Effect of Control Interface Implementation on Operation of a Multi Degree of Freedom Telerobotic Arm

by

Sherrie Alyssa Hall

M.S., Georgia Institute of Technology (2012)

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Signature redacted

Author ..............................................................

Department of Aeronautics and Astronautics

June 2017

Signature redacted

Certified by ............

Leia Stirling
Assistant Professor
Thesis Supervisor

Signature redacted

Accepted by ............

Youssef M. Marzouk
Chairman, Department Committee on Graduate Theses
Signature redacted

Certified by.

Jeffrey A. Hoffman
Professor of the Practice

Certified by....

Julie Shah
Associate Professor
Effect of Control Interface Implementation on Operation of a
Multi Degree of Freedom Telerobotic Arm

by

Sherrie Alyssa Hall

Submitted to the Department of Aeronautics and Astronautics
on June 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In exploration, scenarios can include a human working along-side - or attached to - a robot. For example, concepts of Mars human-robot exploration teams, or extravehicular activity (EVA) on the International Space Station (ISS) with an astronaut fixed to the end of a large robot arm for stability. Robots in these scenarios must be able to be directed in real time to react to environmental unknowns. In this work, use of a fully-wearable gesture system was proposed to provide control of the robot to the human in the field. A wearable gesture interface would allow user mobility in the field, would allow the user full arm range of motion when not in use, and could be built into the user’s clothing to avoid requiring them to carry additional equipment for robot control. This work used the Canadarm2 as a case study for exploring implementations and input mappings for robot operations with a gesture interface in complex environments. Manual control of the Canadarm2 is difficult, involving a complex twin-joystick interface. Although astronauts on EVA often stand fixed at the end of the robot arm for stability, EVA astronauts cannot control the arm themselves, instead relying on teleoperation by a second astronaut inside the ISS. A study was conducted with a simulated Canadarm2, comparing three different gesture implementations to the traditional joystick input method. In order to test gesture control mappings of this case, a gesture interface was needed for operation. The wearable gesture system selected used integrated surface electromyography sensors and inertial measurement units to detect arm and hand gestures. Two gesture mappings permitted multiple simultaneous inputs (multi-input), while the third was a single input method. One multi-input method was inspired and aligned with natural human reach while the other divided controls between different segments of the human arm kinematic chain. The single input method exhibited high workload in addition to reduced efficiency as compared to the joystick control group. The gesture mapping inspired by human motor control showed potential for performance equivalent to traditional joystick controls after training. The multi-input mapping less aligned with natural motor control showed reduced completion rate for certain tasks and higher overall workload as compared to the joystick interface. Unlike the joystick controls, the ges-
ture interface was limited to one rotational input at a time. To investigate potential performance effects due to such limits on controller degrees of freedom (DOF), a second study was conducted that locked different DOF in the joystick interface. Four joystick interfaces were compared: full multi-axis (with nominal six DOF), rotation limited (one rotation at a time), translation limited (one translation at a time), and without simultaneous translation/rotation or “non-bimanual.” This study found no statistically significant differences in performance or workload between traditional controls and reduced rotational DOF, which was comparable to the gesture interface mapping. For the non-bimanual condition, there was an increase in task time combined with decreased multi-rotation, highlighting that non-bimanual operation may have potential in training for rotation efficiency. Two different strategies were observed during translation limiting to overcome inability to visually track, align with, and move toward the target simultaneously. This work highlights the importance of multi-input control for complex robotic teleoperation and provides recommendations for the development of input mappings and implementations of gesture control interfaces, as well as any interfaces that require reduced DOF as compared to the operational environment or system being controlled.

Thesis Supervisor: Leia Stirling
Title: Assistant Professor
Acknowledgments

I have been fortunate to have incredible support in making my PhD a reality. First and foremost, I’d like to thank my advisor, Leia Stirling, for taking me on as her first AeroAstro PhD student. It has been my privilege to have her guidance, as we faced each milestone together. Thank you to my committee members: Jeff Hoffman, for giving me the opportunity to study at MIT to begin with and so many opportunities since. To Julie Shah, both for her guidance on my thesis and for becoming one of the best house-masters a student resident could ask for. You all provided advice and helped me learn to navigate the academic world over the past five years, but more importantly, through conversations, dinners, events, and support, you each helped me truly live at MIT.

To my readers, Andy Liu and Chris Assad: Thank you for your support. Andy, thank you for your perspective and guidance. Chris, thank you for your enthusiasm, your constant determination to make sure the BioSleeve project continues, and letting me be a part of the incredible research opportunities this project has involved. On that note - thank you to the entire BioSleeve team at JPL. You have all been great to work and fly with. Thank you also to Clementine Mitchel, one dedicated UROP who worked through a seemingly endless de-bugging effort, hardware issues, and even learning to use a needle and thread.

The Man Vehicle Lab is an amazing group of people, and I cannot imagine a better lab to work with. From my first week and the 50th anniversary, to Thanksgiving, to every memorable Friday, the camaraderie and kindness of this lab is outstanding. I am lucky to have had such a supportive environment as a graduate student. Thank you to Dava Newman, Chuck Oman, Larry Young, Alan Natapoff, and Liz Zotos for all you do to keep the MVL going. Outside of lab, thank you to everyone from Sidney-Pacific the “best graduate residence in the universe!” as Roger would say. My friends and time at SP made it a second home. One where food, laughter, a cup of tea, and a board game were never far away. I also want to thank everyone at Georgia Tech, especially HPEPL, and from the Girl Scouts of Northeast Georgia, without
whom I would never have made it so far.

Finally, thank you to my friends and family. You’ve supported me through this journey, and seen me through every struggle and success. You’ve been there with phone calls, baked goods, and encouragement. Thank you to my mom, for always believing I’d get this far. Thank you Jack, my best friend, my partner in all things. Thank you for putting up with late nights, for providing coffee, and for reminding me to take a break and see a movie or play a game, even at the busiest of times.

This work was supported by Jet Propulsion Lab Award 1516097. The SSRMS simulator was developed with support from NASA Contract NCC9-58, NSBRI SA1301, NSBRI SA001, and NSBRI NBPF02001. All study protocols were approved in advance by the MIT Committee on the Use of Humans as Experimental Subjects.
## Contents

1 Introduction ........................................ 11
   1.1 Motivation ........................................... 11
   1.2 Previous Literature ................................. 14
      1.2.1 SSRMS Operation ................................. 14
      1.2.2 Creating Intuitive Interfaces ................. 16
      1.2.3 Multi-DOF Interfaces in Six DOF Environments .. 17
      1.2.4 Gesture Control .................................. 21
   1.3 Case Study Experimental Environment ............... 28
   1.4 The SSRMS Simulator .................................. 29
   1.5 The JPL BioSleeve ..................................... 33
   1.6 Thesis Structure ...................................... 35

2 Project Development ................................ 39
   2.1 IMU Integration .................................... 39
      2.1.1 IMU Characterization ............................. 42
   2.2 Gesture Mappings .................................... 45
      2.2.1 Single Input ..................................... 47
      2.2.2 Multiple Inputs .................................. 49
   2.3 Interface Development Study ....................... 51
      2.3.1 Motivation and Hypotheses ...................... 51
      2.3.2 Methods ......................................... 52
      2.3.3 Results ......................................... 59
      2.3.4 Discussion of Results ......................... 62
3 Investigation of Gesture Control Mappings

3.1 Motivation and Hypotheses

3.2 Methods

3.2.1 Participants

3.2.2 Experimental Tasks

3.2.3 Data Processing

3.2.4 Experimental Procedure

3.2.5 Statistical Analysis

3.3 Results

3.3.1 Effect of Input Method on Task Time

3.3.2 Effect of Input Method on Task Success

3.3.3 Effect of Input Method on Workload

3.3.4 Usage of Multiple Axes

3.4 Discussion of Results

4 Effects from Limiting Interface Degrees of Freedom

4.1 Motivation and Hypotheses

4.2 Methods

4.2.1 Participants

4.2.2 Test Environment

4.2.3 Test Conditions

4.2.4 Data Collection and Analysis

4.2.5 Protocol

4.2.6 Statistical Analysis

4.3 Results

4.4 Discussion

5 Conclusions

A Appendix A: Experiment Training Slides
A.1 SSRMS Basic Training for Hardware Development and Gesture Studies
   (Chapters 2 and 3) ........................................ 117
A.2 Task Training for Hardware Development Study (Chapter 2) .......... 124
A.3 Task Training for Gesture Study (Chapter 3) ............................ 127
A.4 Task Training for Degrees of Freedom Study (Chapter 4) ....... 129

B Appendix B: Pre-Experiment Participant Background Questionnaire 137

C Appendix C: Additional Gesture Study Materials 139
   C.1 Joystick Input Handout .................................. 139
   C.2 Single Input Gesture Handout .............................. 140
   C.3 Endpoint Control Gesture Handout ......................... 141
   C.4 Coupled Input Gesture Handout ............................ 142
   C.5 Handout Illustrating Translation Command “Box” for Endpoint Control and Coupled Input ............. 143
   C.6 Training Checklist ........................................ 143
   C.7 Verbal Task Instructions .................................. 144
      C.7.1 Path Following Training ................................. 144
      C.7.2 Track and Capture Training ............................ 144
      C.7.3 Fly-to and Grapple Training ............................ 144
      C.7.4 Path Following Evaluation .............................. 144
      C.7.5 Track and Capture Evaluation .......................... 145
      C.7.6 Fly-to and Grapple Evaluation .......................... 145

D Appendix D: Task Selection Effects in DOF Study 147
   D.1 Effect of Drift Type: Results ................................ 148
   D.2 Effect of Drift Type: Discussion ............................ 149
Chapter 1

Introduction

1.1 Motivation

Exploration scenarios with human-robot collaboration include operations where a robot and human move alongside each other through an environment. This includes scenarios such as concepts for human-rover teaming on Mars surface exploration missions, extravehicular operations on the International Space Station (ISS) with an astronaut fixed to the end of a large robot arm for stability, or earth-based military and law enforcement applications with teams aided by reconnaissance rovers. In such situations, the solution to directing robot movements in reaction to environmental unknowns in real time, is to have a dedicated human teleoperator control robot movement. For space exploration applications, having a dedicated remote operator is not ideal. An Earth-bound operator may be subject to time delay, depending on the distance of the mission from Earth. Using a crew-member to operate from a nearby location such as on the ISS or in a habitat on Mars requires that crew-member to be dedicated to robot teleoperation and monitoring for the duration of robot operations, preventing them from completing other tasks. With a small crew on a spaceflight mission, crew time is a valuable, limited resource. The logical alternative would be to allow humans working alongside the robot to assume control as needed. These scenarios all involve human collaborators who must already carry equipment, be completely mobile, and have hands free for other uses for most of their mission.
In order to accommodate these needs, the concept of fully-wearable gesture control was proposed. Such a system would be worn on a user’s arm, either integrated into or worn under the operator’s clothing (or spacesuit). With gesture control, the human arm and hand become the interface, requiring no additional hand-held or bulky equipment. In this concept, the ISS robot arm may be left stationary, or a recon rover given automation to follow a human team for an extended time. However, if the EVA astronaut needs to change position, or a ground team needs to direct the rover to a point ahead, the human collaborator could then turn on a control system, and direct the robot as needed. Inputs for robot motion would become simple hand gestures, such as pointing the hand or rotating the wrist in the direction the robot needs to move.

Although gesture interface hardware and software development is critical to creation of a functional, reliable system, implementation of gesture commands must also be considered. Even if ideal sensors and robust processing allowed an interface perfect gesture recognition, without a mapping of gestures to command inputs that an operator may learn and use with proficiency, the system as a whole would be rendered ineffective. Determination of strategies for intuitive mappings became a main focus for this work.

There are many robotic operation scenarios that require function and real-time planning in large or unconstrained environments, including reconnaissance operations and space robotics. Use of a fully-wearable gesture control system is well suited to human-robot or human-computer interaction scenarios where users are mobile, must have full range of motion for their hands when not operating the robot, or have limited availability for carrying additional equipment. Operation of the International Space Station (ISS) robotic arm (also called the Space Station Remote Manipulator System (SSRMS) or Canadarm2) to support astronauts on Extravehicular Activity (EVA) aligns with cases well-suited for gesture control. The SSRMS requires robotic movements through all three axes in space with operation through a range of small or large-scale tasks.

Using the SSRMS as a case study, this work investigated methods for mapping
gestures to command inputs for a wearable gesture control interface and implications of broader interface design choices, namely degrees of freedom (DOF) included in a control mechanism. In a broad sense, this meant exploring the relationships between a control interface, the system being operated, and operation in a complex environment, in order to improve future interface design for such environments. A “complex” environment here refers to a machine’s physical operating location which is unconstrained or large scale. Increased environment DOF also contribute to environmental complexity. The DOF of an environment are defined here separately from the DOF of a robot or the DOF of a control interface. While robot DOF refer to the physical degrees of freedom in the robotic system, interface DOF refer to the total simultaneous inputs feasible for that controller, and environment DOF refer to the spatial degrees of freedom for the axes relevant to the operational space. Thus, as an example, if a simple ground rover were maneuvered across a plane, there would be a two-dimensional, three DOF environment (two translation directions and one rotation direction are possible). However, in the case of a flying vehicle or robotic arm with vertical motion through three-dimensional space, there would be a six DOF environment. Meanwhile, if an interface allowed inputs in three translational directions and one rotational direction at a time, that would represent a four DOF interface, regardless of the robot being operated or its environment.

To investigate different mappings of gesture to command in a fully-wearable interface, a system was adapted using a combination of surface electromyography (sEMG) sensors and inertial measurement units (IMUs) to allow multiple inputs at once. With sEMG sensors alone, only one input could be made at a time. IMU integration enabled multi-input operations. Over the course of multi-input mapping development and study, translation inputs were selected to be driven by IMU data while rotation inputs were predominantly by sEMG control. Mapping translation and rotation in this manner was done in order to match operator gestures with an anticipated mental model of end-effector motion, in an effort to make the mappings more intuitive. It was acknowledged that due to the nature of sEMG signal processing, any subset of commands mapped to the sEMG portion of the interface would preclude those
commands from being input at the same time, thus reducing the overall interface DOF. To both better characterize the gesture mappings being investigated and inform future interface/input mapping designs, further investigation would be required to determine how removal of different degrees of freedom from the interface would impact operation in a six DOF environment. Furthermore, understanding the impact on operations of using different DOF limited controller conditions would have an impact on robotic interface design beyond space and gesture applications.

Thus, this work sought to explore the following overarching research problems:

Problem 1: *Can wearable gesture control be used as an effective control interface for a complex robotic arm system?*

Problem 2: *Does a multi-input gesture control mapping provide enhanced performance over traditional single-input gesture control strategy in operation of a high-complexity robotic system?*

Problem 3: *How are performance, workload, and technique affected by the available degrees of freedom in a control interface?*

1.2 Previous Literature

1.2.1 SSRMS Operation

The SSRMS, or Canadarm2, is a 7-degree-of-freedom manipulator system designed to reach as many regions of the International Space Station as possible [69]. Although the joint design is meant to mimic a human arm, with a "shoulder," "elbow," and "wrist," control of the robotic arm is a notoriously difficult challenge. The arm may be operated aboard the ISS from one of two Robotic Work Stations (RWSs), or can take commands from ground controllers. Whether controlled from ground or on station, an SSRMS operator relies on three display screens with views from different cameras outside station and on the Candarm2 itself to guide its direction. The operator must choose camera positions wisely to provide suitable visual feedback, and thus avoid singularities in the manipulator system and collisions [61]. Commands input by the
operator control the manipulator arm at the end-effector, or "hand" segment. The control interface for the SSRMS was designed with two joystick hand-controllers, with three translation DOF controlled by the left hand joystick, and three rotation DOF controlled by the right hand joystick [13].

An important factor for ensuring successful SSRMS operation is that the user maintain awareness of the robot arm's orientation and surroundings. The Candarm2 allows selection of different coordinate systems for controlling the end effector, which changes the mapping between joystick inputs and end-effector motion. Options for operational reference frame, or Command Frame, include axes centered at and moving with the end-effector (referred to here as "internal"), axes fixed with respect to the center of the ISS truss (referred to here as "external"), and others fixed to a point of preference such as payloads attached to the manipulator, points along station, or a point in space. The dynamics and position of the arm is calculated within the SSRMS computer using additional joint frames [13]. One attempt to improve operator awareness during missions was to provide 3D animations of potential robotic maneuvers for controllers to use during training [61]. Such animations would allow operators to view arm motion from varying angles prior to taking the controls, and have a better understanding of arm joint motion during different movements, although the result of implementing this system is yet to be determined. Other attempts to mitigate risk of collisions due to the challenging nature of SSRMS operations have included extensive operator training pre-flight on generic skills such as basic arm navigation and planning [18]. This has included instruction using the "Dynamic Onboard Ubiquitous Graphics" (DOUG) system, a three-dimensional desktop tool that allows astronauts to view the full station, added objects, and Canadarm2 from any angle in a virtual environment [1]. Additionally, long-term, detailed planning of robotic arm movements have been used for specific missions, with SSRMS operation essentially choreographed and rehearsed by the crew. The Canadian Space Agency (CSA) and NASA began pre-mission planning of ISS assemble missions two years in advance, detailing explicit manipulator motions with adjustments based on operator input up-to and through real-time support during operations [49]. As a whole, the
Canadarm2 represents a high DOF robotic system operated in a six DOF, large-scale complex environment, and requiring real-time control capability - a system of the kind that was of interest for this research.

1.2.2 Creating Intuitive Interfaces

As discussed previously, one of the main goals for this study was to understand how interface design decisions affect system performance and workload, with the goal of enabling more intuitive interface designs in the future. An intuitive control interface is both easy for a new user to learn quickly and responds to command inputs in an expected way. Once implemented, such a design should reduce training time for a system, and ideally reduce cognitive and/or physical workload. Some define such intuitive interaction as a "subconscious application of prior knowledge" [37]. This requires matching the system's responses to the operator's idea of what is happening (their mental model of the system), or designing with user expectation in mind.

A long-time proponent of user-centered design, Donald Norman is an expert on human interaction and cognitive perception of products. In his extensive writing, he frequently evaluates the concept of "human error" and provides a simple theory: while error is often blamed on the ability or alertness of the user, they are not the cause of the problem. The design of the system is flawed in a way that leads to errors, or more specifically slips and mistakes [60], by not engaging the human user in an appropriate way [51, 52]. Many interface designs considered "intuitive" are based on previous systems. In his work, Raskin considers intuitive to mean familiar [59]. He points out that just because a system is based on an interface that came before and is easy to learn does not make it effective for its purpose. However, an interface without any basis in the familiar can be frustrating.

In discussing design of gesture-based systems, Norman points out that poor design can still lead to problems when using natural-seeming motions as an input [53]. One highlighted example is the Wii hand-held controller used in a bowling game, which people tend to let go of when swinging their arm as if throwing the ball. This corresponded to a real environment, where people would release their hands bowling.
Nintendo tried instructing players to wear a wrist strap to keep the controller from becoming a missile, but this has merely made Norman's case that effective interfaces must include obvious hints and signs, called signifiers, that make intended use seem simple. Additional cues may be used if not all signifiers are apparent, however adding extra instructions will not fix design [53]. An effective gesture interface design would still have some learning curve, but an explicit conceptual model is key to successful design implementation. Even still, interface designs must include options to reverse actions should the user find that they moved in an unexpected way that the system was still able to interpret, creating an unintended state [53, 66].

1.2.3 Multi-DOF Interfaces in Six DOF Environments

As defined in Section 1.1, it is possible to have a human-machine system in which a machine or robot with multiple degrees of freedom is designed to move through a six DOF environment in a three dimensional way, yet the human interface may have degrees of freedom different from the machine being operated, the environment, or both. Historically, designs for multi-DOF interfaces have reflected either the machine or environment DOF.

Early Design for Multiple DOF

As three dimensional graphics became practical in computer systems during the late 1990s, research into user-driven design of interfaces for maneuvering through three dimensional environments had virtual applications (e.g. simulator environments, 3D graphic arts software) not driven by the mechanical dynamics of a system (e.g. an aircraft) [78]. Interfaces with input decomposed into individual DOF controllers via buttons, sliders, or their virtual counterparts, each operating a single DOF, may be a simple design for mapping inputs to a system's output motion and allow mapping of a two DOF interface such as a computer mouse to six DOF applications. However, a user would be required to coordinate in time the individual activation of each DOF, and as observed by Zhai [78], if a user "has to time-multiplex between the separate
degrees of freedom, it is not possible to form a coordinated movement in the six DOF space.” Furthermore, dividing rotational DOF control into individual axes controls without any form of integrated interface or visual aid was found to be unwise [78, 54]. Humans have been found to be limited in their ability to interpret orientation as a series of individual axes rotations [54] making operation with such decomposed axis controls exceedingly difficult.

Interfaces explored for six DOF virtual environments with computer graphics applications included an array of devices such as hand held “3D Ball” or hand-held trackers one moved freely through the air [32], attempts at computer mice that may be “pivoted” on their edges to add DOF, roller mice allowing vertical input (the modern scroll wheel), rate-controlled spheres with elastic suspension allowing force-feedback, virtual “3D” pen devices, and “Flying Mice” interfaces similar to the “3D Ball” but shaped for the hand and allowing integrated buttons [78]. These interfaces had differences in terms of having position control, rate control, or the presence of force-feedback, but all were fundamentally six DOF interfaces being designed to map to a six DOF environment, or three DOF interfaces in the case where the environment called for translational movements only. In studying the strengths and weaknesses of such devices, Zhai [79] compared user coordination with the same single-handed six DOF input device adapted as either a position or rate controlled interface, asking participants to move a tetrahedron cursor to a target of the same shape in a three dimensional virtual environment as quickly as possible. Operation with a position control device following a master-slave style input (i.e. where system motion is meant to precisely mimic the motion of the input device) resulted in faster task times. However, coordination, measured by directness and speed of trajectory, was higher with the elastic rate control device [79].

