ADVANCED SHARED AUTONOMY FROM CONTROLLED ENVIRONMENTS TO THE REAL-WORLD FOR OBJECT TELEMANIPULATION

by

Michael S. Bowman

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mechanical Engineering).

Golden, Colorado
Date _____

Signed: _____
Michael Bowman

Signed: _____
Dr. Xiaoli Zhang
Thesis Advisor

Golden, Colorado
Date _____

Signed: _____
Dr. Carl Frick
Professor and Head
Department of Mechanical Engineering

ABSTRACT

Successfully teleoperating a robot for an object manipulation task is often difficult and complex due to indirect visualization, indirect manipulation with the robot, and physical discrepancies between a human and robot hand. Shared control injects autonomy into an operator's actions to overcome these issues. Even though the approaching/navigation process is one of the major components of teleoperation, the object manipulation task after approaching is essential and challenging but has not received enough research attention. Object telemanipulation requires more than simple position control, especially when considering real-world adoption. These real-world scenarios hold significant challenges: 1) human motion is ambiguous to decipher, 2) operators may solve the same task differently and there are at least 6 dimensions that they control, thus each dimension requires a different level of assistance to aid in the task's success, and 3) control policies are too strict to adapt under real-world scenarios. Standard shared control in navigation is insufficient to address these issues for telemanipulation, causing low success rates and feeling of a loss of control. Three objectives are developed to overcome these issues: 1) disambiguating the grasp motion goal with task level inference and determining the actions needed to satisfy it, 2) provide a dimension-specific shared autonomy to provide appropriate assistance for good task performance and maintain a feeling in control of the robot, and 3) provide an adaptive filter-based shared autonomy, so an operator's corrective actions are not always rejected when differing from the autonomous agent's policy. Overall, this dissertation presents novel shared control algorithms that can be used to improve a human operator's control of a robotic manipulator to achieve a telemanipulation task that would otherwise be challenging or impossible for the user on their own.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to thank my advisor and mentor, Dr. Xiaoli Zhang, for her continued support during my Ph.D. Her trust in me has allowed me to freely explore new avenues and directions within my research. Her enthusiasm has made it easy and enjoyable throughout the many research projects we have worked on together. Along with the freedom to explore new ideas and strategies, her immense knowledge helped guide me throughout this entire dissertation especially with effectively communicating the dissertation content.

I would also like to thank Dr. Andrew Petruska, Dr. Thomas Williams, and Dr. Anne Silverman, for their professional services in my thesis committee, and their invaluable comments toward strengthening my research and dissertation.

I would like to thank my labmates throughout the years, Songpo, Rui, Xu, Sen, Lingfeng, Kyle, Huiru, Neil, Noopur, Yunsik, Matthew, and Grant for the many conversations, perspectives and insights on how they view life, robotics, and interesting problems or techniques that can be adapted to a variety of fields throughout engineering.

I would like to thank my scientific mentors, Dr. Xiaoli Zhang, Dr. John Steele, and my brother Bobby. The opportunities and insights they have afforded me have been instrumental to my research career and I would not have the level success I do without them. Over the years, their advice has been invaluable on how I approach problems, tie in everything I do to the bigger picture, and taking the right steps to advance my career.

Lastly, I would like to thank my friends and family that have endlessly supported me, listened to any rambling about what I currently am working on, and have helped taken mind off of research. I appreciate you all tremendously for all the moments, laughs, and fun over the years.

This project is based on work supported by the US NSF under grants 1652454 and 2114464. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

CHAPTER 1

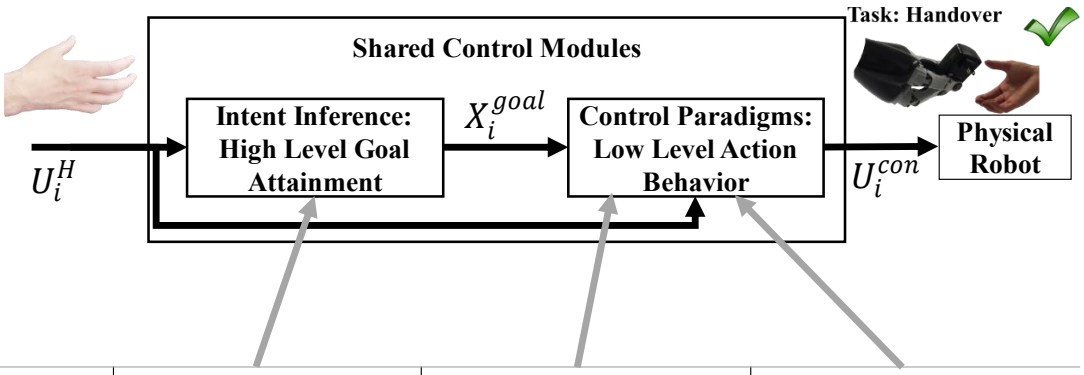
INTRODUCTION

1.1 Telemanipulation and Shared Control

Telemanipulation allows an operator to control a robot from a remote location to complete grasping tasks such as assembly tasks, tool relocation, and tool usage in telesurgery settings. The primary strategy utilized is *manual control* where an operator maintains full control of the robot motion. The state of the art technique in this style of control is Dexpilot [1]. However, successfully teleoperating a robot for an object manipulation task is often difficult and complex due to indirect visualization, indirect manipulation with the robot, and physical discrepancies between a human and robot hand [2] [3]. These challenges cause an operator to make tedious adjustments and results in longer completion times. To overcome these real-world issues, there is a juncture within the robotics field known as *shared control*, which injects autonomy (fully autonomous systems) into an operator's actions (pure teleoperation)[4]. The shared control field focuses on taking advantages of the operator (high adaptability to complete tasks, and ability to overcome uncertainty in the environment) and the autonomous agent (high precision and repeatability, optimal action sequences, and faster execution of tasks) to apply them to a single robotic manipulator [5][6]. Ideally, the tradeoff should augment and improve the operator's performance in a seamless manner. Enabling shared control is requires two main modules: an *intent inference* module and a *control paradigm* module.

The intent inference module aims to determine the goal an operator is expected to go toward, so that the autonomous agent can align with the operator and generate a better action plan. In teleoperation the inferred goal is the object of interest the operator intends to approach. Only inferring the object in a scene is not enough for telemanipulation in real-world scenarios.

Current intent inference strategies lack application task inference for an object. For instance, when grasping an object, different tasks could be used such as relocating the object somewhere else, handing it over to another person, or using the object. This leads to multiple candidate grasp solutions which could be available for a single task. Therefore, models must handle ambiguous human motion to dechiper the intent inference in aiding the autonomous agent.



Task	Chapter 2 & 3 (Intent Inference Module):	Chapter 4 (Blending Control Paradigm):	Chapter 5 (Filtered Control Paradigm):
Real World Problems to Solve	1. Ambiguous intent inference for goal generation 2. Multiple candidate goal solutions	1. Operators solve the same tasks differently, so inappropriate assistance is given to dimensions	1. control policies are too strict to adapt under uncertain real-world scenarios

Figure 1.1 Shared control is broken down into 2 main modules. The first is *intent inference* which allows the autonomous agent to determine the operator's high-level goal. The second is the *control paradigms* which focus on how to determine the low-level actions to accomplish the known/inferred goal. Each module has real world telemanipulation problems which the proposed tasks aim to solve. Intent inference in telemanipulation is ambiguous and difficult to single out a single grasp configuration for the operator's intended task. Control paradigms have two main avenues, Blending and Filtering, which have separate problems to address. Blending is an active assistance approach (one that injects autonomy directly) that requires an appropriate control authority level on each dimension to be effective. Filtering is a passive approach (restricts operator inputs) that requires an operator's acceptable workspace to be adaptive. The adaptive workspace allows the operator to make corrective actions under real-world scenarios to accomplish the task while maintaining operational safety.

A few classes of shared control paradigms exist such as, supervisory control, blended control, filter-based control and manual control, where different levels of autonomy can be seen on a spectrum ranging from full operator control to fully autonomous control (shown in Figure

1.2). The two main control paradigms are Blending and Filtering [4] and are investigated in this thesis to improve the ability for real-world adoption. Each class holds a different fundamental perspective on shared control and holds their own unique problems needed to be solved for real-world adoption. For instance, the filtered-based control is a passive strategy designed to maintain operator control as much as possible, yet current strategies are too strict adapt to uncertainties in the real-world. Filtered-based strategies have been used for navigational tasks and have not been adopted for telemanipulation. Blending is an active strategy which aims to inject autonomy to the system, but requires appropriate assistance levels to be effective. Current Blending strategies have not been introduced to telemanipulation nor are they directly transferrable as they do not account for the rotational components in the linear summation and rely on a universal arbitration term which limits how an operator may want to complete the task.

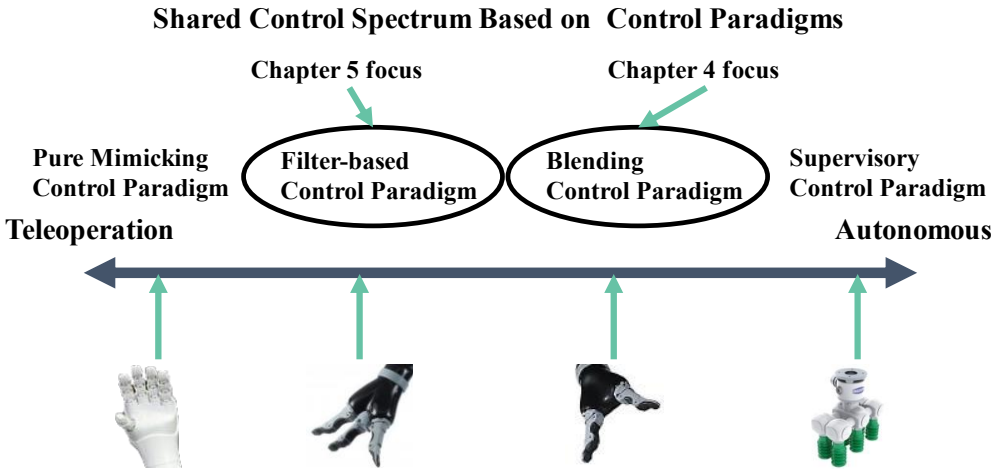


Figure 1.2 A timeline representation demonstrating different classes of control paradigms as they pertain to their general level of autonomy and assistance. The two primary classes of shared control are explored in this thesis: Blending, and Filtering. Generally, the classes follow the showen trend where Filtering strategies provide less autonomy than Blending ones. State of the art approaches in telemanipulation are manual control which purely mimic human motion.

1.2 Challenges and Real-World Uncertainties of Telemanipulation

Even though the approaching/navigation process is one of the major components of teleoperation, the object manipulation task after approaching is essential and challenging but has

not received enough research attention. Object telemanipulation requires more than simple position control, especially when considering real-world adoption. These real-world scenarios hold significant challenges: 1) human motion is ambiguous to decipher, 2) operators may solve the same task differently and there are at least 6 dimensions that they control, thus each dimension requires a different level of assistance to aid in the task's success, and 3) control policies are too strict to adapt under real-world scenarios.

These real-world challenges have been observed firsthand in preliminary works [7] [8]. Standard shared control in navigation is insufficient to address these issues for telemanipulation, causing low success rates and feeling of a loss of control. Accordingly, the following robotic needs for telemanipulation have been identified as a result in conjunction with Figure 1.1: a robot must 1) understand the intended task an operator is trying to accomplish from a high task level and develop suitable low-level actions to achieve it (task disambiguation and a resultant pose) and 2) have a shared control paradigm designed with the consideration for *misalignment* in the low-level actions between the operator and autonomous agent to improve the task performance and maintain the sense of control the operator experiences.

1.3 Limitations of Current Shared Control Strategies

1.3.1 Ambiguous Human Motion is Difficult to Decipher

One of the main barriers is that the uncertainty of intent inference could cause inappropriate robot assistance. If the robot’s modeling and planning process does not consider the uncertainty of intent inference, it can cause incorrect assistance, or task failure [9]. Uncertainty exists due to natural ambiguity of human motion. For instance, when a human grabs the body of a cup with a specific grasp pose in teleoperation, it is difficult to determine the manipulation intent for the grasp pose, although the pose itself could be for drinking for one user or for transferring the cup

to another location for another user. In addition, indirect interaction of human hands with remote objects further increase the ambiguity of human motion and increase the uncertainty in intent inference.

For those ambiguous human motions, there are indeed subtle unique differences for different tasks, which could provide better context to humans in predicting the intended action [10]. These subtle differences are critical to ensure the success of the task such as 1) a “drinking” task requires sufficient room on the top and grasping poses may dominate the handle, 2) a “transfer to another location” task requires sufficient room on the bottom to place the object safely, and 3) a “handover” task requires sufficient room for another person to grab which tends to create a finger dominated grasp. These subtleties vary between different people; however, they are still key components to discern the task. Usually, these subtle differences are difficult to observe, which makes it hard for models to infer the appropriate classification for a task, and coseqeuntly provide inappropriate assistance.

1.3.2 Inappropriate Level of Autonomy Causes Disagreement

Object telemanipulation requires more than simple positional control, realizing the system requires rotational and finger control. These motion components provide a natural split in handling complex tasks. Operators handle these components in their preferred approach. For example, the operator may obtain a position near an object, then begin to handle the rotations and end with finger control. Here an iterative process occurs by the operator to refine each step until they achieve the goal. Certain combinations of refinement will be more intuitive to different operators, like either handling the depth before lining up the robot in an x-y plane or vice versa. Naturally, operators likely only adjust a few components while holding others nearly constant (i.e., holding a position but adjusting the pose angle or finger posture) to succeed in a grasp [6].

The autonomous actions may differ from the operator actions along all or a portion of these motion components. For instance, given the same high-level target pose, the autonomous agent and operator may have differing motion strategies to achieve it. Disagreement occurs due to misperception or preference of the operator. Neither agent is incorrect in their actions to accomplish the task, nor is the robot producing suboptimal action plans. This disagreement is fundamentally inherent in shared control, and precautions are necessary to prevent over-dominance from the robotic agent. Further, the robot and human may only partially agree, where they may agree in the positional component and differ in the rotational component.

Determining the appropriate level of autonomy in teleoperation is the key research topic to realize telemanipulation scenarios fully. The most common state-of-the-art strategy in shared control is to use a linear blending strategy with a universal arbitration term. The control authority is a singular scalar value that influences every dimension of the control rather than individual components. Specifically, it only includes positional components, not rotational ones, to dictate arbitration along a trajectory. This strategy cannot handle partial agreement that arises in telemanipulation scenarios, especially towards the end of the trajectory where conventionally, the robot assumes most control. Consider a case where the robot is near the positional goal, and the operator attempts to refine the position before adjusting the rotation. With the conventional strategies, the robot will dominate not only the positional components but the rotational aspect too. This is despite the operator not yet focusing on the rotational components to grasp an object successfully and restricts their ability to alter them. The over-dominance restricts the operator in how they want to complete the task (i.e., grasp the object from the side rather than the top). This leads to an operator's attempt to refine actions but sees the control as too aggressive thus make drastic changes [11]. Over-adjustments and strong reactionary responses are due to the low

control authority. However, the operator's initial intuitive refinement is to be subtle. This is especially common for telemanipulation applications [12].

1.3.3 Current Control Policies Are Too Strict to Handle Real-World Uncertainties

The optimal actions of the autonomous agent may differ from the operator in real-world applications. The different actions are inherent to the system for two factors: 1) an operator attempting to handle unexpected uncertainties and 2) multiple ways to accomplish a shared control task. Toward the first factor, operators adjust to overcome the uncertainties in perception and the environment. For example, if the robot misses the initial approach of grasping an object by knocking it over in a teleoperation scenario, an operator will attempt to adjust the actions to correct the robot. However, the robot assumes the pose is correct and rejects these sensible operator actions. In the second factor, the robot's optimal policy in a teleoperation scenario may desire a grasp orientation over the top of the object, while an operator prefers the side to see due to visual constraints. Neither agent has an incorrect action, nor is the robot producing a suboptimal action plan. This difference is fundamentally inherent in shared control, and precautions are necessary to prevent filters from rejecting an operator's sensible actions. Thus, filtered shared control must be flexible toward the operator's corrective actions or preference, empowering them to stay in control.

Current filter-based shared control strategies usually reject actions; there is no option other than conforming to the robot policy. They strictly use a directional filter. If the operator's direction differs from the robot policy, the controller will reject the action, and the operator will feel a loss of control. It prevents an operator from 1) resetting to try again, 2) correcting the robot's action, or 3) using it in a preferred way. The consequences are the operator: 1) makes

drastic movements to no avail; and 2) must give up on their strategy as they frequently have "risky" actions rejected.

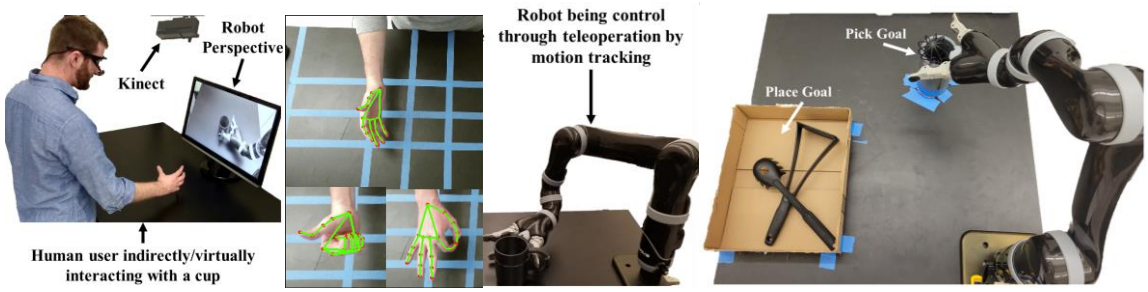
1.4 Dissertation Goal

This dissertation aims to present novel shared control algorithms that can be used to improve a human operator's control of a robotic manipulator to achieve a telemanipulation task that would otherwise be challenging or impossible for the user on their own. To this end, three objectives are developed:

1) Disambiguating the grasp motion goal with task level inference and determining the autonomous actions needed to satisfy it (Chapter 2 and 3).
2) Provide a dimension-specific shared autonomy to provide appropriate assistance for good task performance while maintaining the operator feels in control of the robot (Chapter 4).
3) Provide an adaptive filter-based shared autonomy, so an operator's corrective actions are not always rejecred when differing from the autonomous agent's policy (Chapter 5).

Toward developing these goals, a telemanipulation environment is created to act as the testing platform. Specifically, the application of interest is teleoperating a robot arm for grasping an object to achieve a specific task goal. Both the operator and autonomous agent have pose states. An RGB-D image from an Xbox Kinect tracks an operator's hand where features are extracted (see Figure 1.3.b) using MediaPipe [13]. The tracking can reliably handle open hand configurations and precision grasping motions such as a 2 finger pinch. It also provides predictive features for occluded configurations such as the "thumbs up" motion. These features are then processed into palm pose including position and orientation as well as respective velocities. Lastly, the features can also. The processed features used as inputs for the robot

control of a Kinova MICO arm (Figure 1.3.c and Figure 1.3.d). The operator controls the hand pose velocity (position and orientation) and how much each finger opens and closes. In all, the human directly controls eight variables (position, orientation, and two-finger control variables).



a) b) c) d)

Figure 1.3 a) Operators see the robot perspective on the screen while they move their hand freely over the table. b) The Xbox Kinect captures hand posture to extract the palm pose and the amount each finger is open or closed. The information is then sent to the robot to execute the motion. c) The robot scene for Objective 1 (Chapter 3) is to grasp a singular object (cup) for multiple tasks. d) The robot scene displayed is for Objective 2 and Objective 3 where a simulated pick and place Transfer task is used.

Two telemanipulation scenarios are defined. The first is for the Intent Inference Module (Chapter 2 and 3) to allow the operator to grasp a cup for different tasks(Figure 1.3.c). The singular object multi-task scenario allows for effective evaluation of the Intent Inference Module because it provides easier rule-based evaluation to determine task success rather than grasp success. The different preferred ways an operator may want to perform a given task will provide different intent inferences and thus different grasp solutions.

The second scenario aims to evaluate the Control Paradigm Modules (Chapters 3 and 4). The operator aims to grasp a regular kitchen utensil from a holder and place it in a bin (Figure 1.3.d). This scenario aims to recreate an open pick and place scenario as the operator is free to move about to accomplish the task however they wish. The robot is given a goal position for both the pick and place locations with induced noise to simulate a realistic scenario.

1.5 References

[1] Handa, K. V. Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff and D. Fox, "DexPilot: Vision-Based Teleoperation of Dexterous Robotic Hand-Arm System," in IEEE Internation Conference on Robotics and Automation (ICRA), Paris, France, 2020.

[2] Y. Rybarczyk, E. Colle and P. Hoppenot, "Contribution of Neuroscience to the Teleoperation of the Rehabilitation Robot," in IEEE International Conference on Systems, Man and Cybernetics, 2002.

[3] N. Healey, "Speculation on the Neuropsychology of Teleoperation: Implications for Presence Research and Minimally Invasive Surgery," Presence, vol. 17, no. 2, pp. 199-211, 2008.

[4] A. Erdogan, B.D. Argall, “The effect of robotic wheelchair control paradigm and interface on user performance, effort and preference: An experimental assessment,” Robotics and Autonomous Systems, Volume 94, 2017, Pages 282-297, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2017.04.013.

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

[6] A. Dragan and S. S. Srinivasa, “A policy blending formalism for shared control,” Int. J. Robot. Res., vol. 32, 2013, pp. 790–805.

[7] S. Li, M. Bowman, H. Nobarani and X. Zhang, "Inference of Manipulation Intent in Teleoperation for Robotic Assistance," Journal of Intelligent & Robotic Systems., 2020.

[8] S. Li, M. Bowman and X. Zhang, "A General Arbitration Model for Robust Human-Robot Shared Control with Multi-Source Uncertainty Modeling," in arXiv:2003.05097, 2020.

[9] H. Zhang, C. Reardon, F. Han, and L. E. Parker, “SRAC: Self-Reflective Risk-Aware Artificial Cognitive models for robot response to human activities,” 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016.

[10] John Lin, Ying Wu and T. S. Huang, "Modeling the constraints of human hand motion," Proceedings Workshop on Human Motion, Austin, Texas, USA, 2000, pp. 121-126.

[11] Y. Oh, M. Toussaint, and J. Mainprice, “Learning to arbitrate human and robot control using disagreement between sub-policies,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2021, pp. 5305–5311.

[12] D. Song, C. H. Ek, K. Huebner, and D. Kragic, “Task based robot grasp planning using probabilistic inference,” IEEE Trans. Robot., vol. 31, no. 3, pp. 546–561, Jun. 2015.

[13] Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg and M. Grundmann, "Mediapipe: A framework for perceiving and processing reality," in Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR), 2019.

CHAPTER 2

INTENT-UNCERTAINTY-AWARE GRASP PLANNING FOR ROBUST ROBOT ASSISTANCE IN TELEMANIPULATION

A paper published on *IEEE International Conference on Robotics and Automation 2019*[1]

Michael Bowman[2], Songpo Li[3], and Xiaoli Zhang[4]

2.1 Abstract

Promoting a robot agent's autonomy level, which allows it to understand the human operator's intent and provide motion assistance to achieve it, has demonstrated great advantages to the operator's intent in teleoperation. However, the research has been limited to the target approaching process. We advance the shared control technique one step further to deal with the more challenging object manipulation task. Appropriately manipulating an object is challenging as it requires fine motion constraints for a certain manipulation task. Although these motion constraints are critical for task success, they are subtle to observe from ambiguous human motion. The disembodiment problem and physical discrepancy between the human and robot hands bring additional uncertainty, make the object manipulation task more challenging. Moreover, there is a lack of modeling and planning techniques that can effectively combine the human motion input and robot agent's motion input while accounting for the ambiguity of the human intent. To overcome this challenge, we built a multi-task robot grasping model and

[1] Reprinted with permission from *IEEE International Conference on Robotics and Automation* 2019. © IEEE.

[2] Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines, 1610 Illinois Street, Golden, Colorado, 80401, USA.

[3] Co-author, Postdoctor in the Department of Mechanical Engineering and Material Science at Duke University, Durham, NC 27705 USA.

[4] Corresponding author, Associate Professor, Department of Mechanical Engineering, Colorado School of Mines, 1610 Illinois Street, Golden, Colorado, 80401, USA.

developed an intent-uncertainty-aware grasp planner to generate robust grasp poses given the ambiguous human intent inference inputs. With this validated modeling and planning techniques, it is expected to extend teleoperated robots' functionality and adoption in practical telemanipulation scenarios.

2.2 Introduction

2.2.1 Need of Robot Assistance in Telemanipulation

Teleoperating a robot allows operators to carry out tasks remotely with the robot as a medium. This indirect interaction brings in many advantages including increased motion precision and strength, and remote access to work fields that might be inaccessible or hazardous to the operator. However, successfully teleoperating the robot for a task is often difficult and complex due to indirect manipulation and physical discrepancy between a human hand and robot hand[1][2]. To reduce the control difficulty of teleoperated robots and the operation workload of the operators, the robot agent is being designed to have more intelligence and autonomy, thus, to understand the operator's intent and assist in achieving it. Research has demonstrated that in a target approaching process, the robot agent can infer the target location by observing the operator's motion trajectory and provide motion assistance in approaching the target [3][4]. Much research has been documented on how to precisely infer the target location and how to effectively blend the human input trajectory and robot agent's input.

Even though the approaching process is one of the major components that consist of the teleoperation, the object manipulation task after approaching is essential and challenging, but has not received enough research attention. Successfully manipulating an object requires fine motions (i.e., motion constraints for task success [17]), such as approaching the object in a specific angle, grasping the object at a particular part, and applying the force in a certain manner

[18]. To satisfy the certain constraints for a task, the operator has to use their own mental and physical capability to adjust the robot hand, beyond suffering from the general disembodiment problem and physical discrepancy of teleoperation. Thus, there is a great need of technologies that can enable the robot agents to proactively assist the operator in successfully tele-manipulating objects for various tasks.

2.2.2 Influence of Uncertainty in Robot-Assisted Telemanipulation

One of the main barriers in intent-based robot assistance is the uncertainty of intent inference could cause inappropriate robot assistance. If the robot's modeling and planning process does not consider the uncertainty of intent inference, it can cause incorrect assistance, or task failure [9]. Uncertainty exists due to natural ambiguity of human motion. For instance, when a human grabs the body of a cup with a specific grasp pose in teleoperation, it is difficult to determine the manipulation intent for the grasp pose, although the pose itself could be for drinking for one user or for transferring the cup to another location for another user. In addition, indirect interaction of human hands with remote objects further increase the ambiguity of human motion and increase the uncertainty in intent inference.

For those ambiguous human motions, there are indeed subtle unique differences for different tasks, which could provide better context to humans in predicting the intended action [11]. These subtle differences are critical to ensure the success of the task such as 1) a "drinking" task requires sufficient room on the top and grasping poses may dominate the handle, 2) a "transfer to another location" task requires sufficient room on the bottom to place the object safely, and 3) a "handover" task requires sufficient room for another person to grab which tends to create a finger dominated grasp. These subtleties vary between different people; however, they are still key components to discern the task. Usually, these subtle differences are difficult to observe,

which makes it hard for models to infer the appropriate classification for a task, and coseqeuntly provide inappropriate assistance.

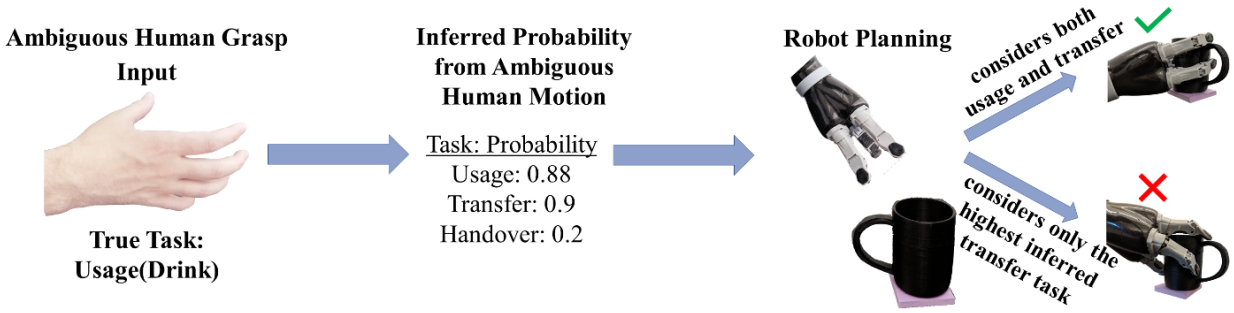


Figure 2.1 Intent interpretation from an ambiguous grasp lead to different solutions based upon the planning procedure. If the ambiguity in the intent is not considered it can lead to unsuccessful grasp configurations which result in the failure of the task. Considering ambiguity allows for subtle adjustments to be made to the grasp configuration to accommodate the task requirements which are close in intent and ensure the robustness of motion planning. Interpretation of ambiguous inputs, and uncertainty in model outputs is crucial for the planning procedures success.

