The long-distance portion utilized Meta Quest 2 controllers for input that are connected to a PC running Unity and ROS2. This PC then communicates with a robot control PC through the local network, which communicates the desired position to a remote PC in Japan through the internet. The remote PC performs the robot control as commanded by the user. Since implementing long-distance video feedback inside the VR headset would be outside the scope of their research, the two cameras placed in the workspace of the NEXTAGE Fillie robot provide visual feedback to the user through Microsoft Teams video conferencing software.

The local teleoperation was required to replicate the feature set as closely as possible with the robots available to the researchers in Canada. The pure teleoperation system described in this study was well-suited for adaptation. While it provided all necessary robot control functionalities, the user input and feedback mechanisms required modification to align with the partners' setup. This adaptation was straightforward due to the pure teleoperation system's design, which only necessitated transmitting the pose of the right controller, the headset's pose, and gripper commands from the controller to the PC running Unity.

Overall, the modified pure teleoperation system performed effectively, demonstrating its adaptability and capacity to facilitate research. Study results revealed no significant difference in user perceptions between local and long-distance robot teleoperation. This included user perception of the robots, experienced workload, and system usability.

4.4 Tracking Performance

Fig. 4.3 illustrates the tracking performance of the pure teleoperation system. The operator's input, captured via the VR controller, guided the robot's end-effector from a starting position in the negative x, negative y, positive z quadrant to a final position in the positive x, positive y, negative z quadrant of the robot's workspace. Simultaneously, the gripper's orientation was commanded to change, rotating from an initial alignment with the positive z-axis to a final alignment with the negative z-axis. The position and orientation inputs do not follow a direct or smooth trajectory, which is characteristic of manual human control.

As shown in Fig. 4.3, the trajectory controller effectively enables the robot to follow

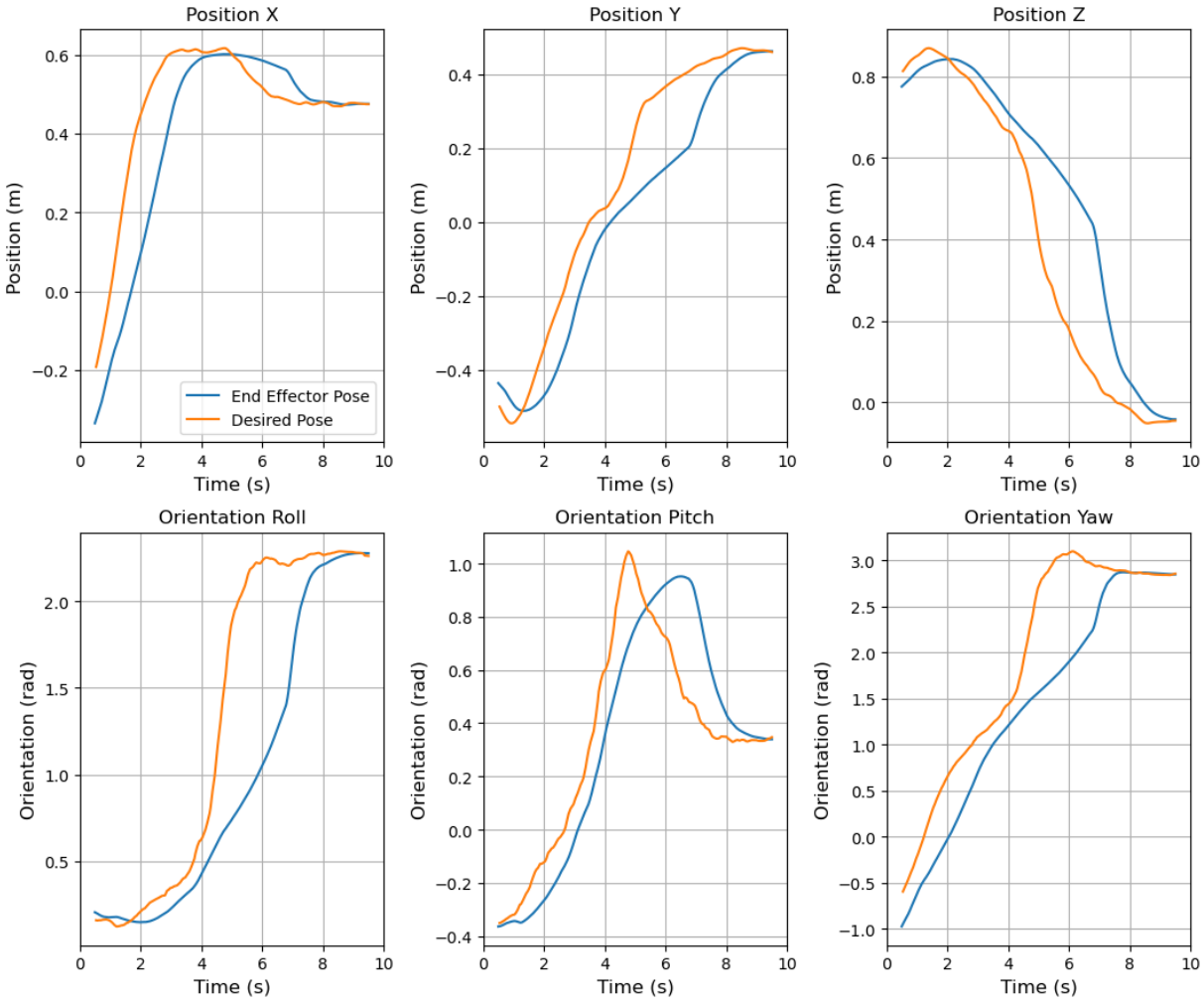


Figure 4.3: End effector pose in response to desired pose input from the user.

the operator's input, albeit with a noticeable tracking delay. The end-effector's movements mirror the commanded trajectory, but with varying degrees of temporal lag. This tracking delay, which is approximately 0.5 seconds between the 2-second and 4-second marks, increases significantly to as much as 2 seconds at the 6.5-second mark. While the linear and angular velocities of the input and the maximum joint speeds of the robot are contributing factors, the exact source of these tracking delays is complex. The manipulability of the robot, which defines the ease of changing the end-effector's position as a function of joint position changes, introduces non-linear effects that complicate the relationship between joint and Cartesian trajectories. A clear correlation exists between increased input velocities and larger tracking delays. This relationship is further supported by the observation that tracking delay is negligible when the input pose is stationary, such as around the 9-second mark.

Despite the significant delays observed in the sample plot, their impact on overall system operation was not substantial. The majority of teleoperation time is spent on precise, low-velocity movements required for tasks like grasping. During these periods of slow input, the tracking delay is minimal. Given that higher velocities, which lead to increased delays, are not representative of typical operational tasks, improving tracking performance at these speeds is unlikely to have a significant impact on the user experience.

Chapter 5

Grasping Assistance System

This chapter describes the details of how Gaze to Grasp is implemented. It leverages the systems developed for the pure teleoperation system as a foundation to implement the systems required for grasping assistance.

The chapter begins by describing how the inputs for Contact-Graspnet are developed. The outputs from the model then need to be filtered and made suitable as a valid selection. The next section provides details of the user interface of Gaze to Grasp. Insight is then provided regarding the implementation of the Gaze to Grasp inside the ROS2 framework. The last section attempts to improve the clarity of the system to the reader by providing an example that includes the operator's view during a task, and the external view of the robot and the operator.

5.1 Grasp Generation

The Contact-GraspNet grasp generation algorithm [60] is selected for its computationally efficient grasp generation, grasp success rate, flexibility to various camera viewpoints, and open-source availability. For best performance, the algorithm requires a depth image and a segmented image which contains segmentation masks for the objects of interest. In this paper, we use an Intel RealSense D435 to obtain the depth and RGB images of the scene.

To generate the segmentation masks for the objects of interest in the scene, the Segment Anything 2 model (SAM2) [52] is chosen due to its high segmentation accuracy, speed, and the ability to generate masks from point prompts. Point prompts act as an additional input to the segmentation model, which selects the region of the image to focus the segmentation

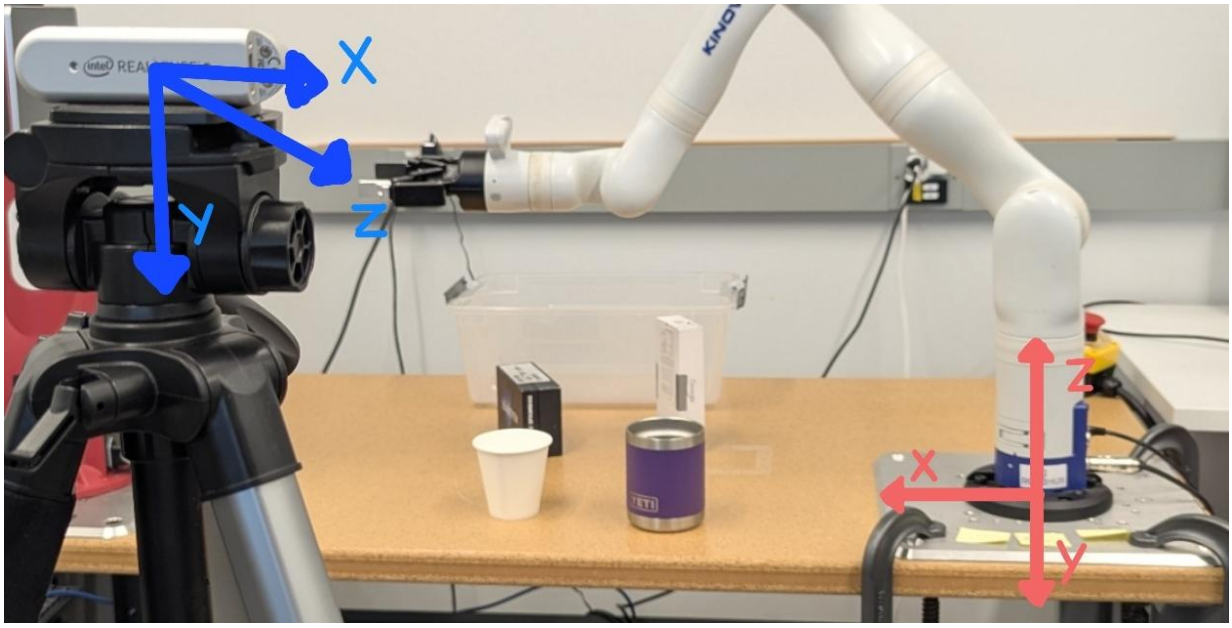


Figure 5.1: The world frame is shown in red, the camera frame is shown in blue. The xy plane is coincident with the origin of the world frame and is on top of the light brown table

around. The point prompts enable more accurate segmentation maps and the ability to select the object of interest during segmentation. This can be combined with gaze data from the Meta Quest Pro to generate accurate segmentation masks focused on the target object. These masks, isolating only the object of interest, ensure that Contact-GraspNet produces relevant grasps using both the mask and depth image as input.