In six DOF telerobotics applications such as the original Candarm or remote handling in hazardous environments, King [40], Fischer [22], and Zhai [78] each proposed or recommended variations on one-handed six DOF devices with rate control such as haptic controllers, joysticks, or suspended ball devices as potential practical interfaces. Fischer [22] argued the importance of single-handed interfaces as a critical
design choice, allowing operators to maintain a free hand for secondary tasks. In design of the original Canadarm interface, several single-handed options were explored [40, 45] and a single-handed joystick-style six DOF interface was recommended by those tasked with interface review [48]. However, the decision to use the dual-joystick interface still in place on the ISS ultimately depended on a combination of factors, such as reliability, cost, weight, and the preferences of astronauts who favored the the two-joystick system due to its similarity to past flight interfaces with which they were familiar. [47].

Although information on operation with interfaces with reduced DOF relative to both robot and environment is lacking, effects have been observed isolating or restricting DOF for training purposes. In part task training, a complex task is broken down into different elements, which are taught individually and then brought back together to train the full task [41, 74]. Different strategies include segmentation (breaking the whole task into a series of sub-tasks, typically by temporal division), simplification (adjusting task characteristics to make easier, training versions), and fractionation (teaching one control element at a time) [74, 73]. Fractionation is most similar to DOF limiting as defined here. As an example, fractionation for a task involving pitch and roll might involve isolating pitch control on an interface and providing instruction on that alone, then isolating roll similarly [74]. Unlike fractionation, in which the training tasks with isolated DOF would only require control of the corresponding input, operations using a controller with reduced rotational DOF would involve a full task (e.g., requiring consideration of both pitch and roll) but the interface would never allow use of multiple rotation inputs simultaneously. Within part task training studies, fractionation was actually found to be the least effective of the three main strategies [74, 73] with early work recommending against further use or practical investigation [74]. For tasks involving bimanual operations, part task training which divides the task into operations with one hand at a time do not contribute to bimanual whole-task performance [41]. Research of bimanual tasks using simple tapping or opening motions have shown that a person must perceive the two-handed motions as one, single task or operation in order to succeed [41, 25], making part task training
with division into unimanual components impractical. Within training research, a related strategy which may more closely align with effects of limiting interface DOF is a technique which became known as “variable-priority training” (VPT) [29, 30]. In VPT, different aspects of the task are emphasized or de-emphasized so the user can focus on specific elements, but the whole task remains intact. The emphasized elements show improved performance which carries over to whole-task performance [10].

**Examples on Earth**

In surgical robotics, there are robotic arms with a wide range of DOF. Surgical robotic arms with 4 DOF (utilizing two-DOF shoulder and two-DOF elbow joints) have used master-slave style interfaces, for which robot motion is meant to mimic the operator’s hand placement and orientation [77]. With the robot endpoint controlled by such devices, the interface provides a feeling of 6 DOF interface control for the surgeon. Higher DOF surgical arms may have seven DOF or more [5, 67], and are often controlled via master-slave type haptic manipulators such as hand-held “stylus” devices, hand-held tools, or even arm exoskeleton systems [67]. With these interfaces, the six DOF motion controls may be accompanied by integrated triggers or other single-action forms of input used to command robotic actions in additional DOF (beyond six) such as pinching or grasping [5].

In construction, backhoes represent four DOF arm-like systems in which both traditional control schemes use four DOF interfaces to maneuver through three dimensional (six DOF) space. Traditional four DOF interfaces include dual two DOF joysticks [36] and systems with a three DOF joystick to control the bucket accompanied by a spin (arm yaw) controller [33]. Proposed experimental interfaces use single-handed six DOF haptic devices, typically following a master-slave scheme, thus matching the interface DOF to the environment DOF [36]. Crane operation involves a variety of control systems, typically using a combination of joystick and lever, button, or foot controls [55]. Depending on design, the crane itself may have three or four DOF, while traditional interfaces vary between one and four DOF depending
on the combination of control devices integrated into the interfaces (e.g. a pendant with buttons only, vs. an interface with a two DOF joystick combined with button controls). Crane operation is made more difficult because systems are traditionally controlled either joint-by-joint or at the overhead point supporting the hook, rather than having the operator commands control motion at the payload location, which is more intuitive [55]. Peng [55] proposed several alternative interfaces for crane operation, all of which were variations on three DOF position tracking of the control device itself, disregarding orientation, once again matching the DOF of the interface to that of the mechanical system itself.

1.2.4 Gesture Control

Gesture control is the use of human movements and positions to interact with computer or robotic systems. The idea of gesture control has been a pervasive concept in technology development partially because gestures are among the most expressive and natural forms of human communication [71, 8]. However, even with a system available that can readily interpret human motion, the problem still remains of determining how to map inputs to commands. Natural gestures can take on very different appearances or meanings depending on context. Separate definitions of gesture to command mappings may be better suited for different applications, cultures, or operational circumstances [71].

Current Gesture Control Methods

Before defining any potential mappings for a gesture control system, it is important to know and understand how current gesture methods are being implemented. This includes strategies for mapping limb motion to interface commands and how such strategies are employed in different technologies. There are a variety of gesture interface technologies available today, from vision-based systems like the Microsoft Kinect, to hand-held manipulators such as the Wii-mote or surgical robotics devices, and wearable technologies including surface electromyography (EMG) sensors and
inertial measurement units. Across industry and hardware types, gesture systems tend to be implemented in two ways: a master-slave paradigm, or single input using one-at-a-time commands.

**Master Slave Paradigm** In the "master-slave" paradigm, user motions are imitated by robot or virtual system that is being actuated. This method tends to be a popular approach in development of telerobotic systems designed to perform a task similar to a human limb. The appeal in this tactic is self-evident: if you need to move a remote robotic arm or hand, moving your own to control the robot seems like it would make operation very intuitive and easy to learn. In commercial applications such as the video game industry, gesture recognition systems like the Microsoft Kinect and Leap Motion controller have been implemented in master-slave operations. These readily available systems use a combination of stereo cameras Infrared (IR) sensors, and IR light projection onto a user to resolve a moving person [26, 6]. Challenges of these types of systems include discerning a limited array of gestures or difficulty resolving the user from background clutter. Despite these challenges, Bassily et al. [6], were able to use the Leap system for master-slave control of a six DOF robotic arm, having the robot mimic basic hand movements and pinching motions.

For most master-slave interfaces, systems are specifically designed to operate in highly constrained environments such as a physically confined volume (on the order of human reach or smaller) [67, 77, 5, 21]. Other environmental constraints can consist of operation along defined pre-constrained paths with no adaptation [2], or designed specifically toward imitation of human arm motion and function for use in such applications as remote handling of objects [20, 21, 15, 50]. Some practical systems involve a combination of such environmental restriction, which aid in their reliability for such tasks including remote surgery [67].

The master-slave strategy has been the control approach for most surgical robots to date - a manipulator arm at the surgical site follows the motion of a surgeon’s hand direction, scaled down to match the dimensions of the site and required precision of the task. The manipulator arms on surgical robots often have six or seven degrees of
freedom [67, 77]. In nearly all designs, the operator input is provided with some kind of tangible interface for the hand to grasp, rather than a visual information capturing method such as those mentioned above. These include surgical systems such as the PHANTOM, da Vinci, and CAST devices [67]. The hand-held grip for these systems is generally attached to the robotic workstation, meaning the control mechanisms have a mechanically confined volume as the input workspace.

This approach leads to interesting challenges especially at the boundary of the input workspace. When a surgeon reaches the edge of the workspace for their hand and still has further to move a robotic arm, they must brake the system in order to reset the hand-held controller at the other end of the interface area, then remove braking to continue motion. Another frequent challenge is the mismatch between human hand motion and the kinematics of the manipulator. For example, the da Vinci was designed for small-scale cardiac surgeries, and as a result doctors have difficulty using the device for larger scale abdominal surgeries, with large arc motions causing collisions of the manipulator arms [5].

Inspired by efforts to improve human-robot interaction for telerobotics in hazardous locations, research has been conducted to improve master-slave robotics based on, essentially, human intuition. By first having a person mimic what motions they think a robot arm is making, Pierce et al. [58] were able to record discrepancies in human and robot motion. Subjects imitated planar arm motions of a virtual PR2 humanoid robot, while subject motion was recorded using a Vicon motion capture system. The discrepancies indicate a difference between actual robotic motion and the motion a person perceives as the command they send when using master-slave motion control. These findings are consistent with those of another study which used motion capture human-tracking to validate a master-slave robotic trajectory control method [44]. Lin [44] found that when the robot tried to exactly follow human arm motion, the mechanical differences between the two resulted in a different endpoint trajectory than was input. Both studies confirm the difficulty of producing intended positional output in master-slave robot-human relationships, which counteracts the ease-of-use they were intended to create.
**Single Input Paradigm** Direct inputs in which separate actions or positions each trigger one command have an advantage over master-slave systems in that they may be applied to a wide variety of systems without concern for matching human arm kinematics. This technique has grown popular in the video game industry, particularly for systems that use hand-held motion controllers. Devices such as the Nintendo Wii and PlayStation controllers provide six DOF in control inputs, and shake to provide the user with haptic feedback [56]. These systems are now readily available and, depending on the mapping employed, can be used as an intuitive interface for human-computer interaction [8]. Though there may have been problem cases such as Wii bowling, in single-input style implementations most of Wii sports, Kinect games, and user-developed Kinect content have been acknowledged for success with strong user reviews. However, these systems use an external optical sensor combined with an internal accelerometer to track the motion of the remote [6], and can only track dynamic motions of the remote itself. There is no actual tracking of hand or arm, and the system cannot detect more refined movements like hand position [71]. In addition to relying on an external sensor, this limits the number of discernible gestures that may be used as commands. In a complex system like the SSRMS, over a dozen motion commands would be required simply to operate in six DOF.

Systems based on electromyography (EMG) sensors distinguish hand gestures from muscle activation patterns in the arm. EMG systems which use the single-input approach to computer interaction, such as the Myo armband and the Jet Propulsion Laboratory (JPL) BioSleeve [75], have the advantage of being able to distinguish static hand gestures made with both the fingers and wrist posture. The Thalmic Labs Myo [68] system is a crowd-sourced commercial effort developing a wearable universal remote control. Myo system accuracy has not been published at this time. With a combination of eight surface EMG (sEMG) sensors and two inertial measurement units (IMUs) arranged uniformly in a single band on the forearm, the Myo will let users make specific motions or hand positions as signals to turn on lights, change the page on e-readers, operate RC vehicles, or otherwise act as a universal remote. The JPL BioSleeve uses between 14 and 26 sEMG sensors to control ground and space
robotics [4]. One of the limits in using sEMGs is that an EMG based system can only reliably detect a finite number of hand and arm positions before the signals become too similar and two or more gestures become difficult to distinguish from one another. This shall be explored further in the following section on wearable systems.

Systems that attempt to provide a more dynamic sense of control by using hand motions as a kind of three dimensional computer mouse still follow this single input paradigm. These have been seen in Microsoft Kinect systems [26], and 3D CAVE virtual environments which leverage motion capture to track human gestures. These systems still only carry out one action at a time, but have an expanded range of command inputs, as they capture both dynamic and static gestures. Additionally, since these systems consist of external sensors examining gestures within a space of view, they often incorporate synchronous gestures made with two hands [26].

Operation of the SSRMS for complex tasks relies heavily on being able to direct motion along multiple axes at once. This means operation through direct input of a single command at a time is unlikely to provide adequate operational performance. The robotic arm has kinematics different from that of a human arm, which makes any attempts at a master-slave system of control highly unlikely to function as desired and impractical. Alternate systems of gesture mapping beyond master-slave and single input for SSRMS commands must be considered if a practical, wearable gesture interface is to be a reality.

Wearable Systems

For an SSRMS gesture system, there is specific interest in developing a fully wearable control interface that may one day be integrated into a spacesuit, in order to allow control while on EVA. Wearable tracking systems do not require any external camera for sensing and, once made wireless, free the user from having a fixed operating volume. Data gloves are a technology worn on the hand, and have great potential in ability to provide information on orientation and motion of hand and fingers [42]. Unfortunately, in order to measure any arm motion or orientation parameters, these systems require connection to additional arm sensor systems or exoskeletons. The
combination can lead to arm fatigue from use that leads to sluggish "gorilla arms" [56] due to the weight of the added structure. As observed by Wachs et. al [71], data gloves are able to record finger joint adduction and abduction angles with precision, but such finger movement is difficult in a stiff space-suit glove. The restricted movements of fingers within a space-suit represent the greatest obstacle toward use of such a system in EVA applications.

As mentioned previously, EMG sensor arrays may be used to distinguish gestures involving both wrist postures and finger positions. EMGs alone have been used in several studies as an effective means of controlling robotic arms in 3D space [3, 4]. Artemiadis and Kyriakopoulos [3] attempted to feed real-time EMG data from subject arm motion through a computer model and have robotic arm movement correspond precisely to human joint motions. Focusing on shoulder and elbow joints, they were able to achieve robotic manipulation in 2 DOF accurate to 2cm² variance.

A frequent challenge in use of sEMG technology is collection and accurate processing of signals from the sEMGs themselves. This impacts the ability of such systems to reliably distinguish similar hand positions, or recognize large numbers of gestures. While roughly 26 different hand positions have been recorded on the JPL BioSleeve system, a maximum of 20 gestures have been independently discernible in a single set, and up to 14 have been used in robotic applications [4, 75, 76]. The Myo arm-band has a built in set of five gestures, and encourages developers to try adding their own. However, of the five presets, three of the hand positions are frequently confused when a user attempts to make commands [68]. One option to improve data collection is to improve sensor-skin contact through techniques like embedding sensors in a shape-forming material that conforms to the skin [39].

Using sEMGs in concert with other sensors expands gesture recognition capabilities to include arm orientation and enhance interface performance. Combining EMGs with an accelerometer has been shown as an effective means of control in video game design with single input control [80]. Another option would be to use inertial measurement units (IMUs) to measure both linear acceleration and angular velocity to supplement the EMG sensor information (as proposed by Wolf et al. and Assad et
enabling use of novel control mappings with multiple inputs at once.

**Control of Human Arm Motion by the Central Nervous System**

One possible path for developing new mapping schemes for gesture control of robotic systems, such as the SSRMS, is to investigate how humans naturally move their own arms and hands. While it is preferred to create a mapping that relies on directional commands rather than a master-slave type paradigm, using input gestures that are inspired by human motor control during reach may lead to a more usable interface. Thus, understanding the central nervous system's control of human arm motion and human perception of reach can inform the creation of future gesture mappings.

In normal, voluntary human motion, signals are sent from the brain through alpha motor neurons to relevant muscle groups, which contract to create movement in a limb. Experimental studies of human reach and grasping have shown repeatedly that arm motion from point to point in a healthy adult population is smooth, tending to follow straight or curved paths with a single, maximal peak in velocity [35, 23, 28]. Different theories have been explored to explain what type of control mechanism the nervous system uses to direct movement.

According to dynamic optimization theory, movement is determined by minimization of an objective function [24]. Some hold that the objective function is designed to reach the target with minimal energy expenditure balanced with time efficiency. In contrast, others have suggested motor function resembles a kinematic model that minimizes jerk [24, 35], which can work well for planar motions but suffer breakdowns when 3D motion (vertical plane motion) is introduced. The equilibrium point hypothesis follows the concept that reaching motion is led by the hand [9]. During reach, the arm trajectory would follow a path made up from a series of equilibrium points, or points at which opposing muscles (activated in tandem) exert force such that the sum are in equilibrium [34]. Reaching motion has been shown to follow calculated equilibrium trajectories closely for simple movements [23]. What the experiments supporting the equilibrium point theory have indicated are that human reach is not directed by the determination of joint forces positioning each consecutive
limb segment, but is directed by the shifting motion of the hand reference frame [23].

Each model of motion, be it optimization of a control function or follow the equilibrium point hypothesis, has different real-world conditions for which it breaks down, but equally has different cases for which it is accurate. More than likely, the true control method used in the human nervous system resembles a model combining elements of these theories. For the models above, a person is able to hit their target of reach regardless of unusual conditions requiring adaptation (such as reach during rotation or in microgravity). However, in reality we know that people with intact motor control can miss the point being reached for in various circumstances. Theories on adaptation indicate that people develop a new internal model for motor control under new force conditions [38, 64]. For implementation of a gesture interface with inputs based on arm/hand position, the location along the kinematic chain at which a user focuses their motion could drastically impact operational performance. The principles of the equilibrium point hypothesis suggest that control focused at the hand would be most appropriate.

1.3 Case Study Experimental Environment

Several existing software systems were used in this research. The virtual test environment used for all studies was an SSRMS simulator originally developed in the mid-2000s [27, 46, 72]. The interface inputs accepted by the simulator were adjusted as needed for gesture control and joystick DOF experiments. The gesture interface developed for this work used a gesture recognition program for surface EMG signal interpretation created as part of the JPL BioSleeve program. Integration of IMUs and development of associated algorithms were conducted at MIT, and are detailed in Chapter 2. However, no major changes were made to the EMG signal processing and gesture detection software from JPL.
1.4 The SSRMS Simulator

All experiments used the SSRMS simulator at the MIT Man Vehicle Laboratory, which has been used in several previous studies [27, 46, 72] with the traditional SSRMS dual-controller interface. The simulator environment was constructed using Vizard V virtual reality software (Santa Barbara, CA). The hardware included three monitors displaying different views of a modeled ISS, representing two fixed exterior camera feeds and a view from the end effector (Figure 1-1a).

Figure 1-1: (a) The SSRMS simulator as traditionally configured. (b) SSRMS robotic workstation on the ISS.

Like the actual Candarm2, the simulated robot arm is composed of three segments (Figure 1-2). However, in the virtual model of the SSRMS one of the shoulder joints remains locked, leaving six DOF. The virtual SSRMS is controlled at the end-effector, and nominally maneuvered using a two-Joystick interface similar to operation of the real Canadarm2 on station.

The nominal joystick interface consisted of a Left Hand Controller (LHC) used to control translation (Figure 1-3a) and a Right Hand Controller (RHC) used to control rotation (Figure 1-3b), mimicking the corresponding joysticks used on the ISS. Also like the ISS joysticks, grappling of a docking interface with the simulated SSRMS end-effector was activated by a trigger on the RHC. Some commands such as the brake toggle or switching between internal and external command frames were input through keyboard commands. The nominal joystick command configuration is shown in Table 1.1.

The simulator was augmented to permit gesture control inputs and to selectively
Figure 1-2: Segment and joint structure of the simulated Canadarm2. Red arrows indicate joint rotation directions.

Figure 1-3: The simulator joystick interface Left Hand Controller (a) and Right hand Controller (b)
Table 1.1: Traditional joystick inputs for the SSRMS simulator.

<table>
<thead>
<tr>
<th>Action</th>
<th>Input</th>
<th>Action</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward</td>
<td>Left Joystick</td>
<td>Roll Right</td>
<td>Right Joystick</td>
</tr>
<tr>
<td>Backward</td>
<td></td>
<td>Roll Left</td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td></td>
<td>Pitch Up</td>
<td></td>
</tr>
<tr>
<td>Down</td>
<td></td>
<td>Pitch Down</td>
<td></td>
</tr>
<tr>
<td>Right</td>
<td></td>
<td>Yaw Right</td>
<td>Trigger (Right Joystick)</td>
</tr>
<tr>
<td>Left</td>
<td></td>
<td>Yaw Left</td>
<td>Foot Pedal</td>
</tr>
<tr>
<td>Brake</td>
<td>&quot;b&quot; Key</td>
<td>Grapple</td>
<td></td>
</tr>
<tr>
<td>Rest</td>
<td>Do Nothing</td>
<td>Answer Message</td>
<td>Answer Message (DOF study only)</td>
</tr>
</tbody>
</table>

limit DOF of the joystick interface. All input methods controlled the movement of the SSRMS end-effector, which could be maneuvered using an internal or external command reference frame. When operating in the internal command frame, axes of motion are relative to the SSRMS end-effector (Figure 1-4a), while in the external command frame axes of motion are relative to the ISS at the base of the SSRMS (Figure 1-4b). Within the simulator, the Canadarm2 could also be operated under two different velocity modes. The default setting, or "Coarse" mode was meant for large-scale movements covering long distances. For sensitive operations, such as controlling the end-effector within 2m of a surface or grappling an object, the SSRMS was operated in the "Vernier" mode, which maintains a slower maximum speed. In the simulator, this was set to one quarter the default maximum input velocity.

There are several errors, or violations, that a user might make while operating the SSRMS simulator, each of which would risk damage to the Canadarm2, ISS, or astronauts if it occurred in the real world. Possible violations consisted of collisions of the arm, clearance violations, joint hard-stops, and singularities. Clearance violations referred to points where the arm was within a one meter distance of hitting the station.
or other non-target object, while joint hard-stops and singularities are properties of
the mechanical limits of the SSRMS - points where a joint has reached its limit of
rotation or the arm cannot be moved in at least one axis direction. A display in a
bottom corner of the center screen showing the SSRMS joint angles and their limits
helps novice users avoid and maneuver out of joint hard-stops and singularities (Figure
1-5).

