Human-Machine Collaborative Telerobotics: Computer Assistance for Manually Controlled Telesurgery and Teleoperation

H. Hawkeye King

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2014

Reading Committee:

Blake Hannaford, Chair

Howard J. Chizeck

Thomas S. Lendvay

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

Human-Machine Collaborative Telerobotics: Computer Assistance for Manually Controlled Telesurgery and Teleoperation

H. Hawkeye King

Chair of the Supervisory Committee:
Professor Blake Hannaford
Electrical Engineering

Teleoperated robots are controlled by human operators and allow us to act in environments that are inaccessible for reasons of safety, scale, or remoteness. Unfortunately, due to lack of tactile sense and dexterity and proprioception, task performance can suffer compared to direct object manipulation with the hands. Still, use of human controlled robots is the only viable option for these dangerous or hard-to-reach applications, since robots do not yet have the capability to autonomously perform most unstructured tasks. This thesis presents new contributions in telerobotics that show the way to improved access and control. The target application domain is telerobotic surgery.

A novel surgical robotic system is presented, the Raven II, which forms the basis of a collaborative telerobotic research network with open open source software and tool interface. The robot is now in use at more than a dozen institutions and is a shared platform for developing and sharing new hardware, software, and methods.

Collaborating on telerobotics research requires agreement on a teleoperation protocol. To this end a new data interface was developed and tested for compatibility among fourteen master-slave teleoperation robots. Results of this study suggest avenues for a common internet standard for teleoperation robots.

To study performance with this system, beyond purely manual control, two types of computer assistance are studied: virtual fixtures and mixed autonomy. Virtual fixtures are spatially varying environment features that provide motion control assistance. Mixed autonomy combines autonomous

robotic motion with human control or commands.

In this work, a new mixed autonomy method is presented based on a division of the workspace into regions that are either safe for computer assisted control, or require pure human control. The new method was evaluated experimentally and compared with novel virtual fixture implementations and pure manual control. Experimental evaluation used uni- and bi-manual peg-in-hole object grasping and manipulation tasks, and several metrics were reported. The aim of these experiments was to determine when and how computer assistance functions will help improve human control of robots. Results showed that performance on some sub-components of object manipulation was improved by the addition of computer assistance. However a performance detriment was measured, due to a mismatch of state information when control is switched from pure human to mixed human-computer operation.

Results of this research demonstrate ways to improve many high-value tasks being carried out today under telerobotic operation; successful application of this research may literally save lives in the operating room, disaster relief situations, nuclear environments and other critical applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

BI-LATERAL TELEOPERATION: Information is sent bi-directionally between the master and slave systems. Motion commands are sent to the slave robot to accomplish a task, while force-feedback information is relayed back to the slave giving the human operator sensory information about the task.

MIXED-AUTONOMY: A method for controlling a telerobot whereby human and computer directives are combined to determine the robot's behavior.

TELEMANIPULATOR: A robotic device remotely controlled by a human operator to perform object/environment manipulation tasks.

TELEOPERATION MASTER: The user-interface used to control a slave robot. Together the two form a master-slave teleoperation system.

TELEOPERATION SLAVE: A robot controlled remotely operated by a human operator or human-computer cooperative controller. Together the two form a master-slave teleoperation system.

TELEROBOT: A remotely operated robot. Comprises teleoperation master and teleoperation slave.

TELESURGERY: Surgery that is performed with robotic instruments under the control of a human operator. The human operator can be in the same room (local telesurgery) or somewhere far away from the patient (remote telesurgery). Synonyms: "telerobotic surgery", "teleoperative surgery", "robotic telesurgery".

VIRTUAL FIXTURE: a spatially defined constraint or guide overlaid on a multidimensional robotic workspace to assist performance of tasks in that environment either through force feedback to a human operator or the imposition of constraints on slave motion.

Chapter 1

INTRODUCTION

Human controlled robots enable us to perform many tasks that are infeasible by hand or entail significant risk to perform directly. Examples of these "teleoperated robots' or "telerobots" include machines for space or undersea operation, micro-manipulation, or handling radioactive materials. Teleoperated robots are an enabling technology in the nuclear industry, since without them humans could not handle the highly radioactive materials. Undersea exploration and operations are similarly dependent on telerobotics, as remotely operated vehicles perform much of the deep sea work.

However, there are major drawbacks when completing tasks via teleoperation compared to direct human operation. All else being equal, performance via telerobot is rarely as accurate, fast, smooth, or error free as the with the hand. To improve this detriment, research is ongoing to make human interfaces, robotic sensors, and end effectors that help a human operator perform closer to his/her best. The present work explores the use of augmented reality and mixed human-computer control of robot manipulators to compensate for lost capabilities and improve performance.

Industrial robotics gives us many examples of robots doing tasks faster, stronger and more accurately than a human operator. But on most tasks, humans are obviously better due to their capacity for scene analysis, path planning, object recognition and manipulation, and all the other innate capabilities we possess. This thesis presumes that a marriage of the two will yield a system that harnesses the strength of both systems.

A major and beneficial application of telerobotics is in robot-assisted surgery. Modern robot assisted surgery systems require human control of a robotic manipulator. The small diameter robot arms are inserted into a patient through tiny incisions to operate inside the human body. They are able to perform precise tasks with minimal scarring and improved comfort for the surgeon. These "telerobotic surgery systems" or "telesurgery" systems are transforming the way we think about surgery, and opening up new possibilities for computer assistance in the operating room.

This thesis describes new work in telerobotics and telesurgery. First, a new research platform is

presented for innovation in telerobotic surgery. Called the “Raven”, this platform is the basis of a global collaboration on surgical robotic innovation. Next, an emerging standard for interconnecting master-slave telerobots is detailed, and experimental evaluation of this communication standard is presented. New methods are explored for adding computer assistance to human operator control, including comparison of new and established methods. Finally methods and apparatus are evaluated experimentally. Results of the experiments report how and when computer assistance methods help or hinder performance, and they offer new insights into human-machine cooperative robotic control.

Through telerobotics, robots are having a real impact on our lives; enabling humans to go places and do things we would not be able to without them. The present research aims to improve our ability to carry out these valuable tasks, today, and have an immediate positive impact on our lives. Ultimately this line of research could give people super-human accuracy and speed; and greatly reduced rates of error and potential harm.

1.1 Telerobotics

A teleoperation robot can be defined as a system of robot arms and end-effectors that are controlled by a human operator. The robot manipulator that interacts with the target environment is often referred to as the “slave” while the human-interface device for taking human commands is called the “master”.

Teleoperation technology originated in the 1940’s for remote handling of radioactive material. As radioactive materials grew more enriched, safety required people to stay farther away. First, long grasping tools were developed to handle the materials. These were later replaced by complex mechanical linkages that evolved into electro-mechanical systems and then human controlled robots. Today the nuclear industry remains a major producer and consumer of teleoperation robots.

Later, a great deal of research in telerobotics was conducted in the interest of space exploration and undersea applications. Today, telerobots have been further applied to disaster relief efforts, explosive ordinance disposal, and with great success robotic minimally invasive surgery.

This type of remotely operated system allows us to be present and perform tasks in otherwise inaccessible environments. For example, micromanipulators inserted through the abdominal wall allow a surgeon to operate internal to the human body without large incisions. Undersea robots

Figure 1.1: Ray Goertz, inventor of telerobotics, safely handles radioactive material using an early mechanical teleoperation system (1951).

allow humans to do tasks on the ocean floor miles below the surface. In addition, size and motion scaling allows us to operate in very small areas, such as in surgery, or large areas such as in mining operations.

Virtually all teleoperation systems use visual feedback via cameras and monitors to show the operator what the robot is doing. Many teleoperation robots are "unilateral", meaning movement or force commands are sent one-way from the master to the slave. In addition, some robots that send tactile, force, or motion data back from the slave to the master are called "bilateral" and allow the human operator additional sensations of touching the environment.

1.1.1 Haptics and Telerobotics

Bilateral teleoperation systems offer the advantage that the operator can feel what he/she is doing in the environment through force, tactile or other touch information. This touch feedback is called "haptic feedback", and can improve task performance (e.g., [40, 90, 112]).

Haptic feedback is most often cartesian (x-y-z) force applied by the master manipulator to the human hand. Force information can be sensed with an end-effector or environment mounted force/-torque sensor; estimated from robot's controller output; or derived from model based interaction with the environment, where force is predicted from an online simulation of robot-environment

contact.

The force feedback bilateral teleoperation system can be implemented using a number of different architectures (see, e.g., [41,46,62]). Choice of architecture depends on a variety of factors such as the available sensor information, delay characteristics of the communication channel, and system dynamics. In addition, "model based" haptic interaction derives feedback information from an environment model, and sidesteps some problems of communication delay or poor sensor information.

In addition to force-feedback teleoperation, other haptics information can assist the user's perception of the remote environment. High-frequency vibrotactile feedback can improve a user's sense of stiff environmental contact (e.g., hard surface contact). Vibration information can come from accelerometer sensing or be generated artificially using pre-recorded or model based methods. [59, 82]. Additional research has shown the positive influence of skin stretch actuation on haptic perception [116].

Furthermore, artificial haptic effects called "virtual fixtures" can be added over the real environment to help the user navigate and perform tasks. This is a major subject of this dissertation, and related work is discussed in detail in Chapter 4.

1.1.2 Telerobotic Surgery

Robotics is revolutionizing the way we think of surgery and interventional medicine. Surgical robotics, coupled with advances in medical imaging, are giving us new therapeutic capabilities with increased access to the body, decreased invasiveness and scarring, enhanced precision and reduced trauma. Most notably, the daVinci Surgical System by Intuitive Surgical Inc, is a teleoperated robot for minimally invasive surgery [37]. It is the only FDA approved teleoperated surgical manipulation robot in widespread use today.

In this paradigm, a human surgeon controls minute (5 or 10 mm) instruments that are inserted into a patient's abdomen. The abdomen is inflated with carbon dioxide, offering enough space for the surgeon to see and operate on the anatomy. Therefore the surgeon is "virtually present" in this environment- using 3D video to see and a human-interface device to remotely control the robot.

This thesis explores the advantages of using additional computer assistance to help the surgeon better perform the technical aspects of the surgery, i.e., precise motion, faster motion, fewer acci-

dental movements, etc.

1.2 Limitations of Teleoperation Robotics

Teleoperated robots are valuable tools that offer substantial benefits in many circumstances. However, when remotely operating a robot, a human operator is at a disadvantage compared to directly acting with his/her hand [40]. The most notable drawbacks are the loss of touch sensation and dexterity.

The human hand is an amazing 22 degree of freedom (DOF) device capable of a wide variety of dexterous grasps. In contrast, robot hands (end effectors) used in telerobotics have far fewer DOF, and often the grasper itself only has one degree of freedom: open/closed. Furthermore, the touch apparatus of the hand is far more sophisticated than any robotic sensors available. Sensory organs of the hand measure pressure, vibration, pain, touch, and temperature and altogether give a very richly detailed perception when handling objects. All of this is lost when operating via teleoperation robot.

Proprioceptive and stereo visual sensing are impaired when using a teleoperation system. Significant engineering may go into making a teleoperation system very natural to use with an accurate mapping between the operator's bot, but it still doesn't match innate human sensing.

Altogether, these disadvantages lead to relatively poor performance of telerobotic tasks compared to the same tasks done with the human hand. Therefore, ways are sought to compensate for these disadvantages.

1.2.1 Limitations of Internet Based Telerobotics

As teleoperated robots proliferate, it becomes more desirable to easily connect and operate them over existing communication channels like the Internet. A major barrier to this is a lack of standards in teleoperation communication. Even more fundamentally, there is no consensus for what capabilities are necessary for interconnecting teleoperation systems. As a general practice in teleoperator design, master and slave systems are developed together, and no efforts have been made to bridge the gap between independently designed systems. This has led to redundant engineering efforts and "siloed" research agendas.

In the course of this dissertation work, fundamental design for a new standard was developed

to allow heterogeneous teleoperation master and slave robots to interconnect. Chapter 5 discusses these new developments in depth. Using common interfaces allows not just interconnectivity among robots, but also the connection of robots to simulations, virtual environments, multiple masters, and other heretofore unrealized configurations. Ultimately this drives innovation and enables the use of the right tool for the job.

1.3 Human Machine Collaborative Systems

Telerobotics is a new paradigm for interacting with our world: one that is mediated by computer in-the-loop. Having an additional agent cooperating with the human offers a new opportunity for that agent to contribute to efficient, accurate, and effective task completion. Through this combination of human and computer action, a new type of system is created for performing environment manipulation: the human machine collaborative system (HMCS) [57].

While this is a very general term for a robot controlled through the combination of human and machine input, this thesis focuses on two architectures for HMCS: haptic virtual fixtures and mixed autonomy.

Virtual fixtures are a type of model-based haptic teleoperation, where force information is calculated from a model of task operation, not a model of the real environment. Therefore, haptic virtual fixtures constitute a tactile augmented reality interaction where the force overlays artificial tactile features over the real environment, e.g., distance from a target, guidance along an optimal trajectory, or a protective boundary around a dangerous area. In this case the computer is assisting the human operator by providing physical cues about the structure of the task and its environment. Virtual fixtures are discussed in detail in Chapter 4.

Through the use of mixed autonomy, some portions of a task are executed with reduced or zero human input. These are the task portions that are easily understood by the machine and safely executed with reduced risk to the operating environment, e.g., patient. Currently there is a wide gap between simple tasks that a machine can do and complex tasks that require the cognitive and dexterous capabilities of a human operator. Mixed autonomy offers a solution whereby advantages of the two might be combined.

1.4 Contributions of This Thesis

This thesis explores virtual fixture and mixed autonomy HMCSs for improving manual teleoperation control of a robot. Research focuses on the application domain of telerobotic surgery, and results are generalizable to all telerobotics uses.

- A new research telesurgery system, the Raven II, was developed in the course of this research. Contributions by the author to the Raven II project are outlined in Chapter 3.

- Virtual fixtures research is thoroughly reviewed and a new framework is synthisized for understanding their attributes and capabilities (Chapter 4).

- A new standard for interconnecting telerobotic systems is developed and presented in Chapter 5. By demonstration, the new standard is proven sufficient for controlling a wide range of heterogeneous teleoperation systems.

- New types of HMCS are described in Chapter 6. A new method for integrating computer controlled motion with human control is described, and new types of virtual fixtures for object grasping and manipulation are presented. These constitute new capabilities for computer assisted object manipulation.

- The assistance functions of Chapter 6 are experimentally studied in Chapter 7. New and existing objective performance metrics for telesurgical robotics are employed to give a detailed analysis of human and human-machine collaborative task completion. Experimental results break down phases of the operation to study how HMCS can help or hinder performance, and to identify the advantages/disadvantages of trading control between the human and the machine.

Altogether, these individual contributions represent a step forward in human-machine collaborative telerobotics. The widespread use of telerobots in many critical, high-value applications means that this work can have a great impact on improving real-world conditions.

Chapter 2

RELATED WORK

2.1 Telerobotics

Through a teleoperation robot, or telerobot, a user can interact with an environment at a distance allowing him or her to reach dangerous, far away, or inaccessible environments [117]. If the operator manipulates the environment and receives force feedback it is called bilateral teleoperation [41, 46]. Under bilateral control, a control loop is closed between the user interface (master) and the robot arm/hand (slave), often over a local or global communication link such as Internet.

Teleoperated robots are in widespread use today in many high-value tasks. For example, in 2011 teleoperated robots from iRobot were sent into the highly radioactive Fukushima Dai-Ichi power plant to assess an ongoing reactor meltdown [122]. The same robots routinely inspect and safely disable improvised explosive devices in combat zones. Undersea exploration and resource extraction are enabled by telepresence robotics [68, 136], while remotely operated robots for telesurgery are revolutionizing the way we think about surgery [105].

Telerobotics is an established commercial technology, yet still an active area of research. One of the most active areas of research is compensating for network time delays to provide a realistic "transparent" sense of remote telepresence (e.g., [30,43,63]). Researchers are also looking for ways to improve users' experience and ability to control the robot by providing rich haptic feedback such as vibrotactile and skin stretch effects [60, 116]. Another research direction is in environment aware controllers for improved task-specific control [95, 135].

This thesis explores the combination of kinesthetic haptic and autonomous assistance for enhancing teleoperation and telesurgery task performance. Haptic assistance is added through a technique called "virtual fixtures" while autonomous robot movement and human control are combined through "mixed autonomy".

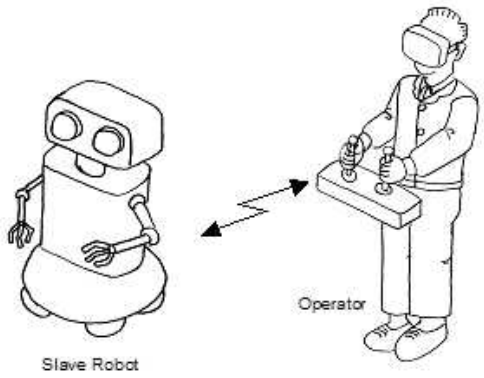


Figure 2.1: Teleoperation robots are controlled at a distance by a human operator. (Image credit: Kazuyo Iwamoto, AIST, Japan.

2.2 Telesurgery

Use of robots for surgery date back to 1985 when Yik San Kwoh et al. used a Unimation Puma 200 for CT guided tumor biopsy [61]. Since then the use of robots in the operating room has grown by leaps and bounds, and is revolutionizing the way we think about surgery.

There are a few notable robots currently in the operating room and commercially available. The Neuromate (Renishaw, Hoffman Estates, IL, USA) is a stereotactic tool holder for brain surgery that helps the surgeon place tools using preoperative imaging for guidance. Robots from Hansen Medical Inc. (Hansen Medical, Mountain View, CA, USA) allow surgeons to perform catheter insertion into the heart from a nearby robot control interface. Surgeons use x-ray fluoroscopy to track the catheter during insertion, and doing the operation remotely reduces surgeons' exposure to harmful ionizing radiation. Mako and Robodoc Surgical Systems are orthopedic surgery systems- a kind of computer assisted CNC milling system that precisely mill bone for hip and knee replacement surgeries [15]. The Cyberknife is a robot that destroys tumors non-invasively by delivering focused beams of high energy X-rays to the tumor [10]. By delivering radiation from many different pre-calculated angles the tumor area gets a strong dose of radiation while surrounding tissue receives only small doses.

The most well known and widely used surgical robot today is the daVinci surgical system (Intuitive Surgical, Sunnyvale, CA, USA). More than 2000 robots have been sold worldwide, in 2012 an estimated 200,000 surgeries were performed using the system, and in total over two million patients have been treated. The daVinci is a master-slave teleoperation system with visual feedback (no force feedback from environment contact) and up to four arms. One arm holds a camera and the surgeon

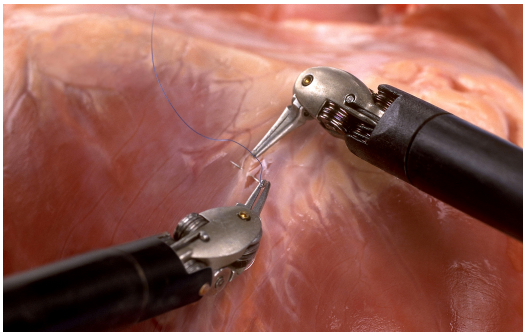


Figure 2.2: End effectors of daVinci surgical system

controls one or two tools at a time. The robot is designed for MIS procedures, and comes with interchangeable tools for cutting, grasping, suturing, and electrocautery. It is the only commercially available FDA approved surgical robot with grasping and manipulation capabilities. However, the robot is limited to manual unilateral teleoperation, i.e., no contact force sensing, and no computer or image guidance.