Once the grasps are generated, they are filtered to retain only those within the manipulator's reachable workspace using a heuristic-based approach. Though calculating the inverse kinematics for all the generated poses would be a simple approach, it would be computationally intensive and limit the speed of grasp generation. As the generated grasps from the Contact-GraspNet model are available as rotation matrices, the first heuristic is identifying whether the grasps are facing toward the center axis of the robot or away from the robot. The poses generated in the camera frame are transformed into the world frame (Fig. 5.1). For each transformed pose P_i, the z-axis vector $\hat{z}_i \in \mathbb{R}^3$ and the position vector $p_i \in \mathbb{R}^3$ is extracted. p_i is then projected onto the xy-plane which is shown in Fig. 5.1 :

$$p_{xy,i} = \begin{bmatrix} p_i(1) \\ p_i(2) \\ 0 \end{bmatrix} \tag{5.1}$$

The angle θ_i between the z-axis $\hat{z}_i$ and the projected position vector $p_{xy,i}$ is computed using the dot product:

$$\theta_i = \arccos\left(\frac{z_i \cdot p_{xy,i}}{\|p_{xy,i}\|}\right) \tag{5.2}$$

A grasp pose P_i is considered inaccessible if the angle θ_i does not satisfy the following condition:

$$\theta_i \leq \frac{\pi}{2} + \delta \tag{5.3}$$

Where delta is an adjustable filtering angle threshold. The secondary heuristic requires the pose to lie within a defined spherical region centred in the robot's workspace. A grasp pose is accessible only if both heuristics are satisfied.

5.2 Gaze to Grasp System Implementation

The image segmentation performed using SAM2 and potential grasp generation through Contact-GraspNet is implemented inside the respective nodes as shown in Fig, 5.2. The Ultralytics package that includes SAM[1] enables easy development using the segmentation model. Specifically, the SAM 2.1 small version of the image segmentation model is used to conserve VRAM on the GPU while still ensuring reliable segmentation. A ROS2 node is created for the SAM 2 model to enable integration with the other elements of Gaze to Grasp. A PyTorch reimplementation[2] of Contact-GraspNet is used instead of the TensorFlow-based original implementation[3] since it resulted in easier package management and reduced memory usage. A ROS2 node is also created for the grasp planner with all data required for planning configured as ROS2-compatible messages.

[1] https://docs.ultralytics.com/models/sam2/
[2] https://github.com/elchun/contact_graspnet_pytorch
[3] https://github.com/NVlabs/contact_graspnet

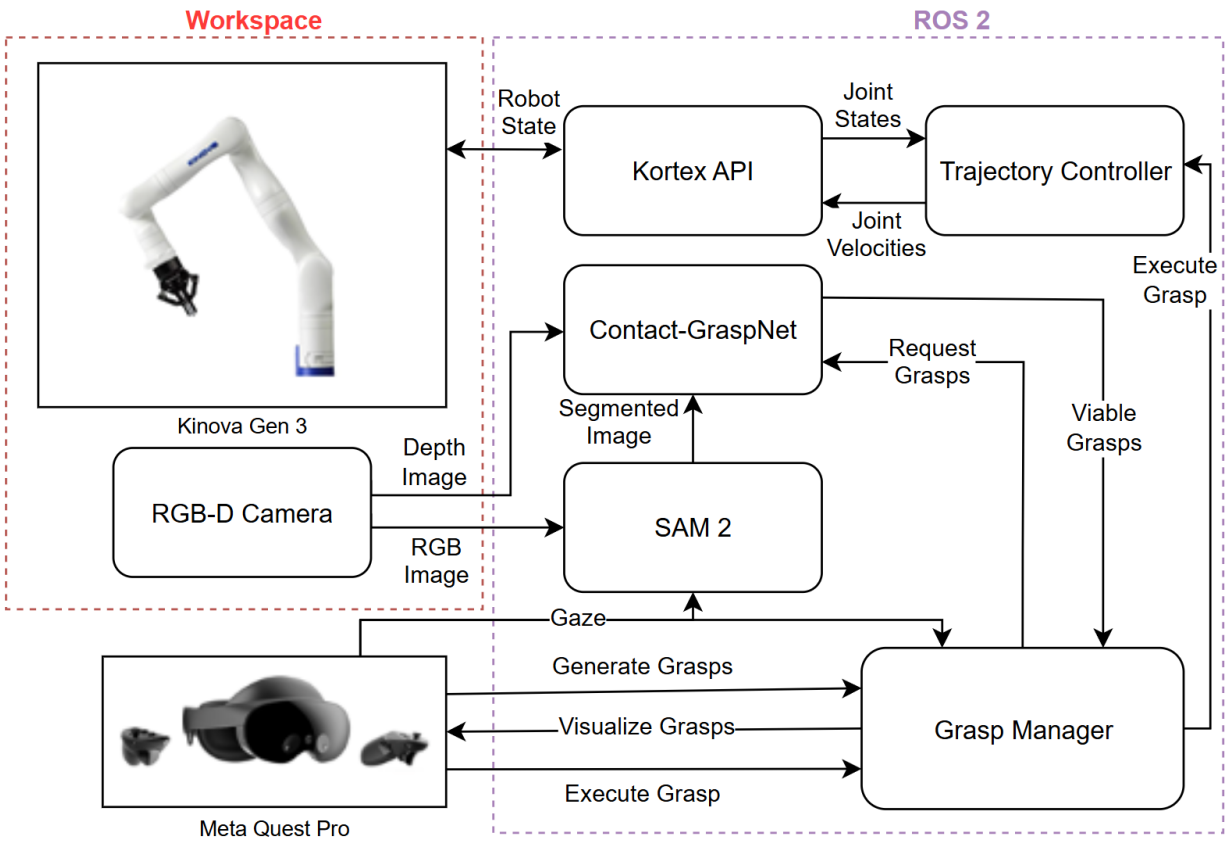
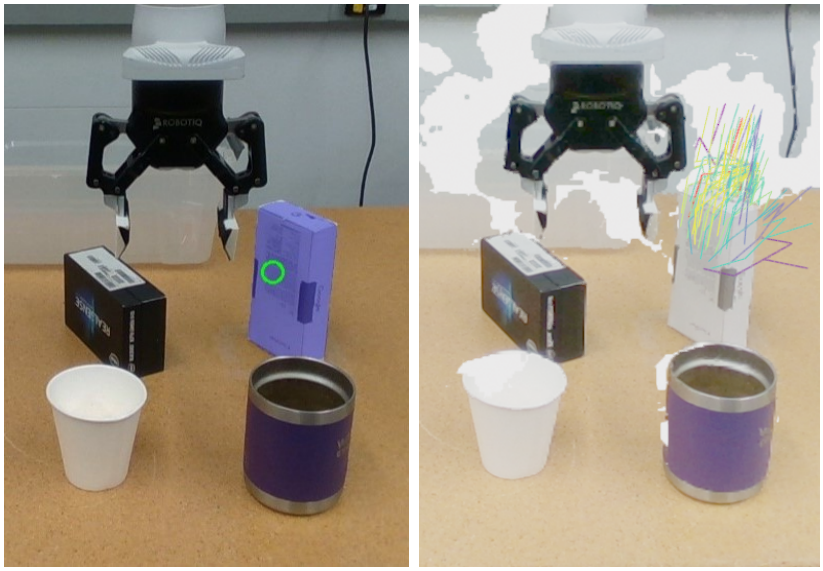


Figure 5.2: System diagram of the grasp assistance systems



(a) Segmentation mask (b) Generated grasps

Figure 5.3: The green circle indicates gaze, the blue highlight shows the segmentation mask, and grasp results use a rainbow scale: red for lowest and violet for highest confidence

Users can request an object grasp simply by looking at the desired object and pressing the grip button on their controller. To aid this, their gaze locations are visualized in the HMD as two red circles, one for each eye, as shown in Fig. 5.4. When a grasp is requested, the system averages the two eye measurements to determine a single gaze point. This point then serves as the prompt for the SAM2 image segmentation model, which generates a segmentation mask. An example of this segmentation from a gaze point is presented in Fig. 5.3a. The generated and filtered grasps from the segmentation mask in Fig. 5.3a are visualized in Fig. 5.3b. The visualization is done using Open3D [71], a 3D data processing library, and it is based on the original implementation by the creators of Contact-GraspNet. Instead of the main camera feed, the potential grasps are shown to the user until they release the index trigger on the VR controller. Once again, using their gaze, the user can look at which grasp they would like the robot to perform and then press the index trigger button to request the execution of the grasp. Upon request, the grasp manager converts the 2D gaze point into a 3D ray and identifies the closest grasps within a defined range of that ray. It then selects the grasp with the highest confidence score from that set for execution and sends it to the trajectory controller. If no grasps are near the gaze point, the grasp with the highest overall confidence is chosen.