![Joint angle display](image)

Figure 1-4: Internal (a) and external (b) command frames

![Arm Joint Angles](image)

Figure 1-5: Joint angle display for user reference in the SSRMS simulator, with angles
indicated in degrees. Dark slider bars represent the current angles of each joint. A
slider reaches maximum (all dark) when the upper limit of the joint is reached. A
slider reaches zero when the corresponding angle is at zero degrees, the lower limit of
motion.
1.5 The JPL BioSleeve

For this research, the JPL BioSleeve algorithms were adapted for use as a wearable gesture control interface to the simulator environment. The BioSleeve gesture recognition program was chosen because of proven reliability in its earlier single-input iterations for up to an hour, and was able to hold 17 individual static hand gestures (Figure 1-6) during that time [4]. “Static” hand gestures recognition referred to detection of specific, fixed poses as opposed to dynamic tracking of hand motion over time. During early assessments of gesture detectability, the 17 gestures shown in Figure 1-6 were made every ten minutes over the period of an hour at three different arm angles, and were recognized by the BioSleeve gesture classification software with 96.6% average accuracy [75]. When this test was repeated using a subset of 11 static wrist and finger gestures, 99.8% average recognition accuracy was observed [75]. Using sEMG sensors arranged evenly around the user's forearm, static hand gestures are estimated from muscle activation patterns via a multi-class support vector machine (SVM) classifier.

Figure 1-6: Set of 17 static hand gestures used in early JPL BioSleeve development

When the nervous system signals the hand to move, it transmits electrical signals that activate muscles, causing them to contract, and resulting in hand motion. Surface electromyography sensors use a pair of electrodes placed in contact with the skin over a muscle of interest (Figure 1-7). During muscle activation, the action potential is seen by each electrode as a voltage signal. The potential difference between the electrodes is passed through an amplifier, and is recorded as a raw EMG signal [11]. Using an array of sEMG sensors, features extracted from the raw signals may be used to identify patterns of muscle activation beneath the skin [11]. For research purposes,
this work used 14 Delsys Bagnoli sEMG sensors (Natic, MA) arranged in two bands of seven, evenly distributed around the forearm, for detection of muscle activation.

Figure 1-7: Surface electromyography sensors are placed upon the skin over muscles for detection of activation potential (Image credit: Delsys)

A prototype BioSleeve static gesture detection program was operated in Matlab with a graphic user interface. With the BioSleeve's data processing (Figure 1-8), raw sEMG signals were processed in 300 ms windows and feature vectors extracted (in this case, signal standard deviations [76]). Windows overlapped by 200ms, such that new data is processed at a rate of 100 Hz. This means that when a new gesture was made, feature data representing the new hand position makes up one-third of the data at the following time-point. The 300 ms time-frame was chosen empirically as an interval long enough to allow clear, reliable signal data, while short enough to preventing a sense of lag between gesture and system response [76, 75]. During calibration, a user held each static hand gesture to be used as a robot command input for a fixed period (six to ten seconds) during which feature data was collected. These data were assigned to a series of class labels, and used to populate the SVM. After calibration, when the system was turned on, features extracted from active signals from the user’s forearm were sent to the SVM, which used binary classification to determine which of it’s labeled classes the current feature set most closely matched. Once classified, the system had to determine if the user’s current gesture was actually the one identified,
or was a false-positive caused by the user making a hand gesture outside of the labeled classes that should not be acknowledged as a robot command. This was accomplished through a false-positive filter, which only allowed the classified input to be sent to the robot controller as a command if the probability of the current gesture matching the gesture classified was greater than a specified value. This probability was typically set to a default value of 0.5, but could be adjusted for person-to-person variation [76]. If needed in the event of poor gesture recognition, the system allowed active control to be turned off, and further calibration data collected and added to “augment” the existing classes. This calibration augmentation increased reliability of gesture recognition over extended time periods (greater than a few minutes) [76].

1.6 Thesis Structure

The first research problem listed in Section 1.1 is focused on the viability of a wearable gesture interface for teleoperation in a complex environment. From previous literature on gesture interface implementations, we see that the existing paradigms of master-slave and single input each have limitations for the operations of a high DOF manipulator through a complex environment. Master-slave is inappropriate for a robot on the scale of the SSRMS, whose large work environment, kinematic differences from a human arm, and limited operation speed would make scaling to human
arm range of motion impractical. Normal SSRMS operation also involves motion along multiple axes at once, involving use of twelve directional inputs (positive and negative inputs for three translation and three rotation directions) along with non-motion related commands such as grappling, which could make a single input gesture mapping feel limiting to the operator. Thus, gesture control of a system such as the SSRMS would require an input mapping somewhere between these two common implementations, using input of discrete motion commands, yet allowing motion along multiple axes at once. This work investigated the effects of “multi-input” mappings on telerobotic operation, as stated in research problem two.

Furthermore, study of the literature revealed that robotic interfaces in general are designed to have the same DOF as either the robot being controlled, or the operational environment (or both, if robot and environment DOF are the same). No study has been conducted to understand the effects of having reduced interface DOF relative to both robotic and environment DOF. The impact of limiting interface DOF on operational performance, workload, or technique is unknown. In the chosen wearable gesture interface for experimental study in this work, matching either the environment DOF or SSRMS DOF for our case study was not feasible. To both understand the impact interface DOF mismatch may have in this research, as well as provide insight for DOF design in future robotic interfaces for any application, we must study the effects of DOF restrictions in interfaces, as stated in research problem three.

The following chapters detail a series of experiments conducted to address the three research problems. Chapter 2 details project development including interface sensor integration, gesture input mapping design, and early testing of these implementations. Chapter 3 covers a user study comparing several gesture input mappings, including two multi-input implementations as compared to single input and standard joystick control. Chapter 4 presents a study examining the effects of interface DOF restrictions on user performance, workload, and strategy in our SSRMS case-study application. Finally, Chapter 5 summarizes the key lessons from the project, as well as further areas of exploration raised by this research, for which future study could
benefit the field of human-robot interaction.
Chapter 2

Project Development

Study of any multi-input gesture mapping strategy could not be conducted without a functional gesture interface enabling multiple simultaneous inputs. Past versions of the JPL BioSleeve used EMGs only and allowed only one input at once. Multi-input operation was accomplished by integrating Inertial Measurement Units (IMUs) to determine hand position in space as well as static hand gestures. After initial development, preliminary testing was conducted to assess the initial IMU implementation, and determine what adjustments would need to be made for mapping studies.

2.1 IMU Integration

In order to modify the BioSleeve to allow multiple inputs at once, four Inertial Measurement Units (IMUs) were integrated, and used to determine a user’s relative hand position in space. The IMUs were wireless “Opal” sensors by APDM (Portland, OR). All IMUs use a combination of three different internal sensors to provide the orientation of the unit at a given moment: gyroscopes, accelerometers, and magnetometers. Data was streamed real-time through Matlab. When streaming live data, these units provided individual orientations and positions in an assumed global reference frame, relative to the wireless access point at which signals were received (at the computer display). Three were used to estimate a virtual hand position and were worn on a user’s chest, upper arm, and lower arm (Figure 2-1). The lower arm IMU was posi-
Figure 2-1: IMU and sEMG sensor placement on the body

 tioned proximal from the EMG sensors at the user's elbow, such that it would not be moved by roll of the arm and radius for hand pronation (palm down) and supination (palm up) (Figure 2-2). The fourth IMU was placed on the wrist (Figure 2-1) and was used directly with the EMG gesture recognition to identify hand pronation and supination gestures.

Figure 2-2: The lower arm IMU does not move during arm roll right or left

The IMUs on the upper and lower arm were used to gather user arm-segment
orientation relative to the chest IMU and a calibrated "zero" or "rest" position. These orientations were used to direct a virtual arm, and the virtual hand position was then used to determine commands. Each IMU outputs its orientation, which was recorded as a set of Euler angles \((\psi, \theta, \varphi)\) relative to a local, IMU-fixed reference frame (Figure 2-3). These raw Euler angles from each IMU were converted into rotation matrices from the global reference frame to each IMU individual reference frame, defined as \(R_{0\rightarrow1}\), \(R_{0\rightarrow2}\), and \(R_{0\rightarrow3}\) for the chest, upper arm, and lower arm, respectively. These rotation matrices were used to define rotation matrices from the chest to the upper arm local frame \((R_{1\rightarrow2})\) and from the chest to the lower arm local frame \((R_{1\rightarrow3}\) as shown in Equations 2.1 and 2.1). These were used to find the Euler angles between the chest local frame and each arm IMU local frame.

\[
R_{1\rightarrow2} = R_{0\rightarrow2}R_{0\rightarrow1}^{-1}
\]  
\((2.1)\)

\[
R_{1\rightarrow3} = R_{0\rightarrow3}R_{0\rightarrow1}^{-1}
\]
Assuming a fixed shoulder position relative to the torso, and the length of the virtual upper and lower arm segments \((l_u\) and \(l_l)\), the virtual hand position, \(h\), can be expressed as:

\[
h = l_u \begin{bmatrix}
  (\sin \psi_{12} \cos \theta_{12}) \\
  \sin \theta_{12} \\
  -(\cos \psi_{12} \cos \theta_{12})
\end{bmatrix} + l_l \begin{bmatrix}
  (\sin \psi_{13} \cos \theta_{13}) \\
  \sin \theta_{13} \\
  -(\cos \psi_{13} \cos \theta_{13})
\end{bmatrix}
\]  

(2.3)

In this equation, the virtual upper and lower arm lengths were fixed at 0.3 m. This definition of a virtual hand position relative to the chest IMU with a fixed shoulder (also relative to torso) accounts for variations in user anthropology and permits the users to turn or shift posture while maintaining gestures relative to their own bodies. Users were also able to calibrate a personal rest region. Hand movement beyond the boundaries of this region would indicate translation.

Finally, the wrist IMU was integrated directly with the EMG data processing and hand gesture detection program. The only orientation data used for this IMU was roll angle, referenced to the wrist IMU's own local reference frame. When the hand is at the rest position, the roll value is 90 degrees. When the IMU roll exceeded 135 degrees (45 deg. to the right), a "roll right" gesture was recognized (supination for a right handed individual). When the IMU roll was less than 45 degrees (45 deg. to the left), a "roll left" was recognized (pronation for a right handed individual).

### 2.1.1 IMU Characterization

To assess reliability of the IMU data processing and translation input detection, a test was conducted placing the experimental IMU sensors on a robot designed to mimic a human upper arm, forearm, and elbow joint. The robot had no shoulder rotation,
and testing was conducted for the right-left direction only. The chest, upper arm, and lower arm IMUs were placed in the appropriate locations on the robot (Figure 2-4). The IMUs were calibrated for right and left motions as they would be when defining the rest region for a participant. During calibration with a human user, the user was provided with a set of reference axes and asked to move their arm in a given direction at the appropriate time. Typically, the rest limits were calibrated to hand placements between 4.5 and 5 inches on either side of center (precise hand placement varied based on user preference). The test robot was commanded right and left by providing an angle input. It would then move in an arc to the designated angle. To mimic calibration with a human user, IMU-based hand position estimates were taken with the robot held at angles of ±23 degrees (0.117m or 4.6in left and right of center) with center equal to zero and “right” as positive. The positions recorded by the IMU algorithm then became the thresholds for right and left translation inputs (0.072m and -0.058m from center, respectively, in the virtual environment). Once calibration was completed, the robot was cycled right and left by 35 degrees (±0.172m or 6.77in) every five minutes for one hour, for a total of 13 right/left cycles. This provided just over two inches of motion beyond the planned thresholds for a duration equivalent to the planned study time. During standard human operation, users held hand positions at a range approximating two to five inches beyond their original calibration limits (positions varied by user preference) when commanding a translational input. Robot motion and IMU data were recorded (Figure 2-5) and used to determine robot horizontal displacement and the IMU estimate of horizontal position, both at a forearm length of 0.3m. Data was collected at a rate of 16Hz and the robot moved at a rate of two degrees per second during right/left cycles.

Following the gesture study (Chapter 3), the IMUs used for experimentation were damaged due to flooding. All sensors of the model used during experimentation were exposed either to excessive moisture or directly to water. Upon collection of the raw IMU data used for characterization here, it was determined that the on-board filter used to provide merged orientation data directly from the sensor units was corrupted. This rendered processed orientation data pulled directly from these IMUs unreliable.
The damaged IMUs were replaced, but the new sensors have different accelerometer and gyroscope specifications. Characterization of our algorithms was performed using a modified experimental set-up. Raw IMU data from the original model IMUs was processed through an Unscented Kalman Filter (UKF) designed in the MVL as part of an unrelated project. This provided orientation data similar to that produced by the on-board filters prior to damage.

Each cycle appropriately triggered a right or left response from the IMU algorithm, with the estimated hand position crossing the thresholds (Figure 2-5), although the estimates do not perfectly align with true robot motion. Actual and IMU-predicted positions were recorded at a rate of 16 Hz. Comparison of the actual robot and IMU-predicted positions at each time point reveal very few false-positives (less than 0.05% of all data each for false right and left responses) (Table 2.1). False negatives were more prevalent. Of all available data, 1.26% of estimates were failures to recognize that the robot had crossed the bounds on the right, while 1.24% were failures to recognize robot position to the left. These errors occurred during time points prior
Figure 2-5: Robot motion and IMU-based estimation of horizontal displacement for an elbow-simulator robot at forearm length equal to 0.3m. Red bounds represent thresholds from IMU calibration to detect “right” or “left” hand position, which would register as translation inputs in gesture operations.

to the estimated position crossing the thresholds. The average (absolute) distance error between the IMU estimate and the appropriate threshold at false-negatives was 0.013m or 0.524in (0.015m/0.602in right, 0.011m/0.444in left). The maximum distances to threshold for false-negatives were 0.044m/1.739in from the right bound and 0.038m/1.488in from the left bound. This was encouraging, as it indicates a user with this calibration would be able to trigger a left or right input with the hand held less than two inches past the original calibration limits, just as the robot did as it moved to the full 35 degree arc each cycle.

2.2 Gesture Mappings

A total of four input methods were selected to be compared as part of the assessment of gesture implementation strategies. Traditional joystick inputs were used as the study control group, along with three gesture mappings. For the SSRMS case study, it was determined early on that a master-slave approach would be inappropriate as a gesture
strategy. As discussed in the previous chapter, mismatch between the kinematics of a human arm and those of the robot leads to the trajectory the user intends the robot to follow and the one it actually does being consistently different [58, 44]. Surgical robotics have documented cases of unintended robot collisions caused by differences in kinematics, range, and scale between the robot and the operator’s hand [5], all of which are present with the SSRMS. Although the single input paradigm has been successful for robots and computer applications with limited complexity, we hypothesize this would be insufficient for complex tasks. To expand gesture interface application to operations in large-scale environments with real-time navigation and decision-making, a new paradigm allowing multiple inputs at once would be more appropriate.

Previous command-based wearable gesture control systems have mapped separate gestures to single commands to create a gesture language, which an operator would use to interact with the desired system. The main drawback of these gesture languages is the use of any set of gestures which may be reliably distinguished by the interface, and arbitrarily assigning these gestures to the desired computer or robotic inputs. This would result in, for example, a set of gestures which might include pointing one, two, or three fingers as separate commands for an arbitrary set of inputs, as was done in early BioSleeve gesture sets such as that in Figure 1-6 [76]. In the development of any gesture mapping for this research, we sought to move away from arbitrary gesture assignment and instead create a gesture language that matched users’ expected mental
model in order to create more intuitive gesture control implementations. Whenever possible, this was accomplished by assigning, for a given command, a gesture which either in some way corresponded to the motion being input, or leveraged a common action or signal associated with that command. For example, for our gesture mappings “roll right” was consistently made by rolling the hand and wrist to the right, and “stop” was indicated by a closed fist - a gesture commonly used as a stop signal in the military, and in music by orchestra conductors. Not all gestures could be assigned in this manner, but the majority were, in the hopes of making our gesture mappings more intuitive.

One of the primary aims in this work was to examine multi-input mappings, which would allow multiple commands at a time, alongside a single input mapping, such as has been used in past gesture control interfaces. To that end, one single input and two multi-input gesture mappings with different approaches to implementation were developed for testing. Elements of the gesture mappings were adapted as the physical interface, calibration protocols, and algorithms for IMU data interpretation evolved between the initial pilot study and the gesture mapping experiment. Differences in command assignment between the two studies are indicated in the gesture tables below.

2.2.1 Single Input

Single Input (Table 2.2) enabled subjects to make one gesture action at a time. This mapping was similar to those currently used in standard gesture-interface technologies. This implementation used only the sEMG sensors. The EMG-based static gestures were able to indicate a single command each, with no variable level of input. Thus, input velocity was fixed. While previous work showed high detection accuracy for seventeen gestures [4, 75, 76], the total number of gestures that may be reliably differentiated becomes more limited in prolonged usage. To accommodate this limitation, Single Input had two control modes to limit the total number of individual gestures required and to permit intuitive use of the same gestures for corresponding translation and rotation direction (e.g. left and yaw left). Four of the single-input
gestures for translational control were repeated as rotational commands. At a given time, a user could either command the system in translation mode or rotation mode, and use a mode-switch gesture to switch between the two. For any non-directional inputs (e.g. brake, grapple), gestures remained the same in both modes. Introduction of two modes allowed the total number of gestures to remain within reasonable numbers for detectability. Introduction of multiple operational modes in any system can lead to problems maintaining mode awareness [43]. A mode indicator was added to the simulator display for single input operations, but such indicators do not eliminate potential for mode confusion [62]. Operational errors in aircraft systems stemming from mode confusion have been well documented [63, 62]. Here, we acknowledged the potential risk of adding multiple modes for motion inputs, and sought to keep them to a minimum (in this case, two).

Table 2.2: Static gestures for the Single Input mapping

<table>
<thead>
<tr>
<th>Action</th>
<th>Input</th>
<th>Action</th>
<th>Input</th>
<th>Action</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rest</td>
<td></td>
<td>Forward</td>
<td></td>
<td>Roll Right</td>
<td></td>
</tr>
<tr>
<td>Brake</td>
<td></td>
<td>Backward</td>
<td></td>
<td>Roll Left</td>
<td></td>
</tr>
<tr>
<td>Grapple</td>
<td></td>
<td>Up</td>
<td></td>
<td>Pitch Up</td>
<td></td>
</tr>
<tr>
<td>Mode Switch</td>
<td></td>
<td>Down</td>
<td></td>
<td>Pitch Down</td>
<td></td>
</tr>
<tr>
<td>Vernier Toggle</td>
<td></td>
<td>Right</td>
<td></td>
<td>Yaw Right</td>
<td></td>
</tr>
<tr>
<td>Pilot study only</td>
<td></td>
<td>Left</td>
<td></td>
<td>Yaw Left</td>
<td></td>
</tr>
</tbody>
</table>

48
2.2.2 Multiple Inputs

The two multi-input gesture mappings represent different implementations of the same control system. One, called Endpoint Control, is inspired by natural human reaching motion. For the other, called Coupled Input, uses were directed to control translation by moving their forearm, focusing input further up the kinematic chain.

Endpoint Control

Endpoint Control enabled simultaneous multi-axis control, where hand motions indicated translation inputs (detected by IMUs) and hand gestures indicated rotation inputs (detected by sEMG). In this mode, the direction of hand position relative to a calibrated "rest" commanded the direction of translation relative to the selected environment reference frame (Table 2.3). These translation positions can be visualized as reaching toward the sides and edges of a box inscribed about a user's hand (Figure 2-6). To maintain consistency with rotation inputs, IMU-based inputs commanded translational motion at fixed velocity. The rest, brake, and grapple gestures were consistent with those for Single Input. Rotations of the hand at the wrist commanded rotation of the SSRMS end-effector. At any time, the operator could simultaneously use their wrist and fingers to signal a rotation with the translation motion. This

![Figure 2-6: Virtual "box" describing translational input points about the user's hand](image-url)
mapping leveraged the intuitive nature of pointing where a user wants to go, and
was aligned with theories of motor control. Inputs were defined by the placement of
the hand in space, having the hand direct arm movement [9, 23]. Furthermore, this
mapping allowed a subject to command up to four simultaneous axis directions.

Table 2.3: Static gestures for the Endpoint Control mapping

<table>
<thead>
<tr>
<th>Action</th>
<th>Input</th>
<th>Action</th>
<th>Input</th>
<th>Action</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rest</td>
<td></td>
<td>Forward</td>
<td></td>
<td>Roll Right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Backward</td>
<td></td>
<td>Roll Left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Up</td>
<td></td>
<td>Pitch Up</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Down</td>
<td></td>
<td>Pitch Down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Right</td>
<td></td>
<td>Yaw Right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Left</td>
<td></td>
<td>Yaw Left</td>
<td></td>
</tr>
<tr>
<td>Brakce</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grapple</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vernier Toggle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Coupled Input

The Coupled Input implementation was algorithmically identical to Endpoint Control,
with different instructions to the participant. Subjects were told that they would
control translation with the movement of their forearm, and were directed to keep
their forearm parallel to the ground and perpendicular to the plane of their torso.
This changed the kinematic solutions of IMU based inputs (Table 2.4). Thus, subjects
were focused on the movement of their rigidly-held forearm to command translations,
shifting attention to a point higher up the kinematic chain for those inputs. From
the subject's perspective, inputs were divided between translation with the forearm
and rotation with the hand and wrist - two sets of commands which they combined, or "coupled" during operations. All rotation and other sEMG based commands were the same as in Endpoint Control, creating a user mental model in which translation was controlled by the arm and rotation controlled at the wrist and hand.

Table 2.4: Static gestures for the Coupled Input mapping

<table>
<thead>
<tr>
<th>Action</th>
<th>Input</th>
<th>Action</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward</td>
<td>![Image]</td>
<td>Roll Right</td>
<td>![Image]</td>
</tr>
<tr>
<td>Backward</td>
<td>![Image]</td>
<td>Roll Left</td>
<td>![Image]</td>
</tr>
<tr>
<td>Up</td>
<td>![Image]</td>
<td>Pitch Up</td>
<td>![Image]</td>
</tr>
<tr>
<td>Down</td>
<td>![Image]</td>
<td>Pitch Down</td>
<td>![Image]</td>
</tr>
<tr>
<td>Right</td>
<td>![Image]</td>
<td>Yaw Right</td>
<td>![Image]</td>
</tr>
<tr>
<td>Left</td>
<td>![Image]</td>
<td>Yaw Left</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

2.3 Interface Development Study

2.3.1 Motivation and Hypotheses

A pilot study with the newly-integrated gesture interface was needed prior to any gesture mapping experiments, in order to pinpoint any challenges with the proposed gesture control interface, including the algorithm for defining IMU inputs, and determine if any changes would need to be made to the calibration procedures. This first study also provided an opportunity to test-run the static hand gestures that would be used for gesture mappings, verify that they could be distinguished from each other
through the EMG-based detection, and ensure that none were confusing or physically
difficult for test participants. In this study, the hypothesis was that different input
methods, and the ability to input multiple commands at once in a gesture system,
would have an effect on user performance, with improvement seen in multi-input over
single input strategies.