Research in surgical robotics is a wide and varied field. In addition to telerobotic surgery, some of the major research areas include stereotaxis and image guidance, surgical automation, addition of haptic feedback, natural orifice endoscopic surgery (NOTES), surgical skill evaluation, design of less invasive manipulators, operating room integration and capsule endoscopy (For an overview of the field see: [104]). Several studies have also shown the feasibility and effectiveness of long-distance telesurgery [12, 79]. Research by this author and others has demonstrated new paradigm for delivering surgical care to underserved regions and extreme environments such as battlefields or disaster relief zones [25, 70].

This dissertation extensively discusses the Raven II Surgical System developed at the University of Washington in the BioRobotics Laboratory. It is an open-source MIS surgical robot with grasping, cutting and manipulation capabilities. With open-source software and hardware interfaces, the Raven II is well suited for medical robotics engineering research. However, Raven II is not FDA approved and is not safety-rated for human operation. See Chapter 3 or [42] for details of the Raven II. Chapter 7 presents new experiments using this system.

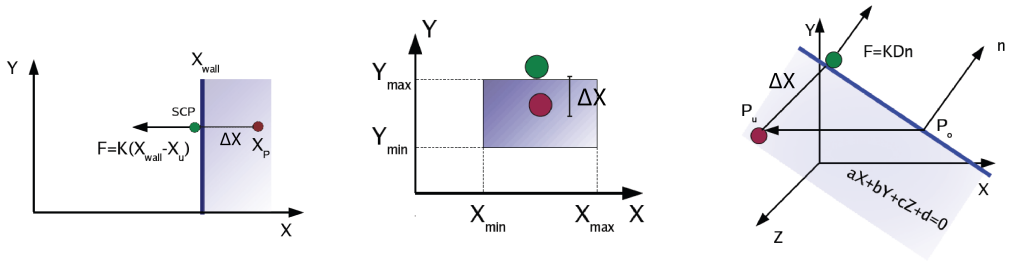


Figure 2.3: Haptic rendering of basic shapes: force calculation for intersection with a wall (left), cube (center), and a plane(right). Red circle: HIP, green circle: IHIP.

2.3 Haptic Rendering

Robotic haptic interfaces allow a user to feel interactions with virtual or remote environments using the tactile senses in addition to visual or other sensory feedback. The addition of haptic force feedback can improve teleoperative control in surgery and other tasks [40, 80, 81, 90]. In addition, with a haptic user interface, an augmented reality system can be created by adding virtual objects and features to a user's workspace.

To provide force feedback to a user's hand and arm, a haptic device will read the position of a backdrivable or impedance controlled robotic end effector held by the user. For virtual environment rendering, the position of the end effector is known as the haptic interface point (HIP). When a collision occurs with a virtual object, the object is rendered as a virtual spring-damper system:

$$\vec{F} = k_1 \Delta \overrightarrow{x} - k_2 \Delta \dot{\vec{x}} \tag{2.1}$$

with force, $\vec{F}$, and the distance and rate of the HIP penetration into the object given by $\Delta\vec{x}$ and $\Delta\dot{\vec{x}}$. The HIP is often displayed visually at the surface of the virtual object (e.g., $\vec{x} - \Delta\vec{x}$) instead of at its location under the surface; displaying what is known as the ideal haptic interface point (IHIP) increases the illusion of realism to the user. This is illustrated in Figure 2.3. For a complete treatment of haptic rendering for virtual environments see [17].

As an aside, when combining haptic rendering with telerobotics, the slave robot can be controlled by either the HIP or the IHIP [86].

This is the most commonly used method for kinesthetic haptic rendering of surfaces and basic geometry, but not the only one. Other approaches have been used, and a full discourse is outside the scope of this paper. For a complete overview of traditional haptic rendering methods see [39, 111]. Recently, haptic rendering of point-cloud data has been studied for generating virtual environments from real, sensed environments [109]. Another interesting example of haptic feedback from virtual environments is [59], where vibrotactile feedback augments force-feedback to improve then realism of hard-surface contact.

2.4 Virtual Fixtures

The term "virtual fixtures" (VFs) was introduced by LB Rosenberg in 1993 and defined as "abstract sensory information overlaid on top of reflected sensory feedback from a remote environment." He made the analogy to using a physical fixture, a ruler, to help draw straight lines. VFs are often implemented as artificial geometric features placed in the telemanipulator workspace that enforce spatial constraints through haptic force feedback [106].

Previously, in 1987, Sato and Hirai suggested using what they called "software jigs", and partial automation to reduce operator burden for complex telemanipulation tasks [115]. In the surgical realm, at Imperial College researchers designed VF constraints using spatially variable motion gains to prevent users from entering "forbidden regions" [49] (see Figure 2.4). This technique was used to constrain operators to a desired region during robotic bone milling under manual control.

VFs for surgery have been studied extensively by researchers at Johns Hopkins University [6,

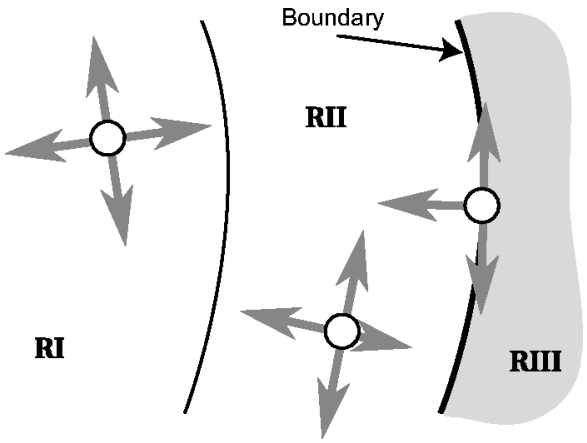


Figure 2.4: Motion scaling virtual fixture. Motion in region RI has uniform motion scaling in all directions, in RII motion towards the boundary is scaled down, and no motion is allowed into region RIII [49].

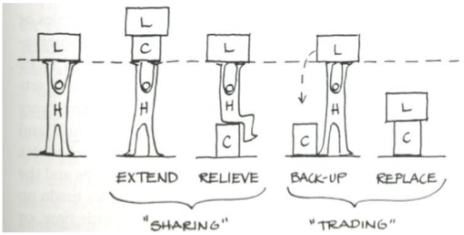


Figure 2.5: Computational agents can improve teleoperation by assisting a human operator in a variety of ways. In this conceptual illustration from [117] we see a human, H, and a computer, C, working together or independently to perform work on load, L.

57, 66]. Initial studies examined geometric VFs. In addition, VFs from CT scans have been made that match anatomical features [65, 130]. Altogether, these studies examined use of VFs for path following, anatomical inspection from different orientations, and bone milling in a safe region. However, they did not study the use of virtual fixtures for grasping and environment manipulation, which are the primary mode of operation for surgeons.

In [103] researchers developed methods for generating virtual fixtures by segmenting preoperative cardiac images. Simulated anatomy was used to evaluate the use of VFs for improving task performance in touching anatomical landmarks, and cutting to a desired depth. This interesting work used simulated tapping and cutting tasks, and did not require object manipulation.

Jesse Dosher [26] demonstrated in one-dimension, and Lee and Hannaford [64] in two-dimensions that haptic feedback in the form of spatially located icons, improved users' performance on target selection tasks. That work used an information theoretic formulation of throughput that is adopted in this thesis. However, that work did not involve complex environment interaction, merely move and touch.

Chapter 4 gives a thorough overview of the state of the art in VFs and gives a taxonomy of types and capabilities.

2.5 Mixed Autonomy

The idea of combined human and autonomous control was widely developed by Thomas Sheridan and William Ferrell [28] who envisioned that space-based robots would be directed by humans at

a high-level, eliminating the need for low-latency feedback to the human. They envisioned several models for mixed autonomy including "shared control", where the human and the computer-in-the-loop would cooperatively control robot hardware simultaneously, and "traded control" where the two would take turns controlling the target hardware.

In 1993, Yokokohji, Ogawa, Hasunuma, and Yoshikawa implemented a mixed autonomy controller for a 1-DoF constant force task [134]. This work demonstrated improvement of constant force tracking when an autonomous controller was combined with a bilateral teleoperation system. One interesting result from their study was exposing a potential discontinuity when changing modes between pure manual operation and combined manual / autonomous operation. Results in Chapter 7 further illustrate the consequence of switching manual and computer assisted modes.

Bumm, et al. [22] developed a system for mixed-autonomy bone drilling in the skull for a sphenoidectomy, using CT scans to plan an approach to the anterior wall of the sinus. A robot made an initial incursion into the bone autonomously, and the surgeon removed remaining bone via telemanipulation. In this case the robot was not used for environment manipulation, since the drilling tool is a single-point end effector. Also, the range of motion required was very limited, since the tool moved within a small continuous area.

Experiments by Griffin, Provancher, and Cutkosky [36] explored shared control to help human operators maintain a minimum grasp force during object telemanipulation. Augmentation of user control was limited to control of the robot grasp force.

Recent work has demonstrated the use of Hidden Markov Models to determine when and which pre-defined, learned autonomous movement should be taken [91]. This exciting research uses offline learning by demonstration from manually segmented data to develop the HMM model and identify autonomous sub-tasks. Meanwhile ongoing research at UC Berkeley is exploring autonomous surgical suturing using learning by demonstration [128] and autonomous surgical debridement [52]. While this hasn't been combined with human telesurgery, that is a clear next step.

2.6 Summary

The synthesis of huzman and machine intelligence for control of remote manipulation robots opens new possibilities for telepresence robotics. Improving human control of robots will provide imme-

diate gains by benefitting important real-world applications of today.

Haptics and mixed autonomy are two technologies that will make this happen, and are explored in depth in the following chapters.

Chapter 3

THE RAVEN SURGICAL SYSTEM®: OPEN PLATFORM FOR SURGICAL ROBOTIC RESEARCH[1]

This chapter presents a new platform for surgical robotics research: the Raven II. This robot provides a testbed whereupon researchers can create new techniques in telerobotic surgery by modifying the hardware and software.

The first generation RAVEN was designed for experiments in long distance, Internet based telepresence surgery [69]. Several studies using RAVEN demonstrated feasibility of Internet based teleoperation to remote and extreme environments [25]. Investigations using RAVEN also measured the impact of common Internet latencies on surgical performance and explored interoperability among a wide range of telesurgery master/slave robots [54, 72].

Raven II is a second-generation system that includes all the same Internet telepresence capabilities as the original, and features many improvements that make it better suited for a wide range of telesurgery research.

With support from the US National Science Foundation's Computing Research Infrastructure program, seven Raven II systems were built, and in February 2012 they were distributed to US based researchers at Harvard University, Johns Hopkins University, University of Nebraska, University of California (UC) Los Angeles, UC Berkeley, UC Santa Cruz, and University of Washington. Having the Raven II hardware creates a new opportunity for groups to share design improvements, replicate results, and collaborate on research. Having a common open-source code base allows new developments to be shared among multiple institutions. This is a significant step towards continued innovation in telerobotic surgery.

Since the initial build of eight robots, additional systems have been delivered to several other institutions.

Design of the Raven II is described in Materials and Methods below. Project completion is de-

[1]Parts of this chapter originally appeared in the Hamlyn Symposium 2012

scribed in the Results section, and the significance of this new platform is treated in the Discussion.

3.1 Design of Raven II[2]

Raven II evolved from the original RAVEN surgical system. Mechanically Raven II differs from the RAVEN in several significant respects. The system inertia, especially due to reduced mass of a linear-actuation guide rail, was significantly reduced from RAVEN to Raven II for improved control performance. Total link mass of RAVEN was 4.6 kg, compared to 2.0 kg for Raven II. In addition, the Raven II mechanism was designed to accommodate either two or four arms. Optimization was performed for mechanism isotropy over the ranges of motion in laparoscopy, as well as maximizing common workspace among the four manipulators. (For complete details see [67].) A new, patented tool design provides six degrees of Cartesian motion and grasping [83]. A unique feature of this tool is a wrist design that eliminates cable coupling between degrees of wrist actuation.

The RAVEN used Maxon DC brushless motors. Brushless motors such as these provided better torque-to-weight ratio than brushed motors. However, they require significant additional wiring and complex expensive motor controllers. Raven II uses Maxon RE40 and RE30 brushed DC motors with a 12:1 and 3.7:1 gear ratio respectively. This has not made noticeable difference in performance, and has reduced cabling and electronics complexity.

Raven II electronics have many of the same features as RAVEN, but in a compact form factor more easily situated in a laboratory environment or carried to the field (Figure 3.1). A single nineteen-inch desktop rack holds the robot power supply, motor controllers and I/O for the two arms, and a Linux PC. As in RAVEN, a key hardware safety feature is a DL05 programmable logic controller that monitors the robot inputs and outputs and has the capability to trigger fail-safe brakes on the first three (gross positioning) mechanism joints. Z6A6 and Z12A8 servo amplifiers (Advanced Motor Control, Camarillo, CA) drive the robot's smaller and larger motors at six and twelve amps peak current output respectively. I/O with the computer is via a custom designed eight channel USB I/O board. This board can read eight encoder signals and write eight analog motor

[2]The author's primary contributions to this project were in software engineering, systems integration, networking, online community management, and debugging of mechanical and electronics problems. Electronics and mechanical engineering, manufacturing and assembly, project management, robot simulation, and other aspects of the projects were carried out by Jacob Rosen, Ji Ma, and Daniel Glozman of UC Santa Cruz; Blake Hannaford, Diana Friedman, Phil Roan, Lei Cheng, Sina NiaKosari, and Lee White of the University of Washington.

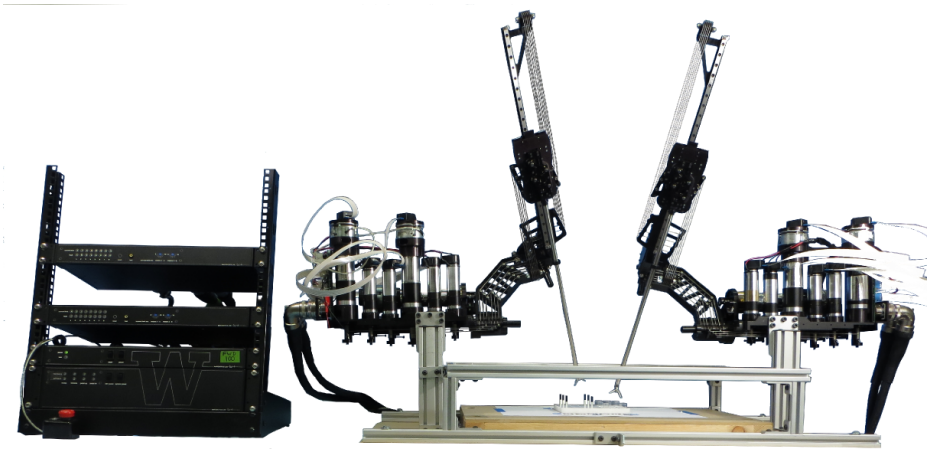


Figure 3.1: Raven-II system including two arms, two instruments, and compact, rack-mount, electronics and control unit.

outputs in less than 125 μs.

Raven II software has been dramatically revised from RAVEN. As with RAVEN, the basic sensor-actuator control loop is closed through the Linux PC at one thousand hertz. To achieve the necessary speed and determinism, RAVEN used an RTAI Linux kernel module. Instead, Raven II uses CONFIG_PREEMPT_RT, a hard real-time patch for the Linux Kernel. The patch satisfies all timing requirements, providing an accurate 1kHz control loop. It also allows real-time software to execute in user space with minimal modification, thus simplifying the software development environment.

In addition, the Raven II software has been integrated with Robot Operating System (ROS) [102]. ROS is a modular, open source robotics middleware package that makes it very easy to combine the Raven II software with other robotics software libraries. For example, Raven II state information is output using ROS message passing mechanisms, and can be plotted in real time using "rosbag" or used by a teleoperation system to calculate force-feedback. Plus, a robot visualization tool (shown in Figure 3.2) was developed using the "rviz" ROS module. The ROS parameter server is used for controller gains configuration of the robot. Despite integration with ROS, raw UDP sockets are still supported for teleoperation control, since laboratory studies showed slightly better

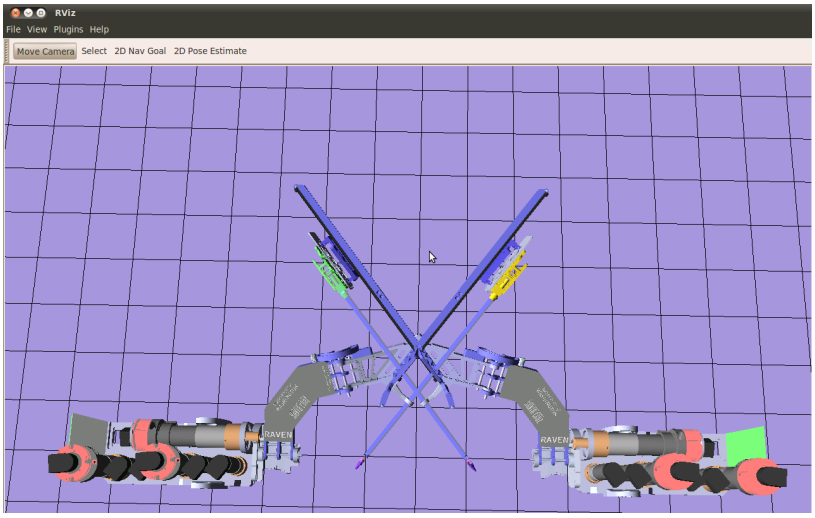


Figure 3.2: Integration with Robot Operating System makes the software compatible with many off-the-shelf robotics packages. This figure shows the use of the "rviz" package for visualization of the robot.

performance this way.

The Raven II control software was rewritten in C++, with the goal of making it easy to incorporate available libraries and for collaborators to implement new controllers and features. At the same time, much of the modularity of the original RAVEN code was maintained. All Raven II partners have access to a common source code repository that will include future contributions from all sites.

Several Internet-based collaboration tools foster communication among research peers and provide peer-to-peer support for the new systems. A wiki has been created to hold documentation and keep it up to date. An online discussion forum, integrated with the wiki is also in place, and has proven crucial to supporting the deployed systems and sharing information. Finally a blog kept participants up-to date as the robots were developed, and is now used for periodic news updates.

3.2 Project Completion

Seven Raven II systems were completed in February 2012 and provided to seven institutions around the United States. One robot suffered shipping damage to two motor encoder shafts which were replaced. In all partner locations, the robots are set up and running and new research is being

devised and implemented.

Community participation via dedicated Internet forums has been active with over one hundred posts in February and March 2012. This hints strongly at further collaboration among groups.

3.3 Discussion and Impact

Raven II represents a state-of-the-art open-source surgical robotics research robot. It is a major improvement over the original RAVEN system and is a robust platform for MIS robotics research. Open-source control software simplifies development of new modules, and ROS integration means it is relatively easy to directly interface with many existing robotics packages. Using ROS bridges the gap between medical specific robotics and general robotics capabilities.

In the future, revisions are planned to fix some minor hardware anomalies. Also, many new avenues of research are being pursued on this platform involving haptic feedback for improved control, machine learning, image guidance and more.

Raven II is now a common surgical robotics research platform in use by seven leading U.S. based research groups with more to follow. This forms the basis of a promising new research network. Community involvement will speed development and eliminate duplication of engineering efforts in platform development.

Chapter 4

TYPES AND CAPABILITIES OF VIRTUAL FIXTURES FOR TELEOPERATION

4.1 Introduction

Robotic telemanipulation entails several disadvantages compared to direct human operation including loss of proprioception, reduced tactile sensing, reduced manipulator degrees of freedom, and worsened visualization. Therefore, there is an imperative to restore the lost capabilities of the human operator and overcome those performance losses.