The selected grasp is executed using a simple strategy to minimize the risk of collision. First, the robot moves to a waypoint offset (set to 10 cm) from the target grasp along the gripper's negative z-axis. From there, it advances along the gripper's z-axis to reach the target pose, closes the gripper, and returns to the position where the user initiated the grasp command. The user can then release the object by releasing the input used to select the grasp. The waypoint system is developed using the trajectory controller mentioned in Chapter 4. It enables the execution of a series of desired end effector poses and associated gripper states (gripper closed or open).

The waypoint system is also used to initialize the workspace camera frame. The view from the workspace camera, the camera on the manipulator, and ArUco markers [24] are used to compute the transformation from the robot to the main camera frame. Once the placement of the main camera overlooking the workspace is selected, a collection of ArUco markers with known parameters is set onto a specified location in the workspace. Upon initialization of the system, the secondary view from the manipulator camera moves to a pose that overlooks the specified location with the ArUco markers. Then the poses of the ArUco markers can be measured and used as the points of reference from which a transformation between the robot's base frame and the main camera frame can be calculated.

The pure teleoperation mode provides full control of the robot positioning and gripper to the user, while the Gaze to Grasp mode features a shared autonomy architecture. Grasping assistance automates the identification of grasps and aligns the robot for the successful execution of the selected grasp while allowing the user to retain control of all other movements.

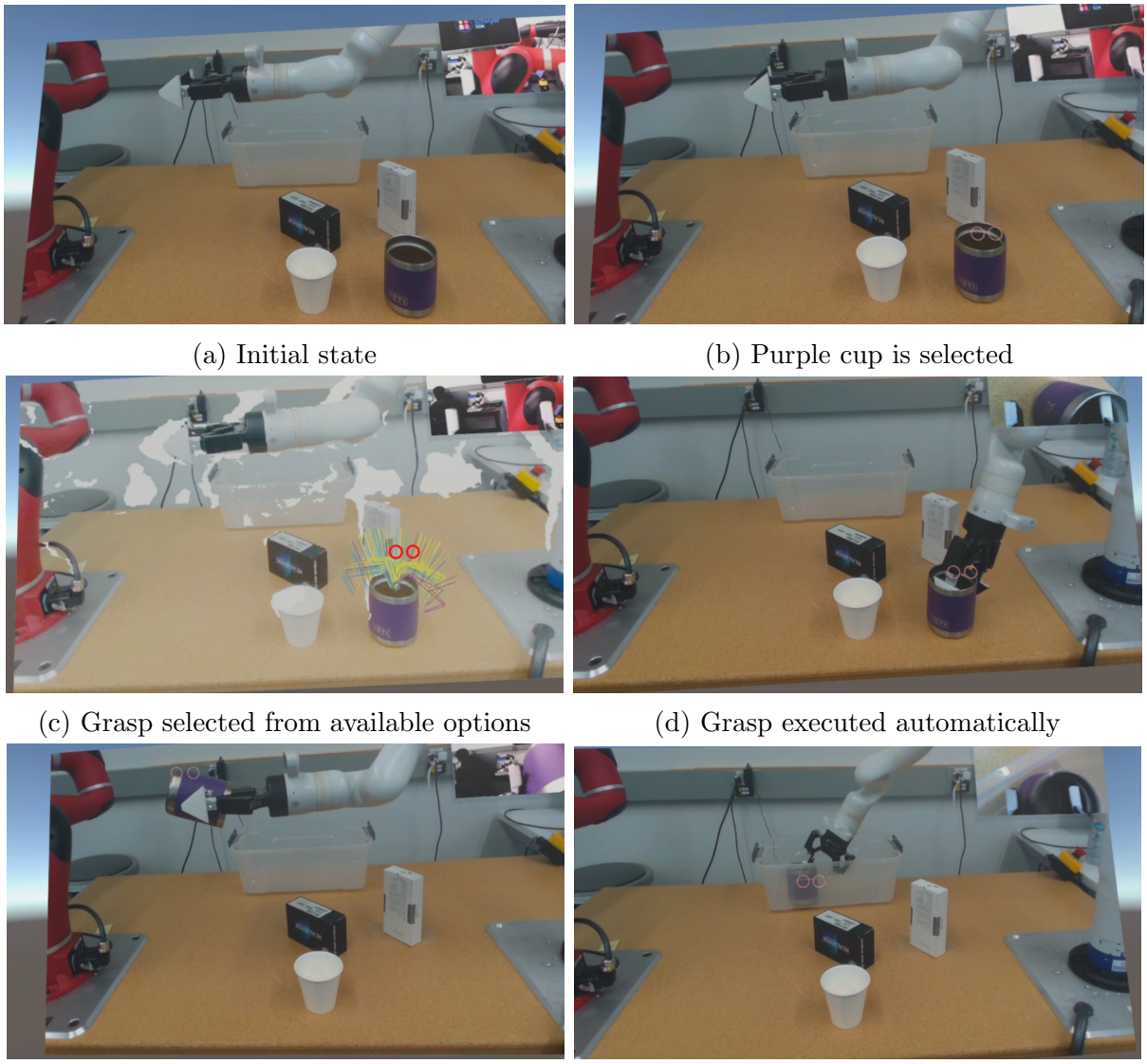
5.3 Grasping Assistance Sequence Example

An example sequence of grasp executions from the operator view through the HMD is presented in Fig. 5.4, and Fig. 5.5 shows the associated sequence of images featuring the operator and the workspace from an external view.

The grasping sequence begins in Fig. 5.4a with the workspace configured for the pick and place task discussed in Chapter 6. The operator initiates Gaze to Grasp by looking at the purple cup in Fig. 5.4b and pressing the grip button on their controller. The gaze points in red determine the target object, and the grasps are generated and visualized as described in this chapter. In Fig. 5.4c, the operator selects their desired grasp with their gaze and presses the index trigger to initiate grasp execution. Gaze to Grasp automatically executes the best grasp at the gaze location (Fig. 5.4d) and returns to the initial position

(Fig. 5.4e). The operator then takes control to complete the remaining task of placing the purple cup inside the bin (Fig. 5.4f).

Unlike the pure teleoperation mode, the operator needs to provide very little input during the grasping process, as seen by the very similar operator pose for the majority of the images in Fig. 5.5. The operator changes position in the final image (Fig. 5.5f) where they resume control in the shared autonomy architecture.



(a) Initial state

(b) Purple cup is selected

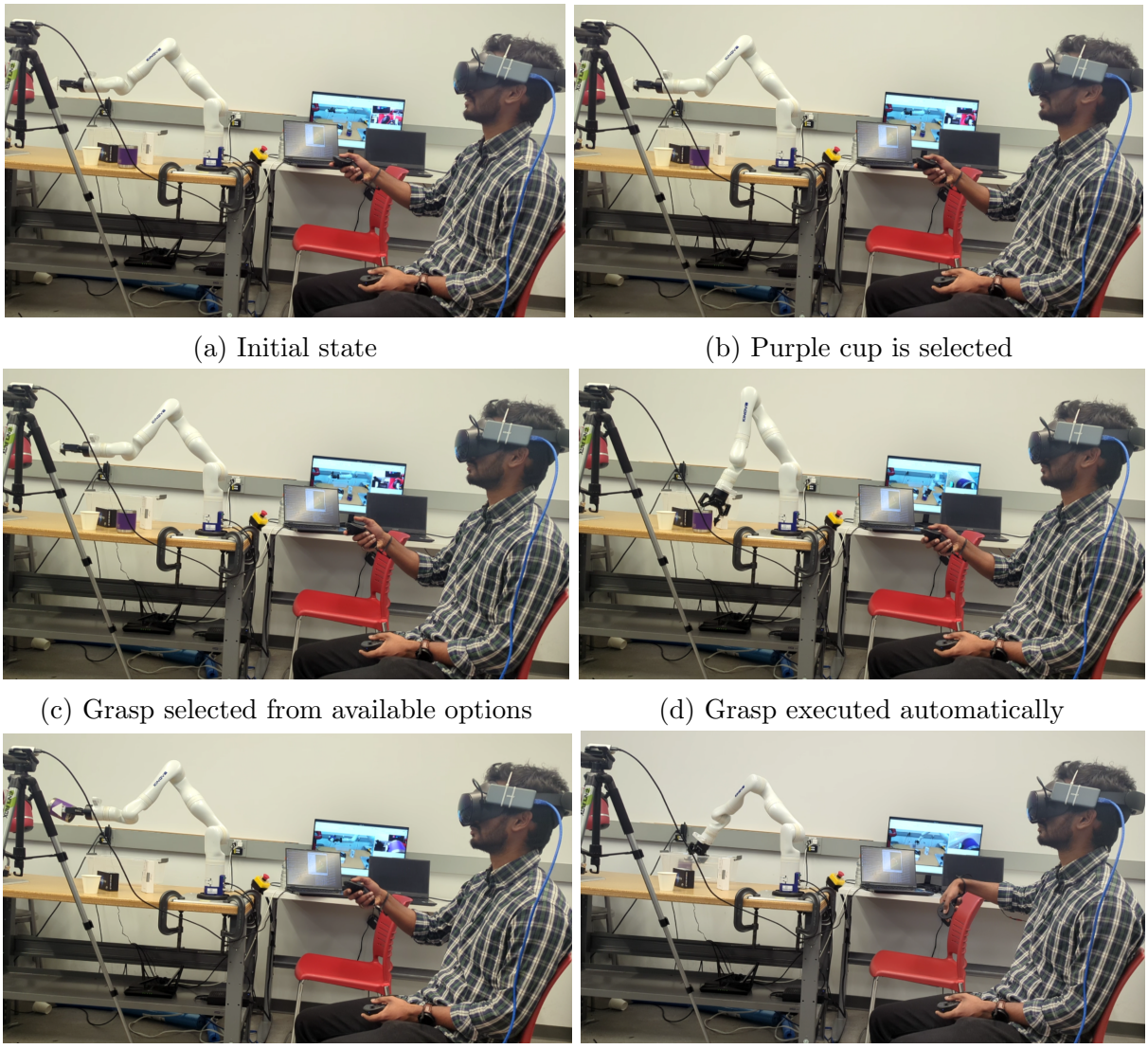
(c) Grasp selected from available options

(d) Grasp executed automatically

(e) Returned to initial position

(f) Operator control to finish task

Figure 5.4: Image sequence showing the operator view through the HMD while using Gaze to Grasp. In Fig. 5.4c, gaze locations are highlighted to enhance clarity for this figure.