2.3.2 Methods

Participants

Four test subjects performed a series of training and evaluation tasks on the simulator.
Subjects consisted of males ages 18-20. The test protocol was approved by MIT’s
Committee on the Use of Humans as Experimental Subjects (COUHES) and all
subjects provided written consent to participate.

Early IMU integration with the JPL BioSleeve

To enable use of multiple simultaneous inputs, the JPL BioSleeve was modified by
integrating Inertial Measurement Units (IMUs) to determine hand position in space.
In the first version of IMU integration, only two IMUs were used: one on the forearm,
and one on the upper arm. The lower arm IMU was placed proximal to the EMGs
at the elbow (Figure 2-7), such that wrist pronation or supination would not affect
IMU orientation. The raw IMU data in the global frame (referenced to the wireless
access point placed at the computer display) was used to determine the upper and
lower arm vectors, assuming a fixed shoulder point, and find hand position.

At a given time-point, the current hand position would be compared to that at the
previous (immediately preceding) time-point. IMU time-points occurred on the order
of 10Hz. If a difference in position was seen in the same direction for five time-points
in a row, this motion was registered as an input gesture and a command for trans-
lation along the corresponding axis direction(s) was sent to the SSRMS simulator.
Translation in a given axis direction would continue until a “stop” command, initi-
ated by a change in hand position opposite that direction (again, consistent over five
consecutive time-points). Requiring a minimum of five sequential changes in position for input to be registered was meant to prevent accidental motions from being seen as inputs, and theoretically allow variation in the rest position. The changes in position had to exceed an empirically set noise value in order to prevent small, unconscious human movement from being registered as inputs. The noise threshold value (0.1 m) was found by having researchers wearing the IMUs record hand position values while holding their arm “still” at the rest position for ten 15 minute periods over several hours. The average movement along each translational axis was used as the noise threshold.

![Figure 2-7: First version of EMG and IMU sensor placement on human arm](image)

In this way, the system was meant to respond such that a user starting from the rest position could, for example, move their arm/hand to a point forward and right of rest, signaling the simulator to initiate SSRMS translation forward and right. If the user wished to continue moving the SSRMS end-effector forward but no longer to the right, they could move their hand back to alignment with the perceived rest position, while continuing to hold their hand forward of rest.
Input Methods

User commands for the three input methods were Joystick controllers (Table 1.1), Single Input (Table 2.2), and Endpoint Control (Table 2.3). As stated previously, the focus of this pilot study was on testing the gesture control interface hardware, the algorithm that was used for defining IMU inputs, and to test-run the hand gestures chosen for the various gesture mappings. Given that the multi-input mappings (Endpoint Control and coupled input) were algorithmically the same, only one (Endpoint Control) was used in the pilot study.

Experimental Tasks

During this study, participants guided the simulated SSRMS though three types of tasks, known as Path Following, Track and Capture, and Advanced Task. To complete a Path Following task, participants guided the SSRMS end effector through a series of nine rings. Participants were instructed to aim for the center of each ring, oriented with a red bar marking the “top” of the ring (Figure 2-8). Track and Capture tasks gave the participants a ninety second window in which to grapple an HTV cargo vehicle in free drift. Participants had to align the green target display with the grapple fixture on the HTV to capture. The center screen of the simulator displayed a view from the end-effector camera with a set of green cross-hairs and guidelines overlaid. To successfully grapple a target, participants had to be close enough to the grapple pin on the HTV such that their outer guidelines were within the white target line on the grapple pin and aligned with the cross-hairs centered in the white target circle (Figure 2-9). The end-effector also had to be oriented such that it was within five degrees of perpendicular to the target surface using the white ball on the target pin for guidance, with the white target bar horizontal. Advanced Tasks combined skills learned in the previous two. Participants were required to fly to an object, grapple it, and move it to a specified location (Figure 2-10). Unlike the other tasks, which did not require use of the Vernier toggle and operated the SSRMS solely with the internal command frame, the fly-to phase of the Advanced Task required subjects
to approach an object from a distance in the external frame, switch to internal frame and Vernier speeds to grapple the object, then toggle out of Vernier and switch back to external frame to move the object once grappled.

Figure 2-8: Right and center camera views at the start of a Path Following task

![Initial View](image1.png) ![Grapple Envelope](image2.png)

Figure 2-9: View of target from the end-effector view (a) at the start of a trial and (b) with cross-hairs aligned for grapple

**Experimental Protocol**

Participants completed a series of three test sessions over a period of three days, using a different input method each day. At the first session, they went through a basic training presentation on the SSRMS simulator. This included information on the robotic arm itself, terminology, basic commands and operation modes present during all input methods, and an introduction to the challenges and strategies common to operations regardless of input method. A second presentation outlined flight rules
and specific procedures for each type of task participants were asked to complete. A printed guide to the relevant input gestures (or joystick commands) was provided participants while completing tests for that input method. These presentations and printed materials may be found in Appendices A and C. During the following two experiment sessions, participants were allowed to review the training presentations if desired.

Prior to the start of simulator operations for any test session using either the Single Input or Endpoint Control gesture methods, the sEMG system was calibrated to recognize the participant’s unique pattern of muscle activations for each of the hand gestures. Participants held each of the hand gestures relevant to the input method for six seconds to appropriately sample the sEMG signal. The initial calibration was augmented to improve gesture distinguishability by repeating the sEMG-training cycle a second time. This process provided participants the opportunity to realize which gestures were prone to confusion, and to make minor changes to their finger positions, further improving differentiation between gestures, or re-calibrate prior to augmentation if necessary.

As part of the set-up process for Endpoint Control, participants were asked to augment the calibration a third time with the arm in the fully “forward” position as they cycled through the rotation, brake, Vernier, and rest gestures. By having participants perform this extra step, the system was able to more accurately differen-
tiate between the hand gestures when a participant’s arm was placed in a non-neutral position to signal IMU-based translation inputs at the same time.

Following set-up, participants completed a set of hands-on training tasks using the relevant input method, followed by an evaluation during which performance data were recorded. Training periods consisted of two Path Following tasks, nine Track and Captures, and one Advanced Task. The evaluation, consisted of one Path Following, three Track and Captures, and one Advanced Task. This pilot study analyzed the evaluation trials. Part way through the training for each task type (e.g., after three Track and Captures, after the first Path Following) a secondary task was introduced that was also present during each evaluation to measure cognitive workload. At random intervals the word “message” would appear highlighted on the screen. Participants were tasked with responding by either pressing the side button on the right joystick with their thumb during joystick command trials, or pressing the “m” key on the keyboard during the other input method trials. They were told that missed responses would reflect negatively on their overall performance.

In summary, after reviewing the PowerPoint slides, participants repeated the following for each input method:

- Set-up/Calibration
- Training (2 Path Following, 1 Track and Capture, 1 Advanced Task)
- Evaluation (1 Path Following, 3 Track and Capture, 1 Advanced Task)

The training period for each experiment session consisted of the same set of task trials across participants, such that each participant practiced each input method with the same scenarios. However, the evaluations for each session consisted of three different sets of task trials, but at consistent difficulty level. While order of input method used in each successive session was randomized, the order of tasks within the evaluations was the same for each participant, maintaining a progression from simplest task type to most complex.
Data Collection and Processing

Several quantitative performance measures were recorded for each experiment task, including total task completion time, number of violations during task, percentage of task time moving along multiple axes directions (always zero for Single Input), and for track and captures the error between the end effector and grapple fixture at the end of each trial. The end of trial errors were split into two metrics: the linear distance from the grapple target or “position error” in meters, and the angular distance from perfect grapple alignment or “angle error” in degrees. If a participant completed the task by grappling the HTV capsule, the position and angle error were recorded as the errors in alignment within the grapple envelope when the participant triggered capture.

From the secondary task, average time to respond to messages was recorded as a quantitative measure of workload. This metric was designed to indicate higher participant workload when average times are larger.

Complete quantitative data sets were collected for three of the four pilot participants (P1, P3, and P4), while qualitative observations were made for all four participants. During testing for participant P2, the system suffered an error resulting in a failure to write data files. Although the error was corrected, repeated exposure to the scenarios rendered even the partial performance data sets collected for P2 inappropriate for statistical analysis.

Statistical Analysis

Analysis of Variance (ANOVA) was conducted for each recorded dependent variable, with input method as the only independent factor. Given the small sample size ($n = 3$) the results would have limited reliability and certain analyses (such as testing for interaction effects) were not appropriate. Although the small $n$ represents a major limitation of this particular study, these pilot experiments informed the later study focused on gesture mappings (Chapter 4). Each task type was analyzed separately. For the Track and Captures and Advanced Tasks, for which more than one trial was
recorded per participant per input method, the ANOVAs included participant as a factor. Participants only had one Path Following trial per input method, meaning “participant” as a factor was included in random variability. If a data set failed a Kolmogorov-Smirnov test for normality, a non-parametric Kruskal-Wallis (KW) test was performed in place of the relevant ANOVA. When an ANOVA test indicated significance, post-hoc comparisons of input method were made, using a Bonferroni correction ($p < 0.0167$ for post-hoc significance). When a KW test indicated significance, post-hoc comparisons were made using the Dwass-Steel-Chritchlow-Fligner method (already corrected), with $p < 0.05$ for post-hoc significance.

### 2.3.3 Results

Given the purpose of this pilot study to identify usability concerns and improve design for IMU integration and software implementation of the gesture interface, both quantitative results (acknowledging the small sample size) and qualitative observational results were considered. The average values for all metrics in which either an ANOVA or Kruskal-Wallis test indicated statistically significant difference are shown in Table 2.5.

**Path Following**

In the path following task, only average task completion time was found to have a statistically significant difference between input methods. The task times for the Joystick command method were significantly shorter than those for either of the gesture methods.

**Track and Capture**

No significant difference between input methods were observed in the number of violations during track and captures, although it was near-significant ($p = 0.059$). Of 18 track and capture trails using either gesture control method, 16 were incomplete, with participants not reaching the grapple fixture in time, while eight out of nine Joystick
Table 2.5: Average values of measures for which analysis of variance or equivalent non-parametric testing indicated significant difference by input method

<table>
<thead>
<tr>
<th>Path Following</th>
<th>Joystick</th>
<th>Single Input</th>
<th>Endpoint Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Time (s)</td>
<td>272.41</td>
<td>547.76</td>
<td>536.46</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Track and Capture</th>
<th>Joystick</th>
<th>Single Input</th>
<th>Endpoint Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Time (s)</td>
<td>52.44</td>
<td>86.82</td>
<td>90.00</td>
</tr>
<tr>
<td>% Multi-axis</td>
<td>86.43</td>
<td></td>
<td>47.86</td>
</tr>
<tr>
<td>Avg. Response Time (s)</td>
<td>0.46</td>
<td>0.90</td>
<td>0.94</td>
</tr>
<tr>
<td>Position Error (m)</td>
<td>0.21</td>
<td>0.53</td>
<td>1.22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advanced Task</th>
<th>Joystick</th>
<th>Single Input</th>
<th>Endpoint Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Time (s)</td>
<td>420.20</td>
<td>634.43</td>
<td>600.00</td>
</tr>
<tr>
<td>Violations</td>
<td>2.5</td>
<td>2.67</td>
<td>56.67</td>
</tr>
<tr>
<td>Avg. Response Time (s)</td>
<td>0.42</td>
<td>1.09</td>
<td>1.26</td>
</tr>
</tbody>
</table>

command trials were completed. Using the joysticks, the percentage of trials using multiple axes was significantly higher than for using Endpoint Control. Endpoint control trials had greater position error than Joystick \( (p < 0.01) \) and Single Input \( (p < 0.01) \) for position error. There was no significant difference between Joysticks and Single Input \( (p = 0.14) \). Both task time and average response time to messages showed a significant difference between Joystick command and the other two methods, with the Joystick times lower than the gesture-based input methods.

**Advanced Task**

For the advanced task, the completion time using the joystick method was significantly shorter than for Single Input \( (p < 0.01) \) and Endpoint Control \( (p < 0.01) \). The Endpoint Control method resulted in significantly more violations from either Joystick command or Single Input. Joystick and Single Input methods showed no significant difference in number of violations from each other \( (p = 0.999) \). Average response time was also significantly lower for Joystick than for Single Input \( (p < 0.001) \) and Endpoint Control \( (p < 0.001) \).

Across all tasks, the message average response time was consistently longer in both gesture methods than with the joysticks \( (p < 0.01) \). However, during the experiment participants were observed to keep their non-dominant hand in their lap or away
from the keyboard during gesture operations. Thus, their hand was not consistently
over the response button as their thumb was during joystick operations. The time
required to move the hand to the keyboard would have impacted the response times.
For following studies, it was determined that any secondary task responses would
use the same response button across all input methods. Furthermore, in all further
experiments the NASA Task Loading Index (TLX) would be used for subjective
workload assessment.

Qualitative Observations

For EMG-based gesture inputs, participants frequently faced difficulty with the two
“pinch” type gestures used for mode switch (thumb and ring finger) and the Vernier
toggle (thumb and forefinger). These gestures appeared both difficult for the BioSleeve
program to distinguish (from each other as well as from other gestures), and were fre-
quently confused by participants such that they would occasionally make the wrong
pinching motion by mistake. Whenever the BioSleeve software interpreted a gesture
incorrectly, participants would become increasingly agitated, and inevitably attempt
many forceful versions of the desired gesture which were different from the posture
used in calibration, resulting in further recognition failure. Rotational overshoot
when aligning to a grapple pin was prevalent, making alignment excessively time-
consuming and difficult. Participants frequently commented that the rotations were
“too fast” when attempting grapple alignment, despite operating at Vernier speeds
for such tasks. With regards to IMU integration, despite augmentation with the arm
fully extended, the simulated SSRMS would appear to spontaneously roll when par-
ticipants moved in the forward direction, despite maintaining a vertical hand position
(rotational rest). Furthermore, variations in subject posture appeared to alter par-
ticipant ability to control translations. When slouching or after adjusting posture,
arm motion in a desired input direction would not always prompt a response from
the simulator. By contrast, participants often discovered that because the IMU al-
gorithm used relative displacement over a short period of time to detect translation
commands, if they moved their hand slowly the system would not recognize a change
in input, or new command, and the SSRMS simulator would continue with its most recently input action. The led to participants providing a translation input (e.g. "forward"), then gradually moving their arm back to the rest position without triggering and input to stop translation. When later needing to stop translation, participants would then find themselves already in the rest position, and have difficulty stopping the translation in time.

2.3.4 Discussion of Results

The study hypothesis anticipated user performance to be affected by different input methods and the ability to operate in multiple axes at once. Although quantitative analysis is limited in a study with \( n = 3 \), trends support the hypotheses that input methods impact user performance, the results and observations in this study favor the Joystick control over gesture inputs, with no evidence supporting improved performance for Endpoint Control over Single Input. This suggested the initial gesture methods needed refinement. Three main areas were identified as critical considerations to improve the interface: (1) distinguishability of hand gestures needed to be robust; (2) for this implementation of IMUs, body posture had an impact on estimated arm kinematics for detecting desired input; and (3) differences in the controller resolution were important in the completion of tasks requiring fine motions. Although completion time and workload were consistently indicated to be lower using the joystick interface, it is hypothesized that with consideration of these three points, a gesture mapping may be developed that allows effective SSRMS control for EVA operations.

Addressing Distinguishability

Excluding the rest position, the Single Input method required twelve individual hand gestures to be stored and distinguished reliably in the sEMG data. Similarities in muscle activation between certain gestures, or activation during transition from one gesture to another, initially led to frequent confusion of commands by the system.
as participants began their first training for Single Input. For example, the gesture
for forward motion (pointing with two fingers) was sometimes detected as a pinch
of the thumb and ring finger instead, and incorrectly toggled the simulator between
translation and rotation modes. The majority of these confusions were eliminated
through augmentation, but certain gestures continued to be prone to false triggers.
Specifically, the mode switch and Vernier toggle gestures tended to become confused
either with each other, or be triggered in the process of making a fist to brake.
During the Endpoint Control trials, similar observations were made with regards to
the Vernier toggle gesture, but to a lesser degree. This became particularly difficult
for controlling track and captures or the grapple phase of the advanced tasks, for
which maintaining Vernier mode is both required by flight rules and critical to making
precise alignments. The system used a 300ms window to determine what gesture is
being made, with 200ms overlap between windows. Extending this time would have
increase the length of feature vectors analyzed by the SVM, providing more data for
comparison and thereby increasing reliability of detection. Windows of 500ms with
400ms overlap have been used in early BioSleeve prototypes [4]. However, the longer
window would mean that only one fifth of the feature data would reflect a new hand
position for the sample immediately after a change in hand gesture, followed by two
thirds at the next time-point, and so on. Thus, a 500ms window corresponded to a
500ms delay before all new feature data reflected a new hand gesture. This balance
between recognition time lag and feature data has been studied previously in EMG-
based prosthetics applications [57]. In general, collection windows of 300ms or less
have been recommended for maintaining recognition time delays low enough for real-
time operation [17]. Adjustments to hand gesture types, replacing the current Vernier
toggle gesture with something more distinct, provided satisfactory differentiation.

Challenges in gesture discernment occasionally led to participant frustration. Un-
fortunately, when frustration is experienced muscles tense, causing the sensors to
record different muscle activation patterns than originally calibrated, resulting in a
lack of response. Participants had to be reminded during the training phase that ges-
tures must be made with the same muscle force and in the same manner as they were
during the EMG-calibration. The general frustration during training led participants to express feeling a lack of control when operating in this input method. Participants occasionally forgot how they had positioned their fingers when making the Vernier mode gesture, further complicating recognition. Going forward, these difficulties were resolved by limiting the gesture set for Single Input to include only one “pinching” gesture.

Addressing Body Posture

Different challenges associated with body posture arose with addition of the IMUs and dynamic arm gestures for the Endpoint Control method. The algorithm determining inputs by changes in position led to participants gradually moving their arms to a position of comfort after commanding translations, leading to kinematically awkward motions when they attempted to halt that translation later. By requiring a series of time-point to time-point movements, this IMU implementation also encouraged rapid movements which could interfere with EMG-based gesture detection. Additionally, with this input method, arm range of motion became essential to operations. For example, when participants grew tired or otherwise assumed a hunched posture, the arm rest position shifted with the elbow low, near the hip and the arm tucked close to the body. Starting from this position, range of motion was limited when attempting to move the hand down or across the body and did not map well to the calibrated motion space. When trying to input a “left” signal if right handed, or “right” signal if left handed, this rendered participants unable to create a change in position large enough to be registered by the system as an input, and the SSRMS remained stationary. When participants hold their arm out from their body and maintain upright posture, motion inputs become more reliable. Furthermore, small perturbations in body angle (and arm angle) relative to the computer display and data access point would lead to changes in system response. This could be addressed by placing a third IMU on the chest, and calibrating the arm motions relative to the torso, instead of relative to a global reference frame. Given the effect of posture shift, and that this system was meant to be fully mobile, it was concluded that neutral alignment of the arm must
be relative to the self (torso) as opposed to the fixed environment for the re-designed IMU integration.

A concern that emerged during Endpoint Control testing was false triggering of the rotation inputs detected by the EMGs while a participant was trying to translate the SSRMS only. Despite the ability to move in multiple axes with this method, the data show no significant difference in most performance measures between Endpoint Control and Single Input methods. In fact, results show a significant increase in track and capture position error at time-out during Endpoint Control over the other two methods. These results reflect the challenge posed by the observed false triggering of undesired rotations, resulting in participants losing control of their direction of motion and intended orientation. This unintentional rotation is believed to have been caused by cross-coupling between the muscle activation patterns when making rotation gestures and those seen when the entire arm is moved in a particular direction. Thus, the end-effector might begin to roll when a participant moves their arm forward. This became a particular difficulty when participants were operating in internal control frame, such that the axes of motion were fixed with respect to the orientation of the end effector, and would change with respect to the environment during rotation. The cross-coupling phenomena was reflected in the advanced task results, with the higher number of violations seen during endpoint operation as the SSRMS spun out of user directed control. These effects are impacted by both the distinguishability between different hand gestures and arm kinematics. While augmentation of the gesture calibration with the arm fully extended helped alleviate some of these effects by improving gesture recognition, the arm motions and subtle changes in participant posture still resulted in some false rotation triggering. An additional IMU on the wrist connected to the EMG gesture detection could disambiguate wrist pronation/supination to prevent roll-triggered actions when the hand was in rest position, and was integrated moving forward. Additionally, having the system compare user arm positions to an initially calibrated rest position, rather than detecting arm direction by comparing current position to the immediately preceding time point would allow users to make arm gestures in a slower and more controlled
manner. This could reduce noise in the sEMG data which can be caused by rapid, sudden motion and contribute to false triggering.
Chapter 3

Investigation of Gesture Control Mappings

3.1 Motivation and Hypotheses

Once a functional gesture interface allowing multi-input mappings was developed, human-subjects experiments could be carried out to investigate the effects of multi-input implementations on telerobotic operation. A human subjects study was conducted both to address the first two research problems set out in Chapter 1, and to explore what unknown factors may influence implementation of a multi-input gesture system. The problems investigated were: (1) Can wearable gesture control be used as an effective control interface for a complex robotic arm system? and (2) Does a multi-input gesture control mapping provide enhanced performance over traditional single-input gesture control strategy in operation of a high-complexity robotic system?