At the same time the computer in-the-loop is an opportunity to assist the user. Computer assistance may take the form of augmented reality feedback to the user by adding visual, auditory or tactile cues. Alternatively, computer assistance might change the control signals sent from master to slave through tremor reduction, motion scaling or automation of subtasks.

This paper explores a class of assistive features called "virtual fixtures". The term virtual fixture was coined by LB Rosenberg who defined them as "abstract sensory information overlaid on top of reflected sensory feedback from a remote environment" [106]. Convention around virtual fixtures is that they are haptic, i.e. force-feedback features in the workspace such as force guidance towards a goal, or restrictive motion constraints around delicate targets. However, this definition has been extended by many to include additional non-sensory physical constraints and guides. For example, a slave can still be limited to a region of interest through direction-dependent motion scaling, or non-tactile position constraints.

Therefore, for the purposes of this review, in order to cast a wide net in understanding the impact of virtual fixtures, the term is defined here as a spatially defined constraint or guide overlaid on a multidimensional robotic workspace to assist performance of tasks in that environment either through force feedback to a human operator or the imposition of constraints on slave motion or both.

The contribution of the present work is to synthesize a bigger picture of the state-of-the-art and future directions in virtual fixtures. First, a review of the literature forms the basis of a new

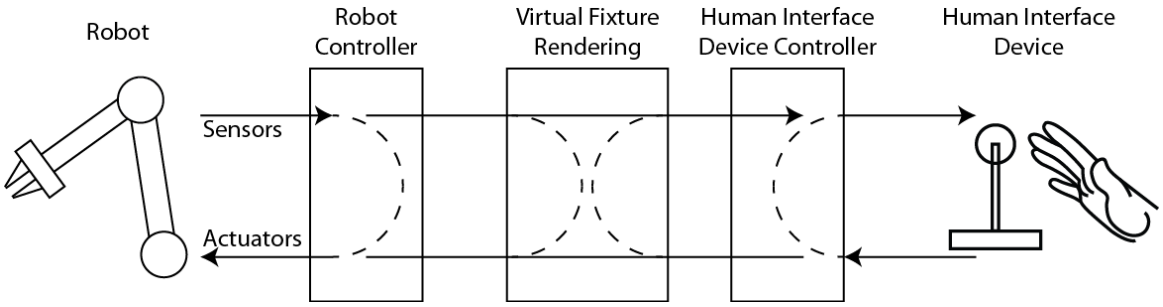


Figure 4.1: Virtual fixtures use the computer in the teleoperation control loop to shape commands from master to slave robot or feedback information to the master.

taxonomy of virtual fixtures, and relevant results from the literature are classified in the new system. Then some new and relevant works are presented, that don't fit the mold of virtual fixtures, but which indicate new directions in virtual fixture assistance for telemanipulation.

Figure 4.1 illustrates the computer-in-the-loop that can modify feedback to the master or commands to the slave. Local control at master and slave sides provide force and position tracking, while the virtual fixture renderer augments commands based on an internal task model (not shown).

4.2 Notation

This paper uses the following notation:

X_m	Position commanded by the master controller.
X_s	Position of the slave.
$\{X_{vf}\}$	Set of points comprising the virtual fixture.
X_{vf}	A point on the virtual fixture boundary optimally chosen for its relationship to X_m, i.e., the closest boundary point.
$P_{\perp}(X)$	Non-preferred motion component. This is the undesired part of commanded motion that a virtual fixture strives to attenuate.
$P_{\parallel}(X)$	Preferred motion component. Represents allowed movement uninhibited or assisted by a virtual fixture.
Z(X)	Impedance or impedance matrix. Admittance $= Z^{-1}$.

K(X) Virtual linear spring constant or matrix.

4.3 Taxonomy of Virtual Fixtures

This section catalogs types and capabilities of virtual fixtures from a review of forty two works in the literature. This taxonomy is meant as a guide to capabilities roboticists can use to create virtual-fixture assisted teleoperation systems.

4.3.1 Guidance vs. forbidden region

Abbot, Marayong, and Okamura identified a distinction between "guidance" and "forbidden region" virtual fixtures [6].

Guidance virtual fixtures (GVF) are task-oriented functions that assist the user by improving accuracy, speed, or trajectory following. Essentially, they guide the robot motion while accomplishing a task. GVFs are used in [4, 20, 21, 51, 58, 76, 78, 85, 88, 92, 94, 97, 101, 132, 137]

Forbidden region virtual fixtures (FRVF) are safety oriented features to prevent the user from entering unsafe areas of the workspace. FRVFs are used in [8, 44, 53, 77, 84, 93, 108, 110, 129, 133]

A combination of the two is used in [6, 14, 32, 33, 87]

4.3.2 Dissipating vs. energy storing

Virtual fixture assistance functions differ in the effect they have on the total system energy. Force feedback assistance functions can be energy storing or energy dissipating. Energy can be virtual potential energy stored in virtual springs or kinetic energy in a moving mass.

Energy storing features are typically implemented as a virtual spring coupling. Impedance type devices, due to their high back driveability and bounded mechanical stiffness, are well suited to energy storing fixture types, where a virtual spring-damper system couples the robotic end-effector to the virtual fixture geometry.

An example of a virtual energy storing virtual fixture is implemented as a spring-damper system:

$$F_m = \bar{K}(X)(X_{vf} - X_m) + Z(X)(\dot{X}_m) \tag{4.1}$$

(4.2)

This is the most common form of virtual fixtures, and some variation of the spring-damper model is used in [8, 47, 58, 84, 85, 88, 92, 94, 101, 103, 106, 108, 110, 129, 133].

Among energy dissipating virtual fixtures, there are two primary methodologies: variable damping and variable motion scaling. Both of these guide the user by canceling motion in a non-preferred direction. Damping dissipates energy through a virtual viscous coupling to mechanical ground, while motion scaling dissipates energy by virtually reducing motion (and thus kinetic energy) transmitted from the master to the slave.

Admittance devices can be used with damping virtual fixtures, since these non-backdriveable master systems can apply high levels of motion damping to the user. Pure damping dissipative virtual fixtures may be implemented as:

$$F_m = \bar{K}_d(x,\dot{x})P_\perp(\dot{x}) + m(x,\dot{x})\ddot{x} \tag{4.3}$$

$$\tau_m = J^T F_m \tag{4.4}$$

Where $\bar{K}_d(x)$ is a position and direction dependent damping matrix and $m(x)$ is a virtual mass; both of these can be increased to the limit of stability as a fixture is encountered. This type of fixture rendering is used in [21, 51, 106].

Motion scaling can be used to dissipate input energy and keep a user inside or outside a target area. Essentially, movements in non-preferred directions are reduced, or

$$\dot{X}_s = K_{d1}P_\parallel(\dot{X}_m) + K_{d2}P_\perp(\dot{X}_m) \tag{4.5}$$

Where K_{d1} and K_{d2} are motion scaling factors. This is not a force-feedback effect, and can therefore be used with non-actuated master devices such as ungrounded motion trackers [4, 20].

A hybrid approach is demonstrated in [87]. An energy storing spring fixture corrects deviation from a desired path while a variable motion scaling feature amplifies motion along the desired path towards the goal.

4.3.3 Hard vs. Soft

A hard constraint cannot be violated, while a soft constraint permits the user to breach the fixture constraints. The hard vs. soft description can be considered a continuum of compliance, with

hard fixtures being at the extreme. [76] examines how varying virtual fixture compliance affects performance of a line-following task. In that experiment, users perform a task that requires violating the virtual fixture constraint and tracking performance is reported as compliance is varied across the continuum.

Hard fixtures can be implemented two ways: on the master side using a device that has low compliance over the relevant range of forces, e.g., a non-backdriveable admittance device; or on the slave side using a proxy.

The former method is equivalent to the following admittance control law:

$$\dot{X}_m = \begin{cases} 0 & , X_m \in \{X_{vf}\}, F_m \, towards \, VF \\ \frac{1}{Z} F_m & , X_m \notin \{X_{vf}\}, \, or \, F_h \, away \, from \, VF \end{cases} \tag{4.6}$$

Where the force output by that master device prevents the human operator from directing the slave motion into the virtual fixture.

The second method for hard virtual fixtures is for impedance type devices that cannot mechanically limit a user's input motion. In this case a virtual position, called a proxy, is used.

$$X_s = \begin{cases} X_m & \text{if } X_m \in \{X_{vf}\} \\ X_{vf} & \text{if } X_d \notin \{X_{vf}\} \end{cases} \tag{4.7}$$

Where the slave position X_s is constrained to the boundary of the virtual fixture X_{vf} when the commanded position X_m is inside the virtual fixture region $\{X_{vf}\}$. This method ensures that the slave robot does not enter a forbidden region virtual fixture, or leave the region allowed by a guidance fixture. Proxy methods that constrain a virtual position to the surface of a virtual object are common in haptic rendering [107, 138]. They were applied to virtual fixtures in [6, 100, 109], including methods for adding simulated physical dynamics to proxy motion.

Work by [110] demonstrates a type of hard virtual fixture with a highly compliant impedance type master device.

Many more examples of hard virtual fixtures can be found in [44,51,65,77,92,93,108,110,131, 132].

In contrast to hard virtual fixtures, soft virtual fixtures can be overcome by sufficient effort on the part of the user. Without a proxy based method, this is the only option for master-side virtual fixtures with a highly-compliant master interface, since a large force applied by the user will overcome the virtual fixture force. Typically a teleoperation slave device will follow a position commanded by the master, so, unless a virtual fixture method applies a hard constraint on the user's motion or strict software limits on commands sent to the robot, a virtual fixture will be soft.

There are many options for rendering soft virtual fixtures. For example, virtual walls can be rendered as spring-damper systems, non-isotropic motion scaling gains can be used, attractive or repulsive force fields can be used with sinusoid or saw tooth or other force profiles (e.g., [27]). The similarity among these is that the user is able to overcome the virtual fixture and move the robot contrary to the preferred directions.

Examples of soft virtual fixtures are found in [4, 14, 20, 23, 32, 53, 58, 76, 84, 88, 94, 97, 101, 137]

4.3.4 Environment modeling and real-time sensing

A virtual fixture assisted system can have real-time sensors, e.g., cameras, depth sensors, ultrasound, etc, to gather information about the environment and alter the virtual fixtures. Others rely completely on user input, such as pre-programmed virtual environments, mouse-clicks to define regions of interest, etc.

Use of sensors for generating or updating virtual fixtures (either off-line or in real-time) is demonstrated in [20, 21, 32, 65, 84, 85, 92, 93, 108, 110, 129]

Furthermore, virtual fixtures require a virtual environment; a spatial model to calculate force feedback or assistance effects. The virtual environment can be generated from off-line or real-time sensor data, a-priori knowledge task geometry, or user input. Many virtual fixture implementations use artificial user generated geometry, while others use accurate environment models from CAD renderings or sensing.

Increased accuracy and completeness of environment modeling can bring advantages in creating task-relevant fixture geometries. For example, [132] relies on user-interaction with a CAD model to generate geometric constraints; while [11, 14] generate potential fields around environment features, and use those to define force feedback across the entire workspace. All three use accurate

environment models.

[110, 133] use point-cloud data from systems such as the Microsoft Kinect to make real-time representations of the environment and generate or register virtual fixtures, while [65, 85] use models generated from pre-operative medical imaging modalities that give a complete picture of the environment. [85, 129] use real-time magnetic resonance imaging.

Several groups have studied combining computer vision with virtual fixtures, e.g. [20, 21, 92]

4.3.5 Combined bi-lateral control and Virtual Fixtures

Theoretical work in [7, 19] combine force sensing with virtual fixtures. Finally, work in [7] examines the combination of environment force, i.e., bilateral teleoperation, with virtual fixture force feedback. Studies have yet to clearly illustrate how this combination will effect real-world task performance.

4.3.6 User Adaptive

Virtual fixtures that alter their assistance based on characteristics of the human operator can be called "user adaptive" and represent an important area of development. Several studies have explored online adaptation of virtual fixture compliance based on human performance [4, 88, 94]. Constraints in [53] deform in response to continuous forces applied by the user.

Work in [44, 77] demonstrates the variation of a protective boundary in response to a user's hand movements. The goal is to decrease user penetration into a forbidden-region feature by pro-actively negating velocity towards the feature.

4.4 New concepts in virtual fixtures

4.4.1 Realtime Virtual Fixtures From Human Intention

Probably the most important upcoming work related to virtual fixtures is human intention recognition, exemplified by [5, 120]. Both of these use hidden Markov modeling to estimate the human's desired action. This points a way towards more sophisticated task-action prediction.

Fascinating work in [84] uses eye-gaze tracking to identify regions of interest and provide haptic feedback to help a user achieve that pose. This is an intriguing use of an additional human input

modality.

4.4.2 Virtual Fixtures vs. Automation

A nagging question in virtual fixture implementations is "why not automate?" If a task can be modeled precisely enough to create a helpful virtual fixture, why simply command robot action towards a goal? The simple answer is that we don't have the manipulation capabilities necessary to perform the tasks.

One way to bridge the gap towards automation is through "velocity virtual fixtures". Conventional virtual fixtures are spatial position constraints pushing the user in a desired direction. Velocity virtual fixtures would constrain a user's motion to move along a velocity profile towards a goal.

Alternatively, autonomy virtual fixtures could be defined. Similar to haptic virtual fixtures, an autonomy virtual fixture moves the robot to a desired location or out of a dangerous area without human intervention. For example, in a path-following task demonstrated in [76] the robot would autonomously follow the specified trajectory whenever the robot crossed or came close to the path.

A big question for mixed autonomous action is how to merge human commands with autonomous ones?

4.4.3 Sensory Substitution

To date, virtual fixtures have mostly used force feedback, but other haptic information like vibrotactile stimuli may be beneficial. Work in [16] demonstrates vibrotactile feedback for teleoperation that could be used for virtual fixture feedback.

4.5 Conclusion and future directions

Virtual fixtures have been studied in a variety of applications and a great number of approaches have been tried. This paper has attempted to boil down the literature to reveal some of the common attributes of virtual fixture implementations. Hopefully, in the future, roboticists can use this work to see how virtual fixtures can be used in their own development.

Successful real-world virtual fixture implementations will require sensing and planning of virtual fixture assistance geometries. Modeling the environment and task requirements will use sensors

like the Kinect, or medical imaging. Planning will probably involve an intuitive interface for the human operator to cooperatively design the virtual environment.

In the present day, robot operation in unstructured environments requires human intervention and control. As robotics capabilities continue to evolve, virtual fixtures are one way new capabilities will be merged with human control to increase the usefulness of robots acting in the real-world.

Chapter 5

GLOBAL INTEROPERABILITY IN TELEROBOTICS AND TELEMEDICINE[1]

Abstract

Despite the great diversity of teleoperator designs and applications, their underlying control systems have many similarities. These similarities can be exploited to enable interoperability between heterogeneous systems. We have developed a network data specification, the Interoperable Telerobotics Protocol, that can be used for Internet based control of a wide range of teleoperators.

In this work we test interoperable telerobotics on the global Internet, focusing on the telesurgery application domain. Fourteen globally dispersed telerobotic master and slave systems were connected in thirty trials in one twenty four hour period. Users performed common manipulation tasks to demonstrate effective master-slave operation. With twenty eight (93%) successful, unique connections the results show a high potential for standardizing telerobotic operation. Furthermore, new paradigms for telesurgical operation and training are presented, including a networked surgery trainer and upper-limb exoskeleton control of micro-manipulators.

5.1 Introduction

Many telemanipulation systems have a great deal in common. The robots usually include one or more master manipulators, and one or more remote manipulators. The manipulators have up to six degrees of freedom in Cartesian position and orientation while kinematics functions translate between Cartesian information and the robots' joint control workspace. Furthermore, the communication architecture for Internet based teleoperation is often an Internet protocol network socket with a well defined data interface to transmit control data in a fixed binary packet.

---

[1]This chapter was published as "Plugfest 2009: Global Interoperability in Telerobotics and Telemedicine" by King et al. in ICRA 2010 [54]

Despite these commonalities there has been no development of a framework for interconnecting such systems, and network interfaces are currently described arbitrarily by individual research groups. In the same way that Internet standards have connected heterogeneous computing systems, we predict robot communication standards will speed research, development and deployment of interoperable teleoperated robots.

Internet-based robotic teleoperation has been explored in many experiments using co-developed master and slave systems (e.g. [35,38]) and network characterizations have shown that under normal circumstances Internet can be used for teleoperation with force feedback [113]. Fiorini and Oboe, for example carried out early studies in force-reflecting teleoperation over the Internet [89]. Furthermore, long distance telesurgery has been demonstrated in successful operation using dedicated point-to-point network links [13,79]. Meanwhile, work by the Tachi Lab at the University of Tokyo sought to create a very general protocol for telerobotics by describing the unique capabilities of each robot [123].

The authors of this paper, representing nine independent telerobotics laboratories, have identified a need for interoperability between heterogeneous, telerobotic master-slave systems. The current work is a new direction, exploiting similarities among teleoperation systems to develop standards, simplify software architecture and improve interoperability.

This paper describes a simple, yet effective solution to the problem using a shared data interface, the Interoperable Teleoperation Protocol (ITP). ITP is very simple for a communication protocol. The simplicity is by design, and we demonstrate that this simplicity allows the easy interconnection of quite different systems. The ITP is essentially a data interface, and is intended as a starting point for development of a robust standard for Internet robotic teleoperation.

Inherent in the interoperability problem is the requirement to support a wide variety of telerobots that differ in kinematics and dynamics, scale, and application. To demonstrate the usefulness of a developed data standard it is necessary to show that it can control a heterogeneous group of robots. Our experiment, titled ”Plugfest 2009”, tested the ITP on fourteen unique master and slave robots, most of them designed for telesurgery, at nine locations around the world. The hypothesis examined was that such kinematically diverse systems could be made to work together by using a common data interface and Cartesian-space teleoperation commands.

“Plugfest” is an industry term for an event where vendors come together and test their products

for compliance with a new standard. We adopted this word and methodology for testing the ITP, creating a one-day event wherein many systems were interconnected and tested for functionality. Performing the testing in a single 24 hour period accomplished two goals: first, it demonstrated that extensive tweaking is not required to switch connections between systems; and second, it showed that engineers could integrate the protocol within a consistently short time from project start in early summer to the end of July. Our primary experimental result is the number of unique, successful master-slave connections, which demonstrates the value of the ITP for interoperable teleoperation. The experiment focuses on the telesurgery application domain by incorporating mostly surgical telerobots, but includes general purpose teleoperators as well.

In this experiment existing, published teleoperation robots were retrofitted to use the ITP. It is important to realize that these are not fourteen systems designed together for compatibility, but rather that a novel communication scheme can tie these disparate systems together with a small integration effort at each site.

It should be further noted that several communications architectures and open-source software packages are available for general robotic operation over the Internet (e.g. [1, 2, 31, 127]). However, these are often designed for mobile robots, and are unsuitable for the high packet rates and extreme delay sensitivity particular to telemanipulators and telesurgery systems. One interesting software application framework is the Surgical Assistant Workstation [50]. The present work differs in that ITP is not a software application or library, but rather a data interface specification that is capable of controlling a wide range of teleoperators and can be implemented on any computing platform. Also, OpenIGTLink is a proposed protocol for medical imaging and medical device description and control [126]. This is interesting research into the medical systems integration problem, but has not yet explored the interoperability question for medical or general teleoperation systems.

5.2 Interoperable Telerobotics

To demonstrate interoperable telerobotics over the global Internet, fourteen unique telerobotic master and slave systems were connected by nine groups in five countries. In the context of surgical telerobotics, the master manipulator is operated by the surgeon in order to control the motions of the slave, or patient-side robot. All the connected systems used the same network facing data interface,

the Interoperable Teleoperation Protocol or ITP.