(a) Initial state

(b) Purple cup is selected

(c) Grasp selected from available options

(d) Grasp executed automatically

(e) Returned to initial position

(f) Operator control to finish task

Figure 5.5: Image sequence showing an external view of the operator and robot that is temporally aligned with Fig. 5.4.

Chapter 6

User Study to Compare Gaze to Grasp to Pure Teleoperation

To evaluate the proposed grasp assistance system, we conducted a user study with a pick-and-place task, comparing Gaze to Grasp with pure teleoperation in terms of effectiveness, usability, and workload. The experiment received ethics approval from the University of Waterloo Human Research Ethics Board. The letter of approval for the user study can be found in Appendix A.

The research questions we aim to answer with the user study are the following:

RQ1 How does the usability of the Gaze to Grasp system compare to that of pure teleoperation?

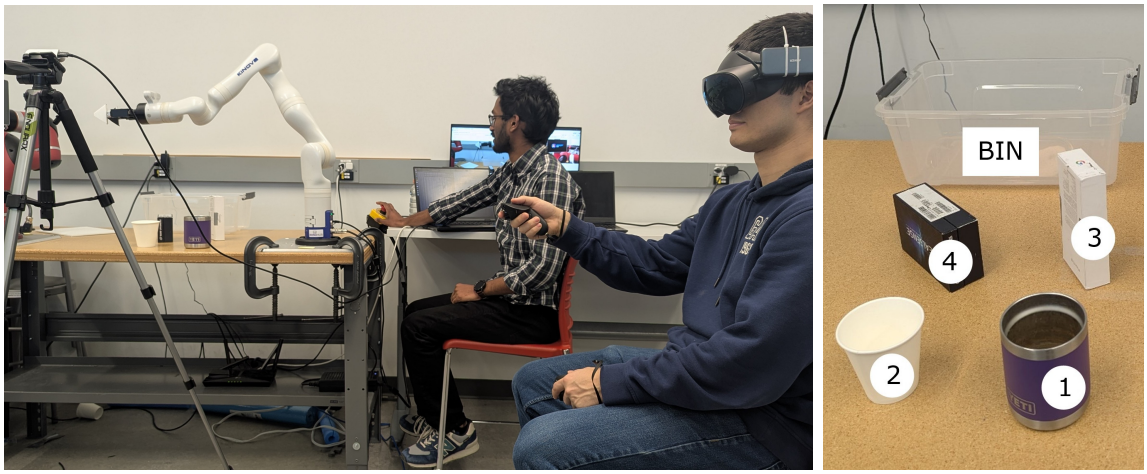
RQ2 Does the Gaze to Grasp system reduce the workload on the operator compared to pure teleoperation?

RQ3 Does Gaze to Grasp improve the task completion score and task completion time compared to pure teleoperation?

RQ4 Does higher perceived system usability correspond to lower perceived workload?

6.1 Experiment Setup

Fig. 6.1a shows the setup used during the user study featuring the relative placement of the user with respect to the robot, the placement of the camera overlooking the scene, and



(a) User setup with VR headset (b) Objects

Figure 6.1: Experimentals setup with the user wearing the VR headset and using the controller to grasp objects in order 1 to 4 and depositing into the bin.

the researcher monitoring the status of the experiment near the robot's emergency stop.

To test the core functionality of Gaze to Grasp, the pick-and-place task requires the participant to grasp four objects in a predefined order from a table and drop them inside a bin on the table. The setup for the task is shown in Fig. 6.1b, where the four objects in the workspace are marked in the order in which they should be completed, and the bin in which the objects need to be placed. Additionally, it challenges the user to maneuver around the other objects in the scene as they are instructed to grasp and place an object inside the bin with minimal or no collision with other objects on the table. The task is designed to be easy for the participants to understand and test the core functionality of the grasping assistance mode compared to pure teleoperation.

Each participant performed the same tasks with both the pure teleoperation mode and Gaze to Grasp mode, in randomized order to avoid biasing effects.

6.2 Experiment Measurements

Task completion time and score are recorded for each object during an experiment session. Specifically, the completion time is the cumulative time users utilize to grasp all objects, and the completion score is computed as follows: half a point for each successfully grasped

object, and half a point for each object successfully dropped into the bin. If the task is not successfully completed, the completion time is set to ten minutes, which is the time limit of the task.

The system usability scale questionnaire (SUS) [13] is used to measure the ease of use of the system. SUS uses a ten-item Likert scale to ask about various aspects of usability concerning their experience using the system under question. This questionnaire is frequently used in literature since it can be quickly and easily completed. The final usability score can be calculated by taking the ten-item scale and applying a series of simple arithmetic described in [13], which generates an overall usability value that ranges from 0 to 100. A higher SUS value indicates a higher usability

The workload experienced by the operator was evaluated with the NASA task load index (NASA TLX) questionnaire [28]. This two-part questionnaire involves a sources of workload evaluation and a rating of load evaluation. The sources of workload portion asks the rater to select which factor of the six factors of workload contributed the most to the task through fifteen pairwise comparisons. This first part yields a series of weights which can be applied to the ratings of load evaluations. The load ratings ask the rater to score the magnitude of load from each of the six factors of workload. The calculated weights are then applied to the ratings of load to generate the NASA-TLX scores that range from 0 to 100. A very low task load would be represented by 0, and 100 represents a very high task load.

Both questionnaires were completed each time after they attempted the task under both modes. For NASA-TLX, the sources of workload evaluation is completed once per participant since both tasks are similar, with only a change in the teleoperation mode, while the sources of load rating scale are done after each attempt of the task.

6.3 Experiment Procedure

Before performing the task with the grasping assistance system, participants perform a calibration process for eye tracking to get accurate measurements of their gaze location. Once they enter the room for the experiment, they are asked to complete a demographics questionnaire, and then they receive an explanation of the task and the controls for the mode they will be using first.

Each task is allotted a maximum of ten minutes, but the participant is provided the option to give up on the task early if they are not able to make progress for any reason.

After the first task, participants are asked to complete the SUS, NASA-TLX sources of workload comparisons, and the NASA-TLX rating of load questionnaires. They then receive an explanation of the second mode and attempt the task again with the new mode. Once the task is done, they complete the SUS and NASA-TLX rating of load questionnaires again for the second task.

6.4 Participants

We recruited 30 participants for the study, with gender distributions of 14 identifying as male, 15 identifying as female, and 1 identifying as non-binary. Their age varied from 18 to 31 ($M = 21.67, SD = 3.50$), with most being students at the University of Waterloo. Of all participants, 21 had little to no experience with robots, 4 had some experience with robots, and 5 had over 20 hours of experience. The experiment configurations are optimized for right-handed users; all 30 participants were comfortable with using their right hand for control-related tasks. Completion time and score data could not be recorded for one participant, so their data were excluded from all analyses involving those metrics. Each participant provided informed consent before the study began.

Chapter 7

User Study Results and Discussion

We consider eight data points from each participant for the analysis: the SUS, NASA-TLX, task completion time, and task completion score for both the pure teleoperation mode and the Gaze to Grasp mode. The significance level for all tests was set at $\alpha = 0.05$. The resulting data for the 30 participants from the study were first tested for normality using the Shapiro-Wilk test. As shown in Table 7.1, alongside the mean, standard deviation, and median of the data, the null-hypothesis of normal distribution is rejected for five out of the eight metrics based on the Shapiro-Wilk test. Therefore, only non-parametric tests, statistical tests that do not require the data to be normally distributed, are used to maintain consistency between tests. To determine significant differences in metrics between the modes, the Wilcoxon signed-rank variance test is performed. The Wilcoxon signed-rank test takes the difference between the data in two groups to determine if the distribution of the data is different from each other, with the null hypothesis being that there is no significant difference between the two groups. Specifically, we performed paired tests between the above-listed metrics for the pure teleoperation case and the Gaze to Grasp case.