The following were hypothesized:

\( H1: \text{Multi-input systems provide improved performance over Single Input.} \)

\( H2: \text{At least one multi-input gesture scheme provides comparable performance to traditional Joystick control.} \)

\( H3: \text{That the multi-input mapping inspired by human reach yields different performance than the multi-input mapping less aligned with natural human reach.} \)
3.2 Methods

3.2.1 Participants

Ten subjects (Eight men, Two women) participated in this pilot study. Procedures were approved by the MIT Committee on the Use of Humans as Experimental Subjects, and participants provided written consent. All subjects were right-hand dominant and between the ages of 18 to 30 years (mean age 23.2, standard deviation 3.12 years). Participants were compensated $40 for completing the experiment. Participants were included if they were within the specified age range and had normal function of both hands. Conditions for exclusion were any recent injury of, prosthetic use on, or condition otherwise limiting function of their hands.

3.2.2 Experimental Tasks

Three task scenarios were implemented in the simulator: Path Following (PF), Track and Capture (TC), and Fly-to and Grapple (FTG). During PF, subjects guided the SSRMS end effector through a series of nine rings. Each ring required passing through the center, with the end effector camera aligned to a red mark indicating the “top” of the ring. For TC, subjects had a ninety-second window in which to grapple an HTV cargo vehicle in free drift. The HTV target might drift translationally, rotationally, or both. Subjects aligned a green target display with the grapple fixture on the HTV to capture. FTG tasks combined skills learned in the previous two, by having subjects fly the end effector to a stationary object, and then approach it to grapple.

3.2.3 Data Processing

The simulator recorded operational task performance data for assessing the hypotheses. The dependent measures for hypothesis one were task trial times, PF time per ring, ring alignment errors, and TC grapple success. Trial times were used to derive percent difference in time from the Joystick control case. For hypotheses two and three, additional dependent measures were the percentage of time during a trial...
spent using two or more translational axes (percent multi-translation), using rotation and translation at once (percent rotation/translation), and any combination of two or more axes (percent multi-axis motion). Data was also taken on grapple pin alignment error for TC and FTG trials, response time to a secondary task, number of failed grapples, and number operational errors (e.g. collisions with station elements, lock-ups caused by reaching the physical limits of the SSRMS). Subjective workload was measured using the NASA Task Learning Index (TLX) survey [31].

3.2.4 Experimental Procedure

Each subject provided signed consent to participate in the experiment, and completed a brief questionnaire. This questionnaire included questions of demographic information (e.g. age, dominant hand) as well background information on past exposure to control systems (e.g. video games, pilot experience). All information was de-identified. Subjects reviewed a series of PowerPoint presentations introducing the SSRMS, its dynamics, flight rules, and task overviews prior to beginning hands-on training or experimental trials (Appendix A).

This experiment tested all four of the input methods described in Chapter 2: traditional Joystick controllers, the Single Input gesture mapping (one input at a time), Endpoint Control (multi-input mapping inspired by human reach), and Coupled Input (multi-input mapping for which participants were instructed to control translation with their fore-arm and rotations with the wrist/hand gestures). Subjects completed a series of trials with each of the four input methods. Order of input method exposure was assigned randomly from a pre-determined counterbalanced set such that no order was repeated (test order became part of subject variability). The study took place over four test sessions, with one input method evaluated on each day.

Based on the input system for the selected session, a printed sheet depicting the gesture or joystick commands for that method was provided to the participant (Appendix C). For gesture sessions, relevant sensors were placed on the subject and calibrations performed. In sEMG calibration, participants practiced holding each static hand gesture to populate the SVM classifier. For sessions with multi-input gesture
mappings, subjects would calibrate the IMU system to their own range of motion, setting the minimum distances from the rest state at which each arm motion triggered a translation command. Participants were instructed to use reaching motions when using the Endpoint Control method. When using the Coupled input method, subjects were instructed to keep their forearm parallel to the ground and perpendicular to the plane of their torso as they moved their arm/hand. A set of straight-line axes on a flat surface were provided to guide users in the appropriate direction when defining forward/back/right/left in calibration. This surface was removed to avoid obstruction when calibrating up/down commands.

Each experiment session was divided into two parts, training and evaluation. At the start of training, subjects were presented with an “empty” simulation of the SSRMS and international space station. Subjects were asked to systematically move the arm through each direction of motion individually, in both command reference frames, and activate each SSRMS command. This checklist was used to confirm that they could move in all directions and engage all inputs. Subjects were also permitted to direct the SSRMS freely through this environment, providing time to learn the difference between the two simulator command frames, and confirm gestures were appropriately calibrated. Training continued with a series of two PF, nine TC, and two FTG tasks. Training trials were presented to all subjects in the same order. During training, only the TC tasks had a time limit. The first three TC were simpler than the following six, with two of the three not including any rotation. Part-way through training and throughout evaluation, participants were given a secondary task of stepping on a button when a flashing “message” cue appeared on the center screen. To prevent complications from running all gesture control algorithms with prototype (MATLAB) software, which would over-burden the computer and risk MATLAB errors if continuously run for more than two hours, participants would re-calibrate the IMUs between tasks during training, and following training completion prior to the start of evaluation. These calibrations had the added benefit of avoiding effects of long-term IMU drift on system response.

Training was ended following the specified set of trials with the intention of captur-
ing participant performance after a fixed amount of exposure to each input method. It was acknowledged that subjects would still be learning at the time of evaluation, depending on the learning curve of each method. The evaluation portion of each test session consisting of one PF trial, five TC trials, and one FTG trial. The PF and FTG tasks each had a ten-minute time limit, while TC had the same ninety-second time limit as during training. At the end of each experimental session, subjects answered the NASA TLX survey [31]. In summary, for each input method subjects repeated the following:

- Calibration
- Training (checklist, 2 PF, 9 TC, 2 FTG)
- Evaluation (1 PF, 5 TC, 1 FTG)
- NASA TLX

As described above in Data Processing, data was collected for a range of variables in an effort to understand what factors merit further investigation in future experiments. These included metrics for performance (time and trial success), accuracy (position and angular error at ring or grapple pin), technique (multi-axis usage), and workload (TLX data). The key results presented in this work were drawn primarily from performance and workload metrics.

### 3.2.5 Statistical Analysis

For performance and workload data, Type III ANOVAs were conducted for each dependent variable with a fixed factor of input method and a random factor of subject. Separate models were developed for training and evaluation data. If the ANOVA indicated a significant effect of input method and an analysis of residuals revealed the model was appropriate, pairwise comparisons were performed with Tukey’s HSD method. If an ANOVA model was not appropriate, a non-parametric Kruskal-Wallis (KW) test was performed, followed by Conovor-Inman pairwise comparisons when appropriate.
For PF and FTG evaluation data, analysis of the percentage of time difference from Joystick trials excluded cases in which subjects exceeded the ten-minute time limit and did not finish. For these tasks, incomplete evaluations were outliers that were included in figures for discussion, but were removed from models for analysis. The total n of data points for each input method are shown by task and metric in Table 3.1.

Table 3.1: Number of trials, n, included in analysis by metric and input method

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PF: Time per ring</td>
<td>180</td>
<td>89</td>
<td>180</td>
<td>83</td>
<td>180</td>
<td>85</td>
<td>180</td>
<td>90</td>
</tr>
<tr>
<td>PF: % Time difference from Joystick</td>
<td>20</td>
<td>9</td>
<td>20</td>
<td>7</td>
<td>20</td>
<td>8</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>PF: Ring angle error</td>
<td>180</td>
<td>89</td>
<td>180</td>
<td>83</td>
<td>180</td>
<td>85</td>
<td>180</td>
<td>90</td>
</tr>
<tr>
<td>TC: Grapple success</td>
<td>90</td>
<td>50</td>
<td>90</td>
<td>50</td>
<td>90</td>
<td>50</td>
<td>90</td>
<td>50</td>
</tr>
<tr>
<td>TC: % Multi-axis motion</td>
<td>90</td>
<td>50</td>
<td>N/A</td>
<td>N/A</td>
<td>90</td>
<td>50</td>
<td>90</td>
<td>50</td>
</tr>
<tr>
<td>TC: % Combined translation/rotation</td>
<td>90</td>
<td>50</td>
<td>N/A</td>
<td>N/A</td>
<td>90</td>
<td>50</td>
<td>90</td>
<td>50</td>
</tr>
<tr>
<td>FTG: % Time difference from Joystick</td>
<td>20</td>
<td>9</td>
<td>20</td>
<td>8</td>
<td>20</td>
<td>5</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>NASA TLX</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>20</td>
<td>5</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

Additional analyses comparing variance were carried out using F-test of equal variance. Usage of multiple axes between Coupled Input and Endpoint Control were compared using two-factor ANOVA with independent variables of input method and type of multi-axis motion (multi-axis translation v. combined translation and rotation).

3.3 Results

3.3.1 Effect of Input Method on Task Time

Kruskal-Wallis tests for both training and evaluation PF time per ring revealed a significant effect of input method ($p << 0.001$). Post-hoc comparisons found that Single Input was the least time-efficient method (Figure 3-1). There was no significant difference between Joystick, Coupled Input, and Endpoint Control cases in average time per ring. However, Single Input time per ring was significantly different from each multi-axis input method ($p < 0.001$). A KW analysis of training data for percentage difference in total trial time from the Joystick method revealed a significant difference
(p << 0.001), while no statistically significant difference was seen in KW testing of evaluation data (p = 0.171) (Figure 3-2). Post-hoc analysis of training data show the Endpoint Control method as significantly different from Single Input (p < 0.01). The variances of Coupled Input and Endpoint Control for percent difference in time were compared using an F-test, which revealed that the evaluation data did not have significantly different variances (p = 0.466), while the training data had a significantly greater variance in the Coupled Input than Endpoint Control (p << 0.001).

KW analyses for the FTG trials for both training and evaluation data show no significant difference between gesture methods for percent difference in time from Joystick trials. An F-test of equal variance between Coupled Input and Endpoint Control showed no significant difference between the variance of these data sets (p = 0.315 for training, p = 0.405 for evaluation).

Figure 3-1: Time per ring in PF trials across input methods. Gray markers indicate subject means, and black markers indicate overall means. Error bars indicate standard deviations for overall means.
3.3.2 Effect of Input Method on Task Success

Ring angle errors during PF trial were analyzed with ANOVA models with initial ring orientation as a covariate. In no axis (pitch, roll, yaw) was the error in angular accuracy significantly different across input method.

KW analysis for grapple success rate (Figure 3-3) found an effect of input method during training ($p << 0.001$) and evaluation ($p << 0.001$). For training, post-hoc comparisons show a statistically significant difference in grapple success rate between each of the input methods tested ($p << 0.001$ for all cases). However, in evaluation there was no significant difference between Coupled Input and Endpoint Control ($p = 0.086$).

The trends in grapple success rate during TC training for each input method (Figure 3-4) show a 100 percent capture rate for Joystick method, with distinct differences
in success rates across trial number for each of the gesture methods. In Single Input, trials with a translation and rotation were not successful. With each successive TC training trial there is an upward trend overall in Endpoint Control successes after the first three “easy” trials. In Coupled Input, the addition of a new rotation type in Trial 6 generated a decrease in success, which was not observed in Endpoint Control.

### 3.3.3 Effect of Input Method on Workload

A one-factor ANOVA for NASA TLX composite score found a significant difference across input method ($p << 0.001$) (Figure 3-5). Post-hoc analysis found Coupled Input had the highest mean workload score of any tested input method. Endpoint Control was not statistically different from Joystick use ($p = 0.257$). Coupled Input
workload was significantly different from the other multi-input methods ($p << 0.001$ and $p = 0.015$ for Joystick and Endpoint Control, respectively). The TLX is an additive score built from several weighted components, including physical demand, mental demand, frustration, temporal demand, performance, and effort. Although not often significant on their own, their sum (the TLX score) provides an overall picture of workload. Of the weighted workload components that make up the TLX score, physical demand was highest for Coupled Input and temporal demand highest for Single Input. The “frustration” component drove the shape of the TLX curve by input method. An ANOVA of the frustration component found a significant effect of input method ($p = 0.001$), with post-hoc tests showing both the Single Input ($p = 0.007$) and Coupled Input ($p = 0.001$) are more frustrating to operate than the Joystick. However, the relative frustration difference between Joystick and Endpoint Control was not significantly different ($p = 0.203$). No significant effects were seen in secondary task response time or its inverse for any task.
3.3.4 Usage of Multiple Axes

The percentage of time using multiple axes during TC training for the Joystick had an overall mean of 94.47 percent with a standard deviation (SD) of 1.43 percent, showing near constant use of multi-axis motion regardless of whether the target was rotating or not. However, neither multi-input gesture method reached above 70 percent (mean: 52.86 percent, SD: 9.17 percent for Coupled Input; mean: 52.79 percent, SD: 10.13 percent for Endpoint Control).

The percent of multi-axis time spent in combined translation/rotation and multi-translation in TC training were compared for Coupled Input and Endpoint Control using a two-factor ANOVA. A significant interaction effect was found for the two factors ($p << 0.001$). Coupled Input had significantly higher percent multi-translation than Endpoint Control ($p = 0.002$). Within the Endpoint Control case, percent of time in multi-translation was significantly lower than the percent combined transla-
tion/rotation \( (p = 0.004) \) (Figure 3-6).

Figure 3-6: Comparison of multi-axis use between Coupled Input and Endpoint Control

### 3.4 Discussion of Results

This study examined several gesture control systems with single and multi-input capabilities to determine the potential of such multi-input systems to be effective mobile robotic interfaces. We hypothesized that (1) multi-input systems would provide improved performance over Single Input, (2) at least one multi-input gesture scheme would provide comparable performance to traditional Joystick control, and (3) that the Coupled Input and Endpoint Control methods would yield different performance results. Investigation of these hypotheses led to two key findings: that even with identical control systems, differences in input mapping as explained to an operator can affect performance and workload, and that a multi-input mapping of static gestures based on natural human reach can provide an effective, intuitive gesture-based interface for robot teleoperation in complex environments.

The first hypothesis was supported, as there were improved task time and TC task success for multi-input methods compared to Single Input. PF allowed subjects to pace themselves and involved no time limits during training, while TC was designed with time as a key component. Within PF, Single Input enabled successful task completion, but within TC in the presence of the ninety-second time limits, Single...
Input was not successful for task completion. As the TC targets move, it is unlikely that many of these trials could have been successfully grappled under Single Input control even if the time limit were removed. In the presence of a time constraint, subjects were observed to use multi-axis movements, which enabled an increased task speed and ability to respond to the HTV motion. These data support the need for multi-input mappings in completion of time-sensitive, complex robotic maneuvering tasks.

Hypothesis two was partially supported. Endpoint Control was not significantly different from Joystick for time performance metrics or NASA TLX workload. In PF ring angle error, the absence of significant difference supports gesture control having no effect on accuracy as compared to Joystick. However, a significant difference was found between Endpoint Control and Joystick use for TC grapple success, with Joystick having a 100 percent TC grapple success rate. These results indicate that Endpoint Control as a multi-input gesture method has potential to provide an effective control interface, although there are still limitations. Endpoint Control training trends toward increased grapple success across trials (Figure 3-4). Further investigation of the learning curve for Endpoint Control could reveal reliable grapple success with an optimized training program.

As noted previously, subjects combined motion along two or more axes over ninety percent of the time when using Joystick, which permitted multi-axis rotation and translation simultaneously. A limitation to the selected multi-input mappings is that while multi-axis translation was possible, it could only be combined with a single rotational degree of freedom at a time. It is unclear if multi-axis rotation with Joystick had a significant impact on control strategy and should be explored further. Another factor that may limit the performance of the selected multi-input gesture mappings is how the users orient their torso relative to their perception of control reference frame and how this alters motion. Although the IMU gestures were designed to be referenced to the body, subjects were observed orienting their forearm and hand to the simulator screens while rotating their torso, leading to un-intended sideways motion and frustration in these cases. This study did not involve data collection directly
pertaining to torso posture, and further investigation will be required.

Hypothesis three was partially supported. Although there was no significant difference between Coupled Input and Endpoint Control for PF time performance metrics, a significant difference was observed for TC grapple success and NASA TLX workload. Endpoint Control was observed to have increased grapple success and lower workload than Coupled Input. The differences between Endpoint Control and Coupled Input during TC training indicate that mapping implementation is an important consideration in gesture interface design. The increased workload for Coupled Input may occur because of the body location being controlled by the user. Although Coupled Input is superficially intuitive, the user was controlling a region of the arm more proximal to the body in a rigid manner to enable translation. The commands in Endpoint Control were based on estimated hand position and were directly inspired by the "endpoint hypothesis" theory [10], [11], [12]. In Endpoint Control, subjects controlled the most distal part of the arm to reach a point in space to control SSRMS translation. In Coupled Input, subjects needed to be aware of individual joint angles and arm segment posture, which Flash and Hogan [11], [12] specifically discuss as unnatural. As subjects combined different translation inputs, this unnatural posturing frequently led to participants attempting to place their joints outside of their comfortable reach envelope, causing further frustration.

The effects of a natural hand-based vs. unnatural forearm-based translation control are reflected in the grapple success data from TC training (Figure 3-5). Subject success rates with Coupled Input dropped each time a new rotation type was introduced, while success with Endpoint Control improves with each practice trial regardless of HTV rotation. This suggested a difference in learning style associated with these methods. The upward trend in Endpoint Control suggested learning through development of a general strategy, which may be applied to any combination of gesture inputs. In Coupled Input, each type or set of input combinations appears to be learned separately, as a unit. New combinations required control of a new set of arm and hand positions, as opposed to being the result of smooth addition of individual input motions. This was consistent with the lack of skill transfer observed in early
part task training studies of fractionated training [74], in which inputs were taught individually before being combined later. This type of training was considered inferior to other part-task methods in which skills transferred smoothly to the whole task [74, 73].

To add to these challenges, the work of Ghez et al. [19] on proprioceptive motion suggested that lateral displacement of the arm out from the body (common in the Coupled Input method) may interfere with forward motion control. They observed, the greater the lateral displacement, the more likely a person moving their arm forward guided by proprioception is to turn their arm outward from the body as they reach. This could add a rightward drift to “forward” commands. From observations, some subjects did have difficulty moving forward without any horizontal motion, leading to frequent corrective movement both right and left. This was reflected in the data as an increased percent multi-translation in Coupled Input over Endpoint Control.

In an effort to identify unknown confounding factors, data was taken over a wide range of metrics, and no significant variation was found for grapple pin errors, failed grapple attempts, operational error counts, or secondary task response time. In the case of secondary task response time, it is believed that any potential use for this measure as an objective workload metric was compromised by task instructions which directed subjects to treat the message response as a priority objective alongside task completion, leading to treatment of responses as a primary task with consistent response time. Study results were limited by the small subject pool. Although data was taken on subject demographics as well as past experiences with systems such as video games or piloting aircraft, none of these factors were correlated with the data. Due to the small sample size of the study, we cannot assume that past experience does not influence use of multi-input gesture interfaces without a large-scale study with target subject recruitment. Prior to a full-scale confirmatory user study, investigations into the factors observed as potentially influencing performance should be conducted (i.e. torso rotation relative to the display, interface limitations on combining rotation commands).

While the results support the potential for command-based multi-input gesture
interfaces for complex human-robot interaction, this study also showed that the gesture mapping was critical. These data suggest that a gesture system mapping would be most effective if aligned with natural human motion. While examined here for the SSRMS, this concept may extend beyond space robotics, and could be well suited to systems involving real-time decision making or large, unconstrained environments. Such potential applications vary from ground-based telerobotic rovers to construction equipment. Inclusion of other common interfaces, such as the Wii-mote or joystick variants that may be used in applicable rate-controlled robotic systems, would be of interest in further study.

In summary, this experiment successfully established that a multi-input gesture control system based on natural human arm motion may provide an effective interface for HRI. We have demonstrated advantages in operational efficiency and task performance when using multi-input systems over Single Input. We have also shown that a gesture-interface implementation strategy based on human motor control that has users control their hand improves performance when compared to a mapping that is less aligned and has users control their forearm. Of the methods tested, the Endpoint Control method had the most potential for future use in operation of robotic systems. Future studies should investigate unanswered questions related to the differences in multi-axis use between Joystick and Endpoint Control systems, as well as the interaction with perceived reference frame.
Chapter 4

Effects from Limiting Interface Degrees of Freedom

4.1 Motivation and Hypotheses

In the wearable gesture interface, our combination of surface electromyography (sEMG) and inertial measurement unit sensors enabled multi-input control. However, the nature of the sEMG signal detection required inputs mapped from these signals to be made one at a time [4, 75, 76]. Thus, any aspect of control (e.g. rotation, translation) mapped to the EMG signal (as opposed to the IMU signals) would be reduced to single DOF inputs. During the gesture mapping study (Chapter 3) for both multi-axis mappings rotation was mapped to the static hand gestures detected by the EMGs in order to match gesture to the user’s expected mental model of SSRMS motion in an effort to make these mappings as intuitive as possible. That is, rotating the hand corresponded to rotation at the end-effector or “hand” component of the arm. The results of that experiment indicated a much higher level of multi-translation used with the joystick input method than with either multi-input gesture mapping. Although the fixed velocity employed by the gesture system would be expected to impact multi-rotation time, the effect of limiting rotational DOF by mapping to the EMGs on multi-translation, or multi-axis use strategy in general, is unknown. Further investigation was required to determine how removal of different DOF from the interface
would impact operation in a six DOF environment. Furthermore, understanding the impact on operations of using different DOF limited controller conditions would have an impact on robotic interface design beyond space and gesture applications, prompting the current work. As discussed in Chapter 1, most interfaces are designed with DOF that match either the DOF of the robot being operated, or the environment in which operations are conducted. The impact of reducing interface DOF such that it does not match either the robot or environment DOF is unexplored.