5.2.1 Interoperable Teleoperation Protocol

ITP is a stateless data description representing commands between the master and slave robot. An ideal protocol is robust enough to work with any new teleoperator independently of its design, and flexible enough to accommodate any new data transforms or teleoperation architectures. Therefore, a mechanism is built into ITP for designating new, numbered data specifications, extending the protocol to new innovations. To make cooperation as simple as possible, the first data interface for the ITP is a low-overhead binary UDP packet structure, and its C implementation is shown in Listing 5.1. This is the data packet used in experiments described in this paper. The "pactyp" (packet type) and "version" fields indicate the packet type in use. For these experiments packet type number one was used, and the ITP version was 0.43 (or 43 as an integer). The small size, 84 bytes, makes this data type suitable for high packet rates.

The ITP requires a common reference frame. From the users' perspective the right-handed ITP reference frame has: the positive Y-axis pointing right, the positive X-axis pointing away, and the positive Z-axis pointing down. Each master and slave implement transformations to convert between their own intrinsic coordinate systems and the common reference frame.

There are many ways to encode movement commands from master to slave. For networked teleoperation it is important that the system be robust to arbitrary network delays and packet loss. In particular, the resulting robot command should be free from significant discontinuities, and should act "safely" when recovering from a period of delay. In the current experiment, motion commands for two arms are encoded as position increments in three Cartesian dimensions (delx, dely, delz in listing 5.1) and orientation increments in roll, pitch and yaw (delroll, delpitch, delyaw in listing 5.1). Position is in units of integer microns and orientations are in integer micro-radian units. The choice of integers sidesteps the complexity of floating point specifications. Increments of position and orientation are calculated as follows:

$$\Delta X_k = X_k - X_{k-1} \tag{5.1}$$

$$\Delta R_k = R_k - R_{k-1}$$

$$X = [x, y, z]^T \tag{5.2}$$

$$R = [roll, pitch, yaw]^T$$

In this representation there is no explicit notion of a time-step, making the scheme more robust to varying network delay and packet rate compared to velocity coding schemes. In response to packet loss the result is position drift. In many situations this is advantageous compared with absolute motion commands which would entail step discontinuities in commanded position. With small packet loss the drift is minor and easily accommodated by the user through motion input or indexing.

```
#pragma pack
#define SURGEON_DISENGAGED 0
#define SURGEON_ENGAGED 1
struct M2S_data {
        unsigned int sequence;
        unsigned int pactyp;
        unsigned int version;
        int delx[2];
        int dely[2];
        int delz[2];
        int delyaw[2];
        int delpitch[2];
        int delroll[2];
        int buttonstate[2];
        int grasp[2];
        int surgeon_mode;
        int checksum;
};
```

Listing 5.1: C implementation of the binary packet structure sent from master to slave robot.

Indexing is a common feature among teleoperation systems and allows the user to reposition the master without moving the slave. The incremental motion scheme simplifies indexing, since

absolute position agreement is not required between master and slave. Two states, “engaged” and “disengaged”, are defined to coordinate indexing and are indicated in the “surgeon_mode” field of Listing 5.1. While in the disengaged state, the slave robot should ignore any motion commands until the engaged state is requested.

Endianness and data types were chosen to conform to 32-bit x86 architecture.

Scaling of the user’s motion is another common feature among teleoperators. The scale factor is not explicitly encoded by the ITP. Instead motion scaling is controlled by the master side and is implicit in the transmitted motion commands.

The lightweight UDP datagram protocol is used, since it does not require the additional overhead of the more common TCP protocol. UDP does not guarantee data integrity, but does provide the lowest possible latency. A sequence number is important for tracking out-of-sequence or lost packets, and corrupt packets are detected by the checksum and discarded. Most importantly, lost incremental motion packets will not lead to unpredictable or unsafe motion of the slave manipulator. Out of sequence packets are ignored.

The ITP does not specify a packet rate. These experiments used packet rates from 10Hz up to 1kHz. 1kHz packet rates are exceptionally high compared to average Internet usage. However, there were no problems reported in our experiments due to the high rate, a result that agrees with [113].

5.2.2 Connected Slave Systems

The six slave systems used in the experiment are described here and shown in Figure 5.1. All systems have two manipulator arms and were operated via UDP/IP using ITP. Each system is previously published and complete technical details of the robots and simulator are found in the references.

The Raven surgical system developed at the University of Washington, BioRobotics Laboratory, Seattle (UW, Seattle, WA, USA) is a six degree of freedom (DoF) capable, cable-driven research robot for remotely operated minimally invasive surgery (MIS). It has a unique spherical mechanism for MIS operation around the trocar point [69].

Located at the Tokyo Institute of Technology, Suzukakedai Campus (TokyoTech, Yokohama, Japan) the IBIS IV system is a pneumatically actuated MIS research robot with 6 DoF plus a gripper [125]. The IBIS IV detects environmental contact force without the use of a force sensor by

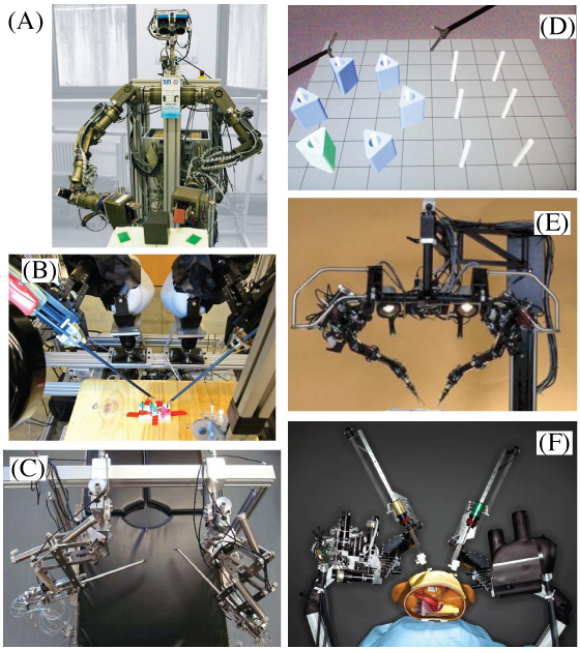


Figure 5.1: (A) TUM general purpose Telerobot. (B) Patient-side robot of the JHU custom version of the da Vinci. (C) TokyoTech IBIS IV surgical robot. (D) RPI VBLaST^{TM}. (E) SRI M7 surgical robot. (F) UW Raven surgical robot.

inference from the pneumatic pressure.

Johns Hopkins University (JHU, Baltimore, MD, USA) connected a commercial MIS robot, the JHU Custom daVinci. The robot hardware is from Intuitive Surgical Inc., Sunnyvale, CA, USA, with custom control software and electronics by Johns Hopkins. The daVinci represents the state of the art in commercial surgical robotics [75].

The M7 surgical research robot was connected at SRI International (SRI, Menlo Park, CA, USA). The M7 is a 6 DoF robot designed for open telesurgery with battlefield and trauma applications [56].

The VBLaST^{TM}laparoscopic training simulator was connected at the Rensselaer Polytechnic Institute (RPI, Troy, NY,USA). It is a virtual reality based bi-manual trainer with haptics that is designed to replicate the FLS tasks. Two virtual MIS tools follow the motion of a physical tool connected to the interface, or may be teleoperated via the built-in network support.

Technische Universität München (TUM, Munich, Germany) connected a general purpose, redundant 7 DoF telemanipulator with a relatively large grasper. The two anthropomorphic arms were designed for general, large-scale manipulation tasks. [119]

This diverse group of slave systems demonstrates the wide variety of robotic capabilities that would be available to surgeons with the adoption of standardized, networked telesurgery capabilities.

5.2.3 Connected Master Systems

Eight master systems, shown in figure 5.2, were used in the experiments. The master systems transmitted ITP data at up to 1kHz via UDP/IP. Although many of the systems are haptic devices capable of force-feedback, that function was not used in these experients. Again, each system is previously published and complete technical details are available in the references.

Three of the master systems at UW, RPI and Imperial College London (ICL, London, UK) comprised two 6DoF Phantom Omni (SensAble Inc, Woburn, MA, USA) haptic devices and surgical console software from UW [114].

Korea University of Technology and Education (KUT, Cheonan, South Korea) used the Phantom Premium (SensAble Inc.) with their own ITP compatible software.

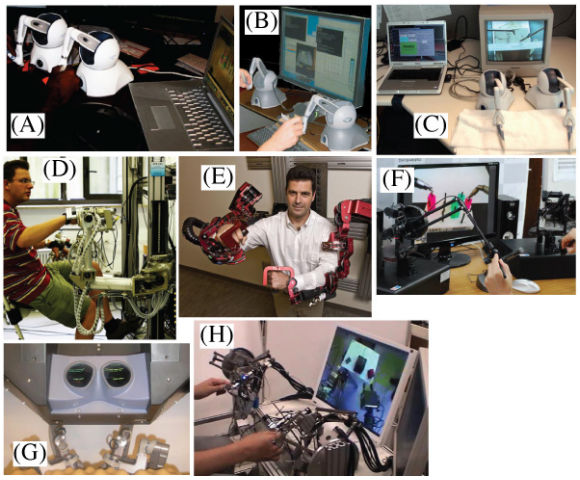


Figure 5.2: (A, B, C) Phantom Omni control station with free software at RPI, ICL and UW respectively. (D) TUM ViSHaRD7. (E) UCSC Exoskeleton. (F) Phantom Premium with custom software at KUT. (G) Master console of the JHU custom version of the daVinci. (H) TokyoTech delta master.

A commercial daVinci surgeon console (Intuitive Surgical, Inc.) with custom controller and software was used at JHU [75].

At the University of California, Santa Cruz (UCSC, Santa Cruz, CA, USA) an upper-limb powered exoskeleton was used, allowing whole-arm motions to scale down to surgery scale tasks [98]. One-arm operation was performed with this system, so bi-manual tasks were simplified to one arm.

At TUM, ViSHaRD7, a custom designed, general-purpose master with 7 DoF was incorporated [96]. The arms are mounted on a mobile platform to form a mobile haptic interface, but in this experiment the mobile base was not activated.

Finally TokyoTech developed a master system with haptic feedback based on a delta motion platform specifically for control of MIS assistant robots [124].

5.3 Experimental Methods

5.3.1 Telerobotic FLS

Wherever possible, participants in this experiment performed the Telerobotic Fundamentals of Laparoscopic Surgery (TFLS) block transfer task, modified for time constraints. Exceptions were made for some robots as described below. TFLS was adapted by Lum [71] from a proprietary scoring method invented by the Society of American Gastrointestinal and Endoscopic Surgeons (SAGES) to evaluate surgeons' laparoscopic proficiency [99]. The test can be seen in Figure 5.3.

The TFLS Block Transfer task is essentially a pick-and-place task. Six plastic blocks approximately one cm^3 are first arranged on the left half of an array of pegs. Subjects grasp each numbered block in order with the left hand, lift it from the peg, transfer the block to the right hand tool, and place the block on a numbered peg on the right side of the board. In this experiment, participants transferred as many blocks as possible in ten minutes.

Due to particular robot configurations, task modifications were necessary in a few cases. The TUM Telerobot slave was designed for manipulating larger objects, so a substitute, bi-manual, pick-and-place task was performed. Users transferred a 4x4x5 cm cube 20 cm from the right to left side of an 8x8x8 cm obstacle, switching hands in the process.

The UCSC Exoskeleton was operated with one-arm only, so the FLS block transfer was simplified; it was performed without the hand-to-hand transfer.

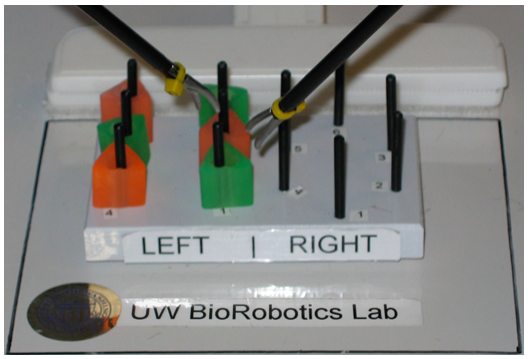


Figure 5.3: The Telerobotic FLS block transfer task.

5.3.2 Plugfest Experiment

Nine groups from North America, Asia and Europe participated in this experiment testing interoperability among their systems. In one 24-hour period on July 30, 2009 participating groups connected over the Internet to control the other robots recording success or failure of the connection and performance on a simple task. Each master-slave connection was restricted to a one-hour slot to connect and complete the experimental task.

With eight masters and six slaves, forty-eight connections were possible. However, labs did not connect their own master-slave combinations, and, to keep the operating schedule within reasonable hours, only global regions separated by less than nine hours were connected. With these constraints thirty two connections were possible. Thirty were attempted, and two were not attempted due to time limitations.

Feedback to the user was visual only and did not include haptics. The focus of this research is on teleoperation rather than video technologies, so Skype^{TM}video was selected due to its ubiquity, ease of use and cross-platform compatibility. The type and location of the camera, PC specification, and video display configuration were not controlled in the experiment.

In each connection the block transfer task was performed one to three times and the number of transferred blocks was recorded, or transfer time when appropriate. Network time delay was measured and recorded in each connection using round trip ICMP Echo (ping) times wherever possible.

5.4 Results

Thirty trials were conducted in which the various master and slave systems connected across the Internet. Of those, 28 resulted in successful task completion, 20 bi manual TFLS, five one-handed TFLS block transfer, and three bi-manual gross manipulation tasks.

Figure 5.4 shows the number of blocks transferred and ping times for each master-slave connection. The average number of blocks transferred in the 20 successful bi-manual TFLS trials was 7.4. The average number of blocks transferred one-handed was 11.6. The average time for block transfer in the gross manipulation trials was 8.2 minutes. The most blocks transferred in ten minutes with bi-manual operation was 16. 30 blocks were transferred using the upper-limb exoskeleton to control the MIS robot at Ttech, IBIS, in the one-handed mode of operation. The highest average

Master Systems \ Slave Systems	UW	JHU	RPI	SRI	TOK	TUM**
UW			(*)12	(34) 14	(133) 15	
ICL		(112) 11	(*)6		(288) 5	(183) 7
JHU	(73) 9		(*)7	X		X
KUT	(180) 6		(*)6	(175) 4	(224) 6	(305) 13
RPI	(*) 8	(*) 13		(*) 2		
TOK	(135) 16			(*) 12		(302) 4.5
TUM		(115) 1	(*)4		(295) 2	
UCSC†	(21) 13	(83) 4	(*)5	(22) 9	(155) 30	

Figure 5.4: Number of blocks transferred in each master-slave trial, and ping response times in parenthesis. *Ping times not taken. **Block transfer task completion time. †One-handed operation only.

blocks transferred per connection for any slave robot was on the UW slave system with an average of 9.5 blocks per 10 minute connection.

There were two unsuccessful trials. In one case a slow packet rate of 10Hz was incompatible with a slave-side controller that assumed 1kHz updates. The resulting behavior triggered a slave-side safety system which harmlessly shut down the robot. For unknown reasons, the behavior was asymmetrical; the right arm moved fine, while the left one caused the fault. In the second unsuccessful case, the orientation mapping between master and slave was too confusing, and the user did not complete the task.

5.5 Discussion

Results show that ITP, providing Cartesian-space motion control commands, is effective for operating a wide range of teleoperation systems. The outstanding result of the experiment is that twenty-eight out of thirty unique attempted connections were successful (93% success rate). In this case, successful operation is defined as an operation wherein a task could be completed, and results given in the table of Figure 5.4 demonstrates that all but two configurations were capable of performing the task.

Besides differences in users' experience level and familiarity with the robots, a major source of variation in task performance was due to the video configuration. Skype™ was easy to use across

platforms and was effective in all thirty connections. However, as mentioned earlier, video parameters including camera type and placement, video codecs, and display configuration were not controlled. Participants also complained about the lack of depth perception demonstrating the need for high-definition and stereo vision. This is an additional avenue of exploration that could benefit telerobotic interoperability, i.e. agreement on video standards, especially high-definition and stereo vision codecs.

Therefore, caution must be taken in considering the numerical results of Figure 5.4. The number of blocks transferred in a given connection was highly influenced by many factors that were not controlled in the experiment, and therefore does not serve as an objective comparison of master-slave performance. What the results do show is whether or not the ITP was an effective data interface and control link between the master and slave robots.

Whats more, Plugfest 2009 demanded that a globally dispersed robot and human operator population to interoperate in a time-critical fashion. In many cases the slave system was "hot-swapped"-operated by several different masters in rapid succession without restarting. Smooth interoperability like this will improve collaborative telerobotic action as, for example, a patient is handed off to a remote surgical specialist for part of a procedure; or multiple users trade control of a telerobot during a long operation.

Furthermore, several interesting telesurgical technologies were explored in this experiment. A virtual reality surgical simulator was controlled in the exact same way as the physical slave systems, demonstrating how surgeons around the world may train on an advanced simulation platform without a specialized simulator on-site. Also, several types of surgical master and slave systems were connected, exploring different modes for human action and control. Use of the UCSC Exoskeleton showed how a surgeon's full range of upper limb motion could be scaled down to control minimally invasive surgical manipulators. Moreover, inexpensive, off-the shelf human interface devices were used with free, open-source software and controlled state-of-the art surgical robots for fine, surgery-like tasks. With a standard protocol, new teleoperation hardware will become easier to integrate into existing applications, and in the future surgeons will be able to select among competing "surgical cockpit" designs.

Twenty-eight out of thirty attempted connections resulted in some degree of task completion. However, even among these there were some difficulties. In many instances, orientation mapping

from master to slave was unnatural or incorrect, in that rotation of the master did not produce the desired rotation of the slave. In at least four cases this was severe enough that orientation degrees of freedom were disabled and robots were operated in three DoF Cartesian space. At the same time, the method of using orientation increments to represent the roll, pitch and yaw leaves open the problem of RPY singularities and the jump discontinuities associated therewith. This will have to be fully addressed in future work, possibly by changing the orientation representation to quaternion increments.

Future work should also address security and session negotiation. Data security is a consideration that could be addressed via encryption. Session managment is another consideration. As an example, in one connection, due to a miscommunication, UCSC and UW sent packets simultaneously to RPI VBLaST resulting in erratic action of the simulated robot. To resolve this, a system for user authentication and exclusion is necessary, possibly by using a stateful session layer protocol that works with the stateless data exchange protocol.

5.6 Conclusions

This work has shown the feasibility of using a simple data interface, the Interoperable Telerobotic Protocol, to control heterogeneous teleoperation systems over the global Internet. In a relatively short span of time in early summer 2009, robots with differing kinematics, hardware and software for medical and non-medical applications were retrofitted to use the ITP, thus becoming seamlessly interoperable. Results of our experiment, titled Plugfest 2009, demonstrated that the uncomplicated ITP protocol is one solution to the interoperability problem. A more complex protocol would be more difficult and take more resources to adapt to the variety of pre-existing computing hardware, programming languages and robot controllers involved in this project.

Although this experiment focused on application to telesurgery, the developed architecture is generalizable to a wide range of telerobotics uses. Future work should focus on incorporating force feedback and more sophistocated bi-lateral control techniques into the ITP, while keeping the data specification simple. The results of this new research direction will be a standardized, robust telerobotics Internet protocol.

The authors predict that increased collaboration will result from this research and expect inter-

esting synergistic results.

5.7 Acknowledgements

The authors acknowledge the following people for their contributions to this work: George Mylonas and David Noonan at Imperial College London, Diana Friedman at the University of Washington, Seattle, and Tomonori Yamamoto of Johns Hopkins University

Chapter 6

COMPUTER ASSISTANCE FUNCTIONS FOR TELEOPERATION

6.1 Introduction

The goal of this research is to make improvements in teleoperation performance, overcome shortcomings of telerobotics technology, and leverage capabilities of the computer-in-the-loop to improve on existing capabilities. Previous chapters described two approaches to this goal, namely virtual fixtures (VF) and mixed autonomy.