7.1 System Usability Scale Results

The box plot in Fig. 7.1 visualizes the distribution of the SUS score. No significant differences were observed (W=207.5, p=0.6120). This indicated that the users found both systems to be similarly usable. The median value of the SUS score for the pure teleoperation mode of 69.5, and 65 for grasping assist, indicating good usability for both modes. To provide some context to the scores, Bangor et al. [9] found that the SUS score for a

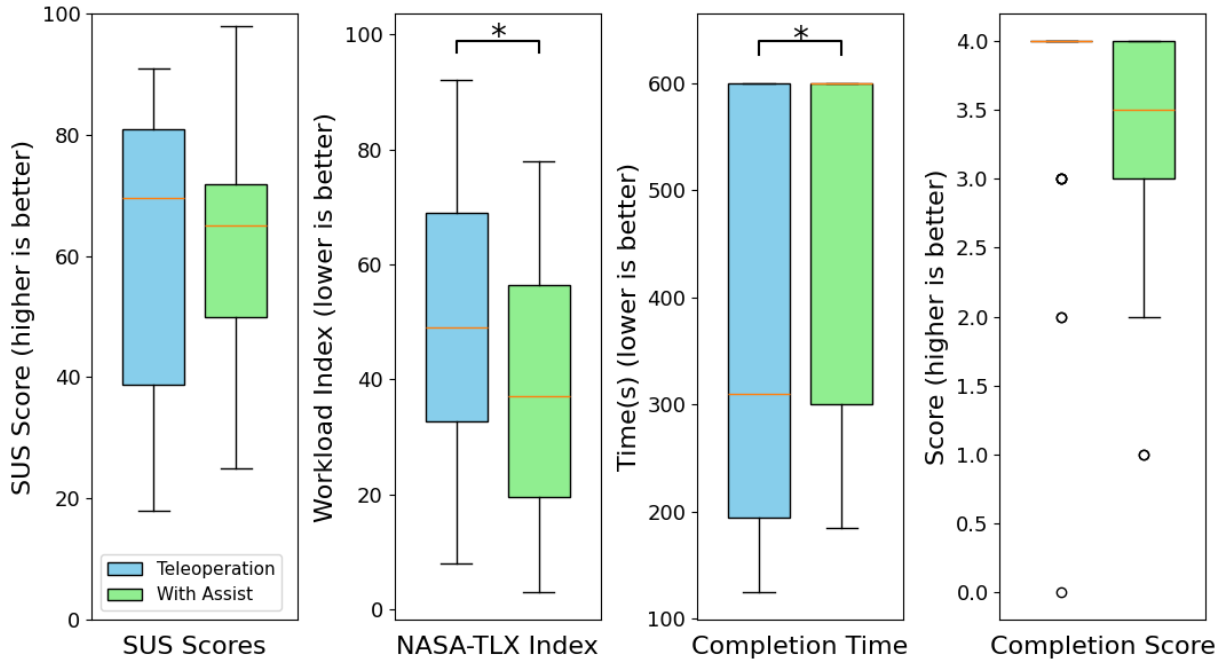


Figure 7.1: Comparison of SUS, NASA-TLX, completion time, and completion score between the two modes

	Pure Teleoperation			
Measure	**SUS**	**NASA-TLX**	**Time**	**Score**
Mean	60.70	50.27	350.00	3.59
SD	18.64	24.53	176.54	0.91
Median	69.50	49.00	310.00	4.00
Shapiro W	0.88	0.95	0.85	0.53
Shapiro p-val	2.5e-3	0.16	6.7e-4	1.7e-8
	With Grasping Assist			
Measure	**SUS**	**NASA-TLX**	**Time**	**Score**
Mean	62.33	38.47	471.10	3.26
SD	18.64	21.34	155.00	0.88
Median	65.00	37.00	600.00	3.50
Shapiro W	0.97	0.96	0.75	0.79
Shapiro p-val	0.64	0.27	1.2e-5	5.5e-5

Table 7.1: Statistics for SUS, NASA-TLX, completion time and completion score for both modes of teleoperation

cellphone interface was 65.9, and 68.2 for a website interface as of 2009. Additionally, based on their study, the usability of the system would range between "OK" and "good" when an adjective-based description is used for scores in this range.

7.2 NASA Taskload Index Results

Results from the NASA-TLX calculations are also shown in Fig. 7.1. From the box plots, we can observe that both pure teleoperation and grasping assist modes have a wide distribution of user-experienced workload. While both modes have a similar range of workload index distribution, the Wilcoxon signed-rank test indicates that the grasping assistance mode resulted in a significantly lower workload for the user (W=131.5, p=0.03841), with a median of 37, while pure teleoperation had a median workload of 49.

Fig. 7.2 shows the distribution of ratings of load recorded from the participants. When comparing between the pure teleoperation and Gaze to Grasp modes, mental demand (W=57.5, p=0.02278), physical demand (W=0, p=6.431e-6), effort (W=102, p=0.002490) and frustration (W=63.0, p=0.03695) ratings showed significant differences between the modes. Gaze to Grasp performed better in all of those factors. As for the sources of workload comparisons, the median measure yielded that performance was the largest factor

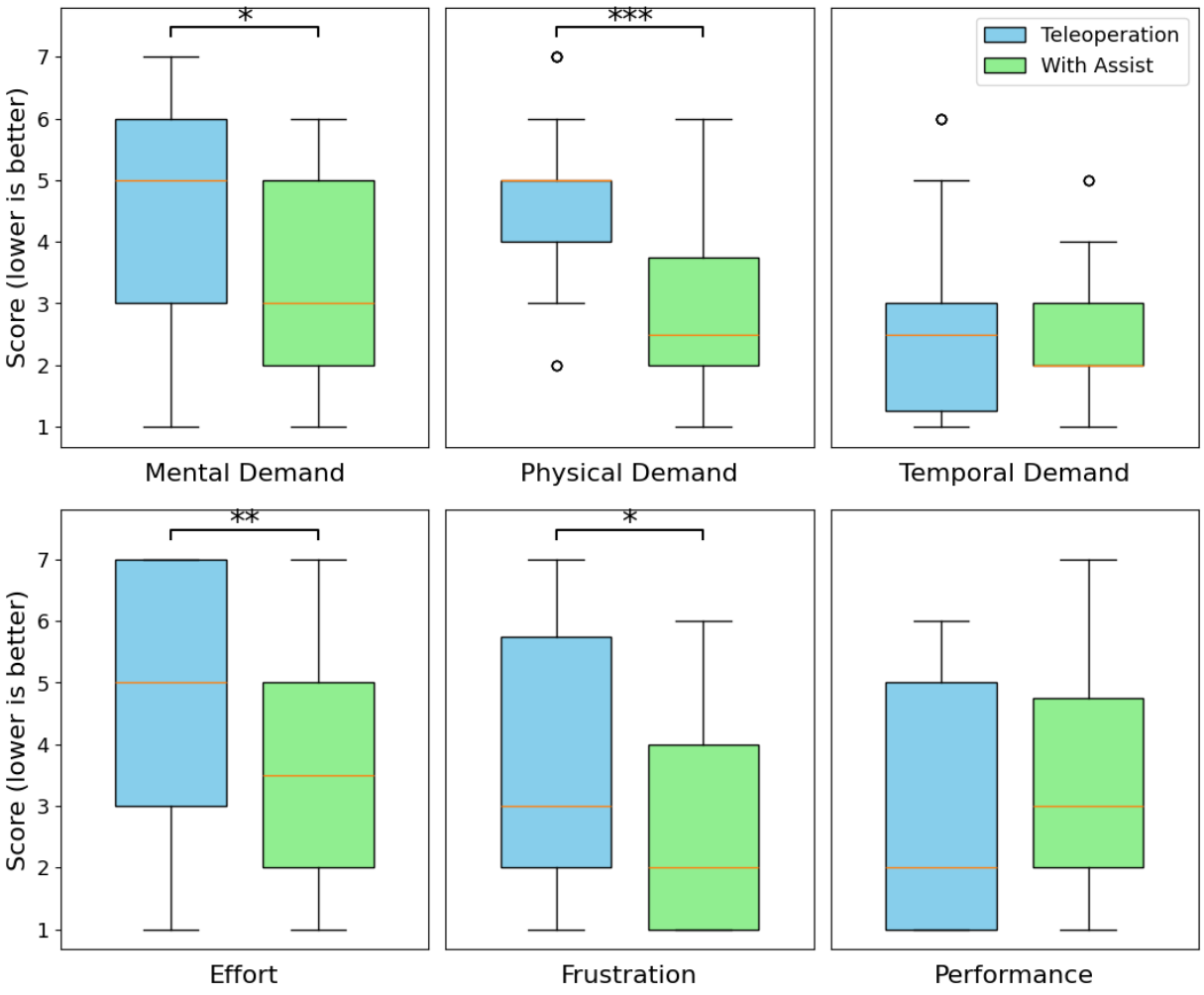


Figure 7.2: Box plots of NASA-TLX sources of load ratings

for workload at 4, effort was at 3, mental demand, physical demand and effort tied at 2, and temporal demand was rated the lowest at 1.5. Sources of workload can range from zero, meaning the lowest factor for workload, to five, meaning the highest factor for workload.

Spearman's correlation was used to compute the correlation between SUS and NASA-TLX to determine how correlated the perceived usability of the system and the workload index are. We found that the SUS score and NASA-TLX workload index were strongly inversely correlated for the pure teleoperation mode (r=-0.8666, p=0.006), and the same for the grasping assistance mode (r=-0.7101, p=0.0001).

7.3 Completion Time and Completion Score Results

Completion time and completion scores are interconnected performance metrics for the study, and the box plots for their distribution are seen in Fig. 7.1. The Wilcoxon signed-rank test indicates that the pure teleoperation mode had a significantly smaller completion time (W=52.5, p=0.005336) with respect to the Gaze to Grasp mode, but the null hypothesis of no difference in completion score wasn't rejected when comparing the completion scores of the two modes (W=44.5, p=0.06170).