A follow-on study was developed to address problem 3: How are performance, workload, and technique affected by locking different degrees of freedom in a control interface? While it was not possible to add further DOF to the experimental gesture system, we could selectively limit different DOF on the SSRMS controllers in order to compare reduced and full DOF conditions using a single interface. By modifying the joystick interface, different input combinations could be selectively prevented from commanding the robot arm, allowing selective limitation of different interface DOF.

There are three types of multi-axis operation that may be considered in a six DOF environment: multi-translation (motion along more than one translation axis), multi-rotation (motion about more than one rotation axis), and combined rotation/translation (motion along a translation axis and about a rotation axis simultaneously). This study examined the effects of removing different multi-axis capabilities from the control interface. That is, interface DOF were reduced to either four DOF, or two non-concurrent three DOF interfaces, depending on which type of multi-axis motion was limited. A human study was conducted, with different types of interface DOF limitations.

We hypothesized that:

\[ H1: \text{Limiting controller DOF would change user performance.} \]

\[ H2: \text{Limiting DOF in rotation would result in a difference in workload.} \]

\[ H3: \text{Operations with limited controller DOF in the translation controller would affect multi-rotation.} \]

\[ H4: \text{Operations with limited controller DOF in the rotation controller would affect multi-translation.} \]
H5: Operations with any combined translation and rotation prevented would affect other multi-axis use.

Hypothesis H1 was investigated through measurement of task success rate and time. By including NASA TLX assessments and a secondary task for workload measurements, both subjective and workload metrics could be used in testing this hypothesis. Hypotheses H3-H5 are each focused on different aspects of multi-axis technique and strategy. Testing of these hypothesis centered on quantifying strategy and multi-axis use. Inputs were grouped to measure the amount of multi-rotation and multi-translation, along with combined translation/rotation, used during a task.

By testing an interface with differently limited DOF in operation of a six DOF system, the work sought to understand how these limitations affect performance, workload, and technique. In doing so, this research provides insight for future interface design on which DOF are necessary for efficient user performance, and which, if excluded, would impact system operation.

4.2 Methods

4.2.1 Participants

The study protocol was approved by the MIT Committee on the Use of Humans as Experimental Subjects. Each participant provided written informed consent and were compensated up to $20 for their time. Study participants consisted of 21 people, one of whom was disqualified from completing the study, leaving 20 tested participants. There were 13 men and 7 women, between ages 21 and 35 (Mean: 25.5, SD: 3.64). Of the test participants, 19 were right hand dominant. Participants were included provided they had self-reported no injury or pathology affecting hand function, and self-reported visual ability to clearly distinguish all elements of the display. Additional screening occurred during the study procedure, in which participants unable to complete three of ten practice trials for each test case were disqualified from continuing with the experiment. The one excluded test participant mentioned previously
was disqualified in this manner.

4.2.2 Test Environment

The study used an SSRMS simulator [55, 46, 27] with the traditional SSRMS dual-controller interface. The simulator environment was constructed using Vizard V virtual reality software (Santa Barbara, CA). The hardware included three monitors displaying different views of a modeled ISS, representing two fixed exterior camera feeds and a view from the end effector. The virtual SSRMS had three structural segments connected by joints, often compared to a human arm structure. Motion was controlled at the end-effector, or “hand” segment at the end of the arm using the two-Joystick interface similar to operation on station (Chapter 2). The simulator was operated in the internal command frame, with input axes as indicated in Figure 1-4b. The simulator was augmented for this study to permit the altered DOF test conditions. The experimental tasks performed for this study were Track and Capture trials, where participants had ninety seconds to grapple a simulated HTV cargo vehicle target in free drift. Participants had to be close enough to the grapple pin on the HTV such that their outer guidelines were within the white target line cross-hairs properly aligned (Figure 2-9) in order to successfully grapple.

4.2.3 Test Conditions

A total of four input conditions were tested in this experiment: full multi-axis (FM), translation limited (TL), rotation limited (RL), and non-bimanual (NB). In this study the nominal joystick configuration was presented to participants as “full multi-axis” to prevent bias. FM allowed standard control of the SSRMS along multiple axes in any combination. The TL condition allowed participants to translate along only one axis at a time. If they attempted to move the THC to input multiple directions, the SSRMS would stop all translation. In TL, the RHC operated along multiple axes and could be used simultaneously with the THC. In contrast, the RL condition allowed rotational motion along only one axis at a time, while the THC allowed multiple
inputs and both controllers could be used simultaneously. The NB condition allowed normal multi-axis inputs in both rotation and translation individually, but the THC and RHC could not be operated in tandem. If participants attempted to apply input from both controllers, no input would be registered to the SSRMS simulator and no motion would occur.

4.2.4 Data Collection and Analysis

All task performance data were collected through the SSRMS simulator. The dependent measures of interest for the first hypothesis included average trial time, grapple success rate, distance to target at time of failure, multi-translation time, multi-rotation time, bi-manual operation time, percent multi-translation (of time spent translating), percent multi-rotation (of time spent rotating), and percent bimanual operation. Workload measures considered in analysis of hypothesis two included an objective measure and a subjective measure. The objective workload measure was average response time for a secondary task described in the protocol, transformed as the inverse of average response time for analysis. Subjective workload was measured using the NASA Task Learning Index (TLX) survey [31]. All previously described measures were pertinent to hypotheses three, four, and five. However, two additional metrics were used to quantify strategy during investigation of TL trials for hypothesis four. Standard Track and Capture strategy involves combining three elements: (1) visual tracking of the grapple pin, keeping it in view of the end-effector camera; (2) closing the distance to the target vehicle; and (3) aligning the end-effector with respect to the grapple pin to maintain the required orientation grapple. The approach then follows a smooth path such as that shown in Figure 4-1a. In the TL condition, with only one translational motion permitted at a time, it is no longer possible to execute all three of these elements throughout the approach to the target.

Instead, two alternative strategies emerge under the TL condition (Figure 4-1b and c). For the first strategy (Figure 4-1b), referred to as “anticipate and wait,” participants alternate between closing the distance and aligning the end-effector - components “2” and “3” of full tracking. In this strategy, participants project where
the target will move, and orient themselves with proper translational alignment ahead of the anticipated motion. They then close in the distance as the target drifts toward the anticipated position. In the other strategy (Figure 4-1c), known as “end-effector steering,” participants use components “1” and “2” of the complete tracking strategy, sacrificing alignment to maintain visual of the target and close distance with forward input at an angle. This involves the participant rotating opposite the direction needed for alignment, and steering the end-effector toward the target without use of lateral or vertical translations. The time spent (or percentage of trial) moving against expected alignment directions became the first strategy metric, known as “against-goal rotation.” The second metric used to define strategy quantitatively was motion along the x-axis (forward input, Figure 4-1c), measured as either time spent or percentage of trial using x-axis motion.

4.2.5 Protocol

Upon providing consent to participate in the study, participants answered questions on basic demographic information and completed the Vandenberg and Kuse Mental
Rotation Test (MRT) for special skills assessment [70].

Participants were trained on the SSRMS Simulator using a PowerPoint guide (Appendix A). Following a description of the SSRMS and simulator system, participants were instructed how to control translation and rotation in the FM condition and were provided with some brief practice with the controllers. To begin this introductory period, participants operated each joystick separately in the “neutral” simulator environment (SSRMS positioned behind the space station truss, with no targets or other added objects). Then they were given a series of three static targets to approach using the FM condition to become accustomed to the joysticks and environment. This introduction also provided an opportunity to describe the experiment side-task, which consisted of pressing a button on the right joystick every time the word “message” appeared on the simulator center screen.

The four different test conditions were introduced, and participants were asked to complete a series of Track and Capture tasks (Table 4.1). For each condition, participants would perform four practice trials, a readiness test consisting of ten trials, and thirteen experimental trials, then move on to the next condition (a total of 27 trials per condition). After each interface condition, participants completed the NASA TLX. For the practice trials, participants were given two trials where the target drifted in two translational directions (without rotation), followed by two trials where the target drifted with one rotational and one translational direction. The number of target translation and rotation drift directions for a given trial was defined as the drift type. The readiness test consisted of ten Track and Capture trials, which served as both practice and a means of determining if participants fully understood the task and techniques for controller operation. Readiness test trials all had the same drift type, consisting of one translational and one rotational direction. Participants needed to successfully complete three of the ten trials to pass the readiness test and continue. If a participant did not pass the readiness test during a test condition, they were considered not trained, or unable to reliably complete the experimental trials with competency, and were ineligible to continue with the study.

Upon completion of the practice trials, participants completed thirteen additional
Track and Capture trials, each with a mix of translational and rotational drift (Table 4.1). Participants were told to complete each trial with the following priorities: (1) grapple success and completion speed, and (2) answering messages. The thirteen trials were randomized into four orders, with an order assigned to a specific test condition such that each participant faced the same trial order for each condition regardless of condition order. This prevented participants from learning a set trial order during the experiment. Test conditions were presented to each participant in one of two orders to allow later analysis of condition order effects. Orders used were the FM-First order (FM, NB, RL, TL) such that participants started with no DOF limitations and different DOF were removed from there. The alternate, TL-First order followed the reverse pattern (TL, RL, NB, FM). Order was assigned based on MRT score as the assessment was completed, such that balance was maintained between the two order groups for even group size and mental rotation ability (FM-First order MRT scores Mean: 22.5, SD: 6.2; TL-First order MRT scores Mean: 21.9, SD: 6.92).

4.2.6 Statistical Analysis

Initial statistical analyses revealed that these data were non-linear and residuals consistently fail Kolmogorov-Smirnov tests of linearity, prompting the use of non-parametric tests. Kruskal-Wallace (KW) tests were conducted with each of the dependent variables (detailed above), first with condition as the grouping variable, and again with “condition × order” as a grouping variable. For this second set of tests, data were grouped into eight levels (4 condition × 2 order). If a KW test was significant, pairwise comparisons were performed following the Dwass-Steel-Chritchlow-Fligner method. Additional KW tests of the performance (time, success rate) and multi-axis technique variables (percent multi-translation, percent multi-rotation, percent bimanual operation) were conducted across drift type for each condition individually with order pooled. The results of these drift-type analyses are not included in this chapter. However, a discussion of the effects of drift type may be found in Appendix D. Although not all are reported here, a total of 30 KW tests were performed, 25 of which were significant. A level of significance correction was used following the
<table>
<thead>
<tr>
<th>Training trials</th>
<th>Translational Drift</th>
<th>Rotational Drift</th>
<th>Drift Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>2T</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>X</td>
<td>2T</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Readiness test (all trials included twice)</th>
<th>Translational Drift</th>
<th>Rotational Drift</th>
<th>Drift Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experimental Trials</th>
<th>Translational Drift</th>
<th>Rotational Drift</th>
<th>Drift Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td>1T-IR</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>X</td>
<td>1T-2R</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>X</td>
<td>1T-2R</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td>X</td>
<td>1T-2R</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
<td>X</td>
<td>1T-2R</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
<td>X</td>
<td>2T-1R</td>
</tr>
<tr>
<td>9</td>
<td>X</td>
<td>X</td>
<td>2T-1R</td>
</tr>
<tr>
<td>10</td>
<td>X</td>
<td>X</td>
<td>2T-1R</td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>X</td>
<td>2T-1R</td>
</tr>
<tr>
<td>12</td>
<td>X</td>
<td>X</td>
<td>2T-2R</td>
</tr>
<tr>
<td>13</td>
<td>X</td>
<td>X</td>
<td>2T-2R</td>
</tr>
</tbody>
</table>

False Detection Rate method [7], such that an adjusted significance level of 0.0417 was used. Position errors for failed trials at time-out and success rates of the two TL strategies were compared via two-sample t-test.

Within TL, trials were labeled as following the anticipate and wait or end-effector steering strategy based on absolute against-goal rotation time. Though higher use of against goal rotation was expected for end-effector steering trials, some value of against goal rotation was known to exist during the course of nominal operation in the FM case. Thus, the dividing point became the range of against goal rotation time for FM trials excluding extreme values, or the upper outer fence of against goal rotation time for the FM condition.
4.3 Results

There was a significant effect of condition x order on grapple success ($p << 0.001$). Pairwise comparisons revealed a statistically significant difference between TL and the other conditions ($p << 0.001$ for all cases) with TL having the lowest grapple success rate (Figure 4-2a). No significant difference was found between FM and RL conditions. Grapple success was lower for TL than for NB, with both showing reduced performance in the FM-first order. Position errors at time-out for NB failed trials ($n = 18$) were compared with that of TL failed trials ($n = 92$). Mean distance from target at time-out was significantly greater in TL failures than NB failures ($p = 0.038$). When position error was broken into individual axes, mean distance from target in the x (forward) direction was significantly greater in TL failures ($p = 0.035$). There was no significant difference in z (up/down) or y (right/left) error between NB and TL failed trials.

Figure 4-2: (a) Grapple success means by test condition. (b) Percent difference in mean trial time from the FM condition. Not shown: significant difference exists between order levels for the RL condition and the TL condition. For each plot, gray markers depict individual participant means, while bold markers and error bars show overall means and standard deviations for the indicated experimental order group, respectively. Significant differences are indicated by the horizontal brackets.

A significant difference was found in average trial time from nominal conditions across condition x order ($p << 0.001$) (Figure 4-2b), with pairwise comparisons.
indicating no statistically significant difference from nominal for RL \( (p = 0.453) \), but significant differences between all other conditions from each other \( (p << 0.001) \).

Analysis across test conditions with order pooled showed overall effects for both NASA TLX score \( (p << 0.001) \) and inverse message response time \( (p << 0.001) \). However, pairwise comparisons indicated no statistically significant difference between RL and FM conditions for either workload metric \( (p = 0.051 \text{ and } p = 0.61 \text{ for subjective and objective, respectively}) \). From NASA TLX scores (Figure 4-3) NB and TL each had significantly different scores than each of the other conditions \( (p << 0.001 \text{ for all cases}) \). FM and RL had the lowest measured workload while TL held the largest, with NB falling in between. The weighted components of the TLX score generally followed the same pattern across test condition as the overall score. Each workload component was consistently largest for the TL condition, with the exception of physical demand which was similar across test conditions. Inverse response time data supported findings from subjective workload analysis.
Table 4.2: Means and standard deviations of input time data (overall and by experimental order)

<table>
<thead>
<tr>
<th></th>
<th>FM (s) (n=260)</th>
<th>NB (s) (n=260)</th>
<th>RL (s) (n=260)</th>
<th>TL (s) (n=260)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td>OVERALL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trial Time</td>
<td>34.39</td>
<td>8.14</td>
<td>46.29</td>
<td>15.45</td>
</tr>
<tr>
<td>Total Translation Time</td>
<td>31.67</td>
<td>7.11</td>
<td>35.47</td>
<td>9.67</td>
</tr>
<tr>
<td>Total Rotation Time</td>
<td>13.33</td>
<td>7.29</td>
<td>5.78</td>
<td>3.72</td>
</tr>
<tr>
<td>Multi-Translation Time</td>
<td>28.75</td>
<td>5.77</td>
<td>31.54</td>
<td>7.54</td>
</tr>
<tr>
<td>Multi-Rotation Time</td>
<td>3.11</td>
<td>4.04</td>
<td>2.13</td>
<td>2.33</td>
</tr>
<tr>
<td>FM-FIRST ORDER (FM, NB, RL, TL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trial Time</td>
<td>37.27</td>
<td>10.56</td>
<td>50.09</td>
<td>19.46</td>
</tr>
<tr>
<td>Total Translation Time</td>
<td>33.96</td>
<td>9.34</td>
<td>37.85</td>
<td>12.25</td>
</tr>
<tr>
<td>Total Rotation Time</td>
<td>15.06</td>
<td>6.71</td>
<td>4.74</td>
<td>10.50</td>
</tr>
<tr>
<td>Multi-Translation Time</td>
<td>30.55</td>
<td>7.27</td>
<td>33.66</td>
<td>9.40</td>
</tr>
<tr>
<td>Multi-Rotation Time</td>
<td>3.96</td>
<td>5.02</td>
<td>2.70</td>
<td>2.82</td>
</tr>
<tr>
<td>Bimanual Time</td>
<td>14.76</td>
<td>7.92</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>TL-FIRST ORDER (TL, RL, NB, FM)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trial Time</td>
<td>31.50</td>
<td>2.18</td>
<td>42.49</td>
<td>8.43</td>
</tr>
<tr>
<td>Total Translation Time</td>
<td>29.37</td>
<td>1.89</td>
<td>33.09</td>
<td>5.12</td>
</tr>
<tr>
<td>Total Rotation Time</td>
<td>11.60</td>
<td>5.34</td>
<td>4.85</td>
<td>1.90</td>
</tr>
<tr>
<td>Multi-Translation Time</td>
<td>26.96</td>
<td>2.72</td>
<td>29.42</td>
<td>4.10</td>
</tr>
<tr>
<td>Multi-Rotation Time</td>
<td>2.26</td>
<td>2.47</td>
<td>1.57</td>
<td>1.52</td>
</tr>
<tr>
<td>Bimanual Time</td>
<td>11.58</td>
<td>5.32</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

Multi-translation time supported an effect of condition (overall effect: \( p << 0.001 \)), with NB having a greater multi-translation time compared to both FM and RL (Table 4.2) \( p << 0.001 \). However, when considering the percentage of total translation time using multi-translation, there was no significant difference between NB and FM. There was a significant effect of interface condition for absolute multi-rotation time \( p << 0.001 \). Pairwise comparisons revealed larger multi-rotation time for TL \( p << 0.001 \) for both comparisons), but no significant difference between NB and FM. Analysis on percent multi-rotation of time spent rotating, however, did reveal significant difference between NB and FM \( p << 0.001 \). Additionally, significant
effects of interface on total rotation time ($p << 0.001$) revealed TL had greater total rotation time than FM ($p << 0.001$) and NB had a lower total rotation time than FM ($p << 0.001$). Effect of interface on absolute bimanual time ($p << 0.001$) found it was largest for TL and lowest for RL, with all conditions significantly different from each other ($p << 0.001$ for all cases). For percent bimanual use ($p << 0.001$) all conditions were significantly different from each other ($p \leq 0.001$ for all cases) with bimanual operation most frequent in FM and least frequent in TL.

Figure 4-4: (a) All TL trials, marked by success or failure, plotted by percentage of trial time using x-axis input and percentage of trial time using against-goal alignment rotational commands. (b) TL trials divided into anticipate and wait or end-effector steering strategy groups. Gray regions represent overlap between successes and failures for x-axis motion, and the region of trials within five percent of 18.412 seconds against goal rotation (absolute time).

Strategy differences with the TL condition, were determined by x-axis motion and against goal rotation (Figure 6a). All trials with x-axis motion for more than 50 percent of trial time were successful, and all trials with x-axis motion below 30 percent of total trial time were failures. The upper outer-fence for against goal rotation time in FM trials was 18.412 sec. Excluding extreme values, it was assumed that anticipate and wait trials in TL (those not utilizing excess against goal rotation) would have against goal rotation times within this range. Thus, all TL trials with against goal rotation below 18.412 seconds were considered as using the anticipate and wait strategy ($n = 230$), while those trials with against goal rotation time exceeding this
value were considered as following the end-effector steering strategy \((n = 30)\) (Figure 6b). The success rate of the anticipate and wait trials (63.4%) was significantly greater than that of the end-effector steering trials (42.9%) \((p = 0.023)\).

### 4.4 Discussion

This study examined five hypotheses related to interface DOF. As specified in the introduction, the first hypothesis anticipated an effect of DOF on performance, while the second pertained to workload effects. The last three hypotheses pertained to effects of particular interface DOF configurations on operational technique (rotation limiting affecting translation, translation limiting affecting rotation, and an effect of the NB condition on technique, respectively). The first two study hypotheses were supported. TL conditions limited performance and increased workload, indicating that such a set of DOF restrictions would be a poor choice in designing a control system. In the Track and Capture task, it was beneficial for participants to constantly move forward (toward the target), regardless of drift type, to reduce total task time. When any condition required the participant to stop forward motion to take other actions, we expected to see an increase in total trial time. Thus, TL and NB having increased trial time, was expected and supports hypothesis one. Increased time and reduced grapple success rates were observed for NB as well as TL when compared to FM trials. However, with NB these measures were affected to a lesser extent than for TL. The different degradations of NB and TL on trial time and workload are indications that strategy was altered differently by TL than by NB, supporting hypotheses four and five.

This study found no evidence to support hypothesis three. There was no significant difference in trial success, trial time, or workload between FM and RL. When developing new control interface designs, hardware limitations may restrict the number of DOF for the input. This finding highlights that mapping rotation to a single command at a time is a viable option in interface design without loss in performance. With order taken into account, for FM and RL, the condition shown first had a higher
workload than the other. This change in workload is the type of variation expected if the same condition were being tested twice at different times - with exposure, workload was reduced. This is encouraging for future study of the gesture interface, as it suggests the Endpoint Control mapping, or any similar implementations in future applications, does not have any innate affect on performance by mapping rotations to static hand gestures.