This chapter explores the design of both types of assistance functions. Presented here are three assistance types: two types of virtual fixtures, and a novel mixed-autonomy method. A general description of the fixture is given, relevant parameters are discussed, and rendering methods are given. The next chapter, Chapter 7, provides an experimental evaluation of these assistance functions to evaluate when and how each makes a difference.

To reiterate the central premise of human-machine collaborative telerobotics, we assume that the robot can perform some aspect of a task better, i.e., more accurately, with less fatigue, more reliably, etc., while the human operator is essential to the operation by providing improved dexterity and high level cognition.

The assistance functions described here assume that the human must control physical contact and manipulation of the environment while the robot can assist with gross level motions and improved proprioception. Furthermore, to give an intuitive idea of their structure or function, they are given suggestive/metaphorical names.1

6.2 "Golf Course" Forbidden-Region Virtual Fixture

Forbidden region virtual fixtures (FRVF) constrain user motion to safe areas away from obstacles, delicate structures, or occluded areas of the workspace. In safe areas there is no assistive force or action by the FRVF. However, as the user approaches and enters a forbidden region the assistance function applies some force or effect to prevent unintended harmful action. There are a few ways

Table 6.1: List of acronyms

VF	Virtual Fixture
MA	Mixed Autonomy
GVF	Guidance Virtual Fixture
FRVF	Forbidden Region Virtual Fixture
GCVF	Golf Course Virtual Fixture
ARVF	Attractive Region Virtual Fixture
JSMA	Lendvay Jetstream Mixed Autonomy

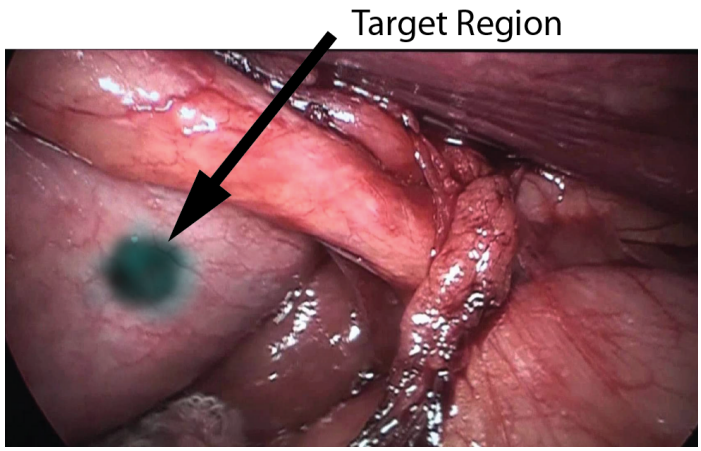


Figure 6.1: Example simulated target site, region of operation is shown.

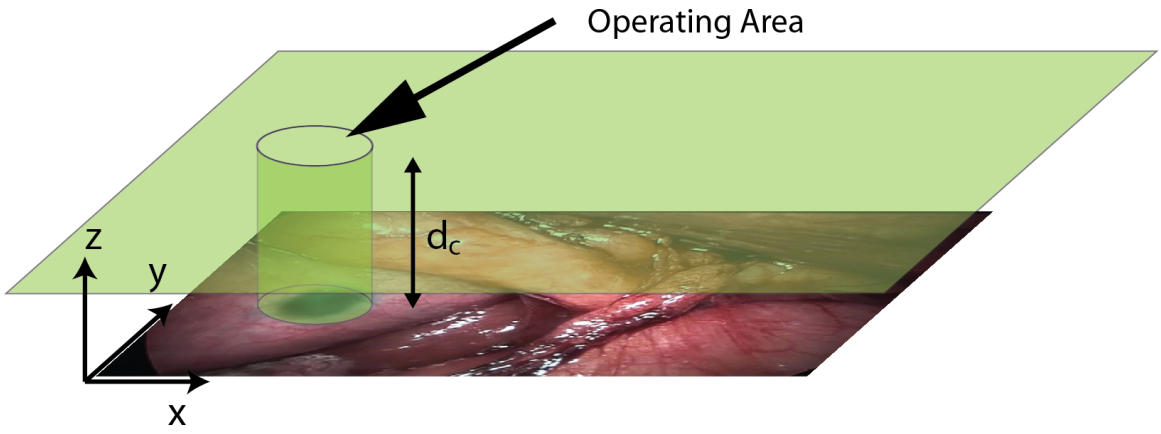


Figure 6.2: Illustration of a Golf Course Virtual Fixture environment with simple planar geometry. Users' motion is constrained to above the plane or inside the cylinder.

these features can be implemented as outlined in Chapter 4.

The Golf Course Virtual Fixture (GCVF) derives an FRVF from the following assumptions:

1. A specific **region of operation** is identified where a task will be performed.

2. The human must control the environmental contact due in the region of operation to the complexity or delicacy required in an operation.

3. There is no need to contact other areas, and contact with other areas of the environment may cause incidental damage.

4. Task performance requires freedom of motion in the region of operation.

5. The human operator can override assistance.

This leads to the following motion constraints and allowances:

Constraints Motions in danger of contact with the environment should be prevented by applying force-feedback away from close contact.

Allowances Contact with the environment *is* permitted in a specified region of operation. Any free space motion not at risk of environmental contact is permitted. Applying sufficient force will overcome force-feedback constraints.

The resulting virtual environment has a protective virtual wall around features in the real environment, with a gap or hole in the wall that allows the user to reach through to the desired task region. During environment contact, motion is constrained to be inside the hole by protective walls around the sides of the hole.

Figure 6.1 shows an example task area. The arrow indicates the region of environmental contact where the surgeon wants to operate. An example GCVF to assist with the safety this operation is shown in Figure 6.2.

A GCVF for this environment is constructed according to the constraint and allowance rules. A barrier (the horizontal plane in Figure 6.2) restricts the robot's cartesian position, X, to distance d_c away from contact. Let d be the distance to contact and X_c is the closest point to X at a distance d_c away from the surface, i.e., outside the barrier, then in the Z direction constraint force F_c is given by the implicit haptic rendering function:

$$F_c = \begin{cases} 0 & ,d \geq d_c \\ k_1(X_c - X) + k_2(\dot{X}) & ,d < d_c \end{cases} \tag{6.1}$$

Where k_1 and k_2 are gains on position and velocity respectively.

To render the allowance region, define the region of operation as a cylinder of radius r centered around line L. If the closest point on the cylinder to the end effector position, X, to L is at point X_a distance d away, then constraint force F_a around the allowance can be calculated by the implicit haptic function:

$$F_a = \begin{cases} 0 & ,d \leq r \\ k_1(X_a - X) + k_2\dot{X} & ,d > r \end{cases} \tag{6.2}$$

And the two can be combined to determine the master-side force, F_m using:

$$F_m = \begin{cases} F_c, |F_c| \leq |F_a| \\ F_a, |F_a| < |F_c| \end{cases} \tag{6.3}$$

Where the minimum magnitude will be zero for motion outside the constraints or inside the allowance.

This is a simple illustrative example of the GCVF. Constraints and allowances can be constructed from any of a variety of haptic rendering methods. Instead of a planar feature, for example, the constraint can be derived from environment sensed point-cloud information. An allowance might come from segmented pre-operative imaging, creating an arbitrary shaped region of operation.

6.3 "Attractive Region" Guidance Virtual Fixture

If forbidden region virtual fixtures are designed to prevent *unintended* action, guidance virtual fixtures (GVF) are designed to assist with *intended* action.

GVF can be designed many ways. For example, direction-dependent motion scaling or restricting movement to an optimal path. This section describes a haptic GVF that uses force feedback to help the user move quickly and accurately to a desired region of operation; termed the "attractive region" (ARVF), since it applies a force that attracts the hand towards a goal.

Design of the ARVF is based on the following premises:

1. A specific region of operation is identified where a task will be performed.

2. Robot confidence is high enough to assist the user in targeting the desired region of operation, but not high enough (or task complexity is too high) that the robot is trusted to autonomously perform the operation.

3. The user's natural proprioceptive and/or visual perception of the robot workspace limit his/her ability to target and stay within the region of operation.

4. A guiding force, pushing the user towards the region of operation, will help the user reach that area and perform tasks there.

5. Control within the region should be handled entirely by the human operator.

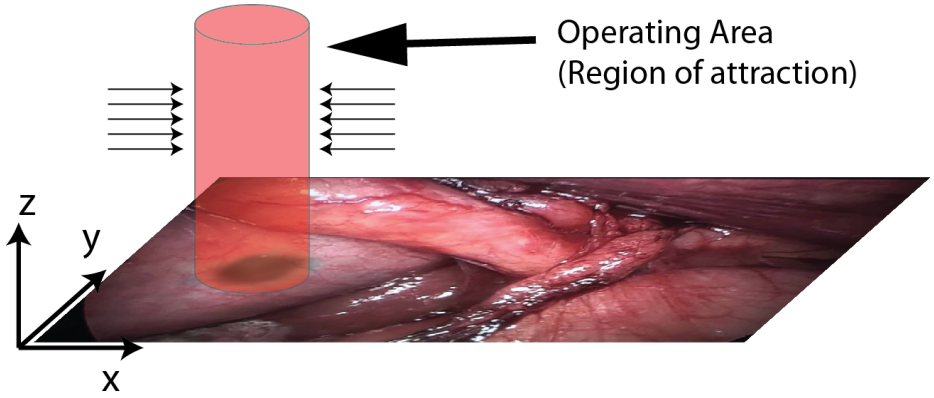
6. The human operator can override assistance.

The resulting virtual environment has an attractive force that guides a user into a desired region of operation, and allows free motion within that region.

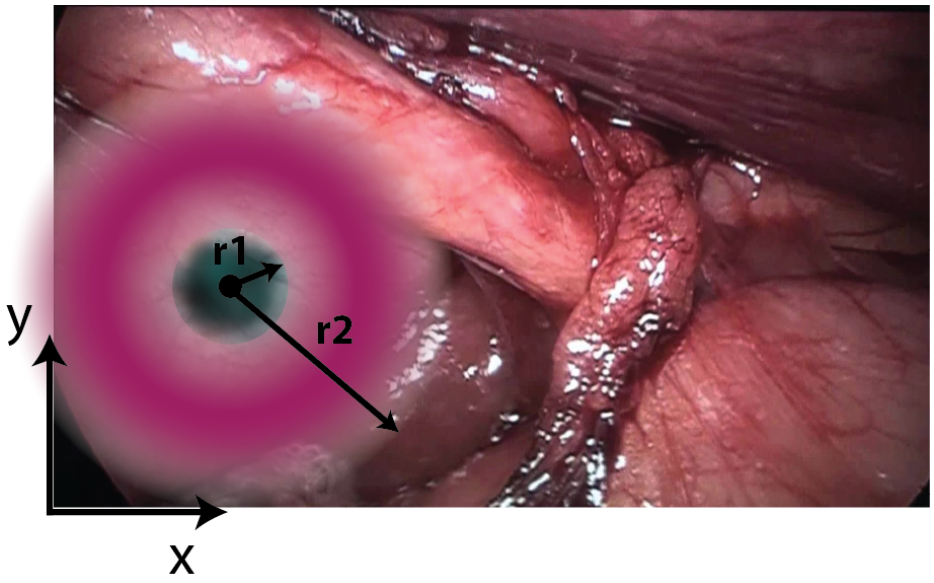
The region of operation is defined by the targeted point(s) of environment contact/action and the radius of free space motion around the target. Specifying the location and shape might be done by the user using virtual environment design software, taken from a pre-operative plan, or recognized online by object / task recognition. Or the operating region might simply be a convenient point in the workspace, e.g., in front of a camera so that a grasped object may be inspected. The area around the target depends on the task, and might be user selectable or learned from prior performance of similar tasks.

The attractive force effect can be described by any of a wide variety of functions. Force feedback should be free of discontinuities that would cause sudden, unexpected acceleration of the master device held by the user. Example force functions would be triangle, parabolic, or sinusoidal force profiles. Also, since the goal of the assistive force feedback is to help with proprioceptive targeting of the region of operation, force feedback can be added only when the user gets close to the target. This avoids discontinuous forces when haptic assistance is turned on/off and assists only with the final targeting of the task region. For example, for a pick and place task, an ARVF with radius 4-5 times the target radius, would help the user with final precise placement, but leave him/her free to select an object to pick up.

Take, as an example, the cylindrical ARVF shown in Figure 6.3. Here, the target region is the same as in Figure 6.1, and the cylindrical volume of radius r_1 above the target is the desired region of operation. Define a sinusoidal force feedback function out to a radius, r_2, around the operating area. Let d be the distance from the robot's end effector position, X, to the center of the cylinder, and $d < r_1$ when the end effector is inside the area of operation. Let X_d be the point on the surface of the cylinder closest to X. Then force feedback to the master, F_m is given by:



(a)



(b)

Figure 6.3: Illustration of the Attractive Region Virtual Fixture.

$$F_{mag} = \begin{cases} 0 & ,d \leq r_1 \\ \frac{1}{2}F_{max}\left(1 - cos\left(2\pi\frac{(d-r_1)}{(r_2-r_1)}\right)\right) & ,r_1 < d < r_2 \\ 0 & ,d \geq r_2 \end{cases} \tag{6.4}$$

$$F_m = F_{mag}\left(\frac{X_d - X}{|X_d - X|}\right) \tag{6.5}$$

This is one possible implementation of an ARVF. Alternatively, for example, a hemispherical force function could easily be defined, or a sawtooth force profile substituted for the sinusoidal one. The general principle remains the same: a user is attracted to the desired region, not via a planned trajectory but via attractive haptic force.

6.4 "The Lendvay Jetstream" Mixed Autonomy Fixture

Virtual fixtures require that the computer has some representation of the task. Under that assumption, it's possible that the computer has sufficient knowledge to execute some part of the task itself.

Mixed autonomy assistance functions share control of the robot hardware between the user and the computer. When the task is suitably well known and movement is sufficiently safe, the assistance function sends autonomous motion commands to the robot. Otherwise, the human must take full manual control.

One way to determine safe action is by segmenting the workspace into safe and unsafe regions; and when action takes place in the safe region, the computer can take control. Thus a *spatially varying autonomy field* function is defined across the workspace delimiting safe and unsafe regions of computer-controlled motion.

The *Lendvay Jetstream* [1] mixed autonomy function (JSMA) is a spatially varying autonomy function for safe, fast robot control in areas sufficiently far from environment contact.

The JSMA approach is based on the following assumptions:

1. A specific region of operation is identified where a task will be performed.

[1] Named in honor Thomas Lendvay, M.D. who came up with the title, contributed significantly to medical inspiration for much of this work, and who will someday be replaced by mixed or fully autonomous robotic surgery.

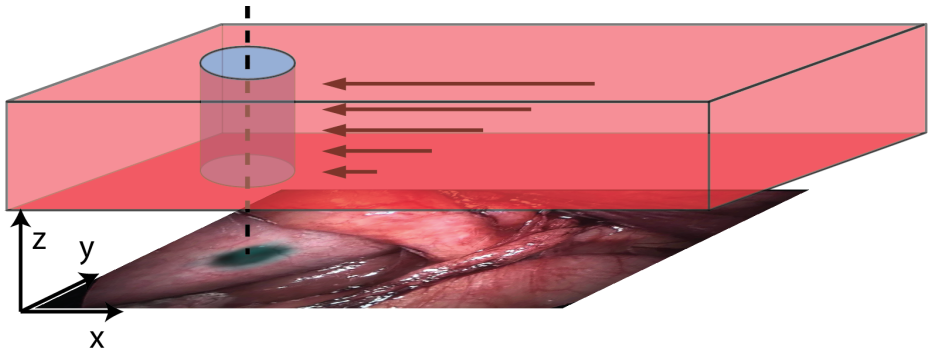


Figure 6.4: Illustration of the Lendvay Jestream Mixed Autonomy Function. Autonomous velocity magnitude is proportional to the elevation penetration into the virtual "jetstream". Figure shows motion left-to-right, but the velocity command is always towards the target. The blue cylinder represents the portion of the region of operation that overlaps the autonomous region. Autonomous movement is disabled there.

2. The robot is not sufficiently trusted to perform environment manipulation.

3. Computer controlled motion in free space is acceptable and safe.

4. No autonomy is allowable in contact or close to contact with the environment,

5. The further the robot is from the environment, the faster it is permitted to move.

6. Autonomy should be disabled within the region of operation for the task.

7. User control and computer control should be shared, not traded.

As a result, two pure-manual regions are defined. The first, for safety, is defined near any objects in the environment. The second is the region of operation for the task, and allows the human operator to perform the necessary environment manipulation. In the remaining space the robot is free to perform autonomous actions.

Figure 6.4 shows an example of how this would work for the task space shown in Figure 6.1. The blue cylinder shows the region of operation, the red zone shows the safe autonomy region, and longer arrows indicate faster autonomous movement towards the region of operation.

Also in this case, autonomous motion is in the horizontal direction, while the vertical elevation is under manual control. This is an interesting example of orthogonal autonomous/manual workspaces. Furthermore, it allows the user to enter/exit the autonomy field.

Taking a height, h, of the robot end effector penetration into the autonomy field, and unit vector, X_d in the direction of the region of operation, the desired robot velocity, $\dot{X}$ can be calculated as:

$$\dot{X} = k_v \sqrt{h} X_d \tag{6.6}$$

The square root of the height is used to prevent the velocity from scaling too quickly, although a different height-varying function (linear, logarithmic, cube root, etc) might be used. Also, like the GCVF the autonomy field geometry might be defined using a surface model from point cloud data, or other environment sensing mode. In addition, different movement functions might be specified, for example a rotation component added, or an optimal path might be applied to maintain a specified elevation in the field.

Experimental evaluation The next chapter, Chapter 7, gives an experimental study of these features. Performance benefits are compared, and effects on particular tasks and sub-tasks are explored.

Chapter 7

EXPERIMENTAL EXAMINATION OF TELEMANIPULATION PERFORMANCE WITH COMPUTATIONAL ASSISTANCE

7.1 Introduction

Previous chapters described frameworks for designing virtual fixtures, and a new type unique to this research, spatially varying autonomy. Also, a new robot Raven II was presented as a research tool for experimental robotics, and related work with the robot was cited.

This chapter presents experiments evaluating three novel computer assistance functions. Previously discussed virtual fixture (VF) and mixed autonomy (MA) types are developed for a simple object manipulation task. Human subjects performed example tasks with the various assistance modes. Results demonstrate when and how virtual fixtures and mixed autonomy may help or hinder operation.

7.2 Evaluation Methods

This experimental study of VF's and MA's for human-machine collaborative operation uses two example tasks to study their impact on performance. The hope is that by studying these tasks in particular, we gain insight into more general task performance. In addition, the study looks for any characteristic behaviors or effects that arise from task performance.