The median completion time for Gaze to Grasp, being 600 seconds, indicates that most participants weren't successful in completing the entire task within the allotted time. This observation connects with the median completion score of 3.25, which also indicates that the majority of people weren't able to grasp all four objects, but were able to grasp at least three of the objects using grasping assistance.

7.4 User Study Findings

The user study allowed us to answer the research questions from chapter 6. (RQ1) While participants found the usability of Gaze to Grasp to be similar to teleoperation, they were both in the "OK" to "good" range for usability. (RQ2) The perceived workload when performing the task with the grasping assistance was significantly lower compared to the pure teleoperation case, suggesting that the gaze-based assistance in our proposed shared-autonomy framework effectively reduced the operator's cognitive burden. Specifically, the decrease in workload is significant in mental demand, physical demand, effort, and frustration ratings for Gaze to Grasp compared to pure teleoperation. Although the participants ranked performance as the most important factor for workload, they perceived

their performance to be similar in both modes, as reflected in the completion score performance metric. Participants considered temporal load to be the least important factor and reported experiencing minimal time pressure, despite a notable difference in completion times between the two modes. These findings suggest that the primary benefit of grasping assistance appears to lie not in perceived performance improvements but in creating a more comfortable and less demanding user experience.

(RQ3) On the other hand, Gaze to Grasp performed significantly lower than the pure teleoperation case in terms of task completion time, but from a statistical perspective they performed similarly on the performance score metric. The technical issues with the grasping generation algorithm that are discussed in the next section may have led to the under performance of the grasping assistance mode. These technical issues reduce the significance of the results around performance metrics and system usability, since without those issues, the system could have performed better. Future solutions for these technical issues should be implemented for better performance.

Finally, (RQ4) the highly significant inverse correlation between the SUS score and the NASA-TLX workload index suggests that a system designed for user usability can significantly reduce operator workload. This finding emphasizes the importance of intuitive system design in not only meeting functional requirements but also in minimizing cognitive strain, which would result in increased user satisfaction. The finding also aligns with the discussion presented in the [30], where they found intuitive control mechanisms to be an essential aspect of reducing fatigue on the operator.

7.5 Performance of the system

Overall, both teleoperation modes allowed the majority of participants to complete most of the task. In the pure teleoperation mode, 22 out of 29 achieved full task completion. The pure teleoperation mode resulted in better completion time and score metrics. Even though most participants made verbal remarks about how the main view lacked depth perception, they were still able to adapt to the view and utilize the camera from the gripper alongside the main camera to compensate for the depth perception. The collision avoidance system also worked well at warning operators by slowing down when close to colliding with the table or with other parts of the robot. No undesired collisions were observed during the experiments.

There were two general recurring technical issues with the tracking system. The Quest Pro would occasionally have connectivity issues with its controllers, which would lead to

interruptions that could last up to 30 seconds. This was outside of our direct control and likely connected to one of the various packages we relied on to establish the connection. Additionally, when the robot is teleoperated towards a position requiring a joint to reach its limit, the joint limit avoidance system sometimes fails, causing the robot to stop. This occurs because joint limit avoidance is a secondary task, and the primary task of pose tracking could only be possible by reaching a joint limit. These limitations and potential alternatives are discussed in [16]. Though these issues were present, they occurred sparsely enough such that the operators could focus on the task at hand.

While SAM2 consistently produced reliable image segmentation, the grasp generation was more variable during testing. It frequently failed to generate grasps on certain objects and was highly sensitive to their orientation. Although the system often suggested a broad range of reasonable grasps, some of these were occasionally infeasible in practice due to being slightly misaligned from the ideal grasp. One contributing factor to the Gaze to Grasp's lower performance was Contact-GraspNet's difficulty with object four. Its width was only slightly smaller than the width of the gripper, for which feasible grasps were difficult to generate, eventually leading to some participants ending the trials after multiple failed attempts. For instance, of the 29 participants, 21 completed objects 1-3 with grasping assistance, but only 13 could also complete the fourth one. In comparison, 24 out of 29 participants completed the first three objects in pure teleoperation mode. Though pure teleoperation still performed better when excluding object four, the difference is less pronounced.

The grasp generation pipeline took 1.7 seconds on average to generate and visualize a grasp once a request was received. The image segmentation model and the grasp generation model ran using an NVIDIA RTX 4060 laptop GPU with 8 GB of memory. Variability in the timing of grasp requests from the Meta Quest Pro VR headset sometimes led to delays in sending the request to the grasp manager, causing users to inadvertently move away from the target object before processing was complete. This timing inconsistency occasionally resulted in grasp requests missing the intended target, thereby affecting system reliability during the study.

Chapter 8

Conclusion

This section summarizes the contributions and results from this thesis and describes potential avenues to continue the research presented.

8.1 Conclusions

This thesis introduces a novel grasping assistance framework that leverages eye gaze tracking to enhance user control over grasp execution and increase system flexibility during interaction. It begins by establishing the need for an improved grasping assistance system within teleoperation contexts. Subsequently, it reviews related works, offering insights into the history of teleoperation, the advantages of VR teleoperation systems, and existing grasping assistance systems for teleoperation. The limitations of previously employed grasp planners in current assistance systems are discussed, contrasting them with ContactGraspNet, a more performant and efficient grasp planner. Eye gaze is then presented as a promising input modality to address existing gaps in grasping assistance literature.

The thesis also provides foundational knowledge regarding the inverse kinematics techniques applied in the teleoperation system's implementation. The system itself utilizes a Meta Quest Pro VR headset for user interaction and controls a Kinova Gen 3 robotic manipulator with velocity control. The eye gaze-enabled grasping assistance system, Gaze to Grasp, is built upon this pure teleoperation system. A user study was conducted to evaluate the system's usability, compare the workload between the pure teleoperation and Gaze to Grasp modes, and assess the performance of both modes in a pick-and-place task.

The implemented grasping assistance system successfully meets the objective of providing increased user flexibility. By integrating a state-of-the-art grasp planner, Gaze to Grasp supports a wider variety of objects and generates a greater number of feasible grasp options. Eye gaze is incorporated as an input to Gaze to Grasp to naturally infer user intent for grasping tasks, thereby mitigating potential workload concerns associated with offering more options. The user study demonstrated that Gaze to Grasp significantly decreased user workload compared to pure teleoperation, though no significant difference was observed in system usability. However, pure teleoperation yielded faster and more consistent task executions than the grasping assistance mode, potentially attributable to technical issues within Gaze to Grasp.

Overall, the research presented in this thesis offers valuable insights for improving existing teleoperation systems. The application of Contact-GraspNet in this work clearly demonstrates that state-of-the-art grasp generation algorithms significantly enhance the capabilities for developing advanced grasping assistance systems. Gaze tracking proved highly beneficial in this research, aligning with its demonstrated utility in other teleoperation systems. Finally, these findings underscore that focusing on creating highly usable systems for operators can effectively reduce user workload.

8.2 Future Work

The presented framework was successful in achieving many of the outlined objectives, but the user studies revealed potential areas of improvement. The framework can also be developed to be more versatile.

Although the joint limit task generally functioned well, it failed when the primary task of pose tracking required the manipulator to be near its joint limits. This limitation and potential solutions are presented in [16], such as reconfiguring the joint limit avoidance task to be the primary task and setting the tracking task as the secondary task. Warning the user when approaching regions where the secondary task is ineffective could also be an approach that could be explored.

While Contact-GraspNet is a highly performant grasp planner, particularly when computational resources or inference time are constrained, testing indicated a need for enhanced reliability in its generated grasps for execution. AnyGrasp [21] presented a promising alternative, demonstrating strong performance; however, its closed-source nature and limited demonstrations of non-top-down grasp execution led to its exclusion. Integrating AnyGrasp as the primary grasp planner could potentially improve the reliability of Gaze to Grasp. An

alternative approach involves enhancing the input to Contact-GraspNet through improved point cloud coverage via more effective filtering.

Although the collision detection implemented in this framework proved effective, its scope was limited to the table and the manipulator, excluding other objects within the scene. In more dynamic environments, pre-defining all collision objects becomes unfeasible. Therefore, ROS2-integrated 3D mapping frameworks, such as OctoMap [33], combined with point cloud data from the RealSense camera, could be leveraged to implement a more dynamic collision checking and avoidance framework.

A significant future implementation target for this research involves deployment on mobile humanoid robots. While the grasping assistance system has been designed with mobile platforms in mind, their integration will likely present both unforeseen challenges and benefits. The dual-armed capabilities of humanoid robots offer a wider range of potential grasps and a greater variety of gripper options, such as the vacuum grip implemented in Dex-Net 3 [44]. While the variable camera location on a mobile robot could pose a challenge, grasp planners like Contact-GraspNet should remain capable of effective operation even with dynamic camera positions.

References

[1] Firas Abi-Farraj, Claudio Pacchierotti, Oleg Arenz, Gerhard Neumann, and Paolo Robuffo Giordano. A Haptic Shared-Control Architecture for Guided Multi-Target Robotic Grasping. *IEEE Transactions on Haptics*, 13(2):270–285, April 2020.

[2] Firas Abi-Farraj, Claudio Pacchierotti, and Paolo Robuffo Giordano. User Evaluation of a Haptic-Enabled Shared-Control Approach for Robotic Telemanipulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, October 2018. ISSN: 2153-0866.

[3] Luis Almeida, Paulo Menezes, and Jorge Dias. Interface Transparency Issues in Teleoperation. *Applied Sciences*, 10(18):6232, January 2020. Number: 18 Publisher: Multidisciplinary Digital Publishing Institute.