To solve these open issues, an intent-uncertainty-aware grasp planning method is developed for robot-assisted telemanipulation. This approach enables robot-assisted telemanipulation in which the robot infers the human operator's manipulation intent and provides grasping motion assistance to achieve the operator's intended task. The contributions of this work are two-fold:

1) Multi-Task Grasp Modeling. The modeling considers the ambiguity of human motion. Instead of building independent task models, some grasp poses share common features which result in satisfying multiple tasks. The models, therefore, contain overlap between features and poses. Additionally, non-overlapping areas are of special interest since they carry fundamentally different features from other tasks. Although these fundamental differences are often subtle, they are of critical need for the robust grasp planning model. Therefore, we developed a multi-task grasp modeling method to allow robots to understand the common features as well as subtle unique features which distinguish tasks.

2) Intent-Uncertainty-Aware Grasp Planning. We developed an intent-uncertainty-aware planner to handle the uncertainty in human intent inference. Given the intent inference probability input, the planner first interprets the ambiguity level of human motion, then generates the grasp based on this ambiguity level. Highly ambiguous input may need a grasp pose that compromises features from all possible tasks. Input with low ambiguity generate a grasp pose that emphasize features for the higher inferred task.

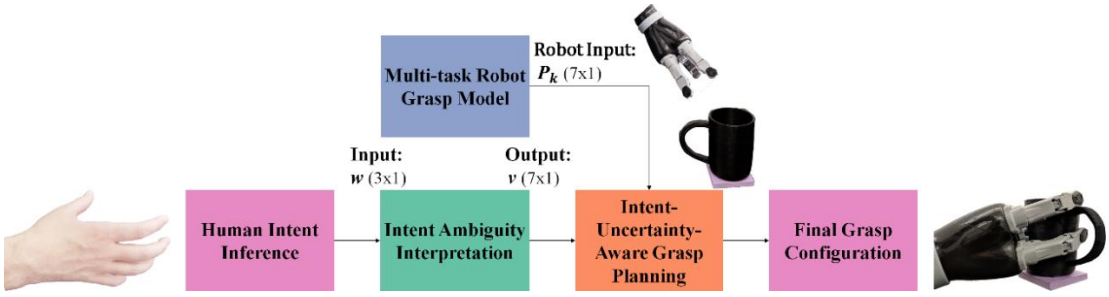


Figure 2.2 This overview demonstrates the framework used to achieve an intent-uncertainty-aware grasp planning. The initial human grasp creates a probability for each principle task, w. To account for ambiguity the intersection of each principle task is considered, v. The robot model will also account for the intersection of tasks by using common poses. Both of these produce probability vectors which are used for the intent-uncertainty-aware grasp planning. A final grasp pose is then obtained from the planning process.

2.3 Methods

2.3.1 Framework Overview

The overall framework process to handle this can be seen in Fig. 2.2. The three main components discussed in this section will be the multi-task robot model, the human intent ambiguity interpreter, and the intent-uncertainty-aware grasp planner. The multi-task robot model will consider the overlapping nature of different tasks due to common motion features shared by their grasp poses and allows the robot to understand the fundamental different features of these tasks. Grasping objects is ambiguous without a clear discrete difference between satisfying specific tasks because of the inherent nature of the manipulation problem where this

varies between users. So, the interpretation of human intent inference provides a descriptor for the ambiguity level among the tasks the robot can satisfy. The planner will attempt to find a robot pose which will pull characteristics from different tasks depending upon the ambiguity levels among tasks.

2.3.2 Multi-Task Robot Modeling

Table 2.1 Subtle Feature Comparison of Different Common Poses Within a Multi-Task Model

Task	**Usage**	**Transfer**	**Handover**	**Usage and Transfer**	**Usage and Handover**	**Transfer and Handover**	**All**
Top View **Left View**							
Subtle Detail	More fingers to dominate handle space. Lower palm to leave sufficient room on top to drink.	Prefers to be perpendicular to table and higher on cup to ensure space on bottom to place cup.	Lower palm, leaves room for another user to grasp handle (180˚ rotation of wrist).	Fingers dominate the handle. Palm position allows room to drink and place cup.	Few fingers in handle space but occupies more of it. Lower palm to leave room to drink.	Few fingers in the handle space for user. Higher palm to ensure space on bottom.	Few fingers in the handle space for user. Enough space to drink and place.

Unlike more traditional models the model structure we propose uses common poses in conjunction with probability distribution. Traditional Bayesian structures [5][6], as shown on the left in Fig. 2.3, do not consider ambiguity[12] of task distinctions, where our method on the right shows the consideration for overlapping tasks. Current robot models create task models separately from one another—to create clear, independent distinctions between tasks—and lack consideration of common poses. This causes difficulties to find continuous features between poses. Common poses have a unique ability to satisfy multiple tasks, however, they may be sub-optimal to satisfying a single principle task and less transparent to independent models.

However, traditional Bayesian structures can obtain probability rather easily [10]. To ensure the probability obtained for our model is correct it is imperative the training data is labeled appropriately and reflects the right side of Fig. 2.3.

2.3.3 Inclusive Task Labeling

Common poses do add a degree of ambiguity compared to those of independent model strategies. Although, our approach takes into consideration the ambiguity by creating a multi-class Bayesian structure [13][14]. Our model considers this by giving an inclusive aspect where poses which satisfy more specific situations should be included in the more general cases. For instance, in Fig. 2.3, all tasks within the Task 1 circle should be considered within the Task 1 class. The example model created was for grasping a cup—as shown in Table 2.1 in conjunction with Fig. 2.3—for principle tasks which include: task 1, usage or drinking from a cup, task 2, transfer of a cup to another location, and task 3, handover of the cup to a person.

Compared with traditional grasp modeling techniques, the classes and labeling of our modeling method will not be the three principle tasks, exclusively, rather they will contain seven classes in which the robot can satisfy. To label these classes correctly, the assumption made is a robot pose which satisfies more principle tasks than necessary is admissible to a more general class because it satisfies the necessary principle task. By creating an inclusive class structure, the model can begin to understand the subtle and unique differences among tasks. For instance, when looking at Table 2.1, we observe a group labeled for the class “Usage and Transfer”. Poses falling under this label satisfy not only the class “Usage and Transfer”, but also “Usage” and “Transfer”, thus the pose should be considered into the training sets for “Usage” and “Transfer” tasks. This duplication, or information sharing, of training sets allows the more inclusive class (such as Task 1, Task 2, and Task 3) to identify the distinctions between unique and common

features. Table 2.2 shows how to appropriately label and consider the training set for the model created by the right of Fig. 2.3 in conjunction with Table I. The letters U, T, and H stands for the principle tasks, “Usage”, “Transfer”, and “Handover” respectively. The columns represent each class the robot model knows, while the rows are the strict subset of events which make up each class. A pose which is used for “Usage and Transfer only” (row 4), can satisfy both “Usage”(column 1) and “Transfer”(column 2) alone as well as “Usage and Transfer”(column 4). Additionally, for instance, all poses which could be used to satisfy a “Usage” task (column 1) include, “Usage only”(row 1), “Usage and Transfer only”(row 4), “Usage and Handover only”(row 5), and “All Tasks”(row 7).

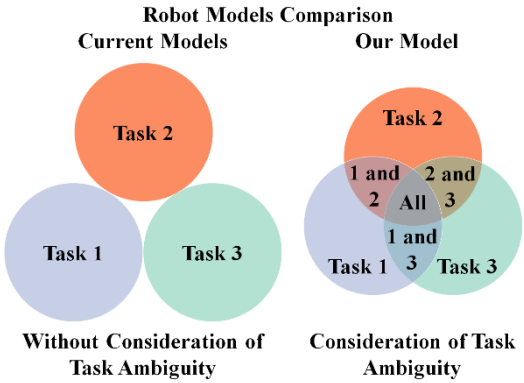


Figure 2.3 Robot models need to consider task ambiguity. Current models(left) do not consider ambiguity or where multiple tasks may be satisfied simultaneously. An alternative model(right) considers tasks being concurrently fulfilled by common poses.

Table 2.2 Inclusive Labeling For Principle Tasks With Consideration of Common Poses

H	T	U	Class k	1: U	2: T	3: H	4: U∩T	5: U∩H	6: T∩H	7: U∩T∩H
0	0	1		1	0	0	0	0	0	0
0	1	0		0	1	0	0	0	0	0
1	0	0		0	0	1	0	0	0	0
0	1	1		1	1	0	1	0	0	0
1	0	1		1	0	1	0	1	0	0
1	1	0		0	1	1	0	0	1	0
1	1	1		1	1	1	1	1	1	1

Principle Tasks: U = Usage, T = Transfer, H = Handover

When observing Table 2.1, we notice between the classes the pose features are subtle, yet they are critical. By labeling the training poses in this manner, we train the model to identify subtle differences and common poses for the principle tasks as well as unique specific cases within the principle tasks. Subtle features can be difficult to observe and determine, however, they provide better context to a human user. Table 2.1 includes the subtle features used for our model including high level concepts such as palm contact, clearance from the top of the cup, and sufficient room for a user to grasp. These subtleties are further discussed in the table.

2.3.4 Multi-Task Robot Model

The goal of the Bayesian model is to obtain the posterior probability, or the probability of each class given a continuous pose[15][16]. Each pose contains continuous features, *x*, which can include position, orientation, force, and object features. Our model included palm center, and palm orientation as well as finger force. The model is a Naïve Bayes classifier (NB), but uses the features to create a multivariate normal distribution as shown in (2.1). The key parameters from the model for each classification zone, *k*, include: the mean value of each feature, μ_k, the covariance matrix, $\sum_k$, and the prior probability, P(k). These parameters are learned from the training set by using the Expectation Maximization Algorithm. In (2.1), it shows the conditional probability of *x* given k, where *d* is the length of *x*.

$$P(\mathbf{x}|k) = \frac{1}{\sqrt{\det(\sum_k)*(2\pi)^d}} e^{-\frac{1}{2}*(\mathbf{x}-\mu_k)^T*\sum_k{}^{-1}*(\mathbf{x}-\mu_k)} \tag{2.1}$$

With this equation, Bayesian models obtain the posterior probability of the class *k* given *x* with (2.2), where the class with the highest probability is the label for *x*.

$$P(k|\mathbf{x}) = \frac{P(\mathbf{x}|k)*P(k)}{\sum_k^K P(\mathbf{x}|k)*P(k)} \tag{2.2}$$

Each probability of a class is then put into what is referred to as the robot probability vector, Pk. The model we created from Fig. 2.3 and Tables 2.1 and 2.2, mean there are seven classes and the probability for one pose to satisfy all seven classes must sum to one.

2.3.5 Intent Ambiguity Interpretation

To handle the ambiguity, the modeling of human manipulation intent follows a concept of multi-label classification, where two, three, or *m* principle task classification model outputs could be satisfied at the same time, shown in Fig. 2.1. Each classification model produces a probability and is put into a vector of mx1 size, where this is referred as the *classification input vector, w*. Since each model output is independently obtained, we can identify each joint probability case. This will generate a vector of size 2^m where *m* is the number of principle tasks. This vector of possible events with their subsequent probability will be referred to as the *human probability vector, u*, shown in (2.3), where $\psi(\mathrm{m})$ is the power set from 1 to *m* task classification models, and Y is each subset of $\psi(\mathrm{m})$.

$$P\left(\bigcap_{i\in Y} w_i \bigcap_{j\notin Y, j\in m} \neg w_j\right) = \prod_{i\in Y} P(w_i) \prod_{j\notin Y, j\in m} 1 - P(w_j), Y \subset \psi(m) \tag{2.3}$$

For example, consider when *m*=3 as shown in Fig. 2.1. The classification probability vector (0.88,0.9,0.2) produces a vector where each combination of the principle tasks is considered true and false. So when event U∩¬T∩¬H occurs, the probability is 0.88*(1-0.9)*(1-0.2) = 0.07, while when the event U∩T∩¬H occurs, the probability is 0.88*0.9*(1-0.2)=0.63.

Since the robot model is unaware of the event where inaction should be taken (¬U∩¬T∩¬H), it is important to eliminate this action and normalize the rest of the probability vector where at least one principle task *m* is satisfied. This is to ensure the robot classes, k, is equal to 2^m-1. This new vector is known as the *target probability vector, v,* and is shown in (2.4).

The target probability vector now has a scenario associated with each class of the robot model previously discussed in Table 2.1 and 2.2.

$$v_i = \frac{u_i}{\sum_j^{2^{m-1}} |u_j|} \; \forall \; i < 2^m \tag{2.4}$$

Alternatively, a designer could force the robot to ask for clarification if the inaction task is sufficiently high. The human probability vector has a distinct advantage over using the classification intent by allowing more descriptive behavior of the type of features the robot model should use. For instance, Fig. 2.1 shows if the classification intents for grasping a cup for Usage and Transfer are similar and high while the Handover task is low, then the decision process should carry features which demonstrate and use features from both higher intents. Additionally, the human probability vector inherently accounts for uncertainty in human intent by allowing for the system to consider multiple classes at once as well as the varying degree the multiple classes may influence the outcome of chosen features.

2.3.6 Intent-Uncertainty-Aware Grasp Planning

The planning process involves taking the ambiguity levels developed in the intent interpretation section and using them as a basis for the determining characteristics needed. Whether the ambiguity levels are high or low will determine which set of features should predominantly be used from the robot model classes developed earlier. The planning process will start from a known pose, which is closest to the ambiguity levels, then continue to refine the features of the pose until a more reflective pose is created. The refining step is iterative and will attempt to use both unique and subtle characteristics from all classes as it needs until convergence.

The planning process includes using a Bayesian structure to best fit the target probability vector. Traditional Bayesian models attempt to maximize the probability of one region or class

over all others, since these models are based on independent model strategies, rather than minimize the difference between the current and target probability vectors. By matching the probability vectors, the subtle and unique features of the robot model should reflect the best combination of intent to make it predictable to the user. The first step is to identify the grasp class out of the defined 7 and select an initial grasp pose from the training set of this class, since this is currently the best approximation the robot model can make. This best approximation is done by taking the highest class of the human probability vector and choosing the best pose which is reflected in the class of the robot model. Determining the best pose within the class can be done by using the objective function as shown in (2.5). The objective function is a least square regression which will attempt to match the robot probability vector to the target probability vector.

$$min \frac{1}{2} \sum_k (v_k - P_k)^2 \tag{2.5}$$

By mimicking the probability, the unique and subtle features of the robot model will reflect those most similarly to the human intent. This is due to the robot model classes containing both the unique and subtle features which the planner takes characteristics. The next step is to refine this initial approximation and determine how the features need to be adjusted. To determine the amount of adjustments the planner will take, from a unique feature to a more common feature or vice versa, a gradient descent method was applied as shown in (2.6).

$$\frac{\partial P_k}{\partial x}(v_k - P_k) \; \forall k \tag{2.6}$$

This can be solved using finite-difference methods [7] or analytically using matrix calculus [8]. This approach will use the ambiguity levels as a guide to determine which subtle or unique features are more critical, or influential in ensuring a successful grasp pose. Afterwards, constraint equations will be applied—which are robot and case dependent based upon features

selected thus will be left to the reader's discrepancy—and the objective function will be evaluated once again. The overall algorithm can be seen in Algorithm 2.1. All together these pieces will provide the best features to match the human intent.

Algorithm 2.1: Intent-Uncertainty-Aware Grasp Planning
Input: Multi-task Robot Model *NB*, Classification Intent Vector $\boldsymbol{w}$
Output: $\boldsymbol{x}$, $\boldsymbol{P_k}$
1: Evaluate $\boldsymbol{w}$ to find ambiguity levels, $\boldsymbol{v}$
2: Obtain starting pose $\boldsymbol{x}$, by choosing closest to known level to $\boldsymbol{v}$ from *NB*
3: **while** iter < iteration_max or obj < tolerance **do**
4: $\boldsymbol{P_k} = NB(\boldsymbol{x})$
5: obj = $\frac{1}{2}\sum_k(\boldsymbol{v}_k - \boldsymbol{P_k})^2$
6: **dobj** = $\frac{\partial \boldsymbol{P_k}}{\partial x}(\boldsymbol{v_k} - \boldsymbol{P_k})$
7: $\boldsymbol{x}$ = updateVariables($\boldsymbol{x}$,*BN*,*obj*,***dobj***)
8: $iter \leftarrow iter + 1$
9: **end while**
10: Return $\boldsymbol{x}$, $\boldsymbol{P_k}$

2.4 Experiment and Results

2.4.1 Experimental Setup

The experimental setup included using a Kinova Mico arm as well as a mug for the object of interaction. The robot model was gathered for three tasks: drinking/using the cup, handing the cup over to a person, and transferring the cup to another location. The robot model only considered features for the final grasp configuration and did not include trajectory or temporal features, however, the force in which the fingers applied was considered. The robot model was created by developing expert analysis of rules to determine which task could be satisfied. The rules developed express extreme cases where there are certain cases of no overlap between principle tasks. The robot was then manually moved to obtain the training data. Additionally, the optimization algorithm used was fmincon [7] available in MATLAB, however, other optimization algorithms may be used to achieve quicker convergence. The classification intent

was simulated for the experiment to represent where one task is dominant, two tasks are codominant, and all three tasks are equal. Two different types of analysis were done to verify the method. The first was an objective result by comparing the initial intent inference and probability distribution with those created by the final pose configuration. The other analysis was the subjectivity and predictability of the pose by a human user.

2.4.2 Objective Results

Due to the planning framework, the resulting poses and solutions are deterministic. The results shown are for three separate cases which include: a single task where the target intent is for usage, a task which holds two likely intents for usage and transfer, and lastly a task where the intent for all tasks are equally likely. Table 2.3 shows the comparison of the target probability vector, *v*, and robot probability vector, P_k. These results show when the individual task is the target task it primarily takes features from this population, while mostly disregarding other populations. This coincides to how independent model strategies exist. It is shown the more ambiguity, or the closer the task intents are to one another, the better the optimization can handle pulling features from the separate populations to achieve a solution. We can also observe how close the final intent inference of the three principle tasks we get to compared to the initial intent inference, *w*, in Table 2.4 below.

Table 2.3 Posterior Probability Vector Comparison

H	T	U	Single Task Target	Single Task Final	Two Task Target	Two Task Final	Three Task Target	Three Task Final
0	**0**	**1**	**0.7933**	**0.8156**	0.0817	0.0998	0.0090	0.0143
0	**1**	**0**	0.0098	0.0336	0.0817	0.1003	0.0090	0.0102
1	**0**	**0**	0.0098	0.0347	0.0010	0.0197	0.0090	0.0044
0	**1**	**1**	0.0881	0.1108	**0.7356**	**0.7533**	0.0811	0.0801
1	**0**	**1**	0.0881	0.0000	0.0091	0.0000	0.0811	0.0809
1	**1**	**0**	0.0011	0.0053	0.0091	0.0269	0.0811	0.0805
1	**1**	**1**	0.0098	0.0000	0.0817	0.0000	**0.7297**	**0.7295**

Table 2.4 shows the comparison of the initial target intent and the final intent given by the final pose. This shows the more ambiguous the initial intent the better the pose can satisfy it. In addition to checking the objective posterior probability and intent of the poses, it is also imperative to observe the grasp pose to determine if it could satisfy the given tasks.

Table 2.4 Principle Intent Inference Comparison

Initial Intent	Single Task Target	Single Task Final	Two Task Target	Two Task Final	Three Task Target	Three Task Final
Usage	0.9	0.9265	0.9	0.8531	0.9	0.9049
Transfer	0.1	0.1497	0.9	0.8805	0.9	0.9004
Handover	0.1	0.04	0.1	0.0465	0.9	0.8953

2.4.3 Subjective Results

For the subjective analysis, there are two main properties to identify. The first, is to determine if poses can satisfy true tasks, while the second is observe the subtle features of the final pose configurations. To analyze both properties, observe Fig. 2.4. It shows the comparison between two ambiguous intents between the transfer and usage task. This figure shows the issues where independent models may fail or possibly be lucky to succeed. The intent for grasp pose *b)* is (0.9,0.9,0.1), so it is difficult to determine which to task to follow. For instance, if the true task is usage and the independent model method incorrectly infers the task is transfer, it will produce the grasp pose *a)* shown on the left. This grasp has subtle differences which fails for usage, such as a finger covering the top of the cup and leaving insufficient room to drink. While grasp *c)* appears to be an optimal grasp for drinking a cup, although it is not optimal for transferring the object because it lacks palm contact with the cup and is a finger dominated grab which may not be a stable hand configuration for a transfer task. Grasp *b)* takes features from both usage and transfer populations, which generate poses *a)* and *c),* to produce a pose which is better equipped to handle both tasks. The palm is close to the cup, which can satisfy a stable grasp, while also

leaving sufficient room on the top to drink from. The subtle differences allow a change in the finger placement, sufficient room to drink, and palm contact for a more stable grasp. These subtle differences of features within a pose allow for a smoother transition of tasks which reduce the risk for inferring the wrong intent. Humans are keen on picking up these subtle differences, so it is critical to account for the vagueness of intent and create models which are more flexible towards it.

Figure 2.4 There are three separate grasp poses a) represents the pose which satisfies the transfer task and not the other two tasks as seen by the intent provided on the top. This grasp dominates the top of the cup to leave room to place and it is unsuitable for usage. b) is the pose which our model generated which carries the subtleties to satisfy both usage and transfer while not satisfying handover. The robot hand dominates the handle by having a majority of fingers around it, while leaving sufficient room to drink from the top. The palm is higher on the cup. c) is the pose for which satisfies the usage task. The fingers dominate the handle space by having a majority of fingers near the handle, sufficient room on the top for drinking.

Additionally, human predictability was also an important factor to consider when analyzing the poses the robot distinguished. The human expert was to identify 50 poses generated by the model and correctly label the pose for which tasks it could satisfy. The expert was able to distinguish 80% of the poses as the correctly labeled ones. The human subject often appeared to believe the pose could satisfy more tasks than it could which may be a result of the user seeing features from the other minor tasks which the optimization may have incorporated as part of the posterior probability vector. This shows the importance of creating continuous models since it is difficult to determine the vague intent presented by grasp configurations.

2.5 Discussion

The ambiguity of distinctions between the different grasp configurations requires robot models to become continuous rather than discrete models. The methods provided demonstrate an effective way to consider and convert the robot models into a more appropriate framework. It is important to understand there are optimal poses which can satisfy a specific task, and there are also sub optimal poses which can satisfy all tasks. The goal of this work is to find sub optimal poses which are better solutions than a default preplanned grasps which can satisfy all tasks by transforming the problem into a continuous system. This is done by interpreting the human intent as independent and transforming the probability to a more descriptive representation. The results also show the subtle feature changes can have an impact in influencing predictability. Obtaining human intent and creating effective human inferencing models creates uncertainty which needs to be accounted for when having a robot system make decisions. This makes it difficult to confidently use independent model strategies compared to probability distribution strategies. All together this system accounts for uncertainty by creating a continuous Bayesian structure from which the final intent distribution can closely mimic the human intent.

2.6 Conclusion

The overall method proves effective due its practicability compared to other modeling methods. However, it can still be improved before being used for true telemanipulation. The evaluation for different levels of overlap between principle tasks is a necessary next step. It is also critical to analyze models for other objects. Lastly, more human subjects are needed to further validate the predictability.

2.7 References

[1] Y. Rybarczyk, E. Colle, and P. Hoppenot, “Contribution of neuroscience to the teleoperation of rehabilitation robot,” in Systems, Man and Cybernetics, 2002 IEEE International Conference on, vol. 4. IEEE, 2002, pp. 6–pp.

[2] A. N. Healey, “Speculation on the neuropsychology of teleoperation: implications for presence research and minimally invasive surgery,” Presence, vol. 17, no. 2, pp. 199–211, 2008.

[3] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, “Continuous role adaptation for human–robot shared control,” IEEE Transactions on Robotics, vol. 31, no. 3, pp. 672–681, 2015.

[4] J. D. Webb, S. Li, and X. Zhang, “Using visuomotor tendencies to increase control performance in teleoperation,” in American Control Conference (ACC), 2016. IEEE, 2016, pp. 7110–7116.

[5] D. Song, C. H. Ek, K. Huebner and D. Kragic, "Multivariate discretization for Bayesian Network structure learning in robot grasping," 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 1944-1950.

[6] D. Song et al., “Predicting human intention in visual observations of hand/object interactions.” 2013 IEEE International Conference on Robotics and Automation, Karlsruhe,2013, pp. 1608-1615

[7] Waltz, R. A., J. L. Morales, J. Nocedal, and D. Orban. “An interior algorithm for nonlinear optimization that combines line search and trust region steps.” Mathematical Programming, Vol 107, No. 3, 2006, pp. 391–408.

[8] K. Petersen and M. Pedersen, “The Matrix Cookbook,” November, 2012.

[9] H. Zhang, C. Reardon, F. Han, and L. E. Parker, “SRAC: Self-Reflective Risk-Aware Artificial Cognitive models for robot response to human activities,” 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016.

[10] Cheng, Jie, Greiner, and Russell, “Comparing Bayesian Network Classifiers,” 23-Jan-2013. [Online]. Available: https://arxiv.org/abs/1301.6684.

[11] John Lin, Ying Wu and T. S. Huang, "Modeling the constraints of human hand motion," Proceedings Workshop on Human Motion, Austin, Texas, USA, 2000, pp. 121-126.

[12] Z. Zhou and M. Zhang, "Solving multi-instance problems with classifier ensemble based on constructive clustering", Knowledge and Information Systems, vol. 11, no. 2, pp. 155-170, 2006.

[13] FA Tzima, M Allamanis, A Filotheou, PA Mitkas , Inducing Generalized Multi-Label Rules with Learning Classifier Systems,- arXiv preprint arXiv:1512.07982, 2015

[14] Zhang, M.-L. and Zhang, K. (2010). Multi-label learning by exploiting label dependency. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 999-1008, New York, NY, USA. ACM.

[15] John, George H., and Pat Langley. "Estimating continuous distributions in Bayesian classifiers." Proceedings of the Eleventh conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1995.

[16] Heckerman D. A Tutorial on Learning with Bayesian Networks. In: Holmes D.E., Jain L.C. (eds) Innovations in Bayesian Networks. Studies in Computational Intelligence, vol 156. Springer, Berlin, Heidelberg (2008)

[17] D Song, K Huebner, V Kyrki, D Kragic, "Learning task constraints for robot grasping using graphical models", IROS 2010

[18] Song, Dan, et al. "Task-based robot grasp planning using probabilistic inference." IEEE transactions on robotics 31.3 (2015): 546-561.

CHAPTER 3

INTENT-BASED TASK ORIENTED SHARED CONTROL FOR INTUITIVE TELEMANIPULATION

Under review to Journal of Intelligent And Robotic Systems (JINT*)* 2023

Michael Bowman[5], Jiucai Zhang[6], and Xiaoli Zhang[7]

3.1 Abstract

Shared control in telemanipulation is a promising yet challenging problem. The unique challenges—beyond teleoperation—include physical discrepancy between human and robot hands and the fine manipulation constraints to constitute task success. We present an intuitive shared-control strategy that generates robotic grasp poses better suited for human perception of success and feeling of control while ensuring a stable grasp for task success. The former is achieved by having user inputs as motion constraints. The latter is achieved by understanding human intent and acting on that inference. The robot's motion follows an arbitration between following the user's motion constraints and accomplishing the inferred task. The arbitration adapts based on the physical discrepancy between the human and robot hands. We have conducted a user study with a telemanipulation scenario to analyze the effects of task predictability, following, and user preference. The results demonstrated that intent-based approaches provide advantages over direct motion mapping strategies.

[5] Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines, 1610 Illinois Street, Golden, Colorado, 80401, USA.
[6] Secondary researcher is with the GAC R&D Center Silicon Valley, Sunnyvale, CA 94085 USA
[7] Corresponding author, Associate Professor, Department of Mechanical Engineering, Colorado School of Mines, 1610 Illinois Street, Golden, Colorado, 80401, USA.