These results also indicate that rotation limiting was not the cause of differences seen in multi-axis use during the gesture study between the Joystick interface and the multi-input gesture methods (Chapter 3). The most likely source of the differences observed in multi-translation and combined translation/rotation are the fixed velocities commanded by the gesture inputs. While the joysticks allow participants to apply partial input and thus have some control over SSRMS speed, the gesture interface commands a fixed speed for each static gesture input. As discussed in Chapter 2, fixed velocity commands were chosen to maintain consistency across all gesture inputs, as the EMGs cannot detect "partial" hand gestures, while promoting reliability of IMU detected translation inputs by defining the "rest" region around the participant's hand. Although participants may often use the maximum input available with the joysticks, being able to use variable velocity during HTV tracking enables near-constant translational tracking of the grapple pin, where commanding a fixed translation velocity faster than that of grapple pin drift would reduce the total amount of translation (and multi-translation) time used during a trial. During hardware integration testing (Chapter 2), it was found that participants perceived rotations as "too fast" and continuously overshot rotational alignment goals, which prompted the decision to reduce the commanded speed of all rotation inputs made with the gesture interface. With participants capable of inputting larger rotation velocities with the joysticks than the gesture system, it also becomes possible for users to spend less of the trial time making rotation alignments (reducing combined rotation/translation time relative to gesture trials) even without making use of higher rotational DOF. Further study would be required to verify these suspected effects of fixing input velocities.
TL conditions limited performance and increased workload, supporting our first two hypotheses and indicating that such a set of DOF restrictions would be a poor choice in designing a control system. In the Track and Capture task, it was beneficial for participants to constantly move forward (toward the target) in order to reduce total task time. We expected to see some increase in total trial time for the NB and TL conditions because they required the participant to stop forward motion to take other actions. These time increases were observed and support hypothesis one. Increased time and reduced grapple success rates were observed for NB as well as TL when compared to FM trials. However, with NB these measures were affected to a lesser extent than for TL. The different degradations of NB and TL on trial time and workload are indications that strategy was altered differently by TL than by NB, supporting hypotheses four and five.

Hypothesis four was supported by a shift in multi-rotation technique associated with the two emergent strategies under this condition. During the TL condition, participants could not simultaneously maintain all three elements of normal tracking (visual tracking, closing distance, and alignment). To overcome the limitations of this condition, participants followed two strategies, each maintaining two of the three tracking components. As described previously, the anticipate and wait strategy focused on alignment and closing distance, while the end-effector steering strategy sacrificed initial alignment maintenance in favor of visualizing the grapple pin and closing distance. The higher success rate for anticipate and wait trials suggests that this strategy is more advantageous, and should be promoted in training for similar target-reaching robotics tasks in which all components of tracking may not be used concurrently. The lower success rate for end-effector steering indicates that this strategy could be detrimental to overall performance under any condition, including FM. However, the total number of end-effector steering trials was small, and given the discrepancy in the size of n between the two strategies, further experimentation with strategy instruction would be required before any definitive recommendations to train against end-effector steering could be made. However, the steering strategy would also be detrimental from an operations perspective for a variety of robotic
systems, as it would require more fuel/power — a limited resource.

Input technique of the same type as end-effector steering is commonly used for basic movement by players in role-playing video games with “look” or “pointing” control [12]. However, the participant group in this study did not have enough variation in video game exposure or large enough sample size to determine any correlation between gaming experience and use of the steering strategy. Among the nine participants who did use end-effector steering during experimental trials, frequency of use varied. Only one participant used the steering strategy consistently throughout their trials, while 16 participants alternated between both strategies using end-effector steering three to five times total, and three used end-effector steering only once before going back to anticipate and wait. The one participant who favored end-effector steering was also the only person to attempt this strategy during the readiness test trials. This suggests that even though participants were all reliably completing the task by the end of the readiness test and considered fully trained, there were still elements of learning and strategy development during the experimental trials.

Finally, the fifth hypothesis was supported by reduced rotation time during the NB condition, indicating an effect on technique from the removal of bimanual control. In the NB condition, participants had to pause forward motion to perform rotation, and while technique was altered, the standard tracking strategy was still present. Participants were still able to maintain visual and close distance while adjusting translational alignment, with occasional rotational alignment adjustments. The effect was a minimization of rotational orientation adjustment, such that increased trial time as compared to FM was accompanied by shorter rotation and multi-rotation time. This reduction in multi-rotation suggests that NB may have potential use in training as a means to optimize rotation efficiency.

These data supported that FM and RL performance parameters were not significantly different. Yet, FM techniques still utilized occasional multi-rotation. Thus, we can infer that for this task, multi-rotation may not be necessary for performance, but it can be used effectively with training. However, a dual-joystick study by Wang [72] recommended that it is desirable to train novice users to avoid unnecessary multi-
rotation, or “train out” unintentional rotation cross-coupling. Given the success of the RL condition in this study, and the efficient rotation seen during NB, it is possible that combinations of RL and NB restrictions on 6 DOF controllers could be useful in early training of novice users. Although part task training through fractionation of the task into concurrent components has been demonstrated as a poor training technique [74, 73], emphasizing and de-emphasizing control elements while maintaining the whole task in variable priority training (VPT) [30] has been shown to improve overall performance for the emphasized elements, with skills transferring to whole-task operation [19, 10]. Removal of simultaneous translation and rotation without altering the task did not directly emphasize rotational input, but reducing emphasis on rotation/translation coordination and added time pressure did effect rotational input efficiency in the manner of targeted VPT. The effects of trial design (drift type) on operations support the above findings, and a detailed discussion of drift type analysis has been included in Appendix D.

There were limitations in this study that may affect generalization of these results. As discussed above, participant spatial abilities were assessed and order group balanced using MRT scores. Most participants in this study scored above-average, thus it is not known how spatial reasoning skill would affect limited DOF performance. The Canadarm2 joystick interface used for this study was a bimanual system. It is unclear if the relationships between translation and rotation observed would apply to a unimanual interface. However, past studies in asymmetric bimanual rhythmic tasks suggest that even when motions between hands are different, the brain can perceive the actions of the hands as a single task, allowing coordination to persist [25]. This indicates that operation using the bimanual joystick configuration used for this study may have been perceived by the brain as a single task. Operations with a unimanual interface may be perceived similarly, suggesting that strategy observations from this study may translate to single-handed interface use, though further investigation with unimanual interfaces is required. Additionally, the simulator interface was rate controlled. Further study would need to be conducted to determine if the design and training recommendations made here would be useful in position controlled systems.
Although some position controlled interfaces follow a master-slave paradigm \[67\], which are based on human hand placement/orientation and less likely to involve interface DOF reductions, other experimental interfaces use position control primarily for the translation aspects of operation \[36\]. Such systems may benefit from incorporation of simultaneous rotations or rotation limiting. The current study evaluated the SSRMS joysticks as a case study. Future investigations are expected to include study of DOF limitations across a range of common interfaces, such as the Wii-mote or other joystick-type controls used for robotic rovers or construction vehicles, to assess for the trends seen here across a broader set of design applications.

In summary, this research aimed to determine the effect of reducing control interface DOF on telerobotic operations. No significant difference in performance or workload was seen between full multi-axis and rotation limited controls. Preventing simultaneous rotation and translation minimized use of multi-rotation and rotation in general, instead favoring time spent translating. It is recommended that future interfaces use such separation in training to optimize rotation use and efficiency. Although translation limiting resulted in poorest performance and highest workload, strategies observed indicate that tactics of anticipation and alignment during goal-reaching tasks are more efficient than steering a robot and aligning later. These findings have potential to guide design and training requirements for future telerobotic interfaces.
Chapter 5

Conclusions

This work has demonstrated which implementation techniques for gesture control of complex robotic systems should be followed or avoided in development of a successful input mapping. It has also shown the effects of restricting various DOF in multi-input interfaces, and the value of limiting rotational DOF when necessary. During experimentation with the Canadarm2 simulator, not only were several key results found which contribute to our understanding of human-robot interaction and interface implementation, observations were also made which pose new avenues for further research on interface implementation techniques.

An input mapping based on natural human reach was the only one of those tested which matched joystick performance for practical gesture interface implementation. The gesture mapping study (Chapter 3) revealed comparable time performance and workload between a traditional Joystick interface and the Endpoint Control mapping inspired by hand-led models of natural human arm motion. Investigation of training data indicated that, with practice, performance in operating the SSRMS with a gesture interface using a human-reach based implementation meets that of operation with the joystick interface. This not only answers research problem one, showing that a wearable gesture interface can be used as a viable controller for telerobotic systems, but goes on to show that implementations based on natural human motion have potential to make use of such interfaces more intuitive.

User implementation in gesture control drastically influenced performance and
workload, even when the software and physical system were unchanged. Despite using the same algorithms for gesture and hand-placement recognition, participants experienced significantly higher workload and reduced task performance with the Coupled Input implementation as opposed to Endpoint Control during experiments in Chapter 4. The only difference between these two gesture mappings was in the instructions to participants on how to hold their arm during operation, and which portion of the arm they were told defined translation inputs. Focusing their movements at a point higher up the kinematic chain of the arm for Coupled Input led to participants maneuvering into awkward, uncomfortable arm positions, and had a negative overall impact on SSRMS operations. The differences observed between these two gesture mappings underline that implementation affects operational performance and workload, and is as critical to creating a successful interface as design.

Limiting rotation DOF of an interface did not significantly affect user performance, workload, or technique. Investigation of interface DOF (Chapter 4) has shown that reducing rotations to one at a time, creating a 4 DOF interface, does not adversely affect operation of a 6 DOF robot through a 6 DOF environment. Based on this evidence, we propose a design criteria for reduced DOF interfaces: that rotational DOF be chosen as limited when DOF restrictions are necessary due to hardware limitation. Additionally, this work supports the use of rotational DOF limiting to potentially improve performance with systems with 6 DOF interfaces, but which are prone to excessive rotational cross-coupling that may be detrimental to performance.

Preventing combined translation and rotation commands in an interface improved rotational input efficiency. In a robotic system such as the SSRMS, operators are trained to use as few inputs as possible when completing a task, in order to improve efficiency and conserve resources (i.e. power, fuel). When bimanual operation was removed from the SSRMS joystick interface (Chapter 4), preventing simultaneous translation and rotation commands, overall rotation inputs decreased despite increased total task time. Though further study is required, there are potential benefits of this DOF separation as a training tool. If a key goal in training of teleoperators is to minimize total inputs, purposefully separating translation and rotation to en-
courage reduced rotational inputs could improve rotation efficiency post-training.

The lessons learned from this research have applications far beyond the case study explored. The guidelines developed and used here for design of a gesture input mapping (basing gestures in the personal reference frame, use of gestures corresponding to robot motion or direction, and use of multi-input schemes for complex systems) may be applied to any mobile gesture interface and robotic system. Gesture interfaces using technologies other than EMGs and IMUs, such as imaging-based systems common in gaming and industry applications. For the JPL BioSleeve project, evidence that rotation limiting does not effect user performance has allowed progress to continue on design of a wireless wearable gesture sleeve integrating EMG and IMU sensors, which would implement gesture mappings in an Endpoint Control type scheme. Such a wearable gesture interface would have applications both in microgravity scenarios such as the SSRMS and ground-based robotic control such as operation of Mars rovers (or reconnaissance rovers on Earth) being used by human exploration teams. Application of DOF study results extend beyond gesture control, with potential use for rotation limiting in construction equipment, remotely operated vehicles, or any velocity-controlled robotic interface involving multi-DOF.

Through experimentation and investigation of the research problems, the following contributions were made to the fields of Human-Robot Interaction and Space Robotics.

1. It was found that neutral alignment of the arm must be relative to the self/torso as opposed to a fixed environment, requiring an IMU sensor array with body reference. However, testing with a non-immersive, fixed, visual environment contributed to user personal reference frame shifting from local to global, causing frustration and requiring further study.

2. Assigning gesture commands to both the hand and to a proximal arm segment along the kinematic chain was detrimental to coordinating multiple inputs, leading to unnatural, physically difficult motions.

3. It was found that the instruction initially given to a user for system operation
had a significant impact on overall operations, underlying the importance of direction in implementation.

4. Single input was not an effective gesture implementation for complex robotic tasks. We have shown that multiple simultaneous inputs is critical for reduced workload as well as improved efficiency for operations in a complex environment.

5. It was found that limitation of rotational DOF did not significantly affect performance. Thus, rotation DOF limiting is recommended in interface design when hardware restrictions prevent six DOF, or the system is susceptible to unintentional rotational cross coupling.

6. It was found that separation of translation and rotation improved rotation input efficiency, and recommend implementation through a variation on Variable Priority Training.

7. It was found that prevention of true target tracking yielded two emergent strategies, for which the anticipate and wait strategy was more effective than a steering technique. These strategies require further study to better understand their applications.

Over the course of the experiments above, further questions arose which require future study. Though understanding the effects of limited interface DOF represented a literature gap given high priority for study, there were three differences identified as potential confounding influences in comparison of multi-input gesture control with traditional joystick interfaces. Though we now have evidence that limiting rotational DOF did not impact multi-axis use during gesture mapping experiments, the influence of fixed velocity input in the gesture system still needs to be verified. With fixed velocity inputs, subjects had to stop translational alignments in order to avoid overshoot, creating a pattern of motion observationally similar to that of the anticipate and wait strategy seen during the DOF study. This indicates that further investigation of the anticipate and wait strategy may provide insight into strategies for fixed-velocity interface operations as well as general tracking tasks.
The effects of operating in an immersive environment versus using a computer screen display is still unknown. Although the multi-input gesture implementations were designed for use while riding on or working alongside a robot in its operating environment, experiments were conducted with the SSRMS simulator on the standard three-screen display. The screens themselves may have unintentionally provided participants with an external reference point, shifting their focus from endogenous to exogenous referencing. It is suspected that an unconscious reference to the center screen as “forward” contributed to the self rotation phenomenon noted in Chapter 4, which in turn may cause unintended translational inputs. Preliminary efforts to study Endpoint Control gesture interface performance in an immersive virtual environment are currently underway. Future study is planned to assess the effect of including and excluding exogenous reference cues in both an immersive heads-up display environment and with the standard simulator three-screen display.

Over the course of this work, there were several limitations. All study participants were MIT affiliated, and as indicated by the MRT score data in Chapter 4, MIT students tend to have above-average spatial reasoning skills. If tested with a larger population, overall performance may vary, and different techniques and strategies may be observed by participants than those seen during our research studies. Fortunately, the intended user for the SSRMS case study are astronauts - a population also expected to have above-average spatial reasoning ability. Furthermore, trends related to participant demographics and past experiences were not observed in either the gesture study or the DOF study. Although the DOF study had a larger sample size than the gesture study, variability of participant experiences and small sample size within demographic groups mean that possible influence of participant background on study results cannot be ruled out.

During the gesture study, EMG and IMU gesture recognition was re-calibrated such that operations with each calibration never exceeded one hour. This improved reliability of the prototype BioSleeve system used, but did not allow characterization of how certain factors would affect operation over time with current available technology. In full implementation of an EMG and IMU based interface, the EMG
signals would be affected by fatigue. In practice, exertion and fatigue change the frequency of raw EMG signals over time [14], and thus have an effect on any features extracted from these signals for gesture identification [65]. Inertial Measurement Unit orientation estimates drift over time, and various methods have been used to counter this drift for on-Earth operations [16]. However, for real time uses the Earth gravity vector measured by IMU accelerometers is often used as a point of reference, and this vector is unavailable in microgravity applications. Ways to orient the IMUs in microgravity, such as using magnetic references, are currently being explored by the BioSleeve group at NASA JPL.

Of specific note, the joystick control task was bimanual, yet the gesture system being tested was uni-manual. While the comparisons were appropriate given that the dual-joystick system is the existing interface for the case study examined, a confirmatory study including comparison of gesture control with a uni-manual interface could be beneficial for understanding implementation effects on operational strategy. Additionally, further research with multiple interfaces would be beneficial to fully explore applications for rotationally limited DOF interfaces. Although we may expect that results from the two-handed joystick interface with varying DOF reductions will also hold true for a single-handed interface, further study with a six DOF single-handed interface would be required to confirm the similarities between six DOF and rotationally limited (4 simultaneous DOF) interface control. As mentioned above, a full training study assessing subjects with and without separation of translation and rotation during training on multiple interfaces would allow understanding of possible benefits to training with translation and rotation divided.

The results of this work may be used to guide design of gesture input mappings for gesture control interfaces, and as design recommendations for DOF restricted interfaces, gesture-based or otherwise. It is hoped that with further study and practical application of the contributions made by this work, future robotic interfaces shall be more intuitive, with designs that consider a wider range of interface DOF than those of the robot or work environment.
Bibliography


Appendix A

Appendix A: Experiment Training Slides

A.1 SSRMS Basic Training for Hardware Development and Gesture Studies (Chapters 2 and 3)
Robotic Terminology

- Links are the rigid bars that form the robotic arm.
- Joints allow two links to rotate with respect to one another.
- The End-Effector is the 'grasping finger' of the arm, made up of multiple small joints and links.

Robotic Terminology

- The arm has:
  - 4 Links
  - 6 Joints

Virtual Environment

International Space Station (ISS)

Components

- Core Module
- Truss & Solar Arrays
- Robotic Arm

The following slides will introduce the virtual environment used in the experiment.
**International Space Station**

- **Truss and Solar Arrays**
  - The truss is mounted at the forward end of the Core Module.
  - It is located on the port, starboard, and center axes of the ISS.

**Basic Controls**

- "R" = Reset arm location
- Change Camera:
  - F1 = camera 1
  - F2 = camera 2
  - F3 = camera 3

**Arm Limitations**

There are two types of limitations on the arm's movement: joint limits and singularities. The following slides depict these limitations and let you see what happens when you encounter them.

**Joint Limits**

- It's very important that you pay attention to what each of the arm's joints is doing, not just to where the end-effector is going.
- The arm has physical limits, which are called limits.
- Limitations are the areas on how far a joint can rotate.
- This notice is displayed on the screen when you encounter a limitation.
- Move in the reverse direction to free the arm.

**Singularities**

The arm also has software limitations. It cannot guide movements in particular places, known as singularities.

**Control Frame**

A control frame defines the direction of arm motion as a result of commanded input. You will be working with either an external or internal control frame during this experiment.
External Control Frame
- An external frame describes movements with respect to the environment.
  - Directions are permanently aligned with the ISS.
  - Forward and aft are along the truss.
  - Port and starboard are along the core modules.
  - Up and down are above and below the station.
- Work in an external frame for large arm movements.

Internal Control Frame
- An internal frame describes movements with respect to the camera on the end-effector.
- Good for small movements at the end stage of grappling tasks (grabbing objects).

Reference Frame Controls
- "T" = Internal Frame
- "E" = External Frame

Review of Other Controls:
- "R" = Reset Arm Location
- "F1" = change camera 1
- "F2" = change camera 2
- "F3" = change camera 3
The Grapple Envelope

- You are in the grapple envelope when the following conditions are met:
  - The white dot is in the white circle.
  - The long green line is within 5 degrees of zero.
  - The inner green line is just inside the edge of the large gray circle.
  - The end-effectors are oriented in the gray circle.

- You should engage grapple as soon as these conditions are met. Do not grapple otherwise.

Alignment Techniques

- Work in the internal Command Frame.
- Apply VelCtor Rate, which will significantly slow motion of the arm. You will have fine control of translation and rotation, allowing for easier (and safer) alignment and grappling.

How to correct alignment errors.

<table>
<thead>
<tr>
<th>Error</th>
<th>Technique 1</th>
<th>Technique 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error in the y-axis</td>
<td>Yaw down</td>
<td>Yaw up</td>
</tr>
<tr>
<td>Error in the x-axis</td>
<td>Pitch down</td>
<td>Pitch up</td>
</tr>
<tr>
<td>Error in the z-axis</td>
<td>Roll down</td>
<td>Roll up</td>
</tr>
<tr>
<td>Error in the end-effector orientation</td>
<td>Grapple and re-align</td>
<td></td>
</tr>
</tbody>
</table>

Quick Review

The following slide reviews some important concepts. If there's something you don't understand, feel free to ask the experimenter or re-read the slides on that material.

Review of Important Concepts

- Arm Limitations
  - Hardstops are the limit on how far a joint can rotate.
  - Limitations in the software cause three types of singularities.
- Control Frame
  - External directions are relative to the base; motion depends on orientation relative to ISS axes.
  - Internal directions are relative to the end-effector.
- Grapple Tolerance
  - White dot in white circle
  - White line level within 5 deg.
  - Inner green lines on edge of large gray circle.
  - End-effectors in gray circle.

Training Overview

Now that you know how the ISS arm and environment work, the following slides will tell you more about what you'll be doing in the experiment.
Training Overview

- Training
  - The training will allow you to practice tasks similar to those you will encounter during the experiment.
  - Over the course of today and the next 2 training sessions, you will become familiar with controlling the arm in the ISS environment and will practice fly-to, grasp, track and capture, and path following tasks.

- Metrics
  - Your performance will be evaluated by and you will receive feedback on:
    - Time to task completion
    - Number of failures
    - Final grip rate and alignment
    - Direction of individual movements and temporal validity
    - The accuracy of your path
    - Number of resets of the arm position
    - Awareness to flight rules, described in the next section

Flight Rules

Two Speeds:
- Venerable (slow) – use when close to target (within 2m) or performing small maneuvers
  - Venerable mode will be indicated on monitor 1 when in use
  - Always use Venerable during grasps

- Cruise (standard/fast) – use for large motions
  - indicated by absence of venerable rag on screen

Flight Rules

- Overall clearance limitation: 1m between the arm and all obstacles
  - A warning will be shown on the screen when any part of the arm comes within this limit
  - Look for what has caused the clearance violation and move away to remove the warning
  - You must avoid collisions between the arm and obstacles or with itself

Things to Be Aware Of

- Cameras
  - Use all of your viewpoints, observe clearances, monitor the task.
- Frames
  - Be careful about how to make the arm in the external or external control frame
- Views
  - Use status features to keep cornerstone orientation/locations
- Joints
  - Be aware of location of joint angle limits and arm singularities

Pre-grip Alignment
- Before gripping, the end-effector should be:
  - 2 meters above the target
  - within 30 degrees from perpendicular with the target

Strategies

- Collision/Clearance Limit Avoidance
  - Monitor both the elbow and the end-effector’s position and clearance
  - Keep checking all of the camera views for potential clearance concerns

- Reducing Task Time
  - Move in more than one direction at once and perform both rotational and translational movements together in order to finish the task more quickly
  - Look for the shortest path between the start position and the target and follow that line as best you can
Strategies

- Moving the Arm:
  - Plan ahead. Before you start working, think about the movements required for the best route and how to avoid obstacles.
  - i.e. where will the elbow be if you move the end-effector toward the post? Will it be too close to a heavy object?
  - If you begin moving the arm, make sure it's doing what you expect. If not, think about what happened and why before moving again.
  - Talk out loud about what you’re doing or planning to do while you’re working.
    - e.g., “I’m pushing the translational hand controller back; I expect the arm to move towards the base. The elbow is going to get closer to the table, but it won’t violate the clearance limit.”
  - If you’re planning to do doesn’t mean sense, you may be able to realize it in advance through hearing yourself say it out loud.