[4] Ainhoa Apraiz, Ganix Lasa, and Maitane Mazmela. Evaluation of User Experience in Human–Robot Interaction: A Systematic Literature Review. *International Journal of Social Robotics*, 15(2):187–210, February 2023.

[5] Stephanie Arevalo Arboleda, Franziska Rücker, Tim Dierks, and Jens Gerken. Assisting Manipulation and Grasping in Robot Teleoperation with Augmented Reality Visual Cues. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–14, New York, NY, USA, May 2021. Association for Computing Machinery.

[6] Susanna Aromaa, Nikos Frangakis, Domenico Tedone, Juhani Viitaniemi, and Iina Aaltonen. Digital Human Models in Human Factors and Ergonomics Evaluation of Gesture Interfaces. *Proc. ACM Hum.-Comput. Interact.*, 2(EICS):6:1–6:14, June 2018.

[7] Reuben M Aronson and Henny Admoni. Gaze Complements Control Input for Goal Prediction During Assisted Teleoperation. In *Robotics: Science and Systems XVIII*. Robotics: Science and Systems Foundation, June 2022.

[8] Adam Banaszczyk, Mikołaj Łysakowski, Michał R. Nowicki, Piotr Skrzypczyński, and Sławomir K. Tadeja. How Accurate is the Positioning in VR? Using Motion Capture and Robotics to Compare Positioning Capabilities of Popular VR Headsets. In *2024 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 79–86, October 2024. ISSN: 2771-1110.

[9] Aaron Bangor, Philip Kortum, and James Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of usability studies*, 4(3):114–123, 2009.

[10] Patrick Barba, Joshua Stramiello, Emily K. Funk, Florian Richter, Michael C. Yip, and Ryan K. Orosco. Remote telesurgery in humans: a systematic review. *Surgical Endoscopy*, 36(5):2771–2777, May 2022.

[11] Luis Basañez and Raúl Suárez. Teleoperation. In Shimon Y. Nof, editor, *Springer Handbook of Automation*, pages 449–468. Springer, Berlin, Heidelberg, 2009.

[12] Nigel Bevan, James Carter, and Susan Harker. ISO 9241-11 Revised: What Have We Learnt About Usability Since 1998? In Masaaki Kurosu, editor, *Human-Computer Interaction: Design and Evaluation*, pages 143–151, Cham, 2015. Springer International Publishing.

[13] John Brooke. SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1986.

[14] John Brooke. SUS: a retrospective. *Journal of usability studies*, 8(2), 2013.

[15] Yu-Chen Chang, Nitish Gandi, Kazuki Shin, Ye-Ji Mun, Katherine Driggs-Campbell, and Joohyung Kim. Specifying Target Objects in Robot Teleoperation Using Speech and Natural Eye Gaze. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–7, December 2023. ISSN: 2164-0580.

[16] Adrià Colomé and Carme Torras. Closed-Loop Inverse Kinematics for Redundant Robots: Comparative Assessment and Two Enhancements. *IEEE/ASME Transactions on Mechatronics*, 20(2):944–955, April 2015.

[17] Oscar Danielsson, Anna Syberfeldt, Rodney Brewster, and Lihui Wang. Assessing Instructions in Augmented Reality for Human-robot Collaborative Assembly by Using Demonstrators. *Procedia CIRP*, 63:89–94, January 2017.

[18] Kourosh Darvish, Luigi Penco, Joao Ramos, Rafael Cisneros, Jerry Pratt, Eiichi Yoshida, Serena Ivaldi, and Daniele Pucci. Teleoperation of Humanoid Robots: A Survey. *IEEE Transactions on Robotics*, 39(3):1706–1727, June 2023.

[19] Simon DiMaio, Mike Hanuschik, and Usha Kreaden. The da Vinci Surgical System. In Jacob Rosen, Blake Hannaford, and Richard M. Satava, editors, *Surgical Robotics: Systems Applications and Visions*, pages 199–217. Springer US, Boston, MA, 2011.

[20] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A Large-Scale Grasp Dataset Based on Simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227, May 2021. ISSN: 2577-087X.

[21] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. AnyGrasp: Robust and Efficient Grasp Perception in Spatial and Temporal Domains. *IEEE Transactions on Robotics*, 39(5):3929–3945, October 2023.

[22] Tomer Fine, Guy Zaidner, and Amir Shapiro. Grasping Assisting Algorithm in Tele-Operated Robotic Gripper. *Applied Sciences*, 11(6):2640, January 2021. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[23] Laura Fiorini, Alessandra Sorrentino, Mattia Pistolesi, Claudia Becchimanzi, Francesca Tosi, and Filippo Cavallo. Living With a Telepresence Robot: Results From a Field-Trial. *IEEE Robotics and Automation Letters*, 7(2):5405–5412, April 2022.

[24] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, June 2014.

[25] Yiannis Gatsoulis, Gurvinder S. Virk, and Abbas A. Dehghani-Sanij. On the Measurement of Situation Awareness for Effective Human-Robot Interaction in Teleoperated Systems. *Journal of Cognitive Engineering and Decision Making*, 4(1):69–98, March 2010. Publisher: SAGE Publications.

[26] Claudia González, J. Ernesto Solanes, Adolfo Muñoz, Luis Gracia, Vicent Girbés-Juan, and Josep Tornero. Advanced teleoperation and control system for industrial robots based on augmented virtuality and haptic feedback. *Journal of Manufacturing Systems*, 59:283–298, April 2021.

[27] Caroline E. Harriott, Tao Zhang, and Julie A. Adams. Assessing physical workload for human–robot peer-based teams. *International Journal of Human-Computer Studies*, 71(7):821–837, July 2013.

[28] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In Peter A. Hancock and Najmedin Meshkati, editors, *Advances in Psychology*, volume 52 of *Human Mental Workload*, pages 139–183. North-Holland, January 1988.

[29] Sandra G. Hart and Christopher D. Wickens. Workload Assessment and Prediction. In Harold R. Booher, editor, *Manprint: An Approach to Systems Integration*, pages 257–296. Springer Netherlands, Dordrecht, 1990.

[30] Kris Hauser, Eleanor 'Nell' Watson, Joonbum Bae, Josh Bankston, Sven Behnke, Bill Borgia, Manuel G. Catalano, Stefano Dafarra, Jan B. F. van Erp, Thomas Ferris, Jeremy Fishel, Guy Hoffman, Serena Ivaldi, Fumio Kanehiro, Abderrahmane Kheddar, Gaëlle Lannuzel, Jacquelyn Ford Morie, Patrick Naughton, Steve NGuyen, Paul Oh, Taskin Padir, Jim Pippine, Jaeheung Park, Jean Vaz, Daniele Pucci, Peter Whitney, Peggy Wu, and David Locke. Analysis and Perspectives on the ANA Avatar XPRIZE Competition. *International Journal of Social Robotics*, 2024.

[31] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning Human-to-Humanoid Real-Time Whole-Body Teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8944–8951, October 2024. ISSN: 2153-0866.

[32] Peter F. Hokayem and Mark W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057, December 2006.

[33] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, April 2013.

[34] Baosheng James Hou, Yasmeen Abdrabou, Florian Weidner, and Hans Gellersen. Unveiling Variations: A Comparative Study of VR Headsets Regarding Eye Tracking Volume, Gaze Accuracy, and Precision. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 650–655, March 2024.

[35] Zunaid Kazi, Shoupu Chen, Matthew Beitler, Daniel Chester, and Richard Foulds. Speech and gesture mediated intelligent teleoperation. In Vibhu O. Mittal, Holly A.

Yanco, John Aronis, and Richard Simpson, editors, *Assistive Technology and Artificial Intelligence: Applications in Robotics, User Interfaces and Natural Language Processing*, pages 194–210. Springer, Berlin, Heidelberg, 1998.

[36] Doik Kim, Bum-Jae You, and Sang-Rok Oh. Whole Body Motion Control Framework for Arbitrarily and Simultaneously Assigned Upper-Body Tasks and Walking Motion. In Katja Mombaur and Karsten Berns, editors, *Modeling, Simulation and Optimization of Bipedal Walking*, pages 87–98. Springer, Berlin, Heidelberg, 2013.

[37] Marco Laghi, Luigi Raiano, Fabio Amadio, Federico Rollo, Andrea Zunino, and Arash Ajoudani. A Target-Guided Telemanipulation Architecture for Assisted Grasping. *IEEE Robotics and Automation Letters*, 7(4):8759–8766, October 2022.

[38] D.A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637, October 1993.

[39] Adam Eric Leeper, Kaijen Hsiao, Matei Ciocarlie, Leila Takayama, and David Gossow. Strategies for human-in-the-loop robotic grasping. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, HRI '12, pages 1–8, New York, NY, USA, March 2012. Association for Computing Machinery.

[40] James R. Lewis. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human–Computer Interaction*, 7(1):57–78, January 1995. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447319509526110.

[41] Tsung-Chi Lin, Achyuthan Unni Krishnan, and Zhi Li. Physical Fatigue Analysis of Assistive Robot Teleoperation via Whole-body Motion Mapping. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2240–2245, November 2019. ISSN: 2153-0866.

[42] Douglas G. Macharet and Dinei A. Florencio. A collaborative control system for telepresence robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5105–5111, October 2012. ISSN: 2153-0866.

[43] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics, August 2017. arXiv:1703.09312 [cs].

[44] Jeffrey Mahler, Matthew Matl, Xinyu Liu, Albert Li, David Gealy, and Ken Goldberg. Dex-Net 3.0: Computing Robust Vacuum Suction Grasp Targets in Point Clouds Using a New Analytic Model and Deep Learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5620–5627, May 2018. ISSN: 2577-087X.