3.2 Introduction

3.2.1 Insufficiencies and Challenges of Telemanipulation

Although teleoperation has great potential to complete tasks in remote environments, current telemanipulation research is insufficient to support task success when grasping an object [1]. It is limited to relatively simple grasping tasks such as caging, pushing, and pulling an object without considering the manipulation requirements. For the above example, when grasping a cup, the user has to go through multiple iterations of grasp, release, and re-grasp actions to adjust the robot's hand posture for one specific task [2] (Fig. 3.1).

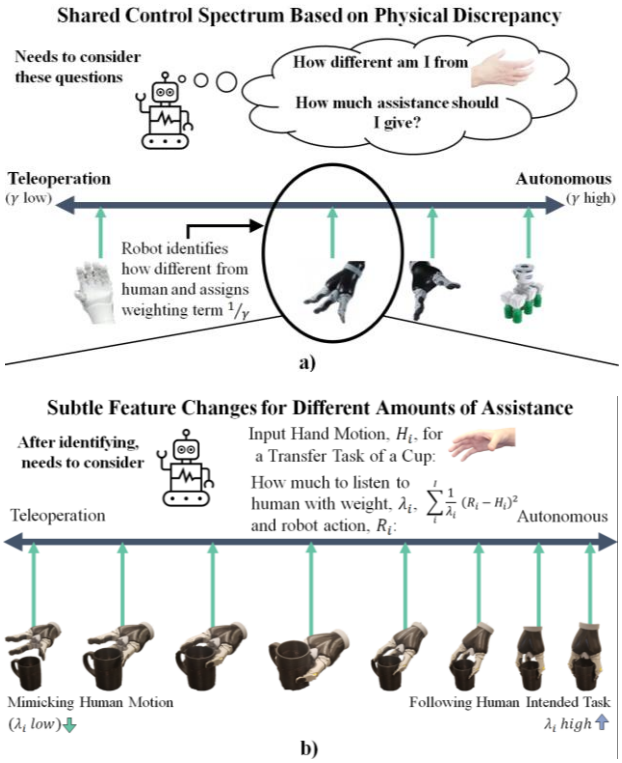


Figure 3.1 a) A representation of how the robot needs to first identify how different it is from the human, where the more similar the structures the easier it is to map and follow the human hand (low γ). b) After identifying the physical difference, the level of authority or autonomy the robot is allowed can produce various potential grasp poses for the intended task and human motion. Mimicking leads to more teleoperated or full user control (low λ_i), while intent-based strategies fall more along the lines of fully automated (high λ_i) for each specific motion constraint. The user may have a preferred option in the middle of the two extremes which is intuitive to their expectations. The approach in this paper aims to achieve that by appropriately setting the γ, and λ_i.

The difficulties in teleoperation stem from the indirect perception and manipulation of the robot's environment and workspace and the physical discrepancy between the operator and the robot [3] [4]. Telemanipulating objects by controlling a robot hand is even more challenging because of 1) the physical discrepancy between robot and human hand structures and 2) object interactions requiring different motion constraints for different tasks. For the first challenge, the physical differences in robot hand size, dexterity, and the number of fingers bring more difficulties in motion mapping from human to robot hands. This discrepancy prevents the robot from directly mimicking capability. For the second challenge, subtle motion differences can cause task failure (i.e., not leaving enough or appropriate open space during a handover task). Although the robot may be able to grasp the object successfully (e.g., caging), without additional context or a way to consider the consequence of robot motion in task space, it cannot ensure the subsequence manipulation task success with the object. Thus, an operator must adjust the hand posture to complete the task successfully.

Current strategies try to either 1) improve immersive human perception [1][5] or 2) achieve accurate motion mapping [2][6]. However, less research attention has been given to improving the robot's intelligence and ability to manage object interaction to solve indirect manipulation and perception. The robot hand is typically treated as a passive follower in teleoperation. Usually, the robot follows a fixed policy/rule to map human hand motion to its hand. The problem with the rule-based passive following is that while the robot interacts with the object, it can result in task failure (e.g., dropping the object) if the human inputs are imperfect. Imperfect inputs are expected due to indirect manipulation and physical discrepancies or if any environmental disturbance occurs around the object. Additionally, existing techniques cannot accommodate the subtle motion constraints for task success. The robot may not understand the

operator's intent behind grasping an object or generate adjustments to satisfy subtle motion constraints essential for task success. The robot's inability to actively remedy imperfect human inputs or provide adjustments to accommodate essential task constraints can easily fail. These limitations force current telemanipulation systems to rely on tedious human adjustments. Due to these reasons, current telemanipulation is limited to relatively simple tasks.

3.2.2 Shared Control: The Identified Open Problems

In the authors' vision, intuitive telemanipulation aims to reduce human adjustments by arbitrating between robot autonomy for task success and the operator's direct motion mapping. Adding autonomous assistance empowered by intent inference is the most viable strategy to augment a person's ability to complete tasks rather than passive motion mapping alone. Motion mapping of finger joints alone cannot guarantee task success, while focusing on only task success may have the human feel like they lose control since they are unaware that the robot will succeed. Arbitrating human actions and task success is the best way to reduce human adjustments for intuitive control (Fig. 3.1). The aim is to rely on the robot's knowledge to adjust. The identified open problems are 1) determining intent inference, 2) building a control scheme using the inference, and 3) determining the tradeoff arbitration between task success (empowered by understanding human intent and providing intent-based assistance) and following the operator (giving them a feeling of control over the robot).

Our previous work investigated a pure intent-based grasp planner to generate task-oriented assistance [7]. This intent-based planner generates grasp motion assistance toward achieving the inferred user's intent. Despite the previous effort, open problems still exist. It lacked awareness of *how* an operator wanted to complete the intended task (i.e., operator motion in the form of motion constraints). Our previous efforts do not address the appropriate way to combine the

intent inference planner with the human motion in a control scheme or the arbitration between the two.

3.2.3 Contributions

This paper focuses on developing a control scheme addressing the specific needs of shared control for object telemanipulation. This includes handling multiple tasks for the same object through intent inference and motion constraints that consider the operator inputs, physical discrepancies of the hands, and the robot's knowledge of the inferred tasks. Lastly, this paper investigates the control's legibility and preference through operators and bystanders. Although this paper does not include haptics or other immersive devices, the control scheme presented is viable for their use. We envision that the autonomous assistance provided by our work can and should be paired with immersive devices for a better-performing shared control system. The main contributions are as follows:

1) *Designed an arbitration approach to blend task-orientated assistance and direct motion mapping.* Adding constraints to consider the explicit motion aims to generalize the framework. These are not formulated as hard constraints but rather elastic, thus making the optimization find a better grasp solution near the human operator motion (shown in Fig. 3.1b as the entire summation term).
2) *Generalized an adaptative arbitration policy across robot structures based on the physical discrepancy.* A policy to determine the strictness of the motion constraints (applied in contribution 1) is paramount to the solutions the optimizer determines. The strictness policy is a penalty term (γ from Fig. 3.1a) based on the dissimilarity level between two separate hand structures. Additional penalty terms (λ_i from Fig. 3.1b) allow the optimizer to adhere

to certain motion constraints more than others (i.e., orientation is stricter than finger actuation). This has been generalized for different hand structures.

3) *Investigated the preference and legibility of control from operators and bystanders.* Ranking questionnaire surveys of the operators (360 trials) and bystanders (1080 trials) gives insight into their view of favorability for the control framework. The operators need to feel in control of the robot as they lack the physical perception of the robot's environment. The bystanders near the robot are vital to consider. They may be wary of the robot's actions and potentially stop what they are doing to assist the robot. This subjective analysis is critical as telemanipulation aims to improve the workflow instead of hindering it.

3.3 Related Work

3.3.1 Passive Motion Mapping for Telemanipulation

Current telemanipulation strategies focus on passive motion mapping techniques between human hands to robotic hands. State of the art approaches come in two forms: 1) Synergy based approaches [2][8] and 2) Neural Network mappings [9][10]. The synergy approaches focus on mimicking human-like interdependence muscle-tendon motion to reduce the needed input dimensions [11] [12]. The mapping follows joint-joint mapping, cartesian mapping, and virtual object mapping [13]. Although these mapping can provide ways to handle the dissimilar structures of a human hand to a robot hand for better accuracy, they do not aid in completing tasks.

Neural Network mappings are an alternative strategy to build the interdependence model. These mapping strategies are feature-to-feature mapping [14] and end-to-end mapping [9]. These approaches aim to extract different levels of information from the collected human dataset. Learning the features rather than defining them as inputs can provide better relationships to be

developed. These approaches lack kinematic constraints and training time, making it difficult to build a sufficiently appropriate model. Although they can provide precise and accurate mappings, they still cannot guarantee task success.

3.3.2 Autonomous Task-oriented Grasping

Semantic grasping approaches inform a robot of functional task-specific grasps [15]. This allows a robot to understand a high-level task success before evaluating a lower-level grasp success. A grasp can be considered successful based on grasp stability and contact pressure [16][13]; however, just because a successful grasp occurs does not mean it is appropriate. For instance, if one were to be at a tea party where it is socially acceptable to drink in a particular manner, others may be confused if you do not conform to their style. There are multiple ways to grab a teacup to drink successfully; however, it would result in a task failure if others do not perceive it as appropriate. Thus, one can think of task success as applying context for a grasp success [17][18]. In [19], they demonstrate how to discretize and classify multiple classes with transitional grasp zones. The context-aware grasping should be extended to what makes a grasp legible to its operator and bystanders (i.e., appropriately grasping the teacup). Further, additional robot models (dependent on hand structures) are needed to succeed in the task as their partner expects. The different grasp models generate different poses due to the physical discrepancy in end-effector design. For instance, the poses generated using the same intent inference for a two-finger robot and a five-finger robot may differ from a human pose.

Pure intent-based formulations without shared control between humans and robots are inherently autonomous control. These approaches aim to use a robot's understanding and grasping models to facilitate completing a human's intended target. These methods determine the most efficient way to accomplish the task from the higher-level information, such as the intended

goal. In other words, it may be able to achieve the same intended task yet accomplish it differently. This makes pure intent-based approaches ideal for implicit control techniques without feedback, such as gaze control [20][21] or gesture control [9][22] but not when an operator uses explicit motion inputs like hand posture control.

3.3.3 Shared Control for Task-oriented Telemanipulation

Intent-based shared control has shown promise and improvement in approaching tasks by placing the burden of task completion on the autonomous system [1]. The intent inference is measured through RGB [23], gaze [20][21], and hand marker data [24]. The intent inference is a posterior probability across a predefined set of tasks for easier classification. To obtain the intent inference, independent data-driven classification models–Neural Networks [21][24], Bayesian Networks [20][24][25], or Support Vector Machines [20][24][26]–are developed from the total pool of human data. These normally employ a winner take all strategy for posterior probability [1][27][28]. These are justified to be used in reaching tasks. It normally pertains to disambiguating the inference between reaching different objects. Still, it may not be suitable to disambiguate the inference between different tasks for grasping the same object (grasping a cup for drinking or transferring a cup to another person).

Afterward, shared control policies such as linear blending are designed to arbitrate between robot inferred goal and human motion[27][29][30]. Shared control strategies have been utilized in approaching tasks [1][31][32] yet are not suitable for manipulation as the motion constraints of manipulation from the operator and fully autonomous perspectives may differ. Other methods of approaching include envelope motion constraints [33][34] and manually selective assistance levels [35][36]. Haptic based strategies have tried to fill the void for grasping strategies but require more extensive knowledge of the environment to provide effective assistance

[37][38][39]. Lastly, the arbitration policy for teleapproaching tasks assumes more autonomy as the intent inference improves (usually towards the end of a trajectory when the robot is near the goal). Since the robot gets added autonomy, the primary focus of the robot is to complete the task over listening to the operator's motion. This behavior causes a sense of losing control of the robot as it generates a predefined grasp motion different from human inputs towards the end of the grasp.

Sets & indices:
i: Feature of grasp configuration 1…I
k: Task of each grasp configuration 1...K

Parameters:
H_i: Human input for feature i
P_k^H: Human probability intent for task k
U_i: Upper bound for feature i
L_i: Lower bound for feature i
$\bar{\bar{H}}$: total human feature set
$\bar{\bar{R}}$: total robot feature set
λ_i: KL divergence value for feature i
γ: total KL divergence for $\bar{\bar{R}}$ and $\bar{\bar{H}}$

Variables:
R_i: Robot input feature i
P_k^R: Probability of satisfying task k given robot features R

Objective:

$$\min \frac{1}{2}\sum_k (P_k^R - P_k^H)^2 + {}^1\!/_\gamma \sum_i \frac{1}{\lambda_i}(R_i - H_i)^2$$

$$s.t. \quad L_i \le R_i \le U_i \quad \forall_i$$

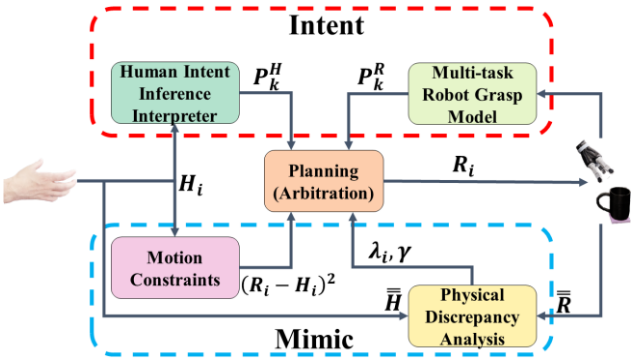


Figure 3.2 The control diagram and corresponding optimization for the "Planning" block. Two parts are separated based on the intent strategy (top) and mimic or motion strategy (bottom). The intent piece is represented in the first part of the objective function while the motion piece is represented in the second part of the objective function.

3.4 Methods

3.4.1 Intent-based Shared Control for Telemanipulation

The main problem of shared control systems is determining the tradeoff or balancing between following the intuitive human motion and assisting in the task's success. Shared control often is defined on a spectrum, where full user control and autonomous control are on opposite ends. A general sense of potential grasp configurations is demonstrated in Figure 3.1. Considering shared control as a spectrum from teleoperation (operator fully controlling the robot) to complete autonomy, we see different potential grasp configurations–some successfully do grasp the cup and others do not–which the operator may have a particular view of what is

most successful. By inferring the intended task, autonomy can take over and focus on the operator's task success. This style of action selection is best suited for control that lacks hand motion, i.e., eye gaze only control. However, all the poses presented in Figure 3.1 could be accepted for a particular operator and interface system. For instance, a well-experienced operator may prefer minimal assistance with hand tracking, while a novice operator may rely on assistance as they get accustomed to the system.

Four separate components are developed and discussed in the following subsections to address the needs of the tradeoff between balancing the task performance and following the operator. The four main components are: 1) disambiguation of the human intent for grasping tasks (section 3.4.2), 2) building a robot model and formulating autonomous action for given intent inference (section 3.4.2), 3) how to introduce motion constraints to generalize the intent planner (section 3.4.3), and 4) how to determine arbitration between the motion constraints and intended task through common variables and physical discrepancy (section 3.4.4). The first two components formulate the intent module (Fig. 3.2), which acts as an autonomous solution, as seen in Fig. 3.1. The latter two components deal with the motion constraint and determining the tradeoff.

3.4.2 Intent-based Control

In [7], we have validated that the disambiguation of multi-inference tasks can be achieved through intent descriptor vectors. The following is how to use intent descriptors to formulate a shared control framework for hands-free telemanipulation, which acts as a summary for [7].

3.4.2.1 Intent Inference Interpreter and Descriptor Vectors

In [24], we have investigated manipulation intent inferencing, where we track human hand poses for principle tasks of four objects. The investigation led to a multi-label classification of

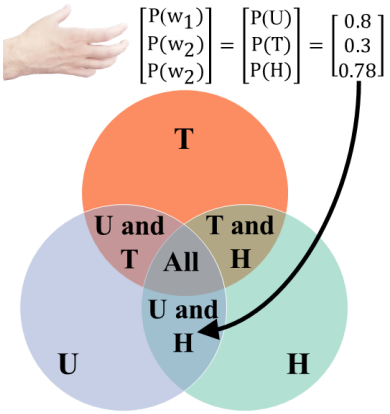
the poses to build independent-task inference models through a Neural Network, Support Vector Machines, and Bayesian Networks. In this paper, we adopt a pre-trained Bayesian Network model with hand pose information to determine intent inference at each time point and use a single object (a cup) as a case study.

Consider the temporal manipulation intent inference where there are three principle tasks, w_m, for grasping a cup (where *m* is the number of principle tasks). w_m holds three unique tasks: 1) using or drinking from the cup (denoted as U), 2) transferring the cup to another location (denoted as T), and 3) handing the cup over to another agent (denoted as H). By stacking these probabilities together, a 3x1 vector is obtained, where we can begin the development of a better descriptor vector. Since each classification model is obtained independently, it is not necessary for the total probability to equal one. For instance, the remainder of the section assumes the manipulation intent vector is w_m=[0.8,0.3,0.78] for a single time point. In this instance, the P(w_1) (the usage/drinking task) and P(w_3) (the handover task) are almost identical to one another. There exist two forms of uncertainty, one from the human input and another from the modeling process; thus, it is critical to deal with this ambiguity by developing a descriptor vector. We establish the descriptor vector called the human target probability vector, P_k^H, by following Eq. 3.1 to combine the manipulation intent inference and reduce the ambiguity. Where P(w) is the probability of the intent inference for the principle tasks (i.e., drinking, transfer, and handover), and Y is a subset of the powerset of all combinations of the task, ψ(m). The powerset, ψ(m), will have the size of $k=2^m$.

$$P_k^H = \prod_{n \in Y} P(w_n) \prod_{j \notin Y, j \in m} 1 - P(w_j), Y \subset \psi(m) \tag{3.1}$$

The index *n* accounts for the tasks in subset Y while *j* accounts for tasks outside the subset of Y. This results in P_k^H being of size $k=2^m$ (in our example, it is of size 8x1) because it accounts

for overlap among multiple tasks. This equation describes each combination of the principle tasks as either true or false. From the example w_m vector([0.8,0.3,0.78]), the first case of the manipulation intent is "usage only." This results in P_1^H=0.8(1-0.3)(1-0.78) = 0.1232. Likewise, if the event were to satisfy all tasks, it would achieve P_7^H = 0.8(0.3)(0.78) = 0.1872. Figure 3.3 demonstrates the relationships between the overlapping regions and how to calculate the probability. The human target probability vector is treated as a reference the robot should match upon calculating all combinations.



Region	P_k^H	Calculation	Ex.
U Only	P_1^H	$P(U) * \big(1 - P(T)\big) * (1 - P(H))$	0.1232
T Only	P_2^H	$P(T) * (1 - P(U)) * (1 - P(H))$	0.0132
H Only	P_3^H	$P(H) * (1 - P(U)) * (1 - P(T))$	0.1092
U and T	P_4^H	$P(U) * P(T) * (1 - P(H))$	0.0528
U and H	P_5^H	$P(U) * P(H) * (1 - P(T))$	0.4368
T and H	P_6^H	$P(T) * P(H) * (1 - P(U))$	0.0468
All	P_7^H	$P(U) * P(T) * P(H)$	0.1872

Figure 3.3 Demonstration of showing how to calculate the probabilities for different overlapping regions of the ambiguity model. The example column shows the resulting probability for P_k^H and how the resulting task will try to accomplish the U and H tasks.

3.4.2.2 Multi-Task Robot Model

The robot planner also needs to produce its own probability vector for satisfying the intent inference distribution; where, for simplicity, it is based on the Naive Bayes robot model ($\bar{\bar{R}}$). Model $\bar{\bar{R}}$ is created by collecting feature vectors **R** for each task and using Expectation Maximization to determine their class distribution. Each vector **R** comprises of individual variables R_i, which contain robot pose (position and orientation) and how much each finger should be open or closed (normalized between 0 and 1 where 0 is considered fully open). It can contain other variables such as the robot joint angles, provided fingertip force, torque, and object

features (shape or size). We obtained our robot model by manually collecting human-defined grasp poses for each individual task. We then used bootstrapping techniques to produce 2000 valid **R** within $\bar{\bar{R}}$ to produce probability distributions. To produce the robot posterior probability vector, P_k^R, with a given **R**, the following two equations can be used where μ_k is the average value for task k, Σ_k is the covariance matrix for task k. Both μ_k and Σ_k rely on the dataset for $\bar{\bar{R}}$ and are obtained after the Expectation Maximization procedure. This allows us to find new **R** solutions. In equation 3.2, d refers to the dimension size of the feature vector **R**. Lastly, P(k) is determined by $\bar{\bar{R}}$ and are the ratios the Expectation Maximization procedure assigned.

$$\mathrm{P}(\mathbf{R}|\mathrm{k}) = \frac{1}{\sqrt{\det(\Sigma_\mathrm{k})*(2\pi)^\mathrm{d}}} \mathrm{e}^{-\frac{1}{2}*(\mathbf{R}-\mu_\mathrm{k})^\mathrm{T}*\Sigma_\mathrm{k}^{-1}*(\mathbf{R}-\mu_\mathrm{k})} \tag{3.2}$$

$$P_k^R = P(k|\boldsymbol{R}) = \frac{\mathrm{P}(\mathbf{R}|\mathrm{k})*\mathrm{P(k)}}{\sum_\mathrm{k}^\mathrm{K}\mathrm{P}(\mathbf{R}|\mathrm{k})*\mathrm{P(k)}} \tag{3.3}$$

3.4.2.3 Formulation of Multi-Task Intent Controller

By building the robot model with this soft assignment, the robot can understand grasping features within its knowledge, which can satisfy the different combinations of tasks. For instance, a single grasp can be used for drinking and handovers of a cup. We can use common grasp poses to satisfy the ambiguous manipulation intent ($\mathrm{P}(w_m)$). The objective function can minimize the difference by using the target and robot probability vectors. An L2 norm, as shown in equation 3.4, is used to minimize the difference between the human target probability vector and the robot probability vector. The objective is to have the robot match the human as closely as possible to ensure it completes the proper task.

$$min\frac{1}{2}\sum_k(P_k^R - P_k^H)^2 \tag{3.4}$$

Compared with the traditional framework for shared control involving approaching tasks, there is a need to handle solutions that are discontinuous, physically different from human inputs, and come from complex interdependent models. The optimization proposed above handles discontinuous models through P_k^R where multimodal distributions can be considered. Further, the model structure of P_k^R can handle complex input interdependence if the probability distributions differ from the Naïve Bayes model.

3.4.3 Arbitration to Blend Task-oriented Assistance and Direct Motion Mapping

Two types of motion constraints must be considered. The first is the physical motion constraints of the robot defining the workspace. The second is the motion constraints imposed by the human operator. The latter determines the objective function of the shared control formulation, which will be presented in the corresponding section.

3.4.3.1 Physical Motion Constraints of the Robot's Workspace

The robot grasp model also contains upper and lower bounds for controllable variables, $R_{i,}$ as shown in equation 5. These are used for two reasons: 1) to bound the problem and help the planning process, and 2) to prevent external harm to the robot and the environment. For instance, the table height can be turned into a lower bound to prevent external harm to the robot, so the robot does not collide with it. The current formulation discussed only implicitly uses human motion to infer intent. However, it is necessary to consider explicit constraints to inform the robot to grasp an object in a particular manner.

$$L_i \leq R_i \leq U_i \quad \forall_i \tag{3.5}$$

3.4.3.2 Motion Constraints Imposed by the Human Operator

An operator's motion constraints are essential to consider in the planning process. In our telemanipulation scenario, the operator has inputs, $H_i \in \mathbb{R}^{11}$, that we map to the dimensions of $R_i \in \mathbb{R}^9$ (i.e., position, orientation, and 1 DOF for each finger of the MICO hand). A simple motion constraint is to force all the operator's inputs to equal the robot's controllable variables to emulate direct telemanipulation, as shown in equation 3.6. This is a strict case where an operator forces the robot to understand how a task should be accomplished—a necessary consideration to increase the effectiveness of assistance.

$$R_i = H_i \quad \forall_i \tag{3.6}$$

However, explicitly dictating a constraint in this manner may lead to unintended errors as the burden of control is placed on the operator. The operator could quickly become overwhelmed in hands-free telemanipulation as they do not have sensory feedback and must determine how to map their inputs to the robot's actions. This ultimately results in more readjustment to complete the task or task failure. Further, this definition of motion constraints leads to the robot controller not providing effective assistance. The explicit nature does not allow the robot to utilize its domain knowledge in exploring an alternative way to accomplish tasks. The imposed motion constraints could be outside the bounds of the robot's grasp model knowledge, leading to an infeasible solution or an incorrect intent inference.

Alternatively, the robot should consider certain variables to adhere to while using its knowledge to determine the best configuration to satisfy the task. Having an agent understand the relative differences between its knowledge of the control variables to another agent's, allows both systems to accommodate their behaviors. Instead of forcing the operator to understand the robot's actions, we have the robot understand the human motion and pair this with the intent

inference. Therefore, the robot uses its knowledge to determine significant constraints to follow and which are relatively less important. The hard motion constraints are formalized as elastic constraints by combining equation 3.4 with an adjusted equation 6 to reformulate the objective function in equation 3.7.

$$\min \frac{1}{2}\sum_k (P_k^R - P_k^H)^2 + \frac{1}{\gamma}\sum_i^I \frac{1}{\lambda_i}(R_i - H_i)^2 \tag{3.7}$$

This allows the robot to use its domain knowledge to alter how a user may want to accomplish a task. Further, the robot violates the mimicking constraint based on the commonality and importance between H_i and R_i. The new components allow the robot to understand common variables between itself and the human operator and how similar they are. If both the human and robot variables are nearly identical, then the weights (γ, λ_i) approach 0; thus, the overall weighted terms approach 1. When the penalty terms become dominant (approaching 1), the formulation begins to follow the mimic formulation—the user has full control. On the other hand, if the hands diverge and differ entirely, the weighted term goes to 0. This makes the intent matching aspect of the objective function dominant, resembling the intent formulation. With an objective function constructed in this manner, we have both components (autonomous actions and mimicking capabilities) needed for shared control and an arbitration weight between the two.

3.4.4 Generalized Adaptive Arbitration Across Robot Structures Based on Physical Discrepancy

Determining appropriate weights for common features is a two-fold process where not only common variables are needed to be determined, but the degree to which the variables are restricted. These weights are static and do not change over time. A way to analyze the similarity is to use Kullback-Leibler (KL) divergence. This divergence aims to determine how two grasp models differ from one another. For this to work, assume the grasp model directly interacting

with the object (the robot grasp model) is the true distribution, and the other (the human grasp model) is the inference/prediction distribution. In other words, the human hand structure is used to infer the robot hand. However, the grasp models may diverge from one another differently. For instance, consider the robot grasp model ($\bar{\bar{R}}$) and the human grasp model ($\bar{\bar{H}}$) where $\bar{\bar{R}}$ contains $\bar{\bar{H}}$ configurations in its own model ($\bar{\bar{R}} \supset \bar{\bar{H}}$). This would make predicting $\bar{\bar{R}}$ from $\bar{\bar{H}}$ easier than determining $\bar{\bar{H}}$ from $\bar{\bar{R}}$.

To determine the commonalities between two populations of variables, λ_i, and the entire hand configuration, γ, KL divergence is used. Since all the grasp models created are assumed to be multivariate normal distributions from the above sections, each feature is univariate normally distributed. Thus, each population contains a mean, μ_{Hi} and μ_{Ri}, and a standard deviation, $\sigma_{Hi,}$ and $\sigma_{Ri.}$ The population parameters are dependent on the structure. The KL divergence between the same feature from two separate populations (two different hand structures) can be defined as the following comparison between two univariate normal distributions.

$$\lambda_i = KL(H_i||R_i) = \ln\left(\frac{\sigma_{H_i}}{\sigma_{R_i}}\right) + \frac{\sigma_{R_i}^2 + (\mu_{R_i} - \mu_{H_i})^2}{2\sigma_{H_i}^2} - \frac{1}{2} \tag{3.8}$$

λ_i are bounded from [0, ∞]. The lower bound of 0 means no divergence, meaning the two populations are identical, while the upper bound ∞ means the distributions are completely different. This can be used to determine the importance level of each grasping feature between two separate hand configurations. Additionally, the multivariate normal distribution between two populations can be used to determine the overall divergence between hand configurations:

$$\gamma = KL(\boldsymbol{H}||\boldsymbol{R}) = \frac{1}{2}\left(Tr(\Sigma_H^{-1}\Sigma_R) + (\mu_H - \mu_R)^T\Sigma_H^{-1}(\mu_H - \mu_R) - d + \ln\frac{|\Sigma_H|}{|\Sigma_R|}\right) \tag{3.9}$$

where d is the length of the feature vector **R** and where γ is also bounded from [0, ∞]. Although these equations are used explicitly for normally distributed variables, there are known configurations for other distributions that may be effective for a more complex system.