Admittedly, results on any particular task may or may not be generalizable to other tasks, however maximum effort is taken to select activities whose measures can generalize to other tasks

7.2.1 Telerobotic Fundamentals of Laparoscopic Surgery

Fundamentals of Laparoscopic Surgery (FLS) is a test suite developed by the Society of American Gastrointestinal and Endoscopic Surgeons (SAGES) to test laparoscopic surgeons' technical skill [99]. Components of the test include a precision suturing task, a precision cutting task, and a gross manipulation task called the FLS block transfer. The block transfer task was adapted by Mitch Lum

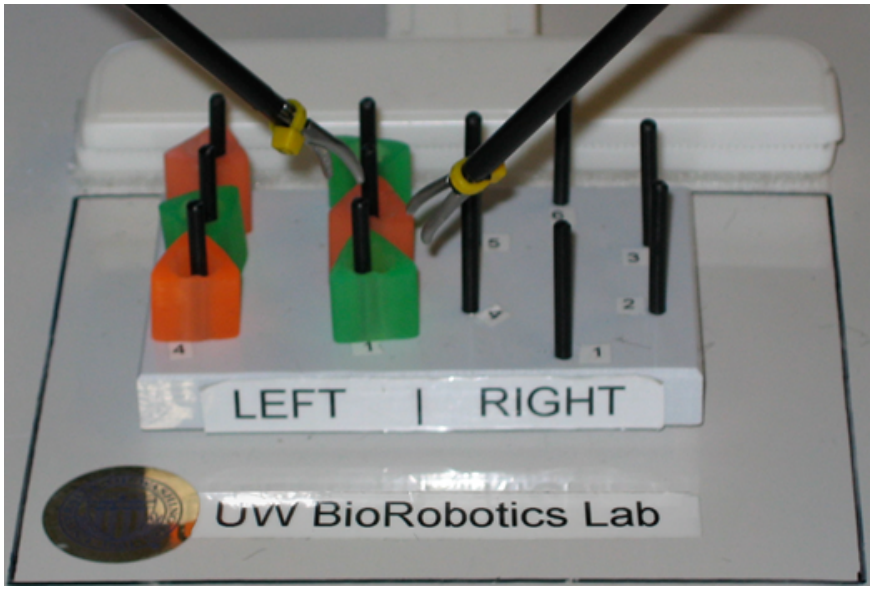


Figure 7.1: Telerobotic FLS task. Blocks are transferred left-to-right and then right-to-left, handing off the blocks between hands in the process [71].

to test robotic task performance [71]. The resulting test is called Telerobotic FLS and is seen in Figure 7.1.

The TFLS task is essentially a pick and place task with the added complexity of a hand-to-hand exchange. Therefore the task requires three actions: pick-up, handoff, and place. Six blocks are first arranged on the left half of an array of pegs. In numbered block order (in contrast to TFLS, FLS blocks can be moved in any order), subjects pick up a block with the left-hand instrument, transfer the block to the right-hand instrument, and place it on the corresponding numbered peg on the right side of the board. They perform the left-to-right transfer for all six blocks, followed by a right-to-left transfer without pausing in between. Thus, the task requires a total of twelve block transfers.

Using collected robot motion data from experimenters can measure the path length and time required for the task. TFLS reports these measures for each of the twelve individual block transfers.

Using the SAGES test gives a surgically relevant measure of human performance manipulating objects in a laparoscopic type environment. However, due to its relative complexity and bimanual requirements TFLS is not a good tool for measuring basic movement control. Therefore, a second,

simpler task is also used.

7.2.2 Fitts' Law and its Extensions

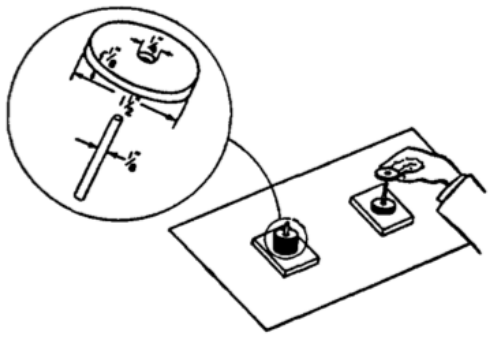


Figure 7.2: From Fitts' 1954 paper [29], "Disc transfer apparatus. The task was to transfer eight washers one at a time from the right to the left pin." Fitts used disks of varying internal diameter to simulate a range of indices of difficulty.

Originally proposed by Paul M. Fitts in 1954 [29], Fitts' Law characterizes the human motor control system, identifying what he called the "information capacity". Fitts measured subjects' performance on simple movement tasks over a range of speed-accuracy conditions. An example of one of these tasks is shown in Figure 7.2, where a human manually transfers disks from one peg to another.

The result of Fitts' experimental analysis was a simple quantitative relationship between task constraints and movement speed (or speed vs. accuracy) during human movement.

$$MT = a + bID \tag{7.1}$$

$$ID = log_2\left(\frac{2A}{W}\right) \tag{7.2}$$

Movement time, MT depends on the "index of difficulty", ID of the task. ID is measured in bits and is derived from the amplitude, A, of the required motion and the width of the target, W. Fitts study included tasks with ID ranging from one to ten bits.

The "Shannon form" of the law described in [74] is similar to Claude Shannon's measure of communications bandwidth [1] and avoids some numerical problems with the original form:

$$MT = a + bID \tag{7.3}$$

$$ID = log_2\left(\frac{A}{W} + 1\right) \tag{7.4}$$

In [121] Stoelen and Akin demonstrated additive effects of combining rotational and translational movement, i.e.,

$$MT = a + b(ID_{translate} + ID_{rotate}) \tag{7.5}$$

In [9] Accot and Zhai also demonstrated additive effects of combining multiple Fitts' Law tasks in series. They examined users' performance on several positioning tasks performed sequentially, and found consistency with Fitts' Law when the ID is of successive movements were added, or,

$$MT = a + b(ID_{task1} + ID_{task2} + ID_{task3}). \tag{7.6}$$

Using these two results gives us an approach to formulating the TFLS task (there are two ways to perform block-transfer handoff, with or without a 90 degree rotation. This analysis assumes the method with rotation is used.)

Furthermore, two measures give a unique overall measure of performance, "throughput" and "index of performance". Index of performance (IP) comes from Fitts' original paper and is given by:

$$IP = 1/b \tag{7.7}$$

The inverse of the slope gives an idea of how movement time will scale up/down with difficulty. However, this measure assumes the performance curve passes through the origin, otherwise it fails to encompass all performance results.

Throughput (TP), as suggested in [118] can is calculated by:

$$TP = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{M}\sum_{j=1}^{M}\frac{ID_{e_{ij}}}{MT_{ij}}\right) \tag{7.8}$$

[1] The analogy to Shannon's theorem is as follows: Amplitude, A, relates to signal power, P; while target width W relates to noise tolerance N. Therefore Shannon's Theorem: $C = log_2((P+N)/N)$ is relatable to the motion throughput of the human motor system: $MT = a + bLog_2((A+W)/W)$. Whether this is an appropriate or generalizable analogy is an excellent question that is beyond the scope of this thesis. For further discussion the user is referred to [74].

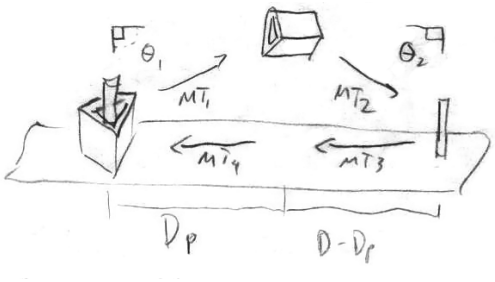


Figure 7.3: Illustration of the three phases required to transfer a block.

N is the number of subjects, M is the number of trials and MT_{ij} is the move time for the trial. ID_e is a corrected index of difficulty for the case where the effective movement amplitude and width are not utilized due to large target widths.

7.2.3 Information Complexity of TFLS

For the grasp segment of movement (move to block grasp):

$$ID_{translate} = log_2\left(\frac{D_p}{W_g}+1\right) \tag{7.9}$$

$$ID_{rotate} = log_2\left(\frac{90}{\theta_1}+1\right) \tag{7.10}$$

$$MT = a + b(ID_{translate1} + ID_{rotate1}) \tag{7.11}$$

The three phases of movement for block i (pick-up, put-down, handoff) can, in theory, be combined:

$$Pick-up: MT_{4_i} = a_4 + b_4\left(log_2\left(\frac{P_{i,i+1}}{2W_g}+1\right) + log_2\left(\frac{90}{\theta_1}+1\right)\right) \tag{7.12}$$

$$Put-down: MT_{2_i} = a_2 + b_2\left(log_2\left(\frac{P_{i,}}{2W_p}+1\right) + log_2\left(\frac{90}{\theta_2}+1\right)\right) \tag{7.13}$$

$$Handoff: MT_{13_i} = a_{13} + b_{13}\left(log_2\left(\frac{(P_i+P_{i,i+1})/2}{2W_g}+1\right) + log_2\left(\frac{90}{\theta_1}+1\right)\right) \tag{7.14}$$

$$Total: MT_{block_i} = MT_{1_i} + MT_{2_i} + MT_{13_i} \tag{7.15}$$

Parameters are described in Table 7.1

Parameter	Symbol	Value
Grasping tolerance	$W_g = w_t - w_g$	3mm = 6mm - 3mm
Grasping angle tolerance	θ_1	5°
Block drop tolerance	$W_p = w_g - w_p$	6.5mm = 9.5mm - 3mm
Block drop angle tolerance	θ_2	5°
Distance (pick→place)	$D_p = \frac{P_i}{2}$	Variable
Distance (place→pick)	$D_p = \frac{P_{i,i+1}}{2}$	Variable

Table 7.1: Measurements of Fitts' Law parameters for TFLS Task.

Block	1	2	3	4	5	6
P_i	32	57	56	50	49	66
$P_{i,i+1}$	32	46	63	89	50	47
ID	26	28.4	29.1	30.4	28.1	29

Table 7.2: Distance required and index of difficulty each TFLS block transferred.

Where MT_{block_i} is the movement time for block i predicted by Fitts' Law and its extensions. Note that MT_{13} is a bi-manual task phase, so we expect b_{13} to be larger.

Computed TFLS task board IDs are given in Table 7.2.

The many assumptions in this analysis (additive IDs, straightforward combination of rotation and translation IDs, bimanual operation) make a weak case for Fitts Law analysis of the task. Therefore, while using TFLS offers one good measurement of manipulation performance on a certain class of tasks, it does not offer generalizable throughput or index of performance metrics available using Fitts Law. To that end an additional simplified task is also used to find those measures.

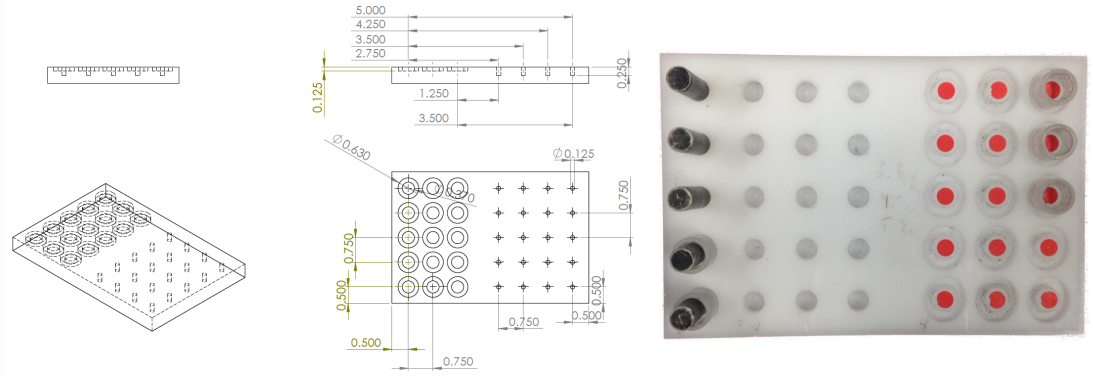


Figure 7.4: (Left) CAD Model of Fitts' Law task board. (Board specifications) movement amplitude: 1.25, 2.0, 2.75, 3.5, 4.25, 5.0 in. Block width of 0.5, .375 in Peg diameter: 0.125, 0.3125 in. (Right) Photo of the board.

7.2.4 Fitts' Law Task

A Fitts' Law type task is used to measure the motor control performance of humans using the telerobot. Three peg boards were constructed with five pegs and blocks. The board can be seen in Figure 7.4.

The task requires a user to transfer cylindrical blocks from wells on one half of the board to pegs on the other. The three boards are usable with two different block sizes (D_b): 0.5 and 0.375 inches; and three unique peg diameters (D_p): 0.125, 0.1875, and 0.3125 inches.

For the fitts law analysis, width W is calculated as $W = D_b - D_p$.

The movement amplitude, A is the center-to-center distance between a block pick-up location and a target peg. The boards accommodate six different A values: 1.25, 2.00, 2.75, 3.50, 4.25, 5.00 inches.

The different combinations of peg diameters, block diameters, and peg-block-distances yields a matrix of 36 IDs ranging from 2.12 to 6.34 bits. From this range, five IDs were selected for the experiments: the maximum and minimum, and three that approximate an equal spacing across the range of values. The selected IDs are: 2.12, 3.17, 4.30, 5.49, and 6.34 bits.

7.3 Computer Assistance Functions

The three computer assistance functions described in Chapter 6 were implemented to assist the performance of the above tasks.

The tasks were divided into component phases and the assistance functions were programmed to change state to assist each phase. The FL task has two phases for each of the five block transfers: block pick-up and block place. This makes a total of ten phases. The TFLS task has three phases for each of the twelve block transfers: block pick-up, block handoff, and block place. This makes a total of thirty six phases.

Phase transitions are controlled by the experimenter. When the user completes a phase, i.e. picks, hands off, or places a block, the experimenter uses the keyboard to command the assistance function to advance to the next phase.

7.3.1 "Golf Course" Forbidden Region Virtual Fixture

The primary assumption behind this assistive feature is that there are particular regions of interest (ROI) in which the surgeon needs to operate while the rest of the environment should not be contacted. For example a surgeon may want to apply electrocautery only to certain tissues and avoid all others.

The resulting virtual fixture design provides motion constraints that limit the robot end effector to safe regions. Safe regions are two overlapping spaces: well away from environmental contact, above a horizontally arranged plane; and vertical cylinder "holes" in the plane allowing the user to reach through the plane into environmental contact near the ROI. For the target tasks this has been developed and called the "Golf Course Virtual Fixture" (GC), and implementation for the Fitts' Law task is shown in Figure 7.5.

Haptic Rendering

The GC comprises two parts: the restrictive surface keeping the robot away from the environment and permissive regions that allow access to specified contact areas.

The permissive regions are cylindrical openings in the plane rendered by:

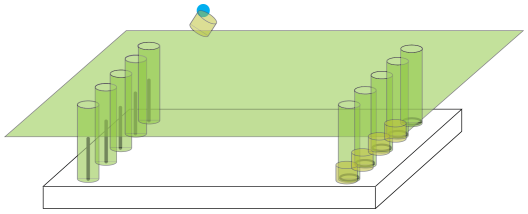


Figure 7.5: Illustration of the Golf Course Virtual Fixture. User motion is restricted to above the plane, or inside the cylinders.

$$d = \left\| \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \\ 0 \end{bmatrix} \right\|_2 - r, \tag{7.16}$$

$$F_d = k_1 d - k_2 \left\| \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} \right\|_2 \tag{7.17}$$

$$F_{cyl} = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{cases} [0] & ,d \leq r \\ \begin{bmatrix} x - x_c \\ y - y_c \\ 0 \end{bmatrix} \frac{F_d}{d+r} & ,d > r \end{cases} \tag{7.18}$$

The restrictive surface is a horizontally arranged plane described by:

$$f_z = \begin{cases} 0 & ,Z >= Z_p \\ k_1(Z_p - Z) - k_2(\dot{Z}) & ,Z < Z_p \end{cases} \tag{7.19}$$

The two are combined by taking the minimum force contribution of the two (which can be zero when out of contact):

$$F_m = min(\|F_{plane}\|, \|F_{cyl}\|) \tag{7.20}$$

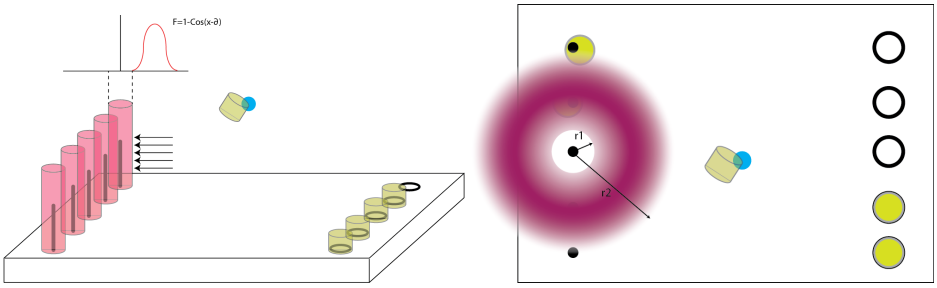


Figure 7.6: Illustration of the Attractive Cylinders Virtual Fixture. A sinusoidal force profile surrounds a dead zone around the block place location. Five cylinders are shown, but only one is active at a time.

7.3.2 "Attractive Cylinders" Guidance Virtual Fixture

The objective of this feature is to assist the user's natural proprioceptive sense by actively pushing their hand towards the goal. There is no assistance added to operation in other areas, except as a side-effect of being encouraged to move towards the objective.

The "Attractive Cylinders" virtual fixture comprises cylindrical volumes surrounding an intended target or targets. A sinusoidal force profile surrounding the region creates a smooth transition from freespace motion outside the target area through the force field, and into the target manipulation region (see Figure 7.6).

Haptic Rendering

The AC is rendered as a single sinusoidal force profile in a radius around the target.

$$d = \left\| \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \\ 0 \end{bmatrix} \right\|_2, \tag{7.21}$$

$$F_d = \begin{cases} 0 & ,d \leq r_1, \\ F_{max}\left(1 - cos\left(\frac{2\pi(d-r_1)}{r_2}\right)\right) & ,r_1 < d < r_2, \\ 0 & ,d \geq r_2. \end{cases} \tag{7.22}$$

$$F_m = \begin{bmatrix} F_x \\ F_y \\ 0 \end{bmatrix} = \begin{bmatrix} x - x_c \\ y - y_c \\ 0 \end{bmatrix} \frac{F_d}{d} \tag{7.23}$$

7.3.3 "The Lendvay Jetstream" Mixed Autonomy Function

Let's assume the computer has some knowledge of *where* the user wants to operate, but not necessarily what the user will do or how to do it. In addition let's assume the computer knows about safe regions in which to move autonomously, i.e, in free space away from environment contact. In this safe region the robot can autonomously carry out non-contact movements. The result is a spatially varying autonomy field with some regions categorized safe for autonomy and others that are not safe for autonomy.

The rules for this mixed autonomy function are as follows:

- Segment space into safe regions and unsafe regions, using sensing or human input.
- Identify task relevant areas either through sensing or human input.
- If the robot is in the safe region, send movement commands to move towards the task relevant area.
- Human movement commands are directly added to assistance function commands.

As implemented for the two tasks, the Lendvay Jetstream function uses a partition function based on height: when the user is away from environmental contact, above the peg board, it is safe

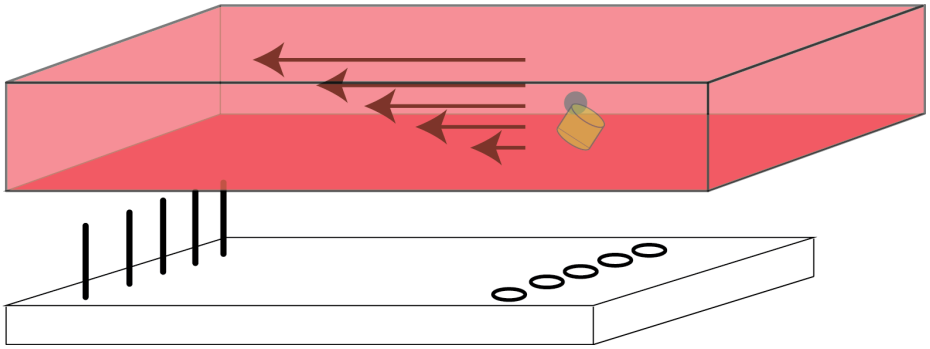


Figure 7.7: Illustration of the Lendvay Jetsream mixed autonomy function. Autonomous velocity magnitude is proportional to the elevation penetration into the virtual "Jetstream". Figure shows motion left-to-right, but the velocity command is always towards the target.

to apply autonomous movement; below that elevation all robot motion is directed manually by the user.