[45] Robin R. Murphy, Tatsuya Nomura, Aude Billard, and Jennifer L. Burke. Human–Robot Interaction. *IEEE Robotics & Automation Magazine*, 17(2):85–89, June 2010.

[46] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, Dieter Fox, and Akansel Cosgun. Deep Learning Approaches to Grasp Synthesis: A Review. *IEEE Transactions on Robotics*, 39(5):3994–4015, October 2023.

[47] Matteo Pantano, Daniel Regulin, Benjamin Lutz, and Dongheui Lee. A human-cyber-physical system approach to lean automation using an industrie 4.0 reference architecture. *Procedia Manufacturing*, 51:1082–1090, January 2020.

[48] Luigi Penco, Nicola Scianca, Valerio Modugno, Leonardo Lanari, Giuseppe Oriolo, and Serena Ivaldi. A Multimode Teleoperation Framework for Humanoid Loco-Manipulation: An Application for the iCub Robot. *IEEE Robotics & Automation Magazine*, 26(4):73–82, December 2019.

[49] Robert Platt. Grasp Learning: Models, Methods, and Performance. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(Volume 6, 2023):363–389, May 2023. Publisher: Annual Reviews.

[50] Mario Prats, Steven Wieland, Tamim Asfour, Angel P. del Pobil, and Rudiger Dillmann. Compliant interaction in household environments by the Armar-III humanoid robot. In *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pages 475–480, December 2008. ISSN: 2164-0580.

[51] Terrin Babu Pulikottil, Marco Caimmi, Maria Grazia D'Angelo, Emilia Biffi, Stefania Pellegrinelli, and Lorenzo Molinari Tosatti. A Voice Control System for Assistive Robotic Arms: Preliminary Usability Tests on Patients. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pages 167–172, August 2018. ISSN: 2155-1782.

[52] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric

Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos, October 2024. arXiv:2408.00714 [cs].

[53] H. Reddivari, C. Yang, Z. Ju, P. Liang, Z. Li, and B. Xu. Teleoperation control of Baxter robot using body motion tracking. In *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, pages 1–6, September 2014.

[54] Mario Selvaggio, Amir M. Ghalamzan E, Rocco Moccia, Fanny Ficuciello, and Bruno Siciliano. Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3617–3623, November 2019. ISSN: 2153-0866.

[55] T.B. Sheridan. Space teleoperation through time delay: review and prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, October 1993.

[56] Xiaohua Shi, Yu Guo, Xuechan Chen, Ziming Chen, and Zhiwei Yang. Kinematics and Singularity Analysis of a 7-DOF Redundant Manipulator. *Sensors*, 21(21):7257, January 2021. Number: 21 Publisher: Multidisciplinary Digital Publishing Institute.

[57] Bruno Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3):201–212, September 1990.

[58] Sebastian Starke, Norman Hendrich, and Jianwei Zhang. Memetic Evolution for Generic Full-Body Inverse Kinematics in Robotics and Animation. *IEEE Transactions on Evolutionary Computation*, 23(3):406–420, June 2019.

[59] Da Sun, Andrey Kiselev, Qianfang Liao, Todor Stoyanov, and Amy Loutfi. A New Mixed-Reality-Based Teleoperation System for Telepresence and Maneuverability Enhancement. *IEEE Transactions on Human-Machine Systems*, 50(1):55–67, February 2020.

[60] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444, Xi'an, China, May 2021. IEEE.

[61] M. Talha, E. A. M. Ghalamzan, C. Takahashi, J. Kuo, W. Ingamells, and R. Stolkin. Towards robotic decommissioning of legacy nuclear plant: Results of human-factors

experiments with tele-robotic manipulation, and a discussion of challenges and approaches for decommissioning. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 166–173, October 2016.

[62] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini Robotics: Bringing AI into the Physical World, March 2025. arXiv:2503.20020 [cs].

[63] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp Pose Detection in Point Clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, December 2017. Publisher: SAGE Publications Ltd STM.

[64] Eleftherios Triantafyllidis and Zhibin Li. Considerations and Challenges of Measuring Operator Performance in Telepresence and Teleoperation Entailing Mixed Reality Technologies, 2021. Version Number: 2.

[65] Himanshu Vishwakarma, Ananyan Sampath, Vinay Krishna Sharma, and Pradipta Biswas. Immersive Teleoperation of Collaborative Robot for Remote Vital Monitoring in Healthcare. In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, pages 210–215, August 2024. ISSN: 1944-9437.

[66] Yuhui Wan, Jingcheng Sun, Christopher Peers, Joseph Humphreys, Dimitrios Kanoulas, and Chengxu Zhou. Performance and Usability Evaluation Scheme for Mobile Manipulator Teleoperation. *IEEE Transactions on Human-Machine Systems*, 53(5):844–854, October 2023.

[67] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. GELLO: A General, Low-Cost, and Intuitive Teleoperation Framework for Robot Manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163, October 2024. ISSN: 2153-0866.

[68] Ali Yousefi, Zoe Betta, Giovanni Mottola, Carmine Tommaso Recchiuto, and Antonio Sgorbissa. Immersive control of a quadruped robot with Virtual Reality Eye-wear. In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, pages 49–55, August 2024. ISSN: 1944-9437.

[69] Andy Zelenak, Robert Reid, Adam Pettinger, and Mitch Pryor. Reactive Motion Control for Real-Time Teleoperation and Semi-Autonomous Contact Tasks. In *International Conference on Intelligent Robots and Systems. IROS*, 2019.

[70] Wei Zhao, Pengxiang Ding, Min Zhang, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. VLAS: Vision-Language-Action Model With Speech Instructions For Customized Robot Manipulation, February 2025. arXiv:2502.13508 [cs].

[71] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A Modern Library for 3D Data Processing, January 2018. arXiv:1801.09847 [cs].

[72] Yaonan Zhu, Bingheng Jiang, Qibin Chen, Tadayoshi Aoyama, and Yasuhisa Hasegawa. A Shared Control Framework for Enhanced Grasping Performance in Teleoperation. *IEEE Access*, 11:69204–69215, 2023.

APPENDICES

Appendix A

Research Ethics Approval Letter

The following letter is the approval letter for the user study ethics clearance from the University of Waterloo Research Ethics Board. It includes approval for both the study included in this thesis and the study mentioned in Section 4.3.

UNIVERSITY OF WATERLOO

Notification of Ethics Clearance to Conduct Research with Human Participants

---

Principal Investigator: Yue Hu (Mechanical and Mechatronics Engineering)

Student investigator: Joseph Kevin (Mechanical and Mechatronics Engineering)

Student investigator: Adam Sanderson (Mechanical and Mechatronics Engineering)

Collaborator: Simeon Capy (Tokyo University of Science)

Collaborator: Tomoya Sasaki (Tokyo University of Science)

Collaborator: Eiichi Yoshida (Tokyo University of Science)

File #: 46919

Title: Human-Robot Interaction via Teleoperation

The Human Research Ethics Board is pleased to inform you this study has been reviewed and given ethics clearance.

Initial Approval Date: 12/18/24 (m/d/y)

University of Waterloo Research Ethics Boards are composed in accordance with, and carry out their functions and operate in a manner consistent with, the institution's guidelines for research with human participants, the Tri-Council Policy Statement for the Ethical Conduct for Research Involving Humans (TCPS2 2022), the Ontario Personal Health Information Protection Act (PHIPA), and all laws and regulations of the province of Ontario (as applicable). Additionally, CREB operates in a manner consistent with the International Conference for Harmonization of Technical Requirements for Pharmaceuticals for Human Use (ICH) Guidance E6(R2): Good Clinical Practice , the International Organization for Standardization of Good Clinical Practices (GCP) as set out by ISO 14155 - Clinical investigation of medical devices for human subjects, Part C, Division 5 of the Food and Drug Regulations, Part 4 of the Natural Health Products Regulations, Part 3 of the Medical Devices Regulations. Both Boards are registered with the U.S. Department of Health and Human Services under the Federal Wide Assurance, FWA00021410, and IRB registration number IRB00002419 (HREB) and IRB00007409 (CREB).

Expiry Date: 12/19/25 (m/d/y)

Multi-year research must be renewed at least once every 12 months unless a more frequent review has otherwise been specified. Studies will only be renewed if the renewal report is received and approved before the expiry date. Failure to submit renewal reports will result in the investigators being notified ethics clearance has been suspended and Research Finance being notified the ethics clearance is no longer valid.

Level of review: Delegated Review

Signed on behalf of the Human Research Ethics Board



Heather Dekker, Ethics Advisor, hdekker@uwaterloo.ca, 519-888-4567, ext. 41506

This above-named study is to be conducted in accordance with the submitted application and the most recently approved versions of all supporting materials.

Documents reviewed and received ethics clearance for use in the study and/or received for information:

file: KinovaGen3_SOP_Version1_20241109.pdf

file: UFACTORYxArm_SOP_Version1_20241109.pdf

file: MetaQuest2_UserManual_Version1_20241109.pdf

file: MetaQuestPro_UserManual_Version1_20241109.pdf

file: RecruitmentPoster_Version1_20241105.pdf

file: RecruitmentEmail_Version1_20241105.pdf

file: NASATLX_Version1_20241109.pdf

file: DemographicQuestion_Version2_20241216.pdf

file: SUS_Version1_20241109.pdf

file: CustomQuestions_Version1_20241209.pdf

file: RoSAS.pdf

file: ConsentLetter_Version3_20241218.pdf

file: AppreciationLetter_Version1_20241108.pdf

Approved Protocol Version 3 in Research Ethics System

This is an official document. Retain for your files.

You are responsible for obtaining any additional institutional approvals that might be required to complete this study.