3.5 Experimental Results

3.5.1 Experimental Conditions, Design, Setup and Hypotheses

We developed two controllers and used a third as a baseline. The first controller is only based on intent inference (equation 3.4) and does not include motion constraints from equation 3.6. This assistance mode is referred to as the Intent Only. The second controller is based on the arbitration strategy in equation 3.7, which balances the pure intent-based control and pure human mimicking control. This assistance mode is referred to as Knowledge Intent Arbitration Optimization (KNITRO). The last one is the pure human mimicking baseline controller (Mimic Only), which includes strict motion constraints of equation 3.6.

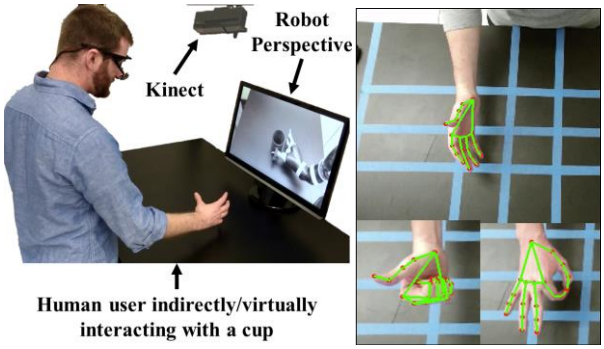


Figure 3.4 Operators see the robot perspective on the screen while they move their hand freely over the table. The Xbox Kinect captures hand posture to extract variables such as palm pose and the amount each finger is open or closed. This information is then used to generate intent inference and/or used in the optimization

The Institutional Review Board (IRB) of the authors' institution approved an exemption for the study. Before participating in the study, written consent was obtained, acknowledging they understood the robot setup, the purpose of the study, and the potential risks involved (arm

soreness). Two distinct groups are evaluated: the operators and the bystanders. The operators were asked to use a 3-finger MICO robot to accomplish three separate tasks with a cup: relocating or transferring the cup, drinking or using the cup, and handing the cup over to a bystander. The cup is used as a case study object as the multi-object inferencing scene is outside the scope of this work and allows a better comparison for the individual controllers. These tasks are tiered in difficulty (Transfer > Usage > Handover) based on empirical rules imposed in [7] and [24]. In summary, the empirical rules restrict finger placement and grasp area allowed to dictate task success. An Xbox Kinect captures images of the operator's workspace (Figure 3.4), which are then fed into Google MediaPipe's [40] hand tracker to extract the hand. The hand is paired with point cloud data from the Xbox Kinect to extract the real-world coordinates of the human hand. The output hand pose (position and orientation) and how much the fingers were open were extracted as inputs for the pre-trained intent inference Bayesian Network model and controller optimization.

The operators were told which task to perform, but they were free to grasp in their preferred way for each given task. The different preferred ways an operator may want to perform a task will provide different intent inferences. The operators were given training time to understand the three separate controllers. The operators were asked to complete six trials for each task for each controller. The total distance and time of the trajectories were recorded. They were also asked questionnaires after each trial. The questionnaire is provided in the Appendix. The human grasp and different controllers for the robot's motion were recorded and saved for the bystanders to analyze later. This experiment was created to test two hypotheses:

1. Intent-based approaches (Intent Only and KNITRO) can outperform pure human control telemanipulation (Mimic Only) in success and completion time.

2. People prefer the arbitration approach (KNITRO) over the pure intent inference controller (Intent Only) and the pure telemanipulation (Mimic Only).

3.5.2 Operator Analysis

Five operators completed the tasks in a predefined order of Mimic Only, KNITRO, Mimic Only, and Intent Only. The Mimic Only approach acts as a baseline since it is pure telemanipulation represented as a cartesian mapping where an operator palm pose controls the end-effector palm pose in a one-to-one mapping of their own hand to the robot hand. The operators must redo the baseline in between assistance modes as a reminder of the behavior of the baseline control mode. For the analysis, both Mimic Only tasks have been averaged together. Participants were allowed unlimited training time to feel comfortable with each control mode before the actual trials began. Participants were permitted unlimited attempts to grasp the object until a successful grasp occurred or the object fell. In total, 360 trials were collected across the control modes (for each of the three tasks, there were 30 for KNITRO and Intent Only each and 60 for the Mimic Only). More information, including examples of successful grasp attempts for each controller, is shown in the accompanying video.

In summary of the following sections: The Intent and KNITRO approaches outperform the Mimic approach in terms of success rate, completion time, and preference, albeit task difficulty plays a factor in the significance of the impact KNITRO is the best for task success, while the Intent Only controller had the fastest completion times. The KNITRO strategy is slightly preferred to the Intent Only controller.

3.5.2.1 Success Rate

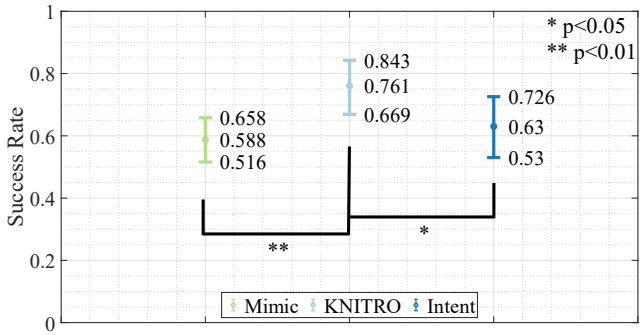


Figure 3.5 Comparison of the adjusted-Wald 95% confidence intervals for the grasping tasks' success rate. The KNITRO controller outperforms the other two strategies. The KNITRO method has statistical significance ($p<0.01$) with the mimic mode and has statistical significance ($p<0.05$) with the intent mode. No statistical significance was found between the intent and mimic controllers.

The KNITRO controller outperforms the other strategies. Due to the sample size, a Laplace estimator ($\frac{x+1}{N+2}$, x is the successful trials, and N is the total trial) was used over the Maximum Likelihood Estimate to better reflect the true success rate. The Laplace estimator is used as a precaution if an operator obtains 100% or 0% success by slightly adjusting their success rate away from these extreme outcomes. The KNITRO controller has the best Laplace estimate (0.76), followed by the Intent strategy (0.63), and the Mimic controller was last (0.59). An adjusted-Wald 95% confidence interval was created for each controller to see the expected bounds of success rate (shown in Figure 3.5). Lastly, an N-1 Chi-Squared test was conducted to determine if any of these controllers is greater than another with statistical significance using a standard p-value of 0.05. No statistical significance was found between the Mimic and Intent strategies. The KNITRO approach has statistical significance with the Mimic ($p=0.002$) and Intent ($p=0.026$) strategies. Adding assistance performs as well, if not better, than traditional telemanipulation (Mimic strategy).

Table 3.1 Success Rate and Confidence Intervals

Task	Controller	Laplace Success	CI Lower Bound	CI Upper Bound
All combined	Mimic	0.588	0.516	0.658
	KNITRO	**0.761**	0.669	0.843
	Intent	0.630	0.530	0.726
Transfer /Relocate	Mimic	0.838	0.737	0.921
	KNITRO	**0.843**	0.697	0.953
	Intent	0.75	0.588	0.885
Usage /Drinking	Mimic	0.468	0.346	0.591
	KNITRO	**0.625**	0.455	0.782
	Intent	0.563	0.392	0.726
Handover	Mimic	0.452	0.331	0.575
	KNITRO	**0.781**	0.623	0.909
	Intent	0.563	0.392	0.726

Table 3.2 Task Breakdowns for p-values of success rates

Task Breakdown	Intent vs. Mimic	Intent vs. KNITRO	Mimic vs. KNITRO
All combined	$2.41\text{x}10^{-1}$	$\mathbf{2.60\text{x}10^{-2}}$	$\mathbf{2.00\text{x}10^{-3}}$
Transfer/Relocate	$1.66\text{x}10^{-1}$	$1.60\text{x}10^{-1}$	$4.17\text{x}10^{-1}$
Usage	$1.87\text{x}10^{-1}$	$3.00\text{x}10^{-1}$	$6.90\text{x}10^{-2}$
Handover	$1.89\text{x}10^{-1}$	$\mathbf{2.70\text{x}10^{-2}}$	$\mathbf{1.30\text{x}10^{-3}}$

Tables 3.1 and 3.2 present the summary of the task breakdown. Table 3.1 shows the success rate and confidence intervals for each task. In each instance, KNITRO has the best success rate. An important note is to remember that the Mimic has more samples when developing the confidence interval, which produces a much tighter interval. The assistance modes are at least competitive, if not better than the Mimic approach.

In Table 3.1, we see the separation of the success rate and confidence intervals grow as the task increases in difficulty (task difficulty level: Transfer<Usage<Handover). Table 3.2 demonstrates the same sentiment through the statistical comparison across the tasks. For the first row, much of the influence in differences across the controllers is based on the Handover task.

The KNITRO outperforms the other two in difficult tasks, while operators may not feel the reliance on assistance for easier tasks.

3.5.2.2 Completion Time

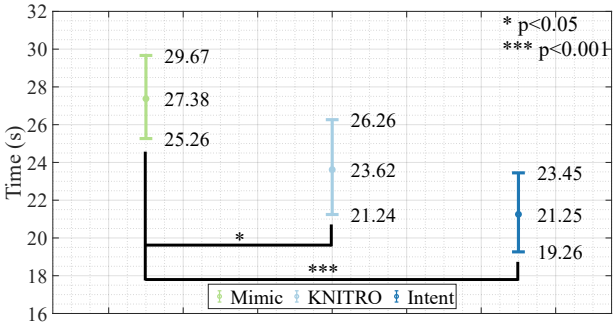


Figure 3.6 Comparison of the 95% confidence intervals for the geometric mean time values. These values are based only on the successful trials. Both the KNITRO and intent controllers outperform the mimic. The mimic method has statistical significance ($p<0.001$) with the intent method and statistical significance ($p<0.05$) with the KNITRO method. The users accomplish the grasping tasks faster with assistance. No statistical significance was found between KNITRO and intent methods.

The Intent and KNITRO controllers outperform the Mimic strategy. Only the successful trials were considered for the time analysis as failure times had high variance and would skew the data. Further, due to the nature of time data being positively skewed and not normally distributed [41], a log transform of the data is more appropriate than the raw data. The geometric mean achieves the log-transformed times and tempers outliers better.

The geometric means and confidence intervals are shown in Figure 3.6. A two-sample t-test was conducted to determine the significance between control modes with a standard p-value of 0.05. No statistical significance was found between the Intent and KNITRO approaches. The Mimic approach had statistical significance with both the KNITRO strategy ($p = 0.0284$) and the Intent strategy ($p = 0.0002$). Providing assistance leads to the operators accomplishing the grasping tasks faster.

Table 3.3 Completion time means and confidence intervals

Task	Controller	Geometric Mean (s)	LB CI (s)	UB CI (s)
All combined	Mimic	27.38	25.26	29.67
	KNITRO	23.62	21.24	26.26
	Intent	**21.25**	19.26	23.45
Transfer /Relocate	Mimic	27.65	24.22	31.57
	KNITRO	26.66	22.42	31.70
	Intent	**23.42**	19.40	28.27
Usage /Drinking	Mimic	24.31	21.06	28.06
	KNITRO	20.34	16.78	24.67
	Intent	**19.79**	16.87	23.22
Handover	Mimic	30.39	26.69	34.59
	KNITRO	23.31	19.16	28.37
	Intent	**20.01**	17.00	23.55

Table 3.4 Task Breakdowns for Completion time p-values

Task Breakdown	Intent vs. Mimic	Intent vs. KNITRO	Mimic vs. KNITRO
All combined	$\mathbf{2.00x10^{-4}}$	$1.45x10^{-1}$	$\mathbf{2.84x10^{-2}}$
Transfer/Relocate	$1.46x10^{-1}$	$3.03x10^{-1}$	$7.31x10^{-1}$
Usage	$5.22x10^{-2}$	$8.15x10^{-1}$	$1.31x10^{-1}$
Handover	$\mathbf{2.00x10^{-4}}$	$2.18x10^{-1}$	$\mathbf{2.52x10^{-2}}$

The task breakdown summary is presented in Tables 3.3 and 3.4. Table 3.3 shows the geometric means and confidence intervals for each task. In each instance, the Intent has the best completion times. The assistance modes outperform the Mimic approach in each task. In Table 3.3, the intent-based controllers separate from the Mimic controller as the task increases in difficulty. Table 3.4 demonstrates the same with statistical comparison across the tasks. The influence in differences across the controllers is based on the Handover task for the first row.

3.5.2.3 Subjective Analysis

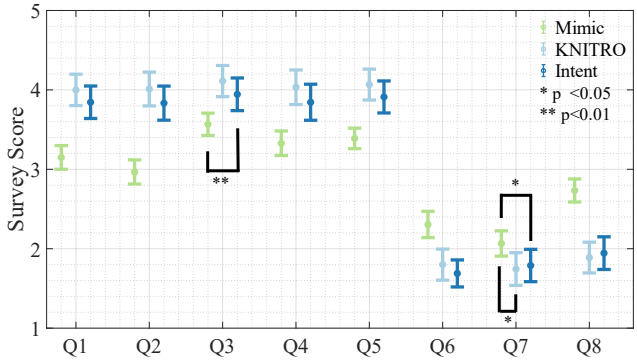


Figure 3.7 Comparison of the combined questionnaire results. The survey questions can be found in the Appendix. For readability, only the exceptions of statistical significance trends are shown on the figure; otherwise, no statistical significance was found between the KNITRO and Intent methods; the Mimic controller has statistical significance ($p<0.001$) with both the KNITRO and Intent controllers. For Q3, the Mimic mode has statistical significance ($p<0.01$) with the Intent controller. For Q7, the Mimic controller has statistical significance ($p<0.05$) with both the KNITRO and Intent controllers. Providing assistance to users is favored.

The KNITRO and Intent approaches are more favorable than the Mimic controller. Operators were asked eight questions after each trial, which they answered on a Likert scale from Strongly Disagree to Strongly Agree. The first five questions have a positive connotation (higher score is wanted), while the last three have a negative connotation (lower score is wanted). The specific questions are in the Appendix. Figure 3.7 shows the t-distribution confidence intervals for each question. For readability, two trends have not been displayed regarding the two sample t-test with a standard p-value of 0.05. The first is that the Mimic is statistically significant with the Intent strategy ($p=0.0001$), while the second is that the Mimic strategy is statistically significant with the KNITRO strategy ($p = 0.0001$). The exceptions to these trends are in Q3 and Q7. For Q3, the Mimic and Intent had a statistical significance of $p = 0.0032$. In Q7, the Mimic strategy has statistical significance with the KNITRO strategy at $p=0.015$ and the Intent strategy at $p=0.0344$. The operators prefer assistance, with no statistical difference between the intent-based strategies. For brevity, the task breakdown has been omitted.

3.5.3 Bystander Analysis

After the operators completed their experiments, their collected trajectories were used in an offline evaluation for bystanders. In that, the operators' inputs were used to produce three output robot grasp configurations, one for each control mode. The operators' hand motion was displayed with the given target task next to the three final robot grasp configurations for the bystanders to view simultaneously. The evaluators' role is to validate whether similar sentiments held by the operators are also held by outside observers. The 20 evaluators were asked to rank 54 different grasp configuration groups (each group consists of an operator input and three output robot grasp configurations) for a total of 1080 trials. The evaluators were not informed how any of the strategies behaved, nor were there any explicit identifying markers to determine the strategies. However, they were given a brief introduction to the MICO arm's abilities and limits to ensure realistic expectations. No evaluator operated the robot to provide a fair comparison. The evaluators can see three potential grasp poses simultaneously instead of trying to recall each assistance strategy to compare. The evaluators were asked to rank the three control strategies based on three criteria: 1) task completion by the robot, 2) following of the operator, and 3) their overall preference.

3.5.3.1 Bystanders' Assessment of Task Completion

The rank choices of the total task completion are shown in Figure 3.8. The distributions of the rankings can inform some expected trends, such as the Intent and KNITRO strategies being more transparent about task completion. Based on the distributions presented in Figure 3.8, a Friedman test with a Least Significant Difference correction factor was determined to see statistical significance across the control strategies. The test was conducted with a standard p-value of 0.05. Table V shows the p-values of the Least Significant Difference statistical

correction between the control strategies after the Friedman test found significance across the data. As expected, there is statistical significance between Intent and Mimic strategies. No statistical significance is observed with either strategy compared to the KNITRO.

Figure 3.8 Bystanders' assessment of task completion across 20 subjects where the trend shows the Intent strategy is the best performer, followed by the KNITRO strategy, and last by the Mimic strategy. This trend aligns with intent methods being better at completing tasks.

Table 3.5 shows the statistical significance across the individual tasks. The Intent strategy is the best across all tasks, the Mimic strategy is the worst, and the KNITRO is in between both. The Intent and Mimic controllers continue to be statistically significant for each task. Greater separation between the Mimic and KNITRO is shown for each task.

Table 3.5 Task Breakdown p-values for Bystanders' Asssessment of Task Completion

Data Breakdown	Intent vs. Mimic	Intent vs. KNITRO	Mimic vs. KNITRO
Total	$\mathbf{4.53x10^{-4}}$	$\mathbf{3.95x10^{-2}}$	$1.48x10^{-1}$
Usage	$\mathbf{5.12x10^{-4}}$	$2.45x10^{-1}$	$\mathbf{2.07x10^{-2}}$
Handover	$\mathbf{4.89x10^{-5}}$	$1.03x10^{-1}$	$\mathbf{1.51x10^{-2}}$
Relocate	$\mathbf{8.03x10^{-5}}$	$8.05x10^{-2}$	$\mathbf{2.81x10^{-2}}$

3.5.3.2 Bystanders' Assessment of Human Following

The rank choices of the total human following are shown in Figure 3.9. The trend is somewhat expected where the Intent Only controller performs the worst. It has an inverse relationship to the bystanders' assessment of task completion. This is expected as the Intent strategy does not follow the operator on how to accomplish the task and may choose a more

well-suited manner to accomplish the task. The Mimic controller is the most favored. The higher third choice count is likely due to the evaluators feeling that the mimicking grasp the robot was attempting was unnatural and off-putting due to finger placement. The KNITRO approach is competitive with the Mimic controller and less susceptible to variational ranking since the grasp pose alters human input to better fit the robot grasp model. The similar first and second choices are likely due to the evaluators looking at more important features to accomplish the tasks, such as the palm direction, instead of more minor features, such as finger placement. The KNITRO and the Intent strategies are near similar for the second choice. The KNITRO strategy sacrifices some of the ability to follow an operator to accommodate its own hand structure. The Intent strategy also implicitly follows the operator when the operator's motion aligns with the robot model rules. No statistical significance was found across the following schemes.

Figure 3.9 Bystanders' assessment of human following across 20 subjects. The Intent strategy performs the worst of the three. While the Mimic strategy is volatile in doing extremely well or extremely poorly. This is most likely due to the evaluators viewing the robot as following the intended action instead of following the motion. The KNITRO strategy is less volatile and does better at following the person.

Table 3.6 holds the individual task results for the human following. Interestingly, the bystanders' assessment for following is more task-dependent. The Intent strategy does not strongly follow the human's action for any task but fairs better in the Handover task. The Mimic controller is consistent for all tasks with a high first and third choice count. The KNITRO strategy is more robust and is generally seen as a good follower of the human across all tasks.

This does not necessarily mean the KNITRO strategy outperforms the Mimic; the rankings suggest it has less variance in following the human. This is likely due to a few factors: 1) the KNITRO strategy listens to an operator's main factors, such as palm direction, yet does not listen to smaller factors, such as position, which ultimately make the difference for evaluators' preference, 2) when addressing this question evaluators subjectively include an ideology of "it does what I want," 3) evaluators could distinguish the subtle differences in how the strategies were following a person and did not prefer the fine tune control the Mimic approach was using. With these trends, the Intent and Mimic strategies have statistical significance for each task, while the Mimic and KNITRO have significance only for the Handover task.

Table 3.6 Task Breakdown p-values for Bystanders' Assessment of Human Following

Data Breakdown	Intent vs. Mimic	Intent vs. KNITRO	Mimic vs. KNITRO
Total	$\mathbf{1.40x10^{-3}}$	$1.18x10^{-1}$	$4.43x10^{-1}$
Usage	$\mathbf{1.50x10^{-3}}$	$7.36x10^{-1}$	$6.22x10^{-2}$
Handover	$\mathbf{1.46x10^{-4}}$	$3.09x10^{-1}$	$\mathbf{4.53x10^{-2}}$
Relocate	$\mathbf{2.41\ x10^{-4}}$	$2.41x10^{-1}$	$8.44x10^{-2}$

3.5.3.3 Bystanders' Overall Preferences

The ranked choices based on the preferences of the bystanders for the total tasks are shown in Figure 3.10. The Intent strategy is preferred over the other control schemes. The distribution is nearly identical to the bystanders' assessment of task completion. The Mimic approach is the least favored. The KNITRO strategy is in the middle of both other strategies. Table 3.7 shows that the Intent and Mimic methods have statistical significance. However, the KNITRO does not have significance with either control scheme.

Table 3.7 shows the statistical significance of the control schemes for bystanders' preferences. The preferences are consistent across all tasks. The Intent controller is the most favored. The Mimic strategy is the least favored. The KNITRO controller is between controllers

for the Transfer and Handover tasks and outright beats the Mimic strategy in the Usage task. There is always a significant difference in preferences between Intent and Mimic strategies, where Intent is preferred. The KNITRO and Mimic strategy is conditional on the task—a significant difference is found for the Handover task only.

The most interesting trend is observed when comparing Figure 3.8 and Figure 3.10. The rankings all bystanders choose are strikingly similar. This suggests bystanders value a successful pose that telegraphs which task the robot is accomplishing over a pose that mirrors a human's action. The bystanders' perspective suggests that autonomous robotic assistance is needed so the robot can correctly complete tasks.

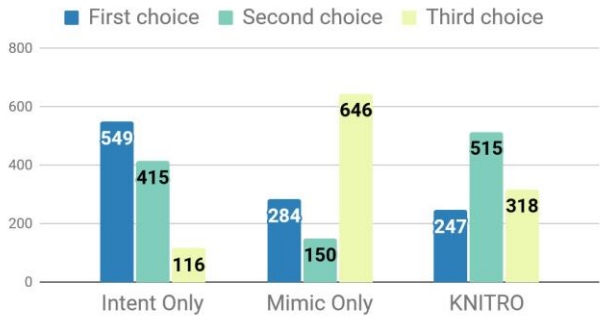


Figure 3.10 Preference of the evaluators across 20 subjects. The Intent only strategy is favored with the KNITRO and Mimic strategies following. This trend closely resembles the rankings of the task completion, which indicates bystanders prefer task completion over following.

Table 3.7 Task Breakdown for Bystander's Preference

Data Breakdown	Intent vs. Mimic	Intent vs. KNITRO	Mimic vs. KNITRO
Total	$\mathbf{4.48x10^{-3}}$	$7.16x10^{-2}$	$2.98x10^{-1}$
Usage	$\mathbf{8.21x10^{-3}}$	$2.87x10^{-1}$	$1.15x10^{-1}$
Handover	$\mathbf{6.74x10^{-4}}$	$1.72x10^{-1}$	$\mathbf{4.18x10^{-2}}$
Relocate	$\mathbf{2.84x10^{-3}}$	$1.53x10^{-1}$	$1.20x10^{-1}$

3.5.4 Arbitration Guided by Physical Discrepancy

It is necessary to quantify the total information difference using the KL divergence since there is a mismatch in the number of fingers between the operator (5 fingers) and the robot (3

fingers). This is necessary as a different number of fingers can lead to different grasp configurations with varying quality to compensate for the hand kinematics [42]. We analyzed the overall similarity values between different hand structures by calculating the KL divergence values. The values generated in Table 3.8 are not look-up values. Rather, we validate our method to show the trend of similarity between hand structures. Each model was built independently for the specific hand structure shown to satisfy the grasping rules defined in previous work. The numbers are not necessarily reflective of the optimization, rather just show a penalty cost toward mimicking the likeness of hands and policies. The leftmost column reflects telemanipulation scenarios. Since $KL(P||Q) \neq KL(Q||P)$, the row indicates P, while the column indicates Q. In a robotics context, agent P will attempt to perform an action based on agent Q's input. The divergence value is associated with how similar the models are. The off-diagonal terms do not necessarily need to be equal. For instance, when P= two-finger gripper and Q= three-finger gripper, the similarity is (5.26×10^{1}), and vice versa is (1.18×10^{6}). This signifies that it is easier for the two-finger gripper model to learn from the three-finger model. This table should be read by comparing columns for a single row. For example, when P= two-finger gripper, where three potential models for Q exist. Of the three potential candidates for Q, we see the three-finger gripper (5.26×10^{1}) is the most similar, while the human hand (1.79×10^{3}) and five-finger gripper (9.85×10^{3}) are orders of magnitude less similar. These differences are due to several factors, such as the characteristic parameters (size, shape, and number of fingers) and grasping configurations (palm orientation, finger contact, and force). In this instance, the values obtained are only for a single task of transferring a cup, although the trend holds for other tasks. Complex hand structures are more similar to their simpler counterparts. For instance, the one in our experiment (P=three finger MICO and Q=Human hand) had an overall divergence of 1.29×10^{3}. Notice how

the complex human hand does the best job relating to the three-finger gripper despite nearly being the same robot as the two-finger gripper. When observing the last row, the divergence value of the complex human hand ($1.41x10^1$) is most similar to the five-finger robot hand, while the three-finger ($1.14x10^5$) and the two-finger gripper ($9.0x10^5$) are less similar.

Table 3.8 Divergence for Different Hand Structures

P: row Q: column KL(P‖Q)	Human Hand	2-Finger Gripper	3-Finger Gripper	5-Finger Gripper
Human Hand		$1.18x10^6$	$9.6x10^5$	$1.22x10^2$
2-Finger Gripper	$1.79x10^3$		$5.26x10^1$	$9.85x10^3$
3-Finger Gripper	$1.29x10^3$	$1.18x10^6$		$3.26x10^3$
5-Finger Gripper	$1.41x10^1$	$1.14x10^6$	$9.0x10^5$	

3.6 Discussion

3.6.1 Operator Objective Analysis and Hypothesis 1 Evaluations

In summary, the KNITRO strategy outperforms the Mimic in terms of success rate (average success rate 0.761 compared to 0.588 with statistical significance $p=2.00x10^{-3}$), completion time (average completion time 23.62s compared to 27.34s with statistical significance $p=2.84x10^{-2}$). We cannot conclude that the Intent and Mimic strategies differ in terms of success rate (average success rate 0.630 compared to 0.588 with no statistical significance $p= 2.41x10^{-1}$), but we can conclude the Intent strategy is faster than the Mimic (average completion time 21.25s compared to 27.34s with statistical significance $p = 2.00x10^{-4}$). These results demonstrate that intent-based strategies can outperform the Mimic strategy.

The first hypothesis was to determine if assistance improves objective outcomes. The results indicate this is true. The KNITRO improves the success rate over the Mimic strategy more than

the Intent Only controller was able to improve over the Mimic strategy. This is likely due to the operator's strong counter-reaction as they perceive a loss of control of the robot as it gains extra autonomy. When considering the completion time, the Intent Only controller shines. Adding the extra autonomy allows the operators to complete the task quicker. The completion time best analyzes the trajectory as cumulative distance traveled, and cosine similarly reflects adding autonomy improves task performance. The robot's trajectory speed is based on the control mode's autonomy level. The Intent Only (the fastest) provides the most autonomy, followed by the KNITRO and then the Mimic Only (the slowest).

3.6.2 Legibility, Preference and Hypothesis 2 Evaluation

In summary, the assistance modes are preferred by the operators over the Mimic (with KNITRO with statistical significance $p=1.0x10^{-4}$ and the Intent controller with statistical significance p= $p=1.0x10^{-4}$). However, the Intent Only is preferred most by the bystanders compared to the Mimic strategies (with statistical significance $p = 4.48x10^{-3}$). There is no statistical significance between the KNITRO and Intent Only controllers. These results are not definitive proof the second hypothesis is true.

The second hypothesis was to compare the preference of the assistance modes and determine if the KNITRO is preferable to the Intent Only and Mimic Only controllers. Hypothesis 2 is only partially true. The operators and bystanders are split on their preferences for assistance modes, with no statistical significance between the modes. It appears the KNITRO is favored by operators in Fig. 3.7, while the bystanders chose Intent Only as the highest rankings in Fig. 3.10. So, we can not say the KNITRO is preferred over Intent Only; however, the assistance modes are preferred over the Mimic strategy.