Autonomous Movement

The Jetstream autonomous movement function as used in this experiment was defined as:

$$H = z - z_j \tag{7.24}$$

$$\bar{X}_d = \frac{\begin{bmatrix} x \\ y \\ 0 \end{bmatrix} - \begin{bmatrix} x_p \\ y_p \\ 0 \end{bmatrix}}{\left\| \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} - \begin{bmatrix} x_p \\ y_p \\ 0 \end{bmatrix} \right\|} \tag{7.25}$$

$$\bar{V} = k_v \bar{X}_d \sqrt{H} \tag{7.26}$$

$$\Delta \bar{X} = \bar{V} \Delta t \tag{7.27}$$

d the distance the user has penetrated into the Jetstream region, z the end effector elevation , and z_j the Jetstream height. The direction $\bar{X_d}$ is computed from the current position $\bar{X}$ to target $\bar{X_t}$.

Velocity $\bar{V}$ is then used to calculate a robot motion increment command $\Delta\bar{X}$ for the next time step Δt.

Mixed Autonomy Implementation

Motion increments, $\Delta\bar{X}$ are sent to the slave robot in parallel with those from the master device, and the human and machine agents have equal control of the robot. On the slave side, motion increments from both sources are integrated to generate a desired robot position.

Combination of the two command sources has two results. First, the human can override the autonomous motion command subject to master-device workspace limits and movement speed. Second, the user has control of the penetration depth into the autonomy field, and therefore is able to take back manual control to operate on the environment.

7.4 Experiment Design

Experiments were designed to test how the different assistance functions affect performance of the FL and TFLS tasks, and to glean insights into how manual and computer control work together. All experiments are approved by University of Washington IRB application #44956 EB.

Twenty two people participated as subjects in this experiment. Seven subjects were participants in a previous study and were already proficient in using the master-slave teleoperation system for purely manual, unaassisted control [34]. The remaining fifteen subjects had little or no prior experience with telerobotics.

7.4.1 Training

All subjects, before starting operation of the robot, were shown a slideshow presentation that described the experiment, experimental objectives, and the three assistance functions.

Due to the difference in experience levels, two different training programs were used: one for novice users and one for proficient users.

Novice users underwent four training phases:

- First, a basic orientation training used a simulator and the user practiced re-orienting the simulated robot end effector five to ten times.

- In the second phase, the novice user transferred three FL task blocks right to left with the right hand. The exercise was repeated eight times, twice with each assistance mode including unassisted operation.

- Next, the user performed the five-block FL task using a unique index of difficulty that was not used in the final assessment. This task was repeated eight times, twice with each assistance mode.

- A final training phase was executed after the FL task trials and before the TFLS trials. In this exercise the user transferred three TFLS style blocks on a training board that resembles the TFLS board, but is geometrically different. The user executeed the bi-manual task four times, once with each assistance mode.

7.4.2 Master Console

The complete master console is seen in Figure 7.8. This experiment used the Mantis Duo (Mimic Inc., Seattle, WA, USA) as the master interface. The Manits is a two handed wire-tension based haptic system capable of rendering forces up to 15.2N. It has 7 DoF input (6 cartesian plus grasp) and a workspace of approximately 790x522x394mm. A motion scaling factor of 0.75 was used from user's hand to robot motion.

A Panasonic monitor placed behind the Mantis Duo provided visual feedback. Video was acquired using a Microsoft LifeCam Studio webcam at 640x480 resolution at 30fps, and displayed on the monitor. The camera viewed the task at an angle of approximately 35° declension.

A footpedal acted as the clutching/indexing control. A user's foot pressed the pedal at all times during operation of the robot.

The user sat in a chair with armrests and was allowed to adjust the height of the chair and position armrests. Also, the Mantis Duo wrist-support was adjusted to improve comfort and range of motion of the hand.

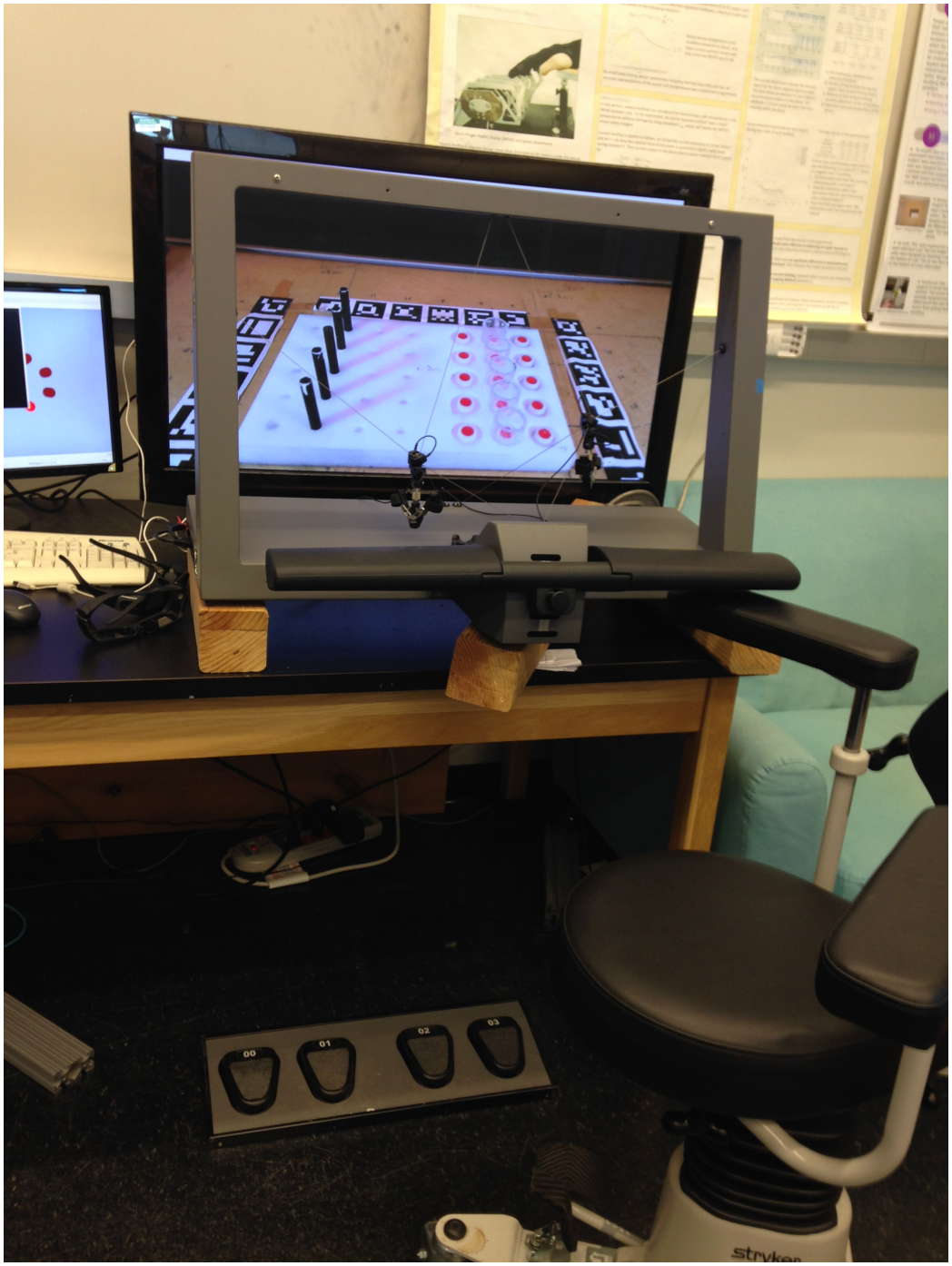


Figure 7.8: The master console shows HDTV visual feedback, the Mantis Duo bi-manual haptic interface, footpedal, and chair.

7.4.3 Trials and Ordering

Each subject performed twenty four trials: twenty FL task iterations and four TFLS task iterations. First the FL tasks were performed followed by the TFLS tasks. The FL task iterations represent twenty unique trial conditions:

$$4\ assistance\ modes\ x\ 5\ indices\ of\ difficulty = 20\ trials \tag{7.28}$$

The FL trial conditions were presented to subjects in random order, randomized for each subject.

Four TFLS task iterations included one iteration for each assistance mode; these were also presented in random order, randomized for each subject.

It should be noted that each FL task iteration required five block transfers, while each TFLS iteration included twelve block transfers. Therefore one task iteration provided a sample size of five or twelve, respectively, for that experimental condition.

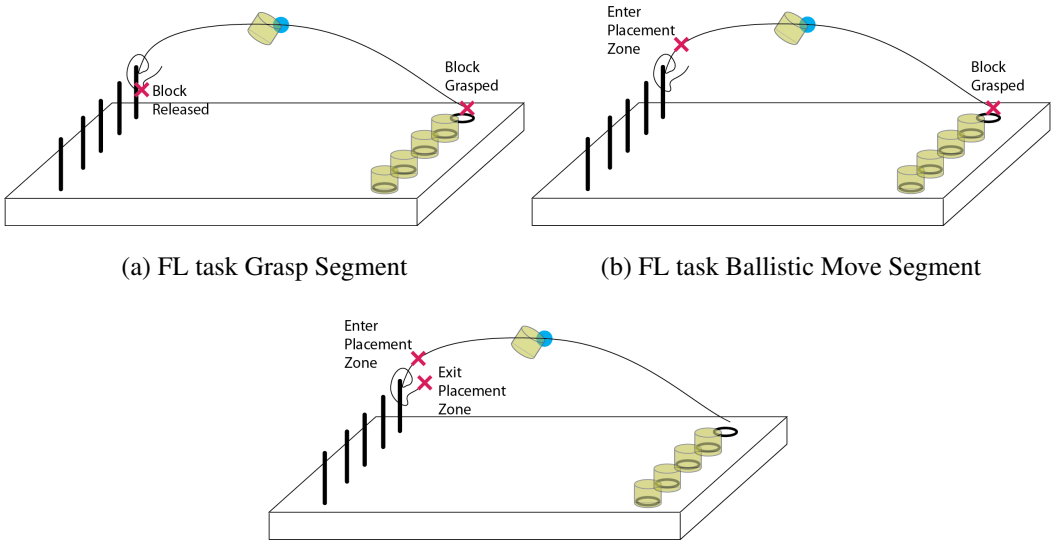
7.5 Data Collection and Analysis

Trial data were automatically collected using the ROS 'rosbag' function. All robot movement data are timestamped and saved for offline analysis. The stored data include: timestamp, task phase, assistance mode, cartesian end effector pose, grasper angle, commanded master-side force, and autonomous movement command.

7.5.1 Data Segmentation for Fitts' Law Task

The simplicity of motion of the FL task allowed a fine-grained segmentation of the resulting data. For each task the data were divided into the following *partially overlapping* component segments:

- "Grasp segment": begins when the block is grasped and ends when the block is released.
- "Ballistic segment": begins when the block is grasped and ends when the end-effector reaches the target area. The target area is defined as a region 5mm horizontal radius from the final block-release location. This segment overlaps the grasp segment.



(a) FL task Grasp Segment

(b) FL task Ballistic Move Segment

(c) FL task Placement Segment

Figure 7.9: Motion segmentation of Fitts' Law task

- "Placement segment": begins when the ballistic segment ends; when the end effector reaches the target area. Ends after the block is released and the robot leaves the target area.

Note that the end of the grasp segment and the end of the placement segment are not the same. The grasp segment ends precisely when the block is released, while the placement segment ends after the block is released and the user moves outside the placement zone.

Data segmentation was performed automatically in data analysis software. The three segments are illustrated in Figure 7.9.

7.5.2 Data Segmentation for Telerobotic FLS Task

TFLS data were segmented into the three phases of task completion shown in Figure 7.3:

- Block pick-up phase

- Block handoff phase

- Block placement segment

These three phases are repeated for each of the twelve block transfers: block pick-up, block handoff, and block place. This makes a total of thirty six phases, and data for each phase is reported.

In addition path length and movement times for the entire block transfer are reported for each of the twelve blocks.

7.5.3 Statistical Methods

Fixed effects analysis of variance (ANOVA) was used to identify differences between assistance modes in both tasks. Response variables (time, path length) are modeled as a function of assistance mode and, for FL metrics, index of difficulty. This tested the signficance of assistance considering the data from all trials.

Before calculating ANOVA completion time and path length, data were log-transformed to satisfy normality and homogeneity. Distributions were examined using Shapiro-Wilk test. The resulting ANOVA model coefficients and confidence intervals were then exponentiated to interpret the resulting calculated differences. The ANOVA model also yields significance measures relative to the intercept (control) case. The resulting significance values account for multiple comparisons.

7.6 Results: Fitts Law Task

7.6.1 Fitts' Law fit

The FL task performance was analyzed in the Fitts' Law paradigm for each of the four assistance modes: Attractive Cylinders (AC), Golf Course (GC), Jetstream (JS), and no assistance (NA). The overall mean grasp segment time was calculated for each of the four AFs at each of the five IDs. Four regression lines were calculated, one for each AF, by fitting to the ID means. Results of the linear regression are shown in Figure 7.10

The movement time (MT) to index of difficulty (ID) relationships for the four operation modes

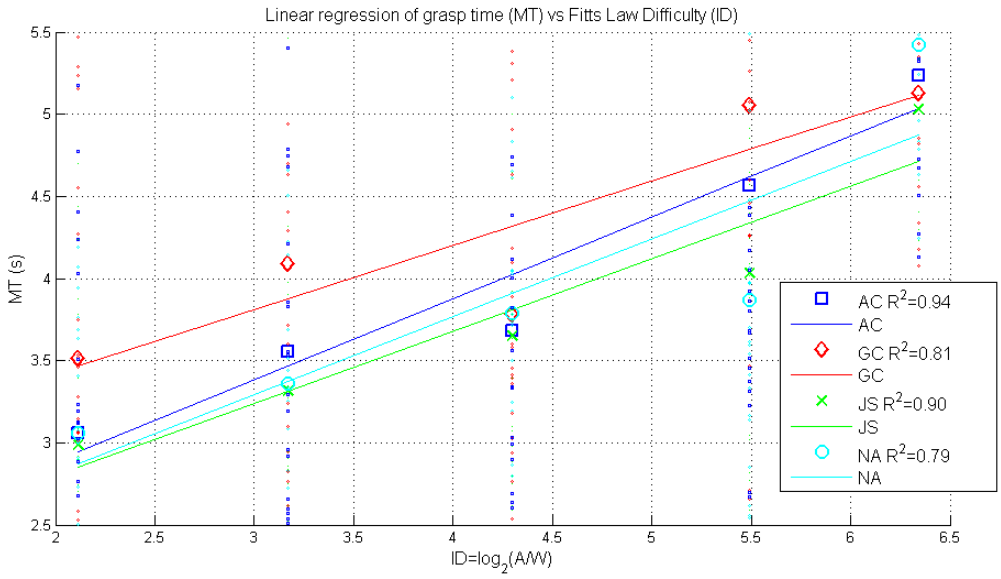


Figure 7.10: Fitts' Law line fitting to all data combined

were found to be:

$$AC : MT = 1.8973 + 0.4951(ID) \tag{7.29}$$

$$GC : MT = 2.6410 + 0.3905(ID) \tag{7.30}$$

$$JS : MT = 1.9177 + 0.4405(ID) \tag{7.31}$$

$$NA : MT = 1.8683 + 0.4743(ID) \tag{7.32}$$

$$\tag{7.33}$$

The R^2 coefficients for the four modes are AC: 0.94, GC: 0.81, JS: 0.90, and NA:0.79.

This can be further analyzed in terms of the "Fitts Index of Performance" (IP) defined as:

$$IP = 1/b \ (bits/sec) \tag{7.34}$$

With a larger value indicating better performance. In this case the IP for the four modes is AC: 2.01, GC: 2.56, JS: 2.27, NA: 2.10 bits/sec.

Throughput (TP) was calculated by Equation 7.8. This analysis shows that TP for the four modes is AC:1.19, GC:1.12, JS:1.27, NA:1.22 bits/sec Higher TP indicates better performance.

7.6.2 Movement Time

The FL task was analyzed in terms of performance in three different segments as described in section 7.5.1. Data were measured for each block transfered. The required times for each task segment are reported in Table 7.3. The times are reported as mean and standard deviation for each assistance mode for each index of difficulty.

ID	A.C.	G.C.	J.S.	N.A.
2.12	3.06 / 0.11	3.51 / 0.13	2.98 / 0.18	3.06 / 0.11
3.17	3.55 / 0.15	4.09 / 0.19	3.32 / 0.14	3.36 / 0.13
4.30	3.68 / 0.16	3.78 / 0.12	3.65 / 0.16	3.79 / 0.12
5.49	4.56 / 0.21	5.06 / 0.23	4.04 / 0.20	3.87 / 0.14
6.34	5.23 / 0.20	5.13 / 0.20	5.03 / 0.27	5.42 / 0.24

(a) Grasp segment

ID	A.C.	G.C.	J.S.	N.A.
2.12	1.70 / 0.08	2.03 / 0.10	1.61 / 0.14	1.79 / 0.06
3.17	1.77 / 0.10	2.13 / 0.10	1.85 / 0.08	1.92 / 0.07
4.30	2.20 / 0.10	2.23 / 0.07	1.84 / 0.08	2.27 / 0.08
5.49	2.13 / 0.12	2.59 / 0.14	1.88 / 0.09	2.33 / 0.11
6.34	2.43 / 0.10	2.86 / 0.13	2.60 / 0.13	3.08 / 0.17

(b) Ballistic movement segment

ID	A.C.	G.C.	J.S.	N.A.
2.12	1.73 / 0.09	1.93 / 0.08	1.82 / 0.13	1.70 / 0.10
3.17	2.26 / 0.14	2.43 / 0.18	2.00 / 0.13	1.98 / 0.15
4.30	2.01 / 0.12	2.01 / 0.11	2.32 / 0.17	2.07 / 0.11
5.49	3.39 / 0.22	3.33 / 0.20	2.78 / 0.21	2.43 / 0.15
6.34	3.75 / 0.20	2.99 / 0.17	3.33 / 0.30	3.31 / 0.21

(c) Placement segment

Table 7.3: Movement Time Mean / Standard Deviation (Sec.). Given for each index of difficulty (ID) for all assistance functions: Attractive Cylinders, Golf Course, Jet Stream, No Assistance.

The distribution of the grasp segment times is shown in Figure 7.11. Distribution of Ballistic Movement and placement segment times are shown in Figures 7.12 and 7.13.