- Arm Alignment:
  - Use the edges of fixed objects such as the targets or the trusses to determine when the arm is vertical or horizontal.
  - Look at all of your views when you are trying to align the arm with a target. This is in most situations, you CANNOT get all of the information you need from a single view.

- Distance Estimation:
  - Remember that the last component of the end-effector is 1 m long. You can use it as a gauge to ensure that you don’t violate the clearance limit and to position the end-effector correctly.

- Maneuvering:
  - If you end up with the arm locked up in a narrow or stopped due to collisions, try to move in the opposite direction to escape the narrowness. If you cannot escape it, reset the arm to its original position by pressing ‘r’; be aware that resets will count against your overall score.
A.2 Task Training for Hardware Development Study

(Chapter 2)
**Track and Capture**

- You have 30 seconds to grapple an HTV module in free drift.
- Time begins as soon as you scan.

- Remove the brake, and approach the HTV.
- Maintain alignment over grapple fixture during approach.
- Trigger grapple as soon as in grapple envelope.
- If you are fast enough, there will be no singularities or joint hard stops.

**Alignment Techniques**

- Correcting Alignment Errors:

  - Technique 1: 
    - Error 1: Left fly-to misalignment.
    - Error 2: Right fly-to misalignment.
    - Technique 3: 
      - Error 1: Left grapple misalignment.
      - Error 2: Right grapple misalignment.

- To avoid a collision with the HTV or target pin due to misalignment, make sure you are aligned with all six arms close to the target.

**Advanced Task**

*Fly-to, grapple, and reposition*

**Advanced Task**

During fly-to, position the end-effector 2 meters above the target box.

Remember: The length of the end-effector is 1m. You can use it as a guide.

**Advanced Task**

- Once above the target, you need to align the end-effector into the pre-grapple position. This is often easiest in the internal control frame.

- When the end-effector is 1.5 meters above the target box, switch to Vision mode for final approach.

**Advanced Task**

- Begin final approach.
- Once in the grapple envelope, engage grapples.
Advanced Task

You will approach and grapple the target. After grapple, you will maneuver the box to a final position as indicated by the task procedure. (Procedure provided for reference during trial.)
A.3 Task Training for Gesture Study (Chapter 3)

**Task Training**
- Path Following
- Track and Capture
- Advanced Task

**Path Following**
- You will pilot the arm through a series of 9 target rings.
- Try to fly through as close to centre as possible.
- A red mark indicates the top of each ring.
- The orange camera box should be aligned with the red ring section.

**Path Following**
- You will pilot the arm through a series of 9 target rings.
- Try to fly through as close to centre as possible.
- A red mark indicates the top of each ring.
- The orange camera box should be aligned with the red ring section.
- 3 targets appear at a time.
  - When one reached, a new target appears.

**Path Following**
As you approach each ring, you may choose to change the center camera (Monitor 2) to end-effector view and use internal control frame.

To change center camera:
- F2 → F6
To change control frame:
- T = internal
- E = external

**Track and Capture**
**Track and Capture**

- You have 90 seconds to grapple an HTV module. Time begins as soon as the task begins.
- Remove the cover and approach the HTV.
- Maintain alignment over grapple fixture during approach.
- Trigger grapple as soon as in grapple envelope.
- If you are too close, there will be no guarantees or just start over.

**Alignment Techniques**

- Correcting Alignment Errors:
  - Technique 1: Turn on Translational
  - Technique 2: Turn on rotational

**Advanced Task**

Fly-to, grapple, and reposition.

**Fly-to and Grapple**

You will fly the arm from its starting position to a target box equipped with a grapple fixture.

**Fly-to and Grapple**

During fly-to, position the end-effector 2 meters above the target box.

Remember: The length of the end-effector is 1m. You can use it as a guide.

**Fly-to and Grapple**

- Once above the target, you need to adjust the end-effector into pre-grapple position. This is often easiest in the internal control frame.
- When the end-effector is 1.8 meters above the target, push switch to remote mode for final approach.
- Begin final approach.
A.4 Task Training for Degrees of Freedom Study
(Chapter 4)
Robotics Terminology

- The arm's structure allows the end effector to move 4 ways in 3-D space:
  - Upright
  - Forward backward
  - Right
  - Left

- The end effector is missing

Virtual Environment

The following slides will introduce the virtual environment used in the experiment.

International Space Station

- ISS (International Space Station)
  - Components
    - Core Modules
    - Truss & Solar Arrays
    - Robotic Arm

Joint Limits

- It's very important that you pay attention to joint limits during the activity.
- You will see physical limits, which are called hard stops.
- Hard stops are the limit on how far a joint can travel.
- This stops the robot if the arm encounters a hard stop.
- If any joint moves past its limit, you will encounter a hard stop.
Singularities

The arm also has software limitations. It cannot guide movements in particular positions, known as singularities.

Clearance Violations & Collisions

- Overall clearance limitation: 2 ft (0.6 m) between the arm and all obstacles
  - A warning will be shown on the screen when any part of the arm comes within this area.

Basic Controls

- To remove the Brake: Press the 'b' Key
- To move: Two hand controllers—translation and rotation

Hand Controllers

The left hand controls translation (forward/backward, wrist, and shoulder), and the right hand controls rotation (positive/negative, pitch, yaw, and roll) of the end-effector.

Hand Controllers (1 of 2)

- Translational Hand Controller

When your left hand controls the translational hand controller, which moves the tip of the end-effector. The arm software calculates how all of the joints must move to move the movement.

Hand Controllers (1 of 2)

It is important to make smooth movements with the hand controllers. Quick motions could damage the robotic arm.

Please ask the researcher to load the simulator to complete this section
Try It Out

- Issue the control signal
  - Use a needle pick with a finger to keep the effector pointing in the desired direction.

Try It Out

- Move the control unit
  - Push the effector up/down to move the end-effector up/down.

Hand Controllers (2 of 2)

- Rotational Hand Controller
  - With your right hand, you control the rotational hand controller, which rotates the arm around the axis of the end-effector. It does not move the end-effector.

Try It Out

- Move the control stick
  - Controls the effector's movement in different directions.

Flight Rules and Tips

The following slides contain rules you must follow and suggested strategies for training.
**Things to Be Aware Of**

- **Camera**
  - Use all of your viewpoints, especially the rear view mirror, to monitor the task area.

- **End-effector**
  - Be aware of location and proximity of all end-effectors and components.

- **Use the end-effector tool kit.

**Strategies**

- **Correct Assistance**
  - Monitor both the position and orientation of the end-effector.

- **Keep checking all the end-effectors**
  - Monitor the environment.

- **Making the Arm**
  - Plan ahead, before you start working.

- **Think about the movements required for the task**
  - When you begin moving the arm, make sure you’re doing the right things.

- **Keep thinking**
  - Think about what’s going on.

- **Rehabilitation**
  - Think about what you’re doing or planning to do while you’re working.

- **Communicate**
  - Make sure the assistant is aware of what you’re doing.

**Responding to Messages**

- **Messages**
  - Periodically, the word “message” will appear in a flashing sign for several seconds on the center screen.

- **Click the thumb button on the Right Joystick to “answer”**
  - The box will disappear when you respond.
Grapple

The following slides introduce grapple procedures and strategies.

The Grapple Envelope

- To grapple, pull the trigger on the right joystick out.
- You are in the grapple envelope, close the following conditions are met:
  - The grapple is on the right side.
  - The grapple is within 100 meters of the target.
  - The target is within a 3-meter radius of the grapple.
You should engage grapple as soon as these conditions are met. Do not grapple otherwise.

Alignment Techniques

1. **Correcting Alignment Error**
   - Technique 1: Adjust the grapple's position to align with the target.
   - Technique 2: Use the grapple's sensors to adjust the grapple's position.

2. **Tracking Targets**
   - Technique 1: Use the grapple's sensors to track the target's movement.
   - Technique 2: Use the grapple's control system to adjust the grapple's position.

Your Task: Track and Capture

- You have 90 seconds to grapple an HAV module in the UT.
- Time limits: You must complete the task within 90 seconds.
- Remove the target, and approach the HAV.
- Use the grapple's sensors to locate the HAV in the UT.
- Trigger the grapple as soon as possible.
- You are not allowed to move the grapple once it is engaged.

Your Task: Track and Capture

- You will use multiple trials.
- When you complete a trial, you must move to the next, on your own.
- You do not need to wait for the researcher.
- If you reach 90 seconds, you must enter the correct box.
- If you reach the box, you must enter the correct box.
- You must re-engage the grapple if it has been disengaged.

Requirements:
- Time limits: 90 seconds per trial.
- Must remove at least 70% of the target.
- You must return the grapple to its original position.
Test Conditions

The following explains the different controller conditions you will test today.

Controller Settings

- Today we are testing four possible settings for the hand controllers:
  - FULL MULTI-AXIS: can operate both controllers in multiple directions at the same time.
  - TRANSLATION LIMITED: can operate THC in one direction at a time.
  - ROTATION LIMITED: can operate THC in one direction at a time. THC has multi-directional operation.
  - NON-BIAXIAL: can operate one controller (either THC or THM) at a time. Each allows multi-directional operation.

Controller Settings

- You will test these conditions by completing a series of Track and Capture trials, and answering a survey for each.
- Your priorities are:
  1. Grapple Completion and Speed
  2. Answering messages
- You will be scored on your success in the task and message responses.

Practice

- To begin each test, complete four practice Track and Capture trials. The first two targets translate only. After the first two, HTV targets will rotate.
- Remember: watch side screens to determine HTV motion.
- When complete, you will be given a series of ten practice trials. To continue you must complete at least three. If you do not complete three, you may be considered unqualified to complete the study.

Trials

- Complete thirteen Track and Capture Trials.

Repeat Practice and Trials procedure for all controller conditions.
Appendix B

Appendix B: Pre-Experiment

Participant Background

Questionnaire
Subject #: 
Age: ___

Please circle one for each of the following

Gender: Male Female

Dominant Hand: Right Left

How often did you play video games in the last year?
Never Once every few months Every few weeks Every week Every day

What type do you play most often?
First-person Third-person Over-head/Strategy Side-view fighter Puzzle

How often did you play video games growing up?
Never Once every few months Every few weeks Every week Every day

What type did you play most often?
First-person Third-person Over-head/Strategy Side-view fighter Puzzle

Rate your experience operating RC aircraft or similar devices:
(No Experience) 0 1 2 3 4 5 (Expert)

Rate your experience flying aircraft:
(No Experience) 0 1 2 3 4 5 (Expert)

Are you working on or do you hold a pilot’s license?
Yes No

Have you ever used the SSRMS Simulator?
Yes No

Rate your experience using the SSRMS joystick controllers or similar joystick devices:
(No Experience) 0 1 2 3 4 5 (Expert/Fully Trained)

Approximately how much sleep did you get last night (hours)? _________
Appendix C

Appendix C: Additional Gesture Study Materials

C.1 Joystick Input Handout

**Joystick and Keyboard Control Scheme**

<table>
<thead>
<tr>
<th>Rest</th>
<th>Do Nothing</th>
<th>Brake</th>
<th>“B” Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grapple</td>
<td>Trigger (Right Joystick)</td>
<td>Answer Message</td>
<td>Foot Pedal</td>
</tr>
<tr>
<td>Forward</td>
<td>Left Joystick</td>
<td>Roll Right</td>
<td></td>
</tr>
<tr>
<td>Backward</td>
<td></td>
<td>Roll Left</td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td></td>
<td>Pitch Up</td>
<td></td>
</tr>
<tr>
<td>Down</td>
<td></td>
<td>Pitch Down</td>
<td></td>
</tr>
<tr>
<td>Right</td>
<td></td>
<td>Yaw Right</td>
<td></td>
</tr>
<tr>
<td>Left</td>
<td></td>
<td>Yaw Left</td>
<td></td>
</tr>
</tbody>
</table>

[Diagram of Joystick]
C.2 Single Input Gesture Handout

<table>
<thead>
<tr>
<th>Rest</th>
<th>Brake</th>
<th>Grapple</th>
<th>Mode Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward</td>
<td>Roll Right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backward</td>
<td>Roll Left</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td>Pitch Up</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Down</td>
<td>Pitch Down</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right</td>
<td>Yaw Right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left</td>
<td>Yaw Left</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## C.3 Endpoint Control Gesture Handout

Endpoint Control Mapping Scheme, Right Handed (8 hand gestures, 6 arm gestures, + rest)

<table>
<thead>
<tr>
<th></th>
<th>Gesture 1</th>
<th>Gesture 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rest</td>
<td>![Rest Image]</td>
<td>![Rest Image]</td>
</tr>
<tr>
<td>Brake</td>
<td>![Brake Image]</td>
<td><strong>Grapple</strong></td>
</tr>
<tr>
<td>Forward</td>
<td>![Forward Image]</td>
<td><strong>Roll Right</strong></td>
</tr>
<tr>
<td>Backward</td>
<td>![Backward Image]</td>
<td><strong>Roll Left</strong></td>
</tr>
<tr>
<td>Up</td>
<td>![Up Image]</td>
<td><strong>Pitch Up</strong></td>
</tr>
<tr>
<td>Down</td>
<td>![Down Image]</td>
<td><strong>Pitch Down</strong></td>
</tr>
<tr>
<td>Right</td>
<td>![Right Image]</td>
<td><strong>Yaw Right</strong></td>
</tr>
<tr>
<td>Left</td>
<td>![Left Image]</td>
<td><strong>Yaw Left</strong></td>
</tr>
</tbody>
</table>
### C.4 Coupled Input Gesture Handout

#### Coupled Input Mapping Scheme, Right Handed (8 hand gestures, 6 arm gestures, + rest)

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Hand Gesture</th>
<th>Arm Gesture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brake</td>
<td></td>
<td>Grapple</td>
</tr>
<tr>
<td>Forward</td>
<td></td>
<td>Roll Right</td>
</tr>
<tr>
<td>Backward</td>
<td></td>
<td>Roll Left</td>
</tr>
<tr>
<td>Up</td>
<td></td>
<td>Pitch Up</td>
</tr>
<tr>
<td>Down</td>
<td></td>
<td>Pitch Down</td>
</tr>
<tr>
<td>Right</td>
<td></td>
<td>Yaw Right</td>
</tr>
<tr>
<td>Left</td>
<td></td>
<td>Yaw Left</td>
</tr>
</tbody>
</table>
C.5 Handout Illustrating Translation Command “Box” for Endpoint Control and Coupled Input

C.6 Training Checklist

The following checklist was used during the “free period” of training prior to any specific tasks, in which participants would attempt various SSRMS motions in the simulator environment without a set target.

<table>
<thead>
<tr>
<th>Input Type: __________________________</th>
<th>Test Session Number: ______</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time: _______ am/pm</td>
<td>Date: _________________</td>
</tr>
</tbody>
</table>

Training Checklist

<table>
<thead>
<tr>
<th>Up</th>
<th>Pitch Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down</td>
<td>Pitch Down</td>
</tr>
<tr>
<td>Right</td>
<td>Yaw Right</td>
</tr>
<tr>
<td>Left</td>
<td>Yaw Left</td>
</tr>
<tr>
<td>Forward</td>
<td>Roll Right</td>
</tr>
<tr>
<td>Back</td>
<td>Roll Left</td>
</tr>
<tr>
<td>Internal Frame</td>
<td>External Frame</td>
</tr>
</tbody>
</table>

143
C.7 Verbal Task Instructions

C.7.1 Path Following Training

You will now complete a series of path following training trials. Training will be un-timed. There will be two trials: the first will not have messages, the second will. Remember to try to fly through the center of the rings with the red bar positioned at the top. Your priorities are completing the trial and answering messages.

C.7.2 Track and Capture Training

You will now complete a series of track and capture trials. The first three trials will not have messages. The remaining trials will. You will have ninety seconds to capture the HTV target. Time begins as soon as the experiment loads. Your priorities are completing the trial and answering messages.

C.7.3 Fly-to and Grapple Training

You will now complete two fly-to and grapple trials. For training, these will not be timed. The first will not have messages, the second will. When you have positioned the end-effector two meters from the target and facing the grapple pin, you will be allowed to change the center camera view to zero, change the command frame to internal, and engage Vernier as instructed. You will then approach and grapple the target. Your priorities are trial completion and answering messages.

C.7.4 Path Following Evaluation

You will now complete one Path Following trial. You will have ten minutes to fly through all nine rings. Your priorities are trial completion and answering messages.
C.7.5 Track and Capture Evaluation

You will now complete five track and capture trials. You will have ninety seconds to grapple the target in free-drift for each trial. Your priorities are trial completion and answering messages.

C.7.6 Fly-to and Grapple Evaluation

You will now complete one fly-to and grapple trial. You will have ten minutes to complete the target. You may not switch the command frame, central camera, and engage Vernier until you are two meters from the target and facing the grapple pin. Your priorities are trial completion and answering messages.
Appendix D

Appendix D: Task Selection Effects in DOF Study

During the DOF study (Chapter 4), experimental trials were designed with varying difficulty by adjusting the target drift directions. While all trials required the participant to move forward in order to capture the target, each target was moving (or "drifting") relative to the participant’s end-effector starting position. Target drift included movement in at least one translational direction and one rotational direction, but in no more than two of each (Table 4.1). Target motion for each trial was classified into four drift types. Three trials had one translation and one rotation (1T-1R), four trials had one translation and two rotations (1T-2R), four trials had two translations and one rotation (2T-1R), and two trials had two translations and two rotations (2T-2R). Additional assessments of trial drift type effects on performance and technique measures were conducted as part of the study. Ultimately, these analyses supported the findings included in Chapter 4. A discussion of select results is included here. All statistical analysis was conducted using Kruskal-Wallis non-parametric tests of metrics across drift type.
D.1 Effect of Drift Type: Results

Trial drift type did not have a significant effect on grapple success rate for the FM, NB, or RL conditions. There was a significant effect of drift type on TL trial success rate \( (p \ll 0.01) \), such that trials with two translational drift directions had fewer successful captures (Figure D-1).

![Figure D-1: Average grapple success rate across drift type.](image)

A significant effect is also seen across drift type for multi-rotation time in the FM, NB, and TL conditions \( (p < 0.001) \) (Figure D-2). For FM, targets for 2T-2R had the highest absolute multi-rotation time \( (p \ll 0.001) \) and multi-rotation for those for 1T-2R was significantly higher than those for 1T-1R \( (p \ll 0.001) \). In the NB condition, multi-rotation was highest for 1T-1R, and lowest for 2T-2R \( (p \ll 0.001) \) with no significant difference between the 1T-2R and 2T-1R drift types. Trials with two translational drift directions had increased multi-rotation over the simplest case (1T-1R) for the TL condition \( (p \ll 0.001) \).

Only the TL condition showed any significant difference in trial time across drift type \( (p \ll 0.001) \). Trials with two translational drift directions had increased total
D.2 Effect of Drift Type: Discussion

Use of multi-rotation during FM trials was low relative to the total rotational motion used, and total rotational use was similar between FM and RL conditions. This indicates that use of multiple rotations at a time was not critical for task success, and was used minimally when available. When examining trials by drift type, multi-rotation use was increased for trials drifting in multiple rotation directions in the FM condition. That is, participants used more multi-rotation during full 6 DOF control when target motion appeared to require it. However, there was no significant difference in success rate between the FM and RL trials across drift type, regardless of target rotational drift. Thus, for the track and capture task with the drift rates selected, multi-rotation was not found to be necessary for task success.

Multi-rotation time for the NB condition decreases as task difficulty increases (Figure D-2). Reduced multi-rotation, and reduced rotational input in general, as the number of drift directions increases for NB is consistent with the idea that increased
Relative increases in multi-rotation time between TL trials were related to higher total trial time (Figures D-2 and D-3). By requiring participants to pause forward motion to make other translations, the target has more time to drift during non-forward translation, requiring more alignments to be made, leading to further pausing of forward motion to make adjustments, allowing more drift, and so on in a diminishing feedback loop. This loop ultimately leads to TL trial times being larger over the other test conditions. Additionally, if a target is drifting in multiple translation directions, more non-forward translations are required for alignment. This leads to higher trial time for these cases, and correspondingly more target drift in all drift directions. Thus, participants must ultimately rotate more when a TL trial takes more time, which is reflected by increased multi-rotation.

It should be noted that study participants were largely novice users. With ad-
ditional experience, allowing more refined rotational input, time efficiency may be improved through multi-rotation. Whether such an improvement would be significant or not would require further investigation. Additionally, target rotational drift was set to a fixed spin-rate of 0.45 deg/s, while the joysticks allowed rotational inputs up to approximately 4.58 deg/s. Slow spin-rate provides the user with time to adjust rotations individually. If the vehicle spin-rate were set to a higher speed, increased multi-rotation may have been necessitated. However, setting the spin-rate above or close to maximum rotational input speed would render the target impossible to capture. In general, high spin represents a limitation for manual operation of such a tracking task. There is an ultimate limit to the speed at which a system such as the Canadarm2 may be controlled manually. If the spin rate of the target exceeds the limits of feasibility in simulation or a real scenario, manual capture would no longer be an option, representing a task outside the focus of this study.