In terms of legibility between the KNITRO and Intent, it would make sense that operators prefer the KNITRO as they have a say over how the robot moves. For the operators, the Intent strategy should have the worst legibility. However, the Intent and KNITRO may show similar motion responses when the desired human approach aligns with the robot model's rules and attempts to follow them. The KNITRO strategy is designed to objectively compromise the tradeoffs of both the Intent and Mimic strategies to reduce the user's mental workload. The actions they take may be easier for the operator to move the robot but lack the task legibility to bystanders. The bystanders prefer the Intent strategy as its final pose best indicates which task is being completed. When designing future control schemes, the discrepancy between the operators and bystanders is imperative. In the authors' view, the operator's preference should have a higher priority since they control the robot.

3.6.3 Limitations and Recommendations

In this paper, we used an ordinary object with simple interaction rules to analyze the methods, where we see the clear expected trend. These trends we observe are possibly more significant depending on the irregular handling of a shape, where the task difficulty will be more serious, and the operator would rely more heavily on the robot's assistance. Additionally, the KNITRO formulation is generalizable because a formal framework has been introduced, and other potential methods exist to determine the weighting scheme to create desired behaviors. There is more room for improvement through operator personalization of the penalty terms, λ_i, γ. Although we present a single approach to impact the arbitration through physical discrepancy, we acknowledge that other factors are important to consider for a more authentic representation of the system [31]. For instance, users should be allowed to focus only on position control (if preferred) while the robot handles the orientation control. Users may have different preferences

for what is considered a natural or intuitive response to the robot. This constraint is already possible within the control system framework proposed by empirically tuning the penalty terms. Fixed values are not the only option, as dynamic values are viable and encouraged. It is possible to design a function to adjust the arbitration based on the historic performance/preference of the operator. Task difficulty may demand it based on the results presented. During the trials, participants often noted that the assistance was not as apparent during the Transfer task (easiest of the three). However, they would later remark for the more challenging tasks that, they felt the robot significantly adjusted (in a positive view) to help them complete the task. This leads to the credence that the penalty terms should be a dynamic function with a new input regarding task difficulty.

However, if an operator encounters an unexpected motion (a rotation or sudden jump in position), they may feel disconnected from the robot. The KNITRO controller may thrive since the robot will not stray from the operators' important input motion, such as palm direction, yet adjust other variables, such as position or finger control. It makes intuitive sense that the more autonomy we inject into the system by adjusting these variables, the faster it can achieve the task.

Nevertheless, too much autonomy can result in a loss of control. It ignores the operator's explicit commands by producing unexpected movements, leading to an overall distrust of the system. These potentially unexpected movements cause operators to overreact in their adaption, thus changing the inputs sent to the robot and finally resulting in the robot failing the task altogether. Therefore, we recommend that the Intent Only strategy should be reserved for control where an operator does not provide direct inputs but rather more high-level or partial control, such as eye gaze only control [20][21], or in situations where the operator has a lack of

understanding of the dynamics of the robot [43]. We believe the KNITRO strategy is most appropriate moving forward when an operator can give more explicit commands (instead of position control, they also give orientation and force) and understand the dynamics better. The KNITRO had the highest operator preference, felt the most natural, and yielded the best success rate, albeit at the sacrifice of some speed (compared to the Intent Only controller).

Although others disambiguate and use independent models for each task, we do not believe performing these methods for all situations is possible. In [27], they attempt to disambiguate the target object from a set of objects with discrete states for approaching tasks; however, this discretization does not exist in grasping. The strategies presented in our paper are used to deal with ambiguity from human operators toward robot systems since this ambiguity is inherent and present in most practical scenarios. Rather than the operator, the robot should deal with environmental and human input uncertainties. Our approach handles human uncertainty through multi-task models, and an optimization process augments the resultant output to complete the task. The robot implicitly must achieve two criteria: 1) complete the inferred task from the person, and 2) display legibility to the operator so they know what task it is accomplishing. This naturally forms a shared control problem. Our approach does arbitration by considering physical discrepancy to achieve the tradeoff between these two criteria. The KNITRO strategy attempts to achieve both goals to determine what people prefer from robotic assistant systems. The Mimic method should theoretically carry the highest level of legibility to the operator as it strictly/explicitly follows the person. However, this makes the Mimic method extremely volatile to task success. A well-experienced operator must ensure the task is successful despite not fully understanding the robot environment. The two assistance modes outperform the Mimic strategy since the level of assistance is increased.

3.7 Appendix A

The following survey questions were comprised from [44] and asked after each trial and answered on a 5-point Likert scale (with 1 being Strongly Disagree and 5 being Strongly Agree):

1) It helps me be more effective in completing the task.
2) It makes the task easier.
3) It worked as I expected.
4) I'm satisfied with it.
5) I trust the robot with this assistance mode.
6) It is annoying or frustrating during the task.
7) I feel I lost control over the robot.
8) It makes the task more difficult to accomplish.

3.8 References

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

[2] R. Ozawa and K. Tahara, "Grasp and dexterous manipulation of multi-fingered robotic hands: a review from a control view point," Advanced Robotics, vol. 31, pp. 1030–1050, 10 2017.

[3] Y. Rybarczyk, E. Colle, and P. Hoppenot, "Contribution of neuro-science to the teleoperation of rehabilitation robot," in IEEE International Conference on Systems, Man and Cybernetics, vol. 4, 2002, pp.6 pp. vol.4

[4] A. N. Healey, "Speculation on the neuropsychology of teleoperation: implications for presence research and minimally invasive surgery," Presence, vol. 17, no. 2, pp. 199–211, 2008.

[5] T.C. Lin, A. Unni Krishnan, and Z. Li, "Shared autonomous interface for reducing physical effort in robot teleoperation via human motion mapping," in2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 9157–9163

[6] G. Salvietti, "Replicating human hand synergies onto robotic hands: A review on software and hardware strategies," *Front. Neurorobot.*, vol. 12, no. JUN, pp. 1–6, 2018.

[7] M. Bowman, S. Li, and X. Zhang, "Intent-Uncertainty-Aware Grasp Planning for Robust Robotic Assistance in Telemanipulation," in International Conference on Robotics and Automation(ICRA). Montreal,Canada: IEEE, 2019, pp. 409–415

[8] M. Santello, M. Bianchi, M. Gabiccini, E. Ricciardi, G. Salvietti,D. Prattichizzo, M. Ernst, A. Moscatelli, H. J¨orntell, A. M. Kappers,K. Kyriakopoulos, A. Albu-Sch¨affer, C. Castellini, and A. Bicchi, "Hand synergies: Integration of robotics and neuroscience for under-standing the control of biological and artificial hands," pp. 1–23, 2016.

[9] S. Li, X. Ma, H. Liang, M. Gorner, P. Ruppel, B. Fang, F. Sun, and J. Zhang, "Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 416–422.

[10] X. Wang, "Hand gesture recognition using sparse autoencoder-based deep neural network based on electromyography measurements," *Proc. SPIE 10597, Nano-, Bio-, Info-Tech Sensors, 3D Syst. II, 105971D*, 2018.

[11] C. Piazza, M. G. Catalano, S. B. Godfrey, M. Rossi, G. Grioli, M. Bianchi, K. Zhao, and A. Bicchi, "The softhand pro-h: A hybrid body-controlled, electrically powered hand prosthesis for daily living and working," IEEE Robotics and Automation Magazine, vol. 24, pp.87–101, 12 2017.

[12] G. Elkoura and K. Singh, "Handrix: Animating the human hand," 2003.

[13] Malvezzi, M., Gioioso, G., Salvietti, G., and Prattichizzo, D. (2015). Syngrasp: a matlab toolbox for underactuated and compliant hands. IEEE Robot. Autom. Mag. 22, 52–68. doi: 10.1109/MRA.2015.2408772

[14] Y. Lin, H. Zhao, Y. Zhang, and H. Ding, "A Data-Driven Motion Mapping Method for Space Teleoperation of Kinematically Dissimilar Master/Slave Robots," *2018 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2018*, pp. 862–867, 2018.

[15] W. Liu, A. Daruna, and S. Chernova, "Cage: Context-aware grasping engine," in2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 2550–2556.

[16] M. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," IEEE Transaction on Robotics and Automation,vol.5,no.3,pp.269–279, 1989.

[17] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in International Conference on Intelligent Robots and Systems, IROS 2010 pp. 1579–1585.

[18] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Task-Based Robot Grasp Planning Using Probabilistic Inference," IEEE Transactions on Robotics, vol. 31, no. 3, pp. 546–561, 6 2015.

[19] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Multivariate Discretization for Bayesian Network Structure Learning in Robot Grasping," in IEEE ICRA. IEEE, 2011, pp. 1944–1950.

[20] S. Li and X. Zhang, "Implicit Intention Communication in Human-Robot Interaction Through Visual Behavior Studies," IEEE Transactions on Human-Machine Systems, vol. 47, no. 4, pp. 437–448, 8 2017

[21] S. Li, X. Zhang, and J. D. Webb, "3-D-Gaze-Based Robotic Grasping Through Mimicking Human Visuomotor Function for People with Motion Impairments," IEEE Transactions on Biomedical Engineering, vol. 64, no. 12, pp. 2824–2835, 12 2017.

[22] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox, "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system," in IEEE ICRA 2020.

[23] D. Song, N. Kyriazis, I. Oikonomidis, C. Papazov, A. Argyros,D. Burschka, and D. Kragic, "Predicting human intention in visual observations of hand/object interactions," in Proceedings IEEE International Conference on Robotics and Automation, 2013

[24] S. Li, M. Bowman, H. Nobarani, and X. Zhang, "Inference of manipulation intent in teleoperation for robotic assistance," Journal of Intelligent & Robotic Systems,2020

[25] N. Hatakeya and K. Furuta, "Bayesian network modeling of operator's intent inference," in Human Factors Meeting. (IEEE), 6 2003, pp. 4–55

[26] H. Park, S. Lee, M. Lee, M. S. Chang, and H. W. Kwak, "Using eye movement data to infer human behavioral intentions," Computers in Human Behavior, vol. 63, pp. 796–804, 10 2016.

[27] S.Javdani, S.Srinivasa, and A.Bagnell, "Shared Autonomy via Hindsight Optimization," Robotics: Science and SystemsXI,vol.2015,72015.

[28] S. Reddy, A. D. Dragan, and S. Levine, "Where do you think you're going? inferring beliefs about dynamics from behavior," in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 1461–1472

[29] A.D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," International Journal of Robotics Research, 2013.

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

[31] M. Young, C. Miller, Y. Bi, W. Chen, and B. D. Argall, "Formalized task characterization for human-robot autonomy allocation," in 2019 International Conference on Robotics and Automation (ICRA),2019

[32] K. Mulling, A. Venkatraman, J.-S. Valois, J. Downey, J. Weiss,S. Javdani, M. Hebert, A. Schwartz, J. Collinger, and A. Bagnell, "Autonomy Infused Teleoperation with Application to BCI Manipulation," Robotics: Science and Systems XI, 2015.

[33] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic Virtual Fixtures for Robot-Assisted Manipulation," in Robotics Research Springer, 2007.

[34] J. D. Webb, S. Li, and X. Zhang, "Using visuomotor tendencies to increase control performance in teleoperation," ACC2016, pp.7110–7116.

[35] D. Feygin, M. Keehner, and F. Tendick, "Haptic Guidance: Experimental Evaluation of a Haptic Training Method for a Perceptual Motor Skill," in HAPTICS IEEE, 2002,pp. 40–47.

[36] M. Li and A. M. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in HAPTICS IEEE, 2003, pp. 125–131

[37] F. Abi-Farraj, C. Pacchierotti, O. Arenz, G. Neumann and P. R. Giordano, "A Haptic Shared-Control Architecture for Guided Multi-Target Robotic Grasping," IEEE Transaction on Haptics, pp. 270-285, 2019.

[38] S. Parsa, D. Kamale, S. Mghames, K. Nazari, T. Pardi, A. R. Srinivasan, G. Neumann, M. Hanheide and G. E. Amir, "Haptic-guided Shared Control Grasping: Collision-Free Manipulation," in IEEE International Conference on Automation Science and Engineering (CASE), 2020.

[39] H. Hu, J. Li, Z. Xie, B. Wang, H. Liu and G. Hirzinger, "A robot arm/hand teleoperation system with telepresence and shared control," in IEEE/ASME International Conference on Advanced Intelligent Mechatronics., Monterey, CA, 2005.

[40] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja,M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T.Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for perceiving and processing reality," in Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR)2019

[41] J. Sauro and J. R. Lewis, "Average task times in usability tests: What to report?" in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, ACM, 2010, p.2347-2350.

[42] G. Salvietti, L. Meli, G. Gioioso, M. Malvezzi and D. Prattichizzo, "Multicontact Bilateral Telemanipulation With Kinematic Asymmetries," in IEEE/ASME Transactions on Mechatronics, vol. 22, no. 1, pp. 445-456, Feb. 2017, doi: 10.1109/TMECH.2016.2606895.

[43] S. Reddy, A. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," in Proceedings of Robotics: Science and Systems, Pittsburgh, Pennsylvania, June 2018.

[44] A.M. Lund, "Measuring usability with the use questionnaire," Usability interface, vol.8, no.2, pp.3–6, 2001.

CHAPTER 4

DIMENSION-SPECIFIC SHARED AUTONOMY FOR HANDLING DISAGREEMENT IN TELEMANIPULATION

A paper accepted for publication on *IEEE Robotics and Automation Letters (RAL) 2023*

Michael Bowman[8], and Xiaoli Zhang[9]

4.1 Abstract

One of the fundamental questions in shared control is how to allocate control power to the human and robot effectively. Conventional arbitration policies often define a uniform singular scalar for all 6 DOFs to blend human input and robot assistance. However, this singular scalar over-dominates some dimensions of the inputs and provides insufficient assistance in other dimensions. Thus, current shared control can support simple telemanipulation tasks such as pushing, pressing, and simple positional control but is limited in tasks with more DOFs like rotational motion. A dimension-specific arbitration policy is developed to customize the control arbitration along each DOF to fill the gap. It looks at whether the robotic assistance is too timid or aggressive along each DOF and determines the arbitration magnitude according to disagreement levels of control allocation and the user's willingness to accept assistance. The user's willingness is estimated from a feedback psychology model. The method has higher similarity and ratio of agreement between the human and robot (lower over-dominance) over

[8] Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines, 1610 Illinois Street, Golden, Colorado, 80401, USA.
[9] Corresponding author, Associate Professor, Department of Mechanical Engineering, Colorado School of Mines, 1610 Illinois Street, Golden, Colorado, 80401, USA.

existing methods and, simultaneously, improves the task performance. This arbitration strategy is expected to increase the adoption of teleoperation for object manipulation.

4.2 Introduction

Teleoperating a robot for a manipulation task is often difficult and complex due to indirect visualization, indirect manipulation with the robot, and physical discrepancies between a human and robot hand [1][2]. Shared control injects autonomy into an operator's actions to overcome these issues. Shared autonomy is desirable for teleoperation applications, including industrial inspection and maintenance, disaster relief, or assistive living.

Object telemanipulation requires more than simple positional control, realizing the system requires rotational and finger control. These motion components provide a natural split in handling complex tasks. Operators handle these components in their preferred approach. For example, the operator may obtain a position near an object, then begin to handle the rotations and end with finger control. Here an iterative process occurs by the operator to refine each step until they achieve the goal. Certain combinations of refinement will be more intuitive to different operators, like either handling the depth before lining up the robot in an x-y plane or vice versa. Naturally, operators likely only adjust a few components while holding others nearly constant (i.e., holding a position but adjusting the pose angle or finger posture) to succeed in a grasp [6].

The autonomous actions may differ from the operator actions along all or a portion of these motion components. For instance, given the same high-level target pose, the autonomous agent and operator may have differing motion strategies to achieve it (Fig. 4.1: human operator and robot agent). Disagreement occurs due to misperception or preference of the operator. Neither agent is incorrect in their actions to accomplish the task, nor is the robot producing suboptimal action plans. This disagreement is fundamentally inherent in shared control, and precautions are

necessary to prevent over-dominance from the robotic agent. Further, the robot and human may only partially agree, where they may agree in the positional component and differ in the rotational component. Fig. 4.2 shows this partial agreement.

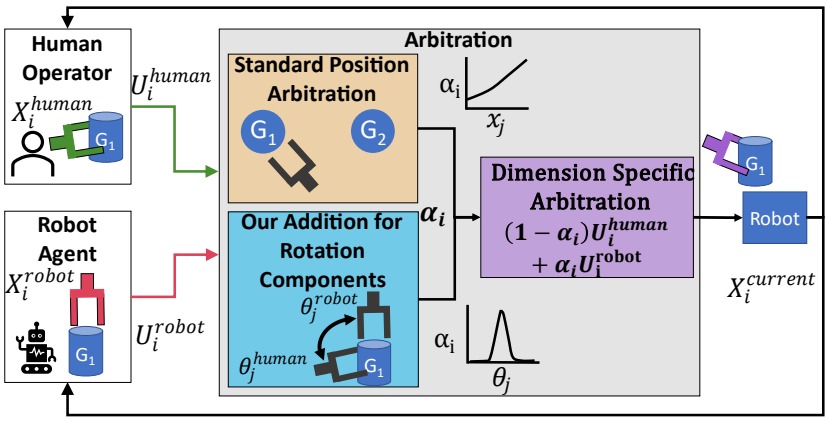


Figure 4.1 The overview of the proposed telemanipulation control paradigm. Our contributions include partitioning dissimilar dimensions, devising independent arbitration policies to handle control allocation disagreement, and applying the appropriate dimension-specific control authority.

Determining the appropriate level of autonomy in teleoperation is the key research topic to realize telemanipulation scenarios fully. The most common state-of-the-art strategy in shared control is to use a linear blending strategy with a universal arbitration term, α. The control authority is a singular scalar value that influences every dimension of the control rather than individual components. Specifically, it only includes positional components, not rotational ones, to dictate arbitration along a trajectory. This strategy cannot handle partial agreement that arises in telemanipulation scenarios, especially towards the end of the trajectory where conventionally, the robot assumes most control (α=1). Consider a case where the robot is near the positional goal, and the operator attempts to refine the position before adjusting the rotation. With the conventional strategies, the robot will dominate not only the positional components but the rotational aspect too as α approaches 1. This is despite the operator not yet focusing on the rotational components to grasp an object successfully and restricts their ability to alter them. The

over-dominance restricts the operator in how they want to complete the task (i.e., grasp the object from the side rather than the top like in Fig. 4.1). This leads to an operator's attempt to refine actions but sees the control as too aggressive thus make drastic changes [3]. Over-adjustments and strong reactionary responses are due to the low control authority. However, the operator's initial intuitive refinement is to be subtle. This is especially common for telemanipulation applications [4][5].

This work aims to provide autonomy to aspects the robot and user agree on and reduce assistance in areas where they disagree. Otherwise, the arbitration leads to operator resistance. Toward this aim, a dimension-specific control authority to focus on individual components to succeed is developed. The contributions are as follows:

1. Break down dimensional aspects with respective motion components, and quantify dimension-specific disagreement and the human's willingness for assistance.
2. Develop a dimension-specific arbitration strategy for non-dominating assistance along different motion components.
3. Validate and evaluate the dimension-specific arbitration strategy in telemanipulation tasks.

4.3 Related Work

Current telemanipulation focuses on improving hand motion mapping capabilities and lacks shared autonomy strategies. These mapping strategies use model-based approaches such as DexPilot [7] or use synergies [8][9]. These methods consider orientation and finger mapping. However, deployment still relies on an operator to fully control a system that is difficult to maneuver and slower than normal actions an operator would take [7]. Despite improving the physical discrepancy issues, the operator struggles to overcome the disembodiment problem,

where they face indirect perception and manipulation of the robot environment and lack dynamic interaction with objects.

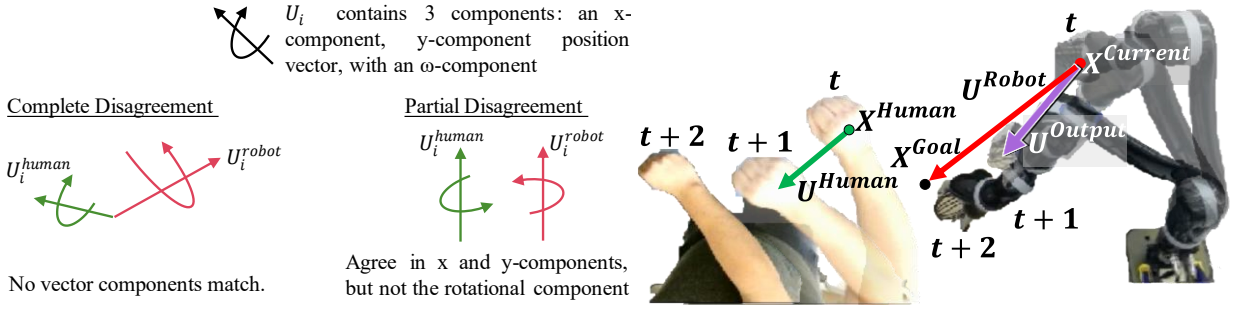


Figure 4.2 Agreed positional components influence and dominate arbitration models, leaving rotational components more robot-dominated than intended. An example of an operator's and robot's actions for 3 timesteps with partial agreement in the vison-based telemanipulation scenario.

Introducing shared control to circumvent the above issues has empowered the autonomous agent with its knowledge of the environment. However, shared control with linear blending has only been used on simplified approaching problems [10]. The arbitration is a single scalar based on the robot's confidence in a human's intended goal [11][12][13] or empirically defined to increase from 0 to 1, where the higher autonomy levels are near the goal [6]. This standard strategy leads to over-dominance of a robot's actions near goal states; an operator may disagree with the output actions. Operators have sometimes reported a desire to retain control despite lower task performance [14][15]. Often seen as inferencing issues to achieve a better arbitration, research lacks questioning the structural issues for linear blending techniques and handling disagreement.

The work in [3] provides insight on reframing a multi-inference sub-policy to quantify disagreement between operators and robotics assistance plans for 2D approaching tasks. Other strategies quantify disagreement between operator and robot actions and then apply minimum assistance to ensure safe control [12][16][17]. The disagreement quantification works for both

discrete and continuous domains [18]. Alternatively, handling disagreement can be viewed as mutual adaptation or adapt to adaptation [19][20][21]. One aim of mutual adaptation is to have an online adjustment of control authority based on the operator's action. However, all these strategies obfuscate the action's individual components by analyzing the entire vector and have rarely been used in telemanipulation fields.

4.4 Methodology

4.4.1 Disagreement of Control Allocation

Both agents have pose states denoted as $X \in \mathbb{R}^6$. Where X is split between 3D positional and three independent rotational dimensions (roll, pitch, yaw), thus $X \in \{x_j, \theta_j\}, j = 1,2,3$. Both the robot and operator apply actions $U \in \{\dot{x}_j, \dot{\theta}_j\}, j = 1,2,3$. We assume the goal state, X^{goal}, is given and known by the robot and the operator, and the optimal robot policy U^{robot} is set as to go towards X^{goal} as seen in Fig. 4.2. The operator is free to choose actions, U^{human}, other than those towards the X^{goal} to achieve the task, thus going to their own state, X^{human} (i.e., the current human hand pose as seen in Fig. 4.2). The standard linear blending arbitration is augmented to a dimension-specific one for both the U^{human} and U^{robot}, as in Eq. 4.1, where i is each dimension of the control vector (i=1,2,…,6). The resultant action, U_i^{output}, applies to the robot dynamics with the current robot state, $X^{current}$, to obtain the next state, X^{next} in Eq. 4.2.

$$U_i^{output} = U_i^{robot}\alpha_i + U_i^{human}(1 - \alpha_i) \tag{4.1}$$

$$X^{next} = f\left(X^{current}, U^{output}\right) \tag{4.2}$$

α_i can be summarized as the L1 distance (or L2 if preferred) towards X^{goal}, from an initial robot state, X^{start}, and $X^{current}$, to determine the control authority. Eq. 4.3 shows an L1 distance using $X^{current}$ in the standard error function (erf).

$$\alpha_i = 1 - \text{erf}\,\frac{\left\|X^{goal} - X^{current}\right\|_1}{\left\|X^{goal} - X^{start}\right\|_1} \tag{4.3}$$

For the baseline case (denoted as the Standard (S) controller), all α_i are the same value as Eq. 4.3. The α_i is only from the robot's perspective as $X^{current}$ solely dictates it to promote as much assistance as possible ($\alpha_i = 1$). However, we must account for misalignment, as previously discussed. So, we alter α_i through quantifying disagreement of control allocation. The disagreement between both the current human (X^{human}) and robot ($X^{current}$) is the difference in expected control allocation. The expected control allocation is where the current robot expects to provide α help based on $X^{current}$ and the operator expects to gain α assistance based on X^{human}. There is a mismatch between $X^{current}$ and X^{human}, thus the expected α is different, due to misalignment. So, Eq. 4.4 makes two changes to Eq. 4.3. The first is to update it as a dimensional difference. The second is to represent each *agent's current state position* (x_j^{human} or $x_j^{current}$) as a probability of expected control allocation (as if each agent were to apply their position to Eq. 4.3 and produce an α_i), where *A* represents either the human (H) or the robot (R) agent and x_j^{agent} is either x_j^{human} or $x_j^{current}$.

$$P(A_i) = 1 - \text{erf}\frac{\left\|x_j^{goal} - x_j^{agent}\right\|_1}{\left\|x_j^{goal} - x_j^{start}\right\|_1} \; \forall i = \{1,2,3\}, j = \{1,2,3\} \tag{4.4}$$

Eq. 4.4 is inappropriate for quantifying rotation dimensions to account for circular variables. Therefore, a more natural, appropriate probability distribution for rotational control allocation is the Von Mises distribution [3] in Eq. 4.5. Functions with similar characteristics that can handle circular variables can also be applied. For instance, our rotations are independent, leading to a univariate probability function in Eq. 4.5.

$$P(A_i) = \frac{e^{\kappa \cos\left(\theta_j^{agent} - \theta_j^{goal}\right)}}{2\pi I_0(\kappa)} \forall i = \{4,5,6\}, j = \{1,2,3\} \tag{4.5}$$

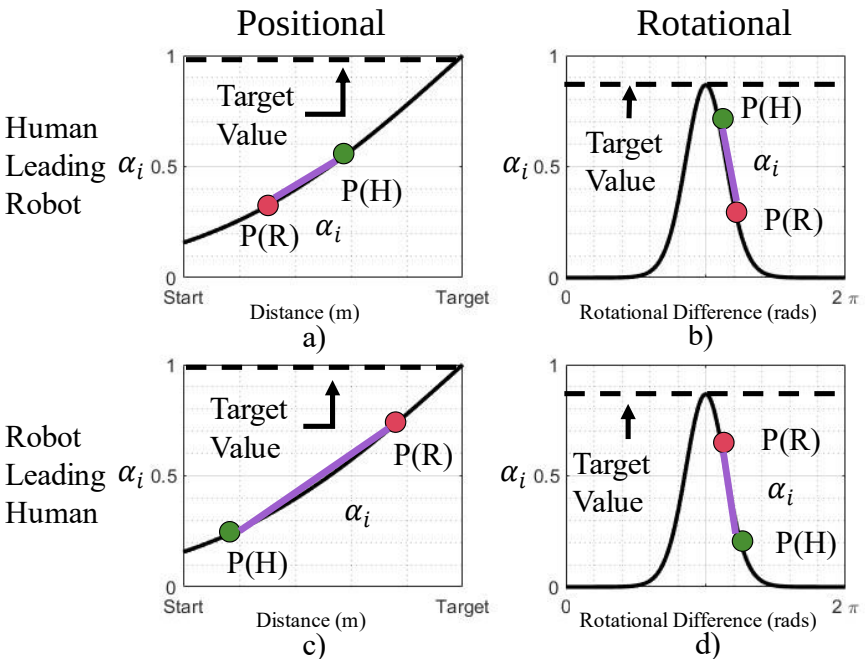


Figure 4.3 The expected control authority by the human (H), and robot (R) using Eqs. 4.4 (left column) and 4.5 (right column) for specific dimensions. The target value corresponds to the goal state known by both agents. Conventional systems take P(R) as the true α. Top row: human leads robot, meaning lower assistance, α, in conventional systems (P(R)<P(H)). Bottom row: robot leads human, meaning higher assistance, α, in conventional systems (P(R)>P(H)). The purple regions are potential α from our strategy in Eq. 4.8. Our approach aims to find a better α than conventional means by evaluating the operator's willingness, B_i, which contextualizes the U_i^{human} based on their current progress in the task. A high B_i will lead α_i being closer to the conventional strategy (P(R)), while a low B_i will lead to α_i to be near P(H).

κ is an empirically defined variable, and $I_0(\kappa)$ is the modified Bessel function, and θ_j^{agent} refers to rotational dimensions of X. Eq. 4.4 and Eq. 4.5 are shown in Fig. 4.3 as solid black lines. In Fig 4.3, the human and robot have differing expectations of control allocation along a singular dimension. $P(H_i)$ and $P(R_i)$ are the bounds of disagreement for control authority seen by the operator and the actual robot pose. In conventional systems, assistance is either *timid* (the robot's provided assistance is behind the human's expected assistance in Fig. 4.3's top row) or *aggressive* (the robot's provided assistance is ahead of the human's expected assistance in Fig. 4.3's bottom row). With our system, we aim to determine whether assistance is truly timid or aggressive (ultimately determining the appropriate level of assistance to provide)

by including the operator's willingness to accept assistance (which contextualizes U_i^{agent} in terms of current progress toward the goal). For instance, if the operator has a low willingness to accept assistance for conventionally aggressive assistance (bottom row of Fig. 4.3), then the α will be set closer to P(H). Alternatively, if they have a high willingness to accept assistance, then the conventionally aggressive assistance is not true, the operator is willing to gain aggressive assistance, and α can be set closer to P(R).

4.4.2 User's Willingness Model

The operator's willingness to accept assistance stems from their desire to change the current control authority in the dynamic process. Determining the change to this authority level can temper too aggressive and boost too timid assistance. The operator's current desire and preference towards the robot's assistance is identified by looking at both agents' actions along each dimension. A probabilistic approach, inspired by psychology feedback models [22][23][24], is developed to determine willingness to accept assistance. The psychology model discusses a proposed feedback mechanism for how people change their behaviors in a task based on current progress, previous experiences, and the current rate of progress. Further, a Beta prime distribution uses these components to determine when a person will likely pick up the pace or back off. Lastly, the model discusses how the pace changes for the individual dimensions; as one dimension slows, another takes precedence. The feedback model was conceptual and discusses the pace in the abstract; we adapt them to a mathematical framework and apply them to the shared autonomy. Compared with the existing model-free data-driven approach that defines adaptability through a Mixed Observable Markov Decision Process [20], the primary benefit of using the mathematical model is to provide the robot with the operator's willingness to accept the control authority in a human-like manner. The proposed model can be adapted in two ways.

The first is to move it into the motion domain rather than a conceptional abstraction of rate changes to complete tasks. The second is to adapt the model to give a predictive or feedforward effect rather than a reactive calculation (i.e., computing acceleration from measured U_i^{human}). Together, both adaptations account for the expected way an operator would move toward the goal given their current progress (correctly transforming or normalizing the U_i^{agent} into the model's concept of speed change). Due to the predictive nature of the model and the transformation, this is referred to as the "expected speed change." To make these adaptations to the psychology model, we identified four components: 1) current actions, 2) previous actions, 3) current progress in the task, and 4) aggregated previous experiences. The first and second components are instantaneous reactionary decisions, while the third term looks at a longer planning window for the current trial. The last term influences the operator's expectations in completing the trial based on previous experiences. The last two terms are what the operator relies on to know if they are slow or fast in this trial compared to previous trials. Eq. 4.6 combines these terms to quantify the agent's expected speed change based on their current (first term) and last actions (second term), current progress (third term), and aggregated previous experiences (last term). Where N is the total number of previous attempts in doing the task, and n is the index for a specific attempt. Eq. 4.6 aims to quantify/normalize whether the operator will increase or decrease their velocity based on their current progress compared to previous experiences in the task.

$$\psi_i^{agent} = U_i^{agent} - \frac{X_i^{current} - X_i^{previous}}{\left\| X^{current} - X^{previous} \right\|_2} - \frac{X_i^{current} - X_i^{start}}{\left\| X^{current} - X^{start} \right\|_2} + \frac{1}{N} \sum_{n=0}^{N} \frac{X_{i,n}^{terminal} - X_{i,n}^{start}}{\left\| X_n^{terminal} - X_n^{start} \right\|_2} \quad (4.6)$$

The variable *agent* refers to the human agent (U_i^{human}) or the robot agent (U_i^{robot}) actions. Each rate (ψ_i^{human} and ψ_i^{robot}) is then used in a Beta prime distribution with hyperparameters *b*,

p, and q. The hyperparameters impact the shape and scale and are empirically chosen based on human speed in manipulation tasks and the robot's velocity limits. The heuristic described in [23] inspires the Beta prime distribution. The distribution allows for normalization and quantification compared to the heuristic in [23] and provides context toward what the operator is attempting to achieve with their speed change. ψ_i^{robot} is necessary as it is the reference to compare the sampling for ψ_i^{human} (which is used to see if it is faster or slower). The robot policy, U_i^{robot}, determines ψ_i^{robot} before the arbitration takes place which is discussed in the next section. ψ_i^{robot} is necessary as it influences the shape of the curve and is the mean value for the probability density function to evaluate the ψ_i^{human} (which is calculated after measuring U_i^{human}). The ψ_i^{human} evaluation produces a single deterministic value (which we call willingness, B_i) for the specific time point. However, this could be extended to a stochastic sampling approach by either 1) taking multiple time point samples or 2) utilizing uncertainty in the measurement of U_i^{human}. It should be noted that the rates are dynamic, in that, at each time point ψ_i^{robot} and ψ_i^{human} will change to new values. The willingness, B_i, to accept assistance is between 0 (fully unwilling) and 1 (fully willing).

$$B_i = \beta(|\psi_i^{human}|, |\psi_i^{robot}|, b, p, q) \tag{4.7}$$

We obtain a different B_i curve for each dimension. Fig. 4.4 shows Eq. 4.7 and its three distinct regions. The left part of the first region is when the operator has little or no ψ_i^{human} compared to ψ_i^{robot}. When the ψ_i^{human} is near 0, two aspects are needed: 1) constant U_i^{human} so the first and second terms of Eq. 4.6 can cancel out, and 2) $X_i^{current}$ configuration that leads to the third and fourth terms being either 0 (the third term is 0 when $X_i^{current} = X_i^{start}$ and the fourth term is 0 with no previous trials attempted) or cancel out ($X_i^{current} =$

X_i^{goal}). ψ_i^{human} near 0 results in an unwillingness to accept assistance ($B_i = 0$). One example of when ψ_i^{human}= 0 occurs is near the goal—resulting in the third and last terms of Eq. 4.6 canceling out—with the operator no longer making adjustments to achieve the task, leading to the first and second terms approaching 0. Note that this is one example, but ψ_i^{human}= 0 could occur anywhere along the trajectory as long as the four terms cancel (e.g. when the operator moves in the y-direction in Fig. 4.5b, the x-dimension ψ_i^{human} is approximately 0 leading to the purple distribution closely resembling the green one due to B_i being approximately 0). The other critical point is when ψ_i^{human} is slightly lower than ψ_i^{robot} ($B_i = 1$). In this scenario, the operator's expected speed change is not far off from the ψ_i^{robot} and the expectation is the operator desires to achieve ψ_i^{robot}.

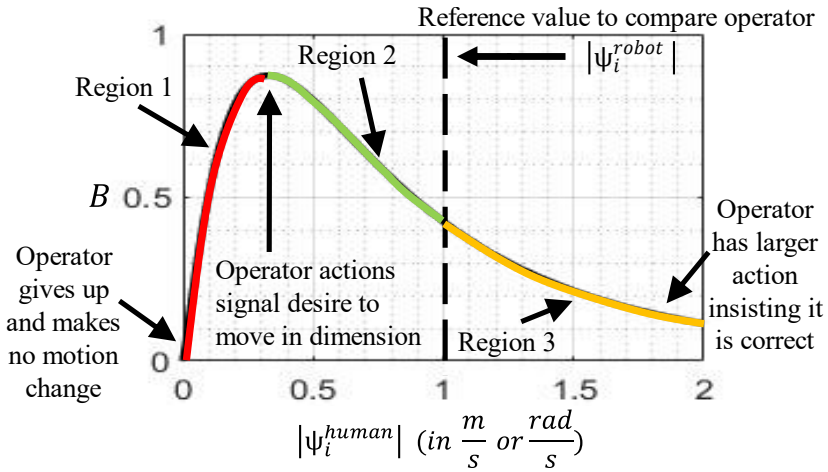


Figure 4.4 The operator's willingness to accept assistance for a single dimension. The dashed line represents the reference which is the expected speed change of the robot, ψ_i^{robot}, from the optimal robot policy (U_i^{robot}). Three regions exist. The first (in red) shows when no expected speed change ($\psi_i^{human} = 0$) occurs, which implies the operator is unwilling to change and accept assistance. As the ψ_i^{human} increases, it implies they are willing to change and accept assistance in this dimension. The second region (in green) shows that as ψ_i^{human} gets closer to ψ_i^{robot}, corrections to ψ_i^{human} are less warranted. The third region (in yellow) shows when the ψ_i^{human} is faster than ψ_i^{robot}, they think their expected speed change is more desired as it is above the ψ_i^{robot}, and that the robot should be following suit. Note that there are two cases B_i can be 0, either when ψ_i^{human}=0, or when ψ_i^{human} is very fast.

The second region shows ψ_i^{human} nearing ψ_i^{robot}, where the operator's B_i is reduced and changes are less warranted In this region, the psychology model of [23] states that the operator decreases their B_i as ψ_i^{human} nears the ψ_i^{robot} because they lose their sense of urgency to achieve it and their expected speed change is in an acceptable bound. The third region occurs when ψ_i^{human} is much faster than ψ_i^{robot}, implying the operator will have a strong desire to move in this direction. In this scenario, the operator does not care as much about the help because it moves slower. Note that in the third region, when ψ_i^{human} becomes extremely fast, B_i will go to 0.

4.4.3 Handling Disagreement in a Dimension-Specific Setting

The dimension-specific arbitration uses the disagreement between control authority allocation and the user's willingness. P(H_i) and P(R_i) act as bounds, where a better arbitration exists between them in Eq. 8. The new α_i tempers too aggressive control while boosting too timid control.

$$\alpha_i = (\mathrm{P}(R_i) - \mathrm{P}(H_i))B_i + \mathrm{P}(H_i) \tag{4.8}$$

After α_i from Eq. 4.8 is calculated, the output action from Eq. 4.1 is calculated. Each dimension will assume a different level of α_i. An example of possible α_i is shown in Fig. 4.3a-d in purple. For instance, consider if the current situation resembles Fig. 4.3c, where an operator is behind the robot. If ψ_i^{human} is near 0 and thus B_i=0 (region 1 of Fig. 4.4), then $\alpha_i = \mathrm{P}(H_i)$. This gives the operator more control and tempering too aggressive assistance. If ψ_i^{human} begins to increase where B_i=1 (near the max of Fig. 4.4), then at this time point, the control authority goes towards the robot at $\alpha_i = \mathrm{P}(R_i)$. Providing more active assistance in tune with the operator. If ψ_i^{human} continues much faster at the next time point where a transition from Fig. 4.3c to Fig.

4.3a occurs, the B_i goes toward 0 (region 3 of Fig. 4.4) which pulls the robot toward the human's expected control authority as $\alpha_i = \mathrm{P}(H_i)$ and boosts too timid assistance ($\mathrm{P}(H_i)$ is dynamic, and changes, so, they are not the same $\mathrm{P}(H_i)$). This does not mean the robot provides zero assistance but uses the operator's expected control authority allocation. In this transition phase, α_i first increases from $\mathrm{P}(H_i)$ to $\mathrm{P}(R_i)$ in Fig. 4.3c then keeps increasing from $\mathrm{P}(R_i)$ to $\mathrm{P}(H_i)$ in Fig. 4.3a. $\mathrm{P}(H_i)$, $\mathrm{P}(R_i)$ and B_i change along each dimension; thus, each α_i will be different. The linear model for Eq. 4.8 should not impact the operator's perception of rotational assistance as they could hold similar α_i but with two different poses. Rather the human-robot policy misalignment would have a higher influence (i.e., the robot wants $\pi - \theta_j^{robot}$, and the operator has the desired $\pi + \theta_j^{human}$).

4.5 Experimental Results

4.5.1 Simulation Setup

For easier visualization of the results and qualitative analysis of each α_i and the outcome of their respective actions, a simplified telemanipulation scenario with three DOF (two translational and one rotational) shown in Fig. 4.5a is used. Success occurs when the robotic hand is radius, **R**, away from the goal position and within a radian tolerance, δ, from the normal direction to the goal position. The tolerances resemble real scenarios where a point goal is not enough. The simulated human trajectories sample a probability distribution that attempts to grasp the object's handle. The robot's target pose has noise at the target position to simulate when a robot is not precise enough to achieve the desired pose. 1000 simulated trajectories were used to compare the baseline S controller (mentioned in III.A) and the Dimension-Specific (DS) controller. Fig. 4.5b shows all three trajectory distributions. The robot policy for the DS and S controllers is to follow

the shortest distance and smallest angle to the goal. The DS controller starts with a B_i of 1 (fully willing) because no previous human actions indicate otherwise.

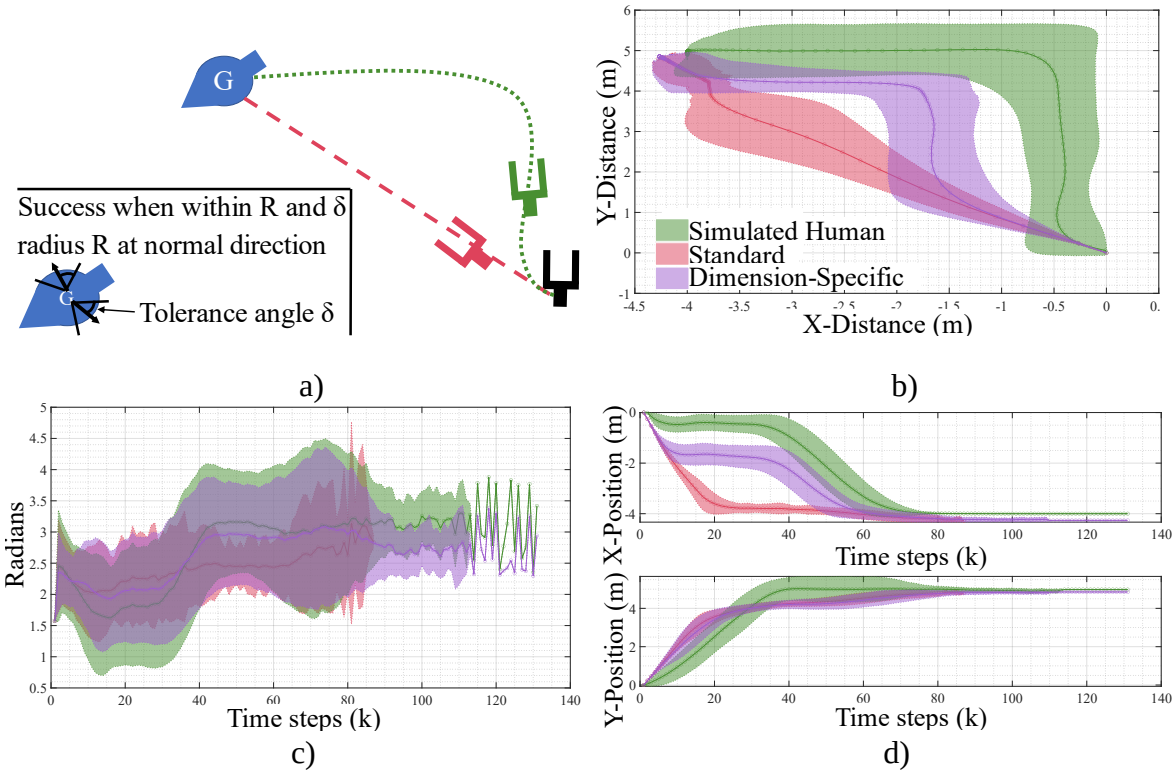


Figure. 4.5 a) The simulation environment. The stochastic human has a policy in green and the robot policy is in red. The robot policy follows the shortest distance and smallest angle to the goal. The S controller follows Eq. 4.3 to set the α_i for all i. The starting pose is always identical (black). b) the distributions of all simulated trajectories. c) is the heading across time, d) is the x and y directions across time. The plots in d) also shows the trends of P(H) in green, and P(R) as red (S) and purple (DS). There is a large difference in the human's and S controller's x-position causing disagreement in control allocation of P(H) and P(R). There is less disagreement between the human and DS.

4.5.2 Simulation Results

Qualitative analysis comes in two forms. The first is the appearance of output robot trajectories for the same human trajectory. Our approach aims to align more with the principal axis the operator is trying to achieve. Fig. 4.5 shows this is evident compared to the singular α that over-dominates the trajectory (i.e., although the operator is moving primarily in the y-direction, the S controller forces motion in the x-direction). Fig. 4.5d also demonstrates the

trends of P(H) in green and P(R) as red (S) and purple (DS). There is a large difference in the human's and S controller's x-position, causing disagreement in the control allocation of P(H) and P(R). There is less disagreement when considering humans and DS. The second qualitative test is to compare α_i.

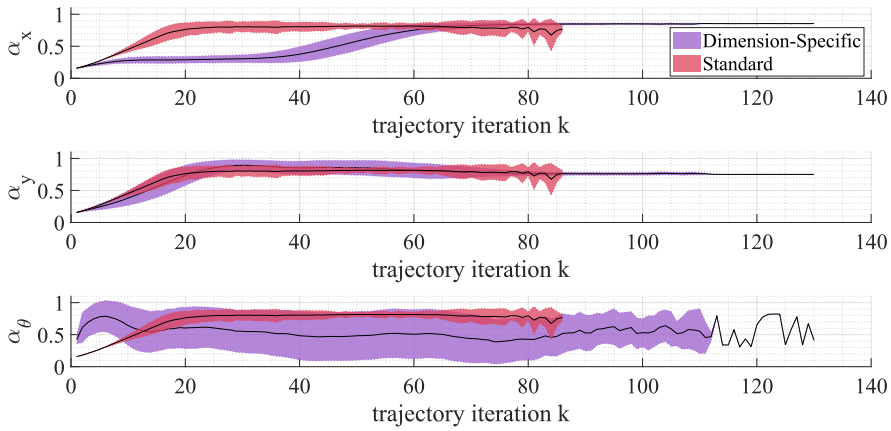


Figure 4.6 α_i distributions for the trajectories shown in Figure 4.5.

The α_i in Fig. 4.6 corresponds to the distributions in Fig. 4.5. The expected operator actions proceed in the y-direction before the x-direction. The DS controller should provide more control authority in y before the x-direction. In the top plot of Fig. 4.6, the α_i is relatively low. As the operator is not moving in this dimension, giving more control to the operator in this dimension can prevent the robot from over-extending its motion in an undesired path. Whereas, in the middle plot, the operator moves toward the component goal making the α_i go higher, and reduces the authority the operator experiences, allowing the robot to provide more assistance in this direction. Further, the S controller calculates Eq. 4.3 before developing an output action with Eq. 4.1, which is why the same alpha is along all components. The DS controller value for α_θ has two trends. The first trend is at the beginning of the trajectory; the human's current and target heading are close to one another. The human angle does not have significant deviations, so the robot is afforded more control with α_θ. The second trend occurs when the human heading begins

to deviate, signaling to the robot that the human wants to assume more control to achieve a desired heading, driving α_θ lower overall. This becomes more apparent in a real-world scenario.

4.5.3 Telemanipulation Setup

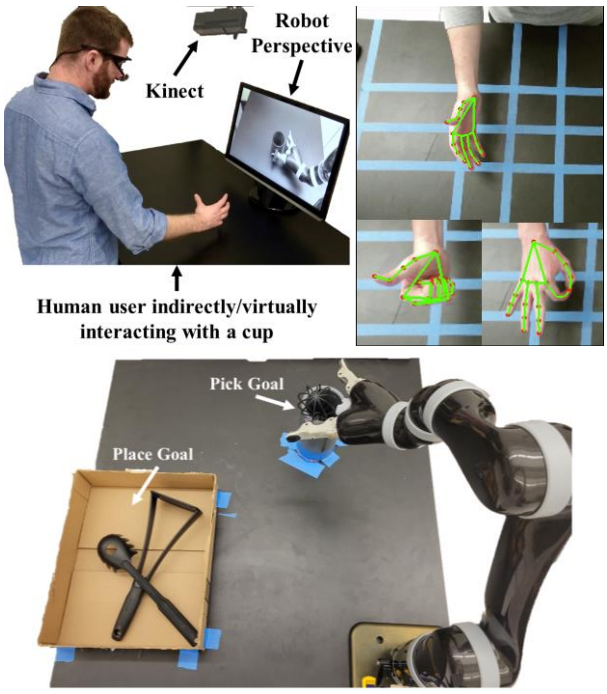


Figure 4.7 Operators see the robot perspective on the screen while they move their hand freely over the table. The Xbox Kinect captures hand posture to extract the palm pose and the amount each finger is open or closed. The information is then sent to the robot to execute the motion.

Figure 4.7 describes the telemanipulation experimental setup. An Xbox Kinect takes an RGB-D image, and Mediapipe [25] extracts features used as inputs for the robot control. The operator directly controls the hand pose (position and orientation) and the amount each finger opens and closes, eight variables in total. The operator aims to grasp kitchen utensils from a holder and place them in a bin. A failure occurs if an object does not reach the bin or lands on the table. The operator is free to move how they wish to accomplish the task. The robot is given pick and place goal poses with induced noise.

The institution's Institutional Review Board (IRB) approved a set of experiments. Before participating in the study, written consent was obtained, acknowledging they understood the robot setup, the purpose of the study, and the potential risks involved (arm soreness). Three trials were conducted for each arbitration strategy for each participant. A trial consisted of three runs; each run had a different object. A randomized order of the control modes was used to reduce learning effects. 10 participants volunteered (a total of 90 runs for each arbitration strategy) in the pilot study. The volunteer breakdown included 25.5 ± 4.15 years of age, with 4 women and 6 men involved.

4.5.4 Telemanipulation Results

Task performance. The quantitative analysis is broken down by success rate, completion time, cosine similarity, and the ratio of agreement. Due to the smaller number of trials compared to the simulation results, the metrics have a different analysis. The success rate uses a Laplace Estimate over the Maximum Likelihood Estimate to reflect the true success rate. Likewise, an adjusted-Wald 95% confidence interval creates the bounds of success. An N-1 Chi-Squared test determines statistical significance. All comparisons are in Table I. The DS controller achieves a higher success rate than the S strategy, but no statistical significance is found ($p=0.092$). The success rate is low for both strategies due to a few factors that are not related to operator error as our scenario was designed to reenact real world challenges: 1) induced noise on the 2 goal locations, 2) induced time delay of the teleoperation scenario, 3) imperfect information as there was a single camera view so operators must overcome the depth perception. The induced noise of the goal state provides a more realistic scenario where imperfect perception causes the robot to have uncertainty about the goal. The induced time delay provides a realistic scenario. Therefore, both uncertainties reduce the success rate of the task.

Time data is notorious for not being positively skewed from a normal distribution [26]; thus, a log transform of the data must be used. The geometric mean and 95% confidence interval is used. A two-sample t-test is used to determine the statistical significance. The time data was the total trial data (the summed three runs); it includes both success and failures to determine if the control strategies limit an operator in multi-step tasks. The DS controller outperforms the S with a statistical significance of p=0.0141. The task performance favors the DS controller with a better success rate and trial time than the S controller.

Table 4.1 Teleoperation Results For Their Respective Means and Confidence Intervals

Mode	Success Rate	Trial Time (s)	Cosine Distance	Ratio of Agreement
DS	0.446 [0.346,0.547]	**152.4** **[125.0,185.8]**	**0.076** **[0.065,0.089]**	**0.981** **[0.980,0.982]**
S	0.326 [0.234,0.425]	223.4 [176.2,283.3]	0.400 [0.385,0.417]	0.895 [0.893,0.897]

DS means Dimension-Specific, S means Standard. Bold means this performs better in this metric by statistical significance.

Assistance quality of the robot control. Objective measures for determining user agreement widen the gap further by exploring over-dominance. Two measures used to identify this are the cosine distance and ratio of agreement. The former is defined:

$$\sum_{actions} 1 - \frac{U^{human} \cdot U^{output}}{\|U^{human}\| \|U^{output}\|} \tag{4.9}$$

The aim is to determine the alignment of the control vectors for the robot and the operator. A score of 0 is considered optimal. The ratio of agreement is defined:

$$\phi = \frac{\sum ag}{number\ of\ actions}, 0 \leq \phi < 1$$

$$ag = \begin{cases} 1, if\ U^{human} \cdot U^{output} > 0 \\ 0, if\ U^{human} \cdot U^{output} \leq 0 \end{cases} \tag{4.10}$$

This metric determines the number of output actions, U^{output}, which aligns with the human operator. The aim is to normalize the number of actions taken and determine a better ratio of agreeable actions. A score of 1 is considered optimal. The geometric mean and 95% confidence intervals for both metrics were generated and placed in Table 4.1. The two-sample t-test is used to determine the statistical significance. The cosine similarity and ratio of agreement demonstrate that the DS controller follows human actions more than the S controller. The DS controller outperforms the S controller for both measures with $p<0.001$. This strongly indicates that the DS controller assists in the direction the operator is moving while further showing the over-dominance of the robot actions occurring in the S controller. The DS controller shows greater alignment with the operator and improved task performance with the objective measures. The conclusion is that the DS controller benefits the operators in their preferred dimensions.

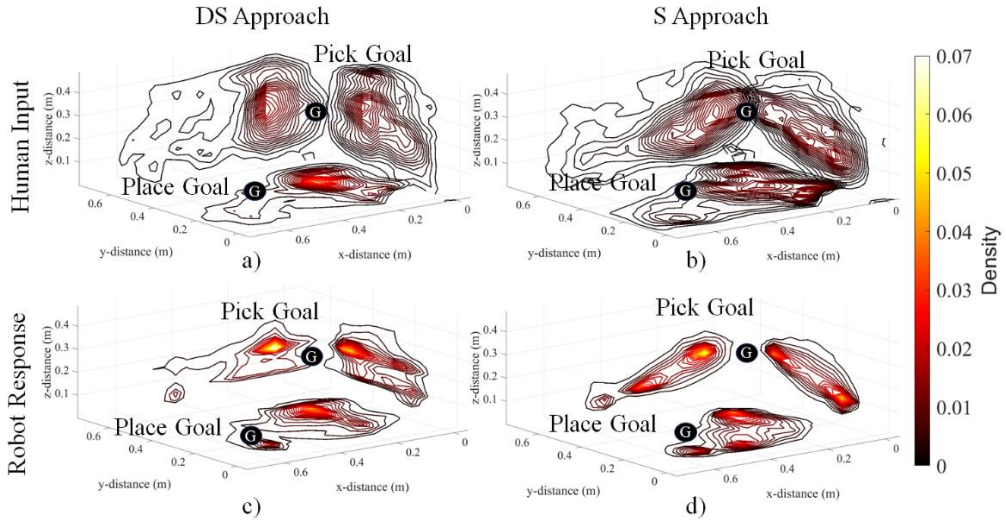


Figure 4.8 Trajectory distributions in $\mathbb{R}^3$. a) Human input for DS, b) Human input for S, c) Robot response for DS, d) Robot response for S. A Jensen-Shannon divergence metric compares the distributions. a) to b) divergence is 0.1293, c) to d) is 0.1896, a) to c) is 0.2905, and b) to d) is 0.3017.

To further assess whether the assistance benefits the operator, qualitative analysis can be done by comparing the operator and robot trajectories. Our video shows the over-dominance issues regarding rotation differences.

The robot output poses differ significantly for two similar human trajectories. The difference has major repercussions on the overall perception and adaptation of the operator as they must contend with a robot that dominates control near a goal location. The over-dominance causes some operators to hesitate as it is not similar to their hand posture; the operator moves to regain control. Another approach is to compare the trajectory differences of the operators and corresponding outputs, $X^{current}$, shown in Fig. 4.8. A Jensen-Shannon divergence measure compares the distributions which are bounded between 0 and 1. Two trends are apparent when analyzing the trajectory distributions. The first is that the S controller forces the robot behavior to act with a smaller deviation compared to the DS controller. This is evident by how Fig. 4.8d shows a rather straight-line trajectory between the 2 goal locations (pick and place goal), whereas Fig. 4.8c shows a more diverse suite of trajectories to aid the operator towards the goals. The divergence between the DS and the S controllers' outputs is 0.1896. The second discernable pattern is how much the robot conforms to the operators. The S controller (Fig. 4.8b to Fig. 4.8d has a divergence of 0.3017) conforms less to the human than the DS controller (Fig. 4.8a to Fig. 4.8c has a divergence of 0.2905). The higher divergence between the S controller means it conforms less than the DS controller. The more concentrated regions in Fig. 4.8b compared to Fig. 4.8a suggest that operators were fighting the robot near the pick goal location, which leads to the credence that the S controller over-dominates the control and does not allow the human to compensate. All the divergence values are relatively low and similar to one another as there are

only 2 goal locations to move between (meaning there are limited strategies for reaching both goals).

4.6 Discussion

A strategy to enable dimension-specific shared control has been developed. By breaking down the problem into subproblems, each dimension has an individual arbitration curve that allows the operator to adapt to the task. Further, we provide a safeguard to prevent over-dominance. This is evident by the improved similarity and agreement for the DS controller over the conventional approach. Although, it is difficult to say with certainty that the proposed strategy improves task performance. The low success rates for both strategies prove that this environment and task were challenging for operators, which may have also contributed to slower reported times. In ideal cases, the S controller should be faster as the robot has higher levels of authority; however, in the real-world, our approach yields faster times. This is likely due to the operator handling uncertainties of the robot and reducing unnecessary adjustment times. The proposed strategy could benefit remote factory and facility maintenance, telenursing, and assistive living tasks. For example, the assistive living tasks where operators may have varying degrees of desired assistance [15], and our approach can provide an assistance level more in line with the operator.

Our strategy aims to give appropriate assistance to improve performance in challenging tasks. The proposed rotational arbitration is necessary for providing an operator with seamless assistance and may benefit users with more limited mobility as the agent can understand when to assume more or less control of the system. The rotational arbitration's limitations are two-fold: 1) the hyperparameter tuning of Eqs. 4.5 and 4.7, and 2) a temporal filter window on both the α_i and B_i to smooth out spikes that could occur in control authority. These are relatively

straightforward limitations and should be tuned based on the interface, setup, and tasks the shared control system needs to handle. Future work directly stemming from this paper should aim to improve the success rate by 1) devising new arbitration curves to handle specific control dimensions (translation vs. rotation) based on properties discussed in this work and 2) investigating alternative strategies for adjusting control allocation on the fly (i.e., alternative models to B_i) in a dimension-specific context to handle the misalignment issues.

Conventional arbitration may hold advantages in a few scenarios: 1) in very controlled settings with low uncertainty, 2) in lower DOF interfaces (i.e., a controller with 2 or 3 DOFs rather than the 6 DOFs), 3) in tasks that do not require a heavy influence on rotation such as tracing tasks on a fixed plane. First, the autonomous agent should yield an optimal action plan where an operator needs limited interaction. Towards the second, it may be beneficial for low-DOF interfaces to use a synergy-style approach [9] where more assistance is warranted. Towards the third, simpler tasks where automated subroutines shine may allow the autonomy to offload all user control. Regardless, our strategy provides an avenue for meaningful shared control in environments where an operator desires more control over telemanipulation systems.

4.7 References

[1] Y. Rybarczyk, E. Colle, and P. Hoppenot, "Contribution of neuroscience to the teleoperation of the rehabilitation robot," in Proc. IEEE Int. Conf. Syst., Man Cybern., 2002, vol. 4, pp. 6, doi: 10.1109/ICSMC.2002.1173273.

[2] N. Healey, "Speculation on the neuropsychology of teleoperation: Implications for presence research and minimally invasive surgery," Presence, vol. 17, no. 2, pp. 199–211, 2008.

[3] Y. Oh, M. Toussaint, and J. Mainprice, "Learning to arbitrate human and robot control using disagreement between sub-policies," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2021, pp. 5305–5311.

[4] M. Bowman, S. Li, and X. Zhang, “Intent-uncertainty-aware grasp planning for robust robot assistance in telemanipulation,” in Proc. IEEE Int. Conf. Robot. Automat., 2019, pp. 409–415.

[5] D. Song, C. H. Ek, K. Huebner, and D. Kragic, “Task based robot grasp planning using probabilistic inference,” IEEE Trans. Robot., vol. 31, no. 3, pp. 546–561, Jun. 2015.

[6] A. Dragan and S. S. Srinivasa, “A policy blending formalism for shared control,” Int. J. Robot. Res., vol. 32, 2013, pp. 790–805.

[7] A. Handa et al., “DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system,” in Proc. IEEE Int. Conf. Robot. Automat., 2020, pp. 9164–9170.

[8] G. Gioioso, G. Salvietti, M. Malvezzi, and D. Prattichizzo, “Mapping synergies from human to robotic hands with dissimilar kinematics: An approach in the object domain,” IEEE Trans. Robot., vol. 29, no. 4, pp. 825–837, Aug. 2013.

[9] D. P. Losey et al., “Learning latent actions to control assistive robots,” Auton. Robots, vol. 46, pp. 115–147, 2022.

[10] H. Wang and X. P. Liu, “Adaptive shared control for a novel mobile assistive robot,” IEEE/ASM Trans. Mechatron., vol. 19, no. 6, pp. 1725–1736, Dec. 2014.

[11] D. Gopinath, S. Jain, and B. D. Argall, “Human-in-the-loop optimization of shared autonomy in assistive robotics,” IEEE Robot. Automat. Lett., vol. 2, no. 1, pp. 247–254, Jan. 2017.

[12] A. A. Allaban, V. Dimitrov, and T. Padir, “A blended human-robot shared control framework to handle drift and latency,” in Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot., 2019, pp. 81–87.

[13] C. Schultz, S. Gaurav, M. Monfort, L. Zhang, and B. D. Ziebart, “Goal-predictive robotic teleoperation from noisy sensors,” in Proc. IEEE Int. Conf. Robot. Automat., 2017, pp. 5377–5383.

[14] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, “Shared autonomy via hindsight optimization for teleoperation and teaming,” Int. J. Robot. Res., vol. 37, 2018, pp. 717–742.

[15] D.-J. Kim et al., “How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot,” IEEE Trans. Syst., Man, Cybern., vol. 42, no. 1, pp. 2–14, Jan. 2012.

[16] A. Broad, T. Murphey, and B. Argall, “Operation and imitation under safety-aware shared control,” in Proc. Int. Workshop Algorithmic Found. Robot., 2020, pp. 905–920.

[17] A. Broad, T. Murphey, and B. Argall, “Highly parallelized data-driven MPC for minimal intervention shared control,” Proc. Robot. Sci. Syst., 2019, doi: 10.15607/RSS.2019.XV.008.

[18] Y. Oh, S.-W. Wu, M. Toussaint, and J. Mainprice, “Natural gradient shared control,” in Proc. IEEE Int. Conf. Robot Hum. Interactive Commun., 2020, pp. 1223–1229.

[19] S. Jain and B. Argall, “Recursive Bayesian human intent recognition in shared-control robotics,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2018, pp. 3905–3912.

[20] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, “Human-robot mutual adaptation in shared autonomy,” in Proc. 12th ACM/IEEE Int. Conf. Hum.-Robot Interact., 2017, pp. 294–302.

[21] R. Balachandran et al., “Adaptive authority allocation in shared control of robots using Bayesian filters,” in Proc. IEEE Int. Conf. Robot. Automat., 2022, pp. 11298–11304.

[22] C. S. Carver, “Negative affects deriving from the behavioral approach system,” Emotion, vol. 4, no. 1, pp. 3–22, 2004.

[23] C. S. Carver, “Control processes, priority management, and affective dynamics,” Emotion Rev., vol. 7, no. 4, pp. 301–307, 2015.

[24] C. S. Carver, “Control theory, goal attainment, and psychopathology,” Psychol. Inquiry, vol. 29, no. 3, pp. 139–144, 2018.

[25] C. Lugaresi et al., “Mediapipe: A framework for perceiving and processing reality,” in Proc. 3rd workshop Comput. Vis. AR/VR IEEE Comput. Vis. Pattern Recognit., 2019.

[26] J. Sauro et al., “Average task times in usability tests: What to report?,” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2010, pp. 2347–2350.

CHAPTER 5

WE-FILTER: ADAPTIVE ACCEPTANCE CRITERIA FOR FILTER-BASED SHARED AUTONOMY

A paper published on *IEEE International Conference on Robotics and Automation 2023*[10]

Michael Bowman[11] and Xiaoli Zhang[12]

5.1 Abstract

Filter-based shared control aims to accept and augment an operator's ability to control a robot. Current solutions accept actions based on their direction aligning with the robot's optimal policy. These strategies reject a human's small corrective actions if they conflict with the robot's direction and accept too aggressive actions as long as they are consistent with the robot's direction. Such strategies may cause task failures and the operator's feeling of loss of control. To close the gap, we propose WE-Filter, which has flexible, adaptive criteria allowing the operator's small corrective actions and tempering too aggressive ones. Inspired by classical work-energy impact problems between two dynamic, interactive bodies, both inputs' properties (direction and magnitude) are inherently considered, creating intuitive, adaptive bounds to accept sensible actions. The model identifies behaviors before and after impact. The rationale is that each timestep of shared control acts as an impact between the operator's and the robot's policies, where post-impact behaviors depend on their previous behaviors. As time continues, a series of impacts occur. The aim is to minimize impacts that occur to reach an agreement faster and reduce strong reactionary behaviors. Our model determines flexible acceptance criteria to bound

[10] Reprinted with permission from *IEEE International Conference on Robotics and Automation* 2023. © IEEE.

[11] Primary researcher and author, graduate student, Department of Mechanical Engineering, Colorado School of Mines, 1610 Illinois Street, Golden, Colorado, 80401, USA.

[12] Corresponding author, Associate Professor, Department of Mechanical Engineering, Colorado School of Mines, 1610 Illinois Street, Golden, Colorado, 80401, USA.

a mismatch of magnitude and finds a replacement action for conflicting policies. The WE-Filter achieves better task performance, the ratio of accepted actions, and action similarity than the existing methods.

5.2 Introduction

5.2.1 Importance and Needs of Filter-based Shared Control

Filter-based shared control strategies are used for teleoperation scenarios where an operator has a strong grasp of the dynamics, controls, and tasks while desiring more freedom to maneuver the robot in a valid control strategy by not over-constraining their actions [1]. One goal is to have the operator maintain as much control over the robot as possible, which would have more human input actions accepted than rejected. Another goal is maintaining safety, which rejects human input incongruous with optimal robot control signals. These two goals need to be balanced to ensure an operator maintains a sense of control while achieving task success.

The optimal actions of the autonomous agent may differ from the operator in real-world applications. The different actions are inherent to the system for two factors: 1) an operator attempting to handle unexpected uncertainties and 2) multiple ways to accomplish a shared control task. Toward the first factor, operators adjust to overcome the uncertainties in perception and the environment. For example, if the robot misses the initial approach of grasping an object by knocking it over in a teleoperation scenario, an operator will attempt to adjust the actions to correct the robot. However, the robot assumes the pose is correct and rejects these sensible operator actions. In the second factor, the robot's optimal policy in a teleoperation scenario may desire a grasp orientation over the top of the object, while an operator prefers the side to see due to visual constraints. Neither agent has an incorrect action, nor is the robot producing a suboptimal action plan. This difference is fundamentally inherent in shared control, and

precautions are necessary to prevent filters from rejecting an operator's sensible actions. Thus, filtered shared control must be flexible toward the operator's corrective actions or preference, empowering them to stay in control.

Current filter-based shared control strategies usually reject actions; there is no option other than conforming to the robot policy. They strictly use a directional filter (Fig. 5.1b). If the operator's direction differs from the robot policy, the controller will reject the action, and the operator will feel a loss of control. It prevents an operator from 1) resetting to try again, 2) correcting the robot's action, or 3) using it in a preferred way. Fig. 5.1b shows the result of the direction-only bounds for a single timestep (preventing sensible actions) when an operator tries to back out and try a new way. The consequences are the operator: 1) makes drastic movements to no avail; and 2) must give up on their strategy as they frequently have "risky" actions rejected.

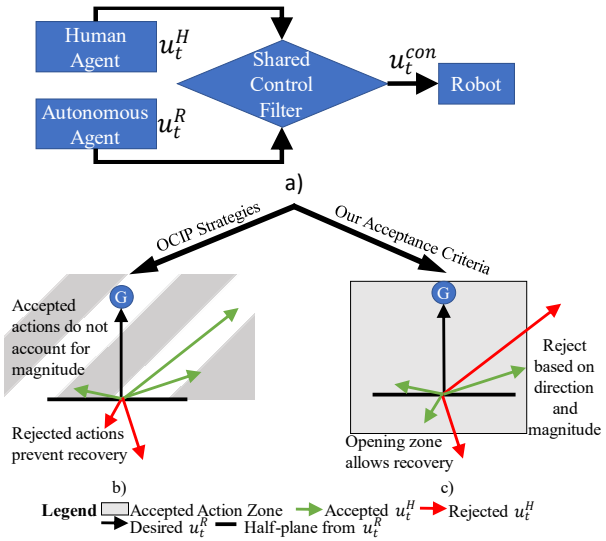


Figure 5.1 a) Overview of the shared autonomy framework, where our contributions focus on providing an optimization procedure for the Shared Control Filter block as in c). b) is the standard approach rejects actions solely on direction which rejects small corrective actions and accepts strong, aggressive actions towards the goal. c) Our strategy simultaneously looks at direction and magnitude to allow smaller actions by an operator to recover, while tempering strong actions of the operator. The floating, adaptive grey zone in c) represents the zone of acceptable actions.

5.2.2 Current State-of-the-Art and Open Problems

Original filter strategies such as virtual fixtures aim to prevent operators from damaging a pre-known environment and rely on the operator's action plan as they do not use an optimal robot policy [2][3]. An alternative strategy augments the human actions based on a model-free learned optimal policy following Deep Reinforcement Learning (DRL) methods [4]. DRL methods assume the operator has an imperfect understanding of the system dynamics and change the operator's actions based on the robot's filter criteria to provide better task performance. Recently, techniques using Maxwell's Demon Algorithm (MDA) provide a model-based structure to determine whether output actions follow the operator or its own optimal policy. The MDA filters take the form of optimal control problems. They use Model Predictive Control (MPC) for nonlinear and discrete systems (rather than DRL). The most recent MDA filter with MPC is known as Sequential Action Control (SAC) [5][6]. It was devised for noise-driven control [7] and has been used for teleoperation [8] by treating operators as noisy rational agents [9][10]. Two open problems exist for filter methods: determining 1) a selection criterion and 2) a post-rejection plan.

Selection criterion: Two state-of-the-art approaches exist for selection criteria. The first is the optimal controller inner product (OCIP), which aims to ensure the inner product is positive, and within an angle tolerance [8]. OCIP calculates a dot product between u_t^R and u_t^H and applies thresholding for acceptable u_t^H [11]. The underlying assumption behind the OCIP is that a user agrees with the optimal control. This makes OCIP best for teaching an operator the optimal policy as it would reject more inputs that do not conform with the optimal policy [12]. Another strategy improves OCIP and is called mode insertion gradient (MIG). Unlike OCIP, the MIG does not assume an operator agrees with the optimal control; instead, operators move in a

descent direction toward a goal. MIG was initially used for scheduling problems to determine when to switch control modes; however, it was adapted to determine whether it should switch [12][13]. MIG accepts more inputs than OCIP. However, it still allows strong motions (potentially causing object knock-over) and rejects adjustments common in shared control (Fig. 5.1b).

Post-rejection plan: The second open problem is determining what to do after rejecting human input [1][8]. There are two main routes: 1) apply no control, or 2) apply the optimal robotic control from MPC (called never-fail solutions). Both have been suggested for different applications and motivations. The first has been used for scenarios requiring higher operator control authority, while the second should be used when an operator is in a supervisory role. The post-rejection plan has no consensus and remains an open problem.

5.2.3 Contributions

This work aims to develop a filter shared control strategy called the WE-Filter inspired by classic dynamics work-energy impact problems. The model explicitly identifies behaviors (velocity actions) before and after impact. The rationale for such a model is that each time step of shared control acts as an impact, where post-impact behaviors depend on their previous behaviors. Thus, previous actions are observed while the current robot response is controllable. As time continues, a series of impacts occur. The aim is to minimize impacts that occur to reach an agreement faster and reduce strong reactionary behaviors. This allows corrective actions to be accepted and rejects too-aggressive actions, so an operator does not feel a loss of control. The operators can refine their actions without conforming to the robot policy. We handle two critical problems current filter strategies struggle with: 1) acceptance criteria incompatible with an operator's senses and 2) post-rejection procedures.

When considering direction and magnitude (as in Fig. 5.1c), we can change bounds in different directions; our new acceptance criteria reflect the realities of the interaction between the operator and autonomous agent. An impact model is developed to account for each agent's magnitude (we call effort denoted as W_t) and the relative difference in direction and magnitude (we call relative disagreement denoted as e_t). With each new human input, new e_t and W_t^c are evaluated. We aim to maximize W_t which drives the operator and autonomous policies to the known goal. The e_t term acts as a constraint to keep the direction and magnitude within an acceptable mismatch range based on the current and previous actions of the robot and human. Since it is temporal and adaptive, it can restrict the set of actions (grey zone smaller in Fig. 5.1c than Fig. 5.1b). It should allow the operator to move backward, albeit in a limited manner. This allows the operator to objectively maintain control of the robot while improving task performance. The contributions follow:

1. Model the interaction between an operator's and autonomous agent's actions using a two-mass dynamic work-energy impact model. It describes current actions to previous ones through effort (magnitude) and relative disagreement (magnitude and direction).
2. Develop adaptive acceptance criteria based on the current autonomous agent and operator's actions through relative disagreement (e_t^R). We accept operator actions outright when an action is admissible ($e_t^R < 1$) and we find a replacement action when it is rejected.
3. Provide an optimization framework when an operator's action is rejected. The controller finds a solution that suits both conflicting policies. We maximize effort (W_t^c) and bound the relative disagreement (e_t^c) to find the action.

5.3 Methods

5.3.1 Current Filter Shared Control Acceptance Criteria

For the remainder of the paper, the current actions of a human and autonomous agent will be denoted as u_t^H and u_t^R, while actions of a previous timestep are u_{t-1}^H and u_{t-1}^R. The applied action to the physical robot will be denoted as u_t^{con} and u_{t-1}^{con}. The OCIP follows a straightforward dot product to determine the direction the control vectors are pointed, shown in Eq. 5.1:

$$u_t^{con} = \begin{cases} u_t^H, & u_t^H \cdot u_t^R > 0 \\ 0, & u_t^H \cdot u_t^R < 0 \end{cases} \tag{5.1}$$

Eq. 5.1 creates a halfplane of acceptable actions, limiting the ability of an operator to back out as the u_t^{con}gives no action. This severely limits accepted actions as dimensions increase. This strategy only looks at the direction of actions, not their magnitude. The OCIP serves as the baseline for evaluation.

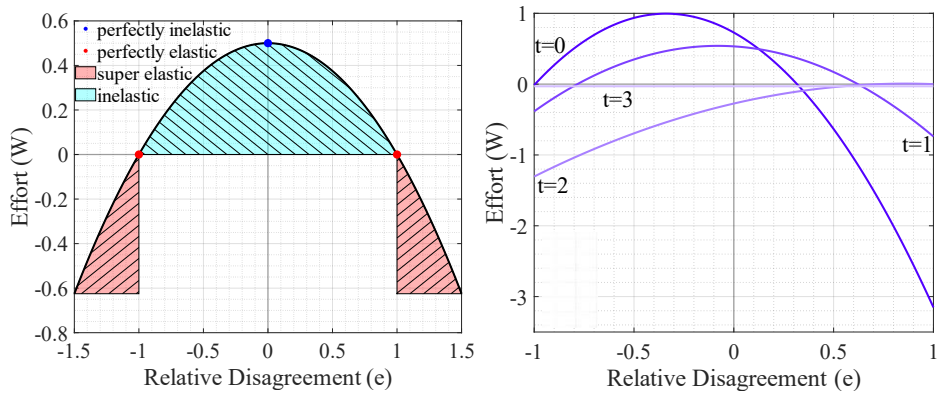


Figure 5.2 An example curve for actions at a single timepoint (left), and a series of them (right). Four scenarios of interest occur on the left. 1) relative disagreement is perfectly inelastic, this action would be identical to their partner's action and maximum effort was put in to agree (blue point), 2) relative disagreement is inelastic which means we have put in positive effort to agree (blue shadow region), 3) the perfectly elastic collision points mean either no change in actions or a complete swap of actions (red points), 4) super elastic collisions result when an agent is actively trying to put in negative effort towards the team (red shadow regions). On the right, a series of 4 impacts occur. Each impact produces a different shaped curve. At t=0 the shape is similar to the left figure. As time continues, the curve becomes flat at nearly 0 (t=3). Meaning that u_(t-1)^H and u_(t-1)^con are the same, and actions reach agreement. The flattening pattern is only true if the team can reach an agreement, otherwise, each time step could stay with sharp convexity.

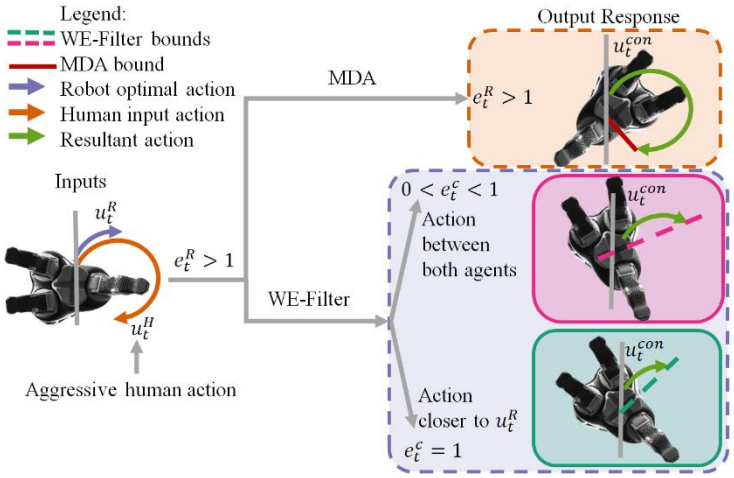


Figure 5.3 Responses of the MDA and WE-Filter to aggressive human rotation in telemanipulation. MDA strategies allow over rotation (top) when both agents agree on the direction of rotation (e_t^R>1). WE-Filter rejects actions when e_t^R>1 and replaces them by bounding relative disagreement (e_t^c). The e_t^c aids in deciding whether u_t^{con} will have an action more like u_t^R or u_t^H. In this aggressive action scenario, we do not want e_t^c=0 or else it will be identical to MDA output (an action, $u_t^{con} = u_t^H$, we already rejected, e_t^R>1). We limit e_t^c so that u_t^{con} is not so like u_t^R that it does not follow the human.

5.3.2 Shared Control Impact Model

A work-energy impact model describes the dynamic, reactive behaviors of two-point masses when they pull apart from one another (u_t^H and u_t^R have different action policies) while adhering to the momentum along a single axis (line of impact). The model explicitly identifies behaviors (velocity actions) before and after impact. The rationale is that shared control acts as an impact at each time step, where new u_t^H and u_t^R behaviors depend on the previous ones (u_{t-1}^H and u_{t-1}^R). The controller is given the four actions and determines whether u_t^H is admissible. As time continues, a series of impacts occur. Two calculations model the interaction.

5.3.2.1 Relative Disagreement: Coefficient of Restitution

The first calculation determines whether u_t^H is admissible or if we need to optimize for an admissible action. It is the coefficient of restitution, e_t, and determines the relative difference in action direction and magnitude. We refer to e_t as *relative disagreement*. e_t^R evaluates u_t^H and u_t^R,

while e_t^c evaluates u_t^H and u_t^{con} if optimization is required for the human input. As the equations are identical in form, in this section, we will demonstrate with e_t^R and is defined in Eq. 5.2.

$$e_t^R = \frac{u_t^H - u_t^R}{u_{t-1}^R - u_{t-1}^H} \tag{5.2}$$

Four cases of Eq. 5.2 must be discussed analogously from the dynamics problem to the shared control context. The explanation will be made with Fig. 5.2 as the mathematical model. Four cases occur: 1) perfectly inelastic collisions, e_t^R =0, 2) inelastic collisions, $0 < |e_t^R| < 1$, 3) elastic collisions, $|e_t^R| = 1$, and 4) superelastic collisions, $|e_t^R| > 1$.

The first case, perfectly inelastic collisions, describes two-point masses stuck together after impact, which mathematically is $u_t^H - u_t^R = 0$. This implies that both agents have a perfect agreement and want the same action to be taken. The second case, inelastic collisions, demonstrates after impact a trend toward full agreement (the blue region in Fig. 5.2). This occurs when the difference between the current actions is smaller than the previous ones, which means the new actions are becoming more aligned and both agents begin to agree. Mathematically, this implies $|u_t^H - u_t^R| < |u_{t-1}^R - u_{t-1}^H|$. This would be seen as slight differences in poses between u_t^H and u_t^R(refinement). The third case is elastic collisions, where $|e_t^R|$=1 (the two red points which cross the x-axis in Fig. 5.2). This implies $|u_t^H - u_t^R| = |u_{t-1}^R - u_{t-1}^H|$. This means after impact complete relative disagreement has been achieved. Last is superelastic collisions, $|e_t^R| > 1$ (the two red regions below the x-axis Fig. 5.2). This case means energy has been introduced into the system, and the actions have become even more polarized, as this implies $|u_t^H - u_t^R| > |u_{t-1}^R - u_{t-1}^H|$. The actions taken have created tension, and an operator overreaction may have occurred.

All cases are important and demonstrate different task behaviors, but none are truly bad. Although case 4 may appear less desirable, it might not be entirely true without context for the

change of actions. For instance, consider u_t^H grasping an object's side while u_t^R grasps the top. The human may make a sudden change in their preferred orientation, which induces case 4 because they want to see the grasp configuration easier. Although, designers will likely aim for case 1 and case 2, as this implies the operator follows the optimal robot policy. In 5.3.3, we describe how our controller would find a replacement action if case 4 occurs as u_t^H would be deemed too large to be a corrective one. Our criteria for an admissible u_t^H is if e_t^R is in case 2, $|e_t^R| < 1$; otherwise, an optimization process is needed to determine a better u_t^{con} so that $|e_t^c| \leq 1$. In Fig. 5.3, we showcase 4 ($|e_t^R| > 1$) occurs if we use the MDA controller. Our method will adjust the u_t^{con} to either be in case 2 ($0<e_t^c< 1$, in the middle row of Fig. 3), or to case 3 ($e_t^c=1$, in the bottom row of Fig. 5.3).

Another critical component is when $e_t^R<0$. This implies that one agent has stayed more dominant for both time steps, such as $u_t^R > u_t^H$ and $u_{t-1}^R> u_{t-1}^H$ or $u_t^R < u_t^H$ and $u_{t-1}^R < u_{t-1}^H$. Lastly, an immediate reaction to$u_{t-1}^R - u_{t-1}^H = 0$ may jump out; however, this is acceptable. When the previous actions align, new actions appear as splitting hairs over fine details. This occurs when the $u_{t-1}^R - u_{t-1}^H$is very small compared to the $u_t^R - u_t^H$, which accounts for and demonstrates the recency bias [16] [17]. This can be limited within our optimization by either applying bounds as written above or adding numerical tolerances to Eq. 5.2's denominator.

5.3.2.2 Effort: Residual Energy Calculation

The second calculation is the residual energy in the system and is used to guide the optimization. The second calculation has two forms like e_t, W_t^R and W_t^c.

$$W_t^c = \frac{m_R}{2}((u_{t-1}^{con})^2 - (u_t^{con})^2) + \frac{m_H}{2}((u_{t-1}^H)^2 - (u_t^H)^2) \tag{5.3}$$

Here, we refer to W_t^c as the *effort* both agents put in towards the interaction as it quantifies the actions' magnitude changes. Ultimately, the goal is to have W_T^c converge to 0 (where t = T is

the terminal time horizon) as they progress in the task, implying both agents have settled on a velocity action (seen in the right of Fig. 5.2 with the faint t=3 line). Toward this high-level goal, it is achieved by maximizing W_t^c at each time step, which would be equivalent to minimizing u_t^H(which we cannot control), and u_t^{con}. Since these actions are in terms of velocities, it would be ideal if u_t^{con} and u_t^H are driven to zero over time, which implies they settled on a state. The parameters m_R and m_H are the weighting terms which can act as role allocation. In a peer-peer setting, they are set to 0.5, while a leader-follower scheme may differ to values such as 0.9 and 0.1. More details on roles can be found in [14] and [15]. In this work, both terms are set to 0.5.

With Eqs. 5.2 and 5.3 established, interesting relationships can be seen if we consider the system controllable through the lens of the u_t^{con} action as a u_{t-1}^{con}, u_{t-1}^H, and u_t^H are observed. Formally, we can look at the relationship of W_t^c and e_t^c by combining Eqs. 5.2 and 5.3 through u_t^{con} in Eq. 5.4 and Fig. 5.2.

$$W_t^c = -\frac{1}{2}m_R(u_{t-1}^H - u_{t-1}^{con})^2 e_t^{c2} - m_R u_t^H (u_{t-1}^H - u_{t-1}^{con}) e_t^c + \frac{1}{2}m_R (u_{t-1}^{con})^2 + \frac{1}{2}m_H (u_{t-1}^H)^2 - \frac{1}{2}(m_R + m_H)(u_t^H)^2 \quad (5.4)$$

With Eq. 4, u_t^{con} impacts variable e_t^c directly as all other actions are given. The quadratic relationship of e_t^c and W_t^c is helpful as we can achieve a global maximum (seen in Fig. 5.2).

The parabolic shape means the behavior can shift left, right, up, or down, and the convexity changes too. This means that the maximum of W_t^c is not always at e_t^c=0. The maximum W_t^c could reside in $|e_t^c|$>1. It is common that current u_t^H have greater magnitude than the previous u_{t-1}^H due to corrective refinement motions meaning the maximum will be $W_t^c < 0$ or $|e_t^c|$>1. Therefore, within acceptable bounds (-1 < e_t^c < 1), the aim is to obtain the maximum W_t^c. Due to this possibility, many interactions could result in $|e_t^c|$=1.

5.3.3 WE-Filter: Acceptance Criteria and Optimization

The optimization is designed for the telemanipulation scenario; an operator has an independent one-to-one mapping to the robot. u_t^H, u_t^R, and u_t^{con} are $\mathbb{R}^7$, where the first 6 DOFs are the end-effector pose velocities (position and orientation velocities), and the last DOF controls the velocity to open and close the hand. So, an operator can assume direct control over the robot's end-effector pose. For each component, we check if it is within the e_t^R bounds. If it is, we accept it and move on to the next component. If it is not, a suitable solution is found. This process occurs for each new human input received.

$$u_t^{con} = \begin{cases} u_t^H, & |e_t^R| < 1 \\ Eq.\,5.6, & |e_t^R| \geq 1 \end{cases} \tag{5.5}$$

The optimization aims to determine a new u_t^{con} that is within an e_t^c range (thus relatively similar to u_t^H) while maximizing W_t^c. Fig. 5.2 shows W_t^c is a convex function in terms of e_t^c. Evaluating each DOF creates a box around possible solutions.

$$\begin{gathered} u_t^{con} = argmax\, W_t^c \\ s.t. \\ -1 \leq e_t^c \leq 1 \\ u_t^{con} \in U \end{gathered} \tag{5.6}$$

A parallel optimization is employed for each dimension of u_t^{con} where U are the environmental bounds and physical limitations of the robot. In our experimental scenario, the bounds were set such that u_t^{con}'s operation range was between u_t^H and u_t^R. The initial value was set to u_t^R. At each time step, the optimization observes a new u_t^H and the previous command, u_{t-1}^H, are given. The u_{t-1}^{con} is the previous command sent to the robot. This makes u_t^{con} the only variable to be optimized is thus the controller's output to run the physical robot. The rationale for

this optimization strategy is two-fold: the output action will be 1) an action that is between both u_t^R and u_t^H within e_t^c bounds while 2) the bounded convexity can ensure an optimal action. This provides an effective bound to allow an operator to make corrective actions (refinement of pose near a goal) and tempers their aggressive erroneous ones (sudden jumps in a posture like our example of $|e_t^R|$>1 when u_t^H decides to grasp the object on the side rather than the top). This ensures the robot does not jump ahead of the operator (potentially knocking the object over) or stall out, resulting in them losing control.

5.4 Experimental Results

5.4.1 Telemanipulation Setup

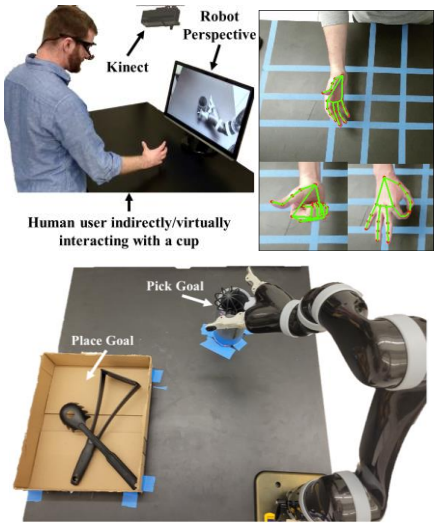


Figure 5.4 Operators see the robot perspective on the screen while they move their hand freely over the table. The Xbox Kinect captures hand posture to extract the palm pose and the amount each finger is open or closed. The information is then sent to the robot to execute the motion.

The telemanipulation experimental setup is described in Fig. 5.4. An RGB-D image is taken with an Xbox Kinect, and features are extracted using MediaPipe [18] before being used as inputs for the robot control. The operator controls the hand pose and the amount each finger opens and closes. In all, u_t^R is $\mathbb{R}^7$. The operator aims to grasp a regular kitchen utensil from a

holder and place it in a bin. This aims to recreate an open pick and place task so the operator can move however they wish to complete the task. The operator can complete tasks with different strategies and recovery plans. With no time constraints, operators have multiple attempts to grasp and place the object in the bin. The robot is given a noisy goal pose (within ±2cm) for both the pick and place locations. Operators were given unlimited training time to familiarize themselves with both control strategies to mitigate bias.

The institution's Institutional Review Board (IRB) approved a set of experiments. Before participating in the study, written consent was obtained, acknowledging they understood the robot setup, the purpose of the study, and the potential risks involved (arm soreness). Three trials were conducted for each filtered strategy by each participant. A trial consisted of three runs; each run was a different object to grasp. A randomized order of the control modes was used to reduce learning effects. As a pilot study, 10 participants volunteered (a total of 90 runs for each mode). The volunteers were 24.1 ± 3.14 years old and included five women and five men with no prior experience controlling the robot. Due to being a pilot study for the viability of the control paradigm, the data analysis had unique scrutiny for the smaller sample size.

5.4.2 Telemanipulation Results

Four quantitative criteria are used to evaluate the effectiveness of the filter shared control strategies: 1) the success rate, 2) completion time, 3) cosine similarity, and 4) the ratio of accepted actions. Table I shows the summary statistics for the measures. Due to the smaller number of trials, necessary precautions must be considered for each measure.

The success rate uses a Laplace Estimate rather than the Maximum Likelihood Estimate to reflect the true success rate. An adjusted-Wald 95% confidence interval was used. An N-1 Chi-squared test was conducted to determine statistical significance. The WE-Filter (64%)

outperforms the MDA filter (39%) with statistical significance ($p < 0.001$). The operators had a better success rate in completing the task. This is likely due to the operator's actions being accepted near the goal as they have different desired poses. The operator can handle the uncertainty, and the robot affords them the chance to complete the task in their way.

Table 5.1 Teleoperation Results For Their Respective Means and Confidence Intervals

Mode	Success rate	Completion time (s)	Cosine distance	Ratio of accepted actions
WE	**0.64 [0.54,0.74]**	**38.60 [32.53,45.80]**	**0.206 [0.187,0.228]**	**0.973 [0.971,0.974]**
MDA	0.39 [0.30,0.49]	56.23 [43.97,71.90]	0.589 [0.560,0.619]	0.681 [0.678,0.684]

WE means WE-Filter, MDA means Maxwell's Demon Algorithm. Bold means this mode performs better in this metric by statistical significance.

The analysis for the time data is not normally distributed but rather positively skewed [19]. Therefore, a log transform is necessary. The geometric mean and a 95% confidence interval are used. A two-sample t-test is used to determine the statistical significance. The time data are only the successful completion time trials. The WE-Filter (38.60 seconds) outperforms the MDA filter (56.23 seconds) with statistical significance ($p < 0.05$). The faster time data shows operators are not constantly being rejected by the filter and can functionally make actions to refine the robot pose to complete the task. Both task performance metrics show improvements over the MDA filter. However, the two other measures can reinforce the notion of an operator having better control over fine-tuning the robot pose to complete the task.

The other two measures of cosine distances and the ratio of accepted actions are described in Eqs. 5.7 and 5.8, respectively, with u^H and u^{con}. The cosine distance measure is:

$$\sum_t 1 - \frac{u_t^H \cdot u_t^{con}}{\|u_t^H\| \|u_t^{con}\|} \tag{5.7}$$

The aim is to determine the alignment of the control vectors for the controller output and the operator action. A score of 0 is considered optimal (they perfectly align). The geometric mean

and 95% confidence intervals were generated and placed in Table 5.1. A two-sample t-test was used to determine the statistical significance. The WE-Filter outperforms the MDA filter ($p < 0.001$) as the output controller's actions are more similar to human actions. This is because of the filter design, where our approach aims to align actions more closely with an operator rather than outright reject them. This is more evident when analyzing the ratio of accepted operator actions:

$$\phi = \frac{\sum ag}{number\ of\ actions},\ \ 0 \leq \phi < 1,\ \ \text{a}g = \begin{cases} 1, if\ u^H \cdot u^{con} > 0 \\ 0, if\ u^H \cdot u^{con} \leq 0 \end{cases} \tag{5.8}$$

This metric determines the number of operator actions accepted by the controller. The aim is to normalize the number of actions taken and determine a better ratio of agreeable actions. A score of 1 is considered optimal as this would imply that all user actions were accepted. The geometric mean and 95% confidence intervals were generated and placed in Table 5.1. The two-sample t-test is used to determine the statistical significance. The WE-Filter outperforms the MDA filter (p<0.001) as the ratio of accepted actions is greater. This is expected as the MDA filter rejects u_t^H rather than augments the parts of u_t^{con} to align with the operator. Thus, it is sensible to conclude that our filter benefits the operator in their preferred strategy. To further assess whether the assistance benefits the operator, we can look at u_t^H and u_t^{con} for qualitative analysis.

Qualitative analysis can be shown in two ways: 1) how human input relates to robot output response and 2) how an MDA output compares to the WE-Filter output. Fig. 5.5 shows a successful trial for the task. Although actions are opened up using this adaptive criterion in some areas, it also restricts the operator's actions, as seen in the $\dot{\gamma}$ where both actions are nearly similar at 0. Our approach accepts more actions than the MDA filter. Notice large blocks where the MDA filter would reject the operator inputs flat out for $\dot{z}$ and $\dot{\gamma}$, while our approach will allow refinement in the $\dot{z}$ (from ~11s to~26s). This is seen in the $\dot{\gamma}$ from 0s to ~8s. The WE-Filter

rejects sudden reactions while accepting ones more in line with previous actions. Strong reactions will knock down the object if they are not limited. The WE-Filter accepts more human actions and allows greater refinement over the MDA strategy.

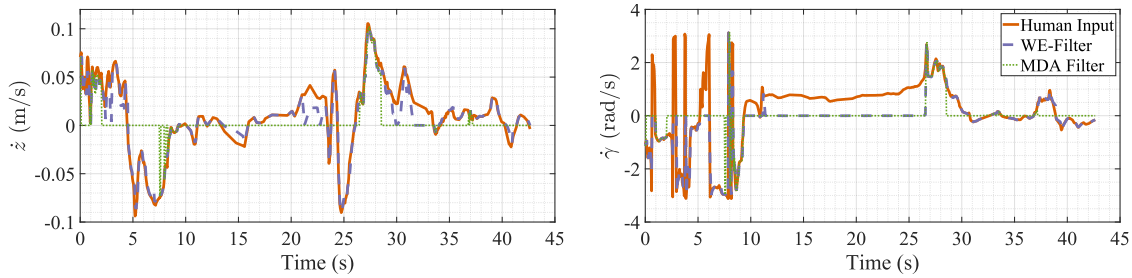


Figure 5.5 A single successful trial for the operator input actions (in orange), and the resulting robot response with our behaviors (purple) and what would be the MDA response(green). Only 2 dimensions (actions in height and yaw) are shown for easier viewing. If our approach deems the actions too aggressive, the output action will restrict the motion (e.g., in $\dot{z}$ during ~20s to ~23s). When the MDA filter rejects an action, the action is set to 0. Our approach accepts more actions on the whole (e.g., in $\dot{\gamma}$ from ~26s to 45s), yet rejects too aggressive motions (e.g., in $\dot{\gamma}$ during the first 8s) and replaces them with acceptable actions.

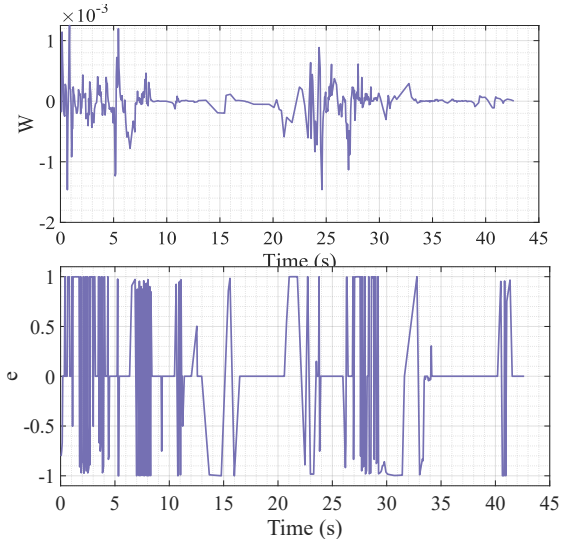


Figure 5.6 The W_t^c(effort) and the e_t^c (relative disagreement) for the $\dot{z}$ in Fig. 5.5. Oscillations in the beginning occur as the operator is unsure of their actions. Our approach continues to maximize W_t^c at each timestep in the e_t^c bounds. Ultimately, our solution reduces W_t^c to 0. This holds when a goal is changed (the second oscillation happens when the operator moves to the place goal at ~23s). When actions appear similar in Fig. 5.5, if e_t^c=0, our strategy deems the best action is the operator's, while an e_t^c=1 or e_t^c=-1 is seen as splitting hairs.

Fig. 5.6 shows the W_t^c and e_t^c plots for the left side of Fig. 5.5 (the $\dot{z}$). Our approach drives W_t^c to 0 over time to a goal. The oscillations at the beginning of the trajectory and in the middle indicate that the operator is moving towards the pick goal and place goal, respectively. The oscillations demonstrate when an operator is unsure of their actions. Our filter aims to settle them by finding the maximum W_t^c within the e_t^c bounds at each timestep. By doing so, the W_t^c is driven to 0 over time. The periods when e_t^c are 0 shows when u_t^{con} deems the most effective solution is the u_t^H (i.e., alignment of u_t^H and u_t^R). The timepoints where e_t^c=|1| demonstrate the u_t^{con} aligns with u_t^R more than u_t^H because the best action to take is closest to the u_t^R. e_t^c does not prevent faster speeds; rather, it rejects sudden speed changes. When W_t^c is very small and smooth, this is seen as the output action, and the team is splitting hairs over minor details. In conclusion, our filter works as expected and provides an effective solution to drive high-effort actions within acceptable agreement bounds. A supplemental video shows examples of both strategies.

5.5 Discussion

Two cases of operator behavior need to be discussed: 1) large decisive actions and 2) small refinement actions. The MDA filter allowed the operator to make larger motions than the WE-Filter. This sometimes allowed an operator to approach the object quicker, but it would also cause them to knock the object over more often (shown by a lower success rate). The WE-Filter reduces the magnitude of the actions, so it may not allow large actions like the MDA filter, which would theoretically limit the speed of completing tasks. However, in practice, WE-Filter tempers the operator's aggressive actions resulting in higher success rates and better completion times. Refinement actions are when the WE-Filter truly shined, as evident by the higher ratio of accepted actions and higher similarity for control vectors. Qualitatively, this is best seen in the

last recovery case in the video, where an operator adjusts the object pose to prevent collisions. It also allows operators to adjust the rotation of the end-effector while making small adjustments in position to line up the grasp. The MDA struggles with small refinement actions needed to succeed in telemanipulation as more actions would be rejected near the final goal states resulting in the robot stalling out and rejecting operator motion. These refinements by the operator simply exist (i.e., the operator wants a pose they can view on the camera to line up a better grasp or think a certain pose will yield a more stable grasp). These refinement actions are critical in telemanipulation [20].

Although the WE-Filter does benefit the operator by adjusting output actions, there are still shortcomings with the current strategies. Potential limitations of our approach are determining 1) hyperparameters for m_R and m_H, 2) the best temporal points to choose (i.e., comparing u_t^H to u_0^H over u_{t-1}^H), and 3) the sensitivity towards noisy inputs by an operator. Investigating these limitations further will allow a model predictive control such as SAC with appropriate roles to shine. The role allocation is outside this work's scope but influences how much the robot will limit its actions over the operator. The second is determining when to consider the impact model, especially if a SAC strategy is adopted. The strategy adopted in this work was using every new operator input (i.e., u_t^H and u_{t-1}^H), but a further investigation must be done to consider more viable time points. For instance, rather than the previous action, like our strategy, it may be the last intervention action applied by the SAC strategy. The third limitation is that noisy human signals occurring when the operator holds their hand steady could cause a vibrating effect on the robot. Although the optimization would attempt to reduce this behavior, a stochastic implementation needs further inquiry. Bounds on e should be task-dependent. Although we follow a classical dynamics threshold, it is not unreasonable to say the selection should be either

$|e| \leq 0.5$ for a narrower set of actions, or $|e| \geq 1$ as an adversarial agent to act against the operator for rehabilitation purposes.

Overall, the strategy presented in this work has viability for telemanipulation scenarios. The limitations and proposed future directions demonstrate there is room to improve the model before being used for an MPC-style strategy like [21]. However, the WE-Filter should be considered for future strategies where two agents are attempting to assume control, and a resultant intuitive action is desired.

5.6 References

[1] A. Broad, T. Murphey and B. Argall, "Operation and Imitation Under Safety-Aware Shared Control," in Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13, pp. 905-920, 2020.

[2] J. Webb, et. al, "Using visuomotor tendencies to increase control performance in teleoperation," in American Control Conference (ACC), Boston, 2016.

[3] F. Ryden, H. J. Chizeck, S. N. Kosari, H. King and B. Hannaford, "Using kinect and a haptic interface for implementation of real-time virtual fixtures," in Robotics: Sciences and Systems, Los Angeles, 2011.

[4] S. Reddy, A. Dragan and S. Levine, "Shared Autonomy via Deep Reinforcement Learning," in Robotics: Science and Systems, Pittsburgh, 2018.

[5] A. R. Ansari and T. D. Murphey, "Sequential Action Control: Closed-Form Optimal Control for Nonlinear and Nonsmooth Systems," IEEE Transactions on Robotics, vol. 32, no. 5, 2016.

[6] I. Abraham, G. De La Torre and T. D. Murphey, "Model-Based Control Using Koopman Operators," in Robotics: Science and Systems, Cambridge, 2017.

[7] E. Tzorakoleftherakis and T. D. Murphey, "Controllers as Filters: Noise-Driven Swing-Up Control Based on Maxwell's Demon," in IEEE Conference on Decision and Control, Osaka, 2015.

[8] K. Fitzsimons, E. Tzorakoleftherakis and T. D. Murphey, "Optimal Human-In-The-Loop Interfaces Based on Maxwell's Demon," in American Control Conference (ACC), Boston, 2016.

[9] M. Kwon, et. al, "When Humans Aren't Optimal: Robots that Collaborate with Risk-Aware Humans," in International Conference on Human-Robot Interaction (HRI), Cambridge, 2020.

[10] D. Fridovich-Keil et al., "Confidence-aware Motion Prediction For Real-Time Collision Avoidance," International Journal of Robotics Research, 39(2-3), pp.250-265., 2019.

[11] A. Broad, I. Abraham, T. Murphey and B. Argall, "Data-driven Koopman Operators for Model-Based Shared Control of Human- Machine Systems," International Journal of Robotics Research, 2020.

[12] K. Fitzsimons, A. Kalinowska, J. P. Dewald and T. D. Murphey, "Task-based Hybrid Shared Control for Training Through Forceful Interaction International Journal of Robotics Research, 39(9), pp.1178-1195, 2020.

[13] A. Kalinowska, K. Fitzsimons, J. Dewald and T. D. Murphey, "Online User Assessment for Minimal Intervention During Task-Based Robotic Assistance," in Robotics: Science and Systems, Pittsburgh, 2018.

[14] A. Mortl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan and S. Hirche, "The role of roles: Physical cooperation between humans and robots," International Journal of Robotics Research 31, no. 13: 1656-1674, 2012.

[15] D. P. Losey et .al, "A review of intent detection, arbitration, and communication aspects of shared control for physical human-robot interaction,"Applied Mechanics Reviews, 70(1) 2018.

[16] P. Kaniarasu, et. al, "Robot Confidence and Trust Alignment,"in International Conference on Human-Robot Interaction (HRI), Tokyo, 2013.

[17] M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld and H. Yanco, “Impact of Robot failures and feedback on real-time trust," in International Conference on Human-Robot Interaction (HRI), Tokyo,2013.

[18] C. Lugaresi et al., "Mediapipe: A framework for perceiving and processing reality," in Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR), Long Beach, 2019.

[19] J. Sauro and J. R. Lewis, "Average task times in usability tests: what to report?," in Proceedings of the SIGCHI conference on human factors in computing systems.,Atlanta, 2010.

[20] M. Bowman, S. Li and X. Zhang, "Intent-Uncertainty-Aware Grasp Planning for Robust Robot Assistance in Telemanipulation," in IEEE International Conference on Robotics and Automation (ICRA), Montreal, 2019.

[21] A. Broad, et. al, "Highly parallelized data-driven MPC for minimal intervention shared control," in Robotics: Science and Systems, Freiburg im Breisgau, 2019.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Contributions

In this dissertation, new approaches to advance telemanipulation from controlled lab settings to more realistic uncertainties in real-world applications have been proposed. In order to evaluate the new approaches, a hands-free telemanipulation system for the experimental setup was developed. This includes designing human and robot grasp models, incorporating human hand tracking for direct teleoperation methods from the human to the MICO robot hand, and extending this system to handle autonomous assistance.

The contributions of the new strategies for assistance include: 1) disambiguating the grasp motion goal with task level inference and determining the actions needed to satisfy it (Chapter 2 and 3), 2) provide a dimension-specific shared autonomy to provide appropriate assistance for good task performance and maintain a feeling in control of the robot (Chapter 4), and 3) provide an adaptive filter-based shared autonomy, so an operator's corrective actions are not always rejected when differing from the autonomous agent's policy (Chapter 5). Overall, this dissertation presents novel shared control algorithms that can be used to improve a human operator's control of a robotic manipulator to achieve a telemanipulation task that would otherwise be challenging or impossible for the user on their own.

One aim of the dissertation was to tackle how to use intent inference for telemanipulation within shared control. The unique challenges—beyond teleoperation—include physical discrepancy between human and robot hands and the fine manipulation constraints to constitute task success. By promoting a robot agent's autonomy level through understanding the human operator's intent and providing motion assistance to achieve it has demonstrated great

advantages to the operator. This dissertation advances the shared control technique one step further beyond approaching phases of teleoperation to deal with the more challenging object manipulation task. To achieve this next step, a multi-task robot grasping model was built and an intent-uncertainty-aware grasp planner was developed to generate robust grasp poses given the ambiguous human intent inference inputs. The user inputs act as motion constraints, while the optimization determines actions based on human intent. The robot's motion follows an arbitration between following the user's motion constraints and accomplishing the inferred task. The arbitration adapts based on the physical discrepancy between the human and robot hands. The techniques presented extended the teleoperated robots' functionality in a more practical telemanipulation scenarios.

Another fundamental aim of the dissertation is present a more effective arbitration scheme through a dimension-specific means. Conventional arbitration policies often define a uniform singular scalar for all 6 DOFs to blend human input and robot assistance. However, this singular scalar over-dominates some dimensions of the inputs and provides insufficient assistance in other dimensions. A dimension-specific arbitration policy is developed to customize the control arbitration along each DOF to fill the gap. It looks at whether the robotic assistance is too timid or aggressive along each DOF and determines the arbitration magnitude according to disagreement levels of control allocation and the user's willingness to accept assistance. The user's willingness is estimated from a feedback psychology model. The method has higher similarity and ratio of agreement between the human and robot (lower over-dominance) over existing methods and, simultaneously, improves the task performance. This arbitration strategy is expected to increase the adoption of teleoperation for object manipulation.

The last aim was to provide adaptive bounds for Filter-based shared control systems. Filter-based shared control aims to accept and augment an operator's ability to control a robot. Current solutions accept actions based on their direction aligning with the robot's optimal policy. These strategies reject a human's small corrective actions if they conflict with the robot's direction and accept too aggressive actions as long as they are consistent with the robot's direction. Such strategies may cause task failures and the operator's feeling of loss of control. To close the gap, WE-Filter was proposed, which has flexible, adaptive criteria allowing the operator's small corrective actions and tempering too aggressive ones. Inspired by classical work-energy impact problems between two dynamic, interactive bodies, both inputs' properties (direction and magnitude) are inherently considered, creating intuitive, adaptive bounds to accept sensible actions. The model identifies behaviors before and after impact. The rationale is that each timestep of shared control acts as an impact between the operator's and the robot's policies, where post-impact behaviors depend on their previous behaviors. As time continues, a series of impacts occur. The aim is to minimize impacts that occur to reach an agreement faster and reduce strong reactionary behaviors. Our model determines flexible acceptance criteria to bound a mismatch of magnitude and finds a replacement action for conflicting policies. The WE-Filter achieves better task performance, the ratio of accepted actions, and action similarity than the existing methods.

6.2 Future Directions

A few directions should be considered in future works with the first are immediate next steps while latter suggestions are much larger systematic changes.

The immediate future directions for the intent inference modules (Chapter 2 and 3) is to extend the technique to produce predictive trajectory actions toward the goal rather than a single

grasp goal location. The predictive trajectory allows more possibilities for optimization strategies to be utilized. For instance, if a predictive trajectory can be obtained, Model Predictive Control (MPC) strategies can be adopted as a means for trajectory reference tracking. Other aspects which are more challenging and require more complex hardware, is expanding the grasp model for post-grasp manipulation. Currently, the work shown in Chapter 2 and 3 focuses on an initial single grasp goal. Expanding toward the post-initial grasp would require a dexterous hand that can make small subtle adjustments while maintaining force or form closure on the object. The benefits of this direction would allow a robot to adjust the object in hand during transitions from initial grasp to final goal states as the operator intends rather than the operator trying to move the object from a single grasp point.

Immediate improvements to the arbitration proposed in Chapter 4 are further tuning hyperparameters for specific shape and scale of the Beta prime distribution, as well as developing a temporal smoothing window to ensure there are no spikes in either the willingness or the arbitration values. The tuning would fundamentally change the shape and scale of the Beta prime distribution to give unique behavior where perhaps the willingness can never reach >0.5. The benefits of smoothing the arbitration ensure a transititory phase occurs which mitigates the chances for rapid switching of control. The potential for rapid switching is undesirable it would likely lead to the operator struggling with their own adaption to the system when completing the task. Another change that would provide better assistance is improving arbitration curve policies. For instance, the curve used in Chapter 4 is known as a positive trending curve (which is the most difficult arbitration scheme to use in high uncertainty environments and was chosen for this reason), where other work is dedicated to improving performance by having negative or bell-shaped curves. Applying the same techniques on these alternative curves would yield higher

success rates, but likely demand more operator control and residuals (i.e., higher mental burden and engagement). Along with developing new arbitration curves, new willingness ones should be considered to provide the robot with a model to predict the operator motion.

The future work for Chapter 5 would be tuning hyperparameters and selecting better temporal actions to consider when intervening. For instance, one suggestion is to tune the mass weighting terms based on role allocation. Yet, alternatively, these weighting terms could encompass other factors such as performance, task, and expertise level. Selecting actions to consider is critical for the relative disagreement filter. Properly determining whether the current actions should be compared to the last action, initial action, or last intervention time point is imperative to be investigated as this would alter the acceptable and unacceptable actions.

Although some more difficult design work has already been suggested, the more extensive and impactful work follows. Adapting the WE-Filter to a MPC scheme such as Sequential Action Control (SAC) which takes into account not only the optimal action but *when* to enact that action. SAC blends very well with the rationale of the WE-Filter where the criteria determines whether actions are admissible, and could be used to determine when should intervention actions occur. Further, a necessary addition to both Chapter 4 and 5, is a stochastic implementation. The stochastic implementation introduces uncertainty in the human motion, which need to be accounted for in the optimization strategies. The stochastic approaches propagate uncertainty and can determine optimal actions in the face of it. However, extending to these techniques requires an extensive look at developing a model to predict human actions.