Statistical Analysis ANOVA was performed on the movement times with fixed effects of AF and ID. The data were log transformed to ensure normality and homogeneity. Shapiro-Wilk normality

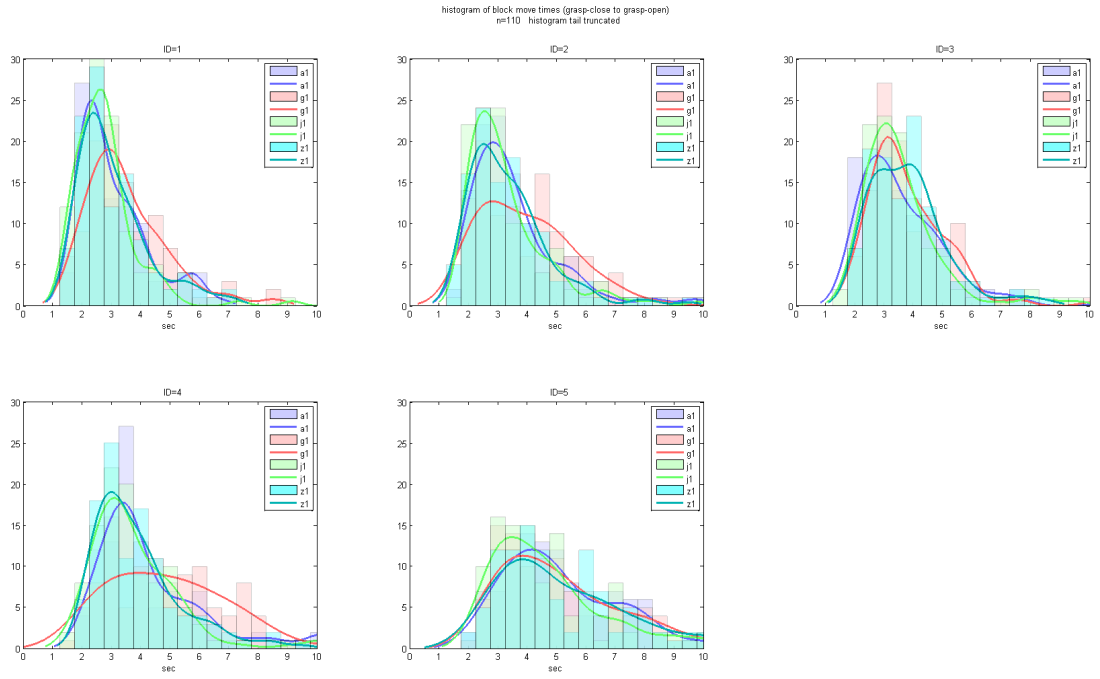


Figure 7.11: Grasp segment time (sec) histogram (FL task). One plot for each ID, lines represent assistance functions: purple=attractive cylinder, red=golf-course, green=Jetstream, blue=no assistance.

test on the log transformed Grasp, Ballistic Move, and Placement Times yielded significance values of p=1.01e-5, 2.2e-16, 5.11e-15, indicating the data distributions are well suited for linear modeling with ANOVA.

Table 7.4 reports the results of ANOVA of segment times. The reported estimated difference in means (Est) is of log-transformed, data. Exponentiating the estimate yields the expected multiplicative difference of means. Or, in other words, given the treatment, the expected movement duration is e^{Est} times the no-assistance duration. The percentage difference is also reported. Negative percentage indicates faster performance.

Little variation is observed in the overall grasp time, ranging from 4% slower G.C. performance to 2% faster J.S. performance. Ballistic movement time is much faster for the A.C. and J.S. modes (11% and 16% respectively), while placement times are slower for A.C. and J.S. (13% and 2%). This is significant in the A.C. case. Golf course in slower in all segments of operation.

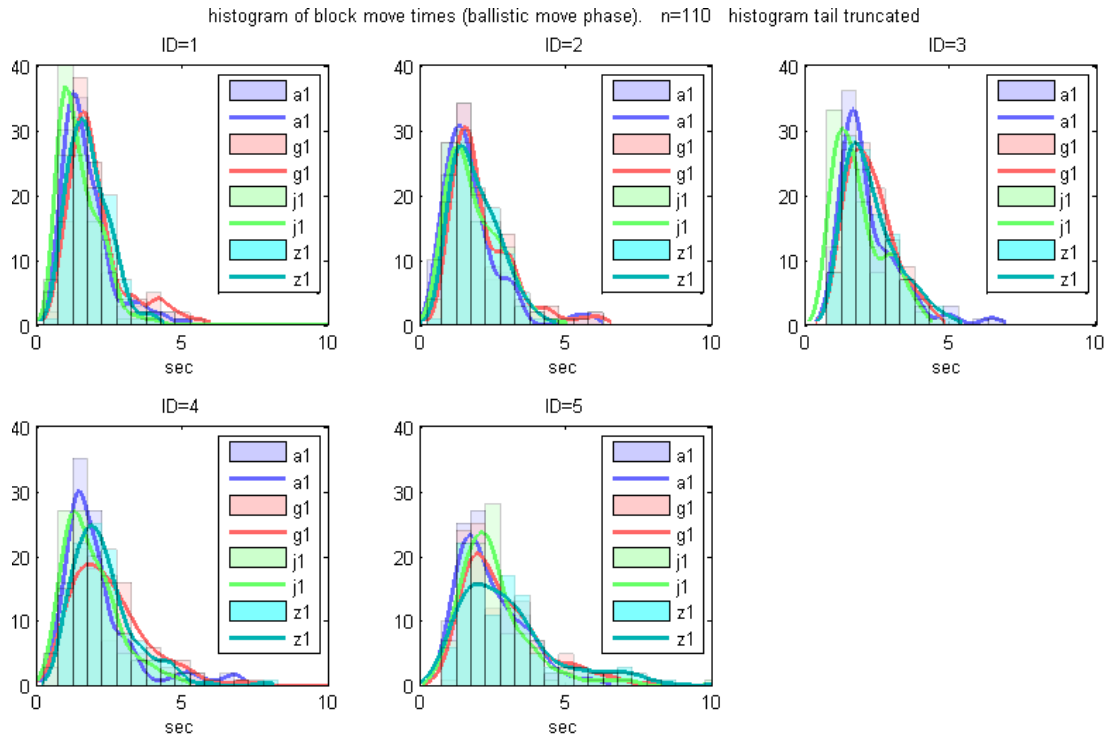


Figure 7.12: Ballistic movement segment time (sec) histogram (FL task). One plot for each ID, lines represent assistance functions: purple=attractive cylinder, red=golf-course, green=Jetstream, blue=no assistance.

7.6.3 Path Lengths

The mean and standard deviations of the distance travelled (path length) for each block are reported in table 7.5. The path lengths are for each assistance mode for each index of difficulty.

Figures 7.14, 7.15, and 7.16 show the distribution of block transfer path lengths for the grasp, ballistic movement, and placement segments respectively. The distributions are separated by ID, and histogram fit curves represent the four assistance modes.

Statistical Analysis ANOVA was performed on the path lengths with fixed effects of AF and ID. The data were log transformed to ensure normality and homgenaity. Shapiro-Wilk normality test on the log transformed Grasp, Ballistic Move, and Placement path lengths yielded significance values

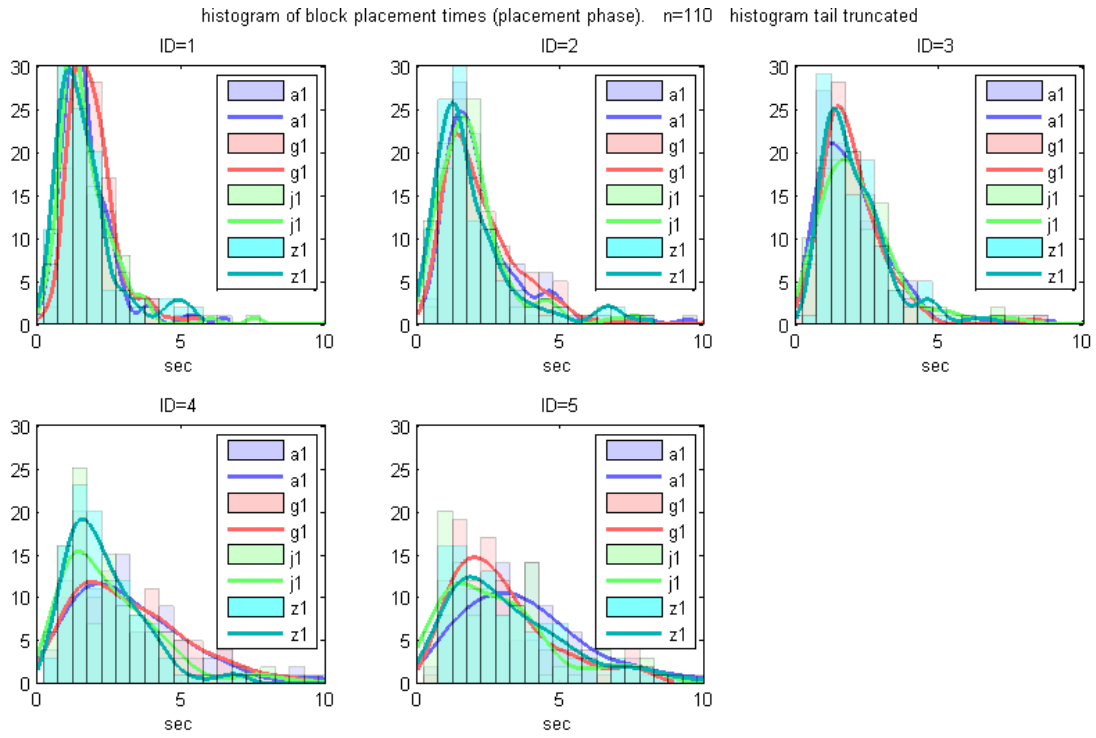


Figure 7.13: Placement segment time (sec) histogram (FL task). One plot for each ID, lines represent assistance functions: purple=attractive cylinder, red=golf-course, green=Jetstream, blue=no assistance.

of p=1.68e-6, 6.6e-06, 6.7e-4, indicating the data distributions are well suited for linear modeling with ANOVA.

Table 7.6 reports the results of ANOVA of path lengths. Reported estimated difference in means is of log-transformed data. Exponentiating the estimate yields the expected multiplicative difference of means, or in other words, given the treatment, the expected path length is e^{Est} times the no-assistance duration. The percentage difference is also reported. Negative percentage indicates faster performance.

All assistive modes show a longer grasp-path length compared to N.A. and for A.C. and G.C. this is a significant difference. J.S. shows a slightly shorter ballistic movement segment path, but longer placement path. A.C. shows a much longer, 20%, placement path, while J.S. and G.C. are 8% and 17% longer respectively- all significant differences.

	Estimate	e^{Est}	%	p
A.C.	0.010	1.01	1%	0.338
J.S.	-0.018	0.98	-2%	0.077
G.C.	0.043	1.04	4%	<0.001

(a) grasp segment

	Estimate	e^{Est}	%	p
A.C.	-0.119	0.88	-11%	<0.001
J.S.	-0.171	0.84	-16%	<0.001
G.C.	0.041	1.04	4%	0.126

(b) Ballistic Movement Phase

	Estimate	e^{Est}	%	p
A.C.	0.126	1.13	13%	<0.001
J.S.	0.023	1.02	2%	0.529
G.C.	0.127	1.13	14%	<0.001

(c) placement segment

Table 7.4: ANOVA between movement times. Each AF is compared to the no-assistance case. Columns are: Estimated difference of log-transformed means, exponentiated estimate, percent difference of means, P-value significance.

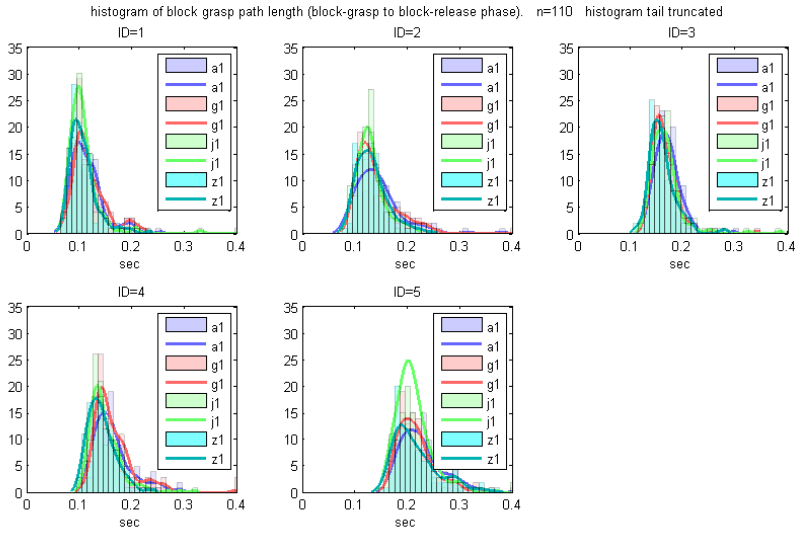


Figure 7.14: Grasp segment path length (m) histogram. One plot for each ID, lines represent assistance functions a1=attractive cyl., g1=Golf Course, j1=Jetstream, z1=no assistance.

ID	A.C.	G.C.	J.S.	N.A.
2.12	0.116 / 0.003	0.120 / 0.003	0.111 / 0.005	0.108 / 0.003
3.17	0.145 / 0.004	0.142 / 0.004	0.129 / 0.003	0.133 / 0.003
4.30	0.170 / 0.002	0.165 / 0.003	0.167 / 0.003	0.160 / 0.003
5.49	0.164 / 0.004	0.170 / 0.005	0.151 / 0.003	0.143 / 0.003
6.34	0.227 / 0.004	0.216 / 0.003	0.218 / 0.006	0.217 / 0.004

(a) grasp segment

ID	A.C.	G.C.	J.S.	N.A.
2.12	0.080 / 0.002	0.083 / 0.003	0.077 / 0.004	0.076 / 0.002
3.17	0.099 / 0.003	0.097 / 0.002	0.096 / 0.002	0.099 / 0.002
4.30	0.137 / 0.002	0.130 / 0.002	0.125 / 0.002	0.126 / 0.002
5.49	0.114 / 0.002	0.120 / 0.003	0.109 / 0.002	0.113 / 0.002
6.34	0.169 / 0.002	0.174 / 0.002	0.170 / 0.003	0.173 / 0.003

(b) Ballistic Move Phase

ID	A.C.	G.C.	J.S.	N.A.
2.12	0.049 / 0.003	0.049 / 0.002	0.048 / 0.003	0.044 / 0.003
3.17	0.061 / 0.004	0.060 / 0.004	0.048 / 0.003	0.050 / 0.003
4.30	0.047 / 0.002	0.050 / 0.003	0.056 / 0.004	0.047 / 0.002
5.49	0.070 / 0.005	0.069 / 0.004	0.057 / 0.004	0.048 / 0.003
6.34	0.079 / 0.005	0.060 / 0.003	0.069 / 0.006	0.063 / 0.004

(c) placement segment

Table 7.5: Path Length Mean / Standard Deviation (m).

For each index of difficulty (ID) For all assistance functions: Attractive Cylinders, Golf Course, Jet Stream, No Assistance.

7.6.4 Excursions From Target

Another interesting metric is how much the operator deviated from their intended target, called here "excursions from target". An excursion is defined as a trajectory performed during the placement segment wherein the robot end-effector moves to more than 1 cm away from the target, defined by the final grasp-release location. Two measures are reported here: the number of excursions, and the summed path-length of all excursions.

The number of excursions for the four modes were AC:182, GC:110, JS:140, NA:163. Total path length for the four modes was AC:6.4, GC:2.9, JS:4.0, NA:4.1. Table 7.7 reports the excursions

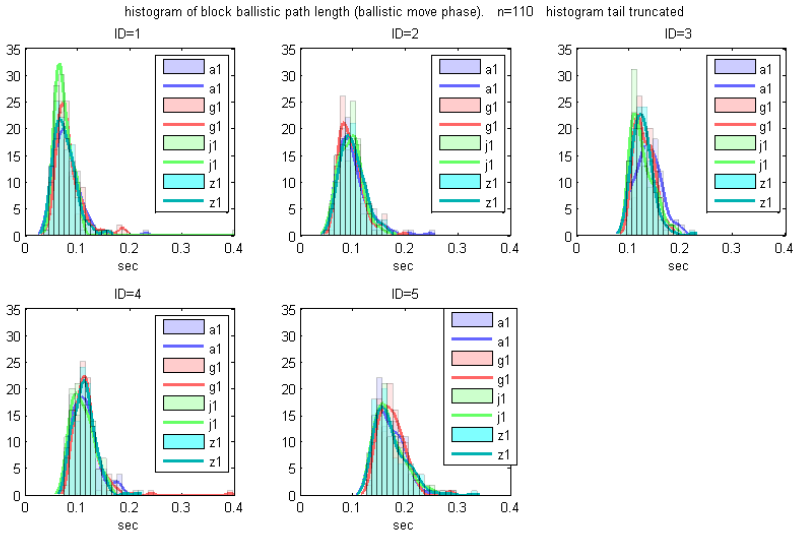


Figure 7.15: Ballistic movement path length (m) histogram. One plot for each ID, lines represent assistance functions a1=attractive cyl., g1=Golf Course, j1=Jetstream, z1=no assistance.

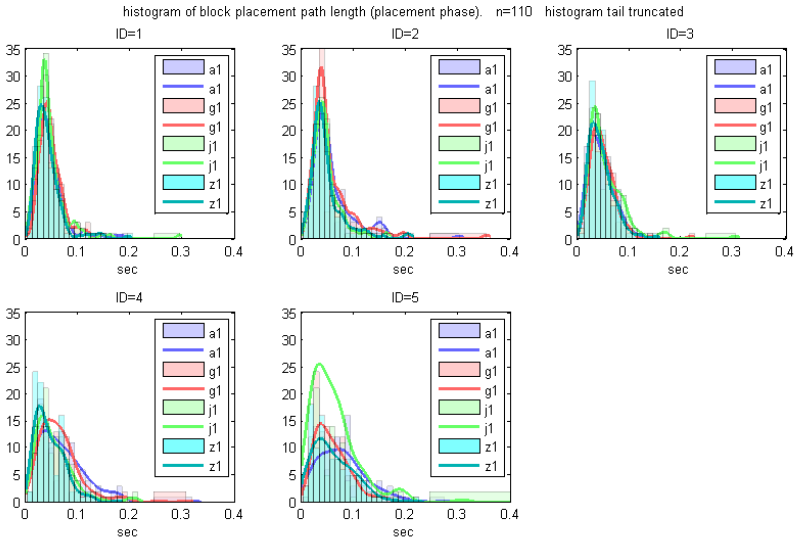


Figure 7.16: Placement segment path length (m) histogram. One plot for each ID, lines represent assistance functions a1=attractive cyl., g1=Golf Course, j1=Jetstream, z1=no assistance.

	Estimate	e^{Est}	%	p
A.C.	0.034	1.03	3%	<0.001
J.S.	0.073	1.08	8%	0.264
G.C.	0.153	1.17	17%	<0.001

(a) grasp segment

	Estimate	e^{Est}	%	p
A.C.	0.018	1.02	2%	0.147
J.S.	-0.020	0.98	-2%	0.116
G.C.	0.030	1.03	3%	0.018

(b) Ballistic Movement Phase

	Estimate	e^{Est}	%	p
A.C.	0.181	1.20	20%	<0.001
J.S.	0.073	1.08	8%	0.034
G.C.	0.153	1.17	17%	<0.001

(c) placement segment

Table 7.6: ANOVA between path lengths. Each AF is compared to the no-assistance case. Columns are: Estimated difference of log-transformed means, exponentiated estimate, percent difference of means, P-value significance.

measures and a percent difference compared to the NA case.

7.7 Conclusions: Fitts' Law Task

7.7.1 Interpretation of linear fit

The Fitts' Law regression line describes the relationship between movement time (MT) and index of difficulty (ID), and the R^2 statistic reports how much predictive power the line provides.

Compared to prior studies [118] the R^2 coefficients for two of the four modes are low, with coefficients of .79 (NA) and .9 (GC). The AC mode had the best fit with R^2 coefficient of .94, while JS had a R^2=.90.

Low R^2 of the Fitts Law fit for any of the treatments indicates that MT is only partly explained by ID. In this case there are additional confounding factors: the three dimensionality of the task, coupled with a lack of stereo vision may cause variation in MT. Also, the task has a rotational component, i.e., the block must be aligned with the peg, and this may be an interfering factor. These additional requirements are the same for all treatments, so it is unclear why there would be a difference in R^2.

Nonetheless, the range of correlations indicates that ID has some predictive power for movement

Table 7.7: Excursions From Target. % Difference is the change in excursion metric compared to the No Assisstance mode.

	A.C.	G.C.	J.S.	N.A.
# Excursions	182	110	140	163
% Difference	11%	-33%	-14%	—
Total pathlength	6.402	2.92	4.006	4.125
% Difference	55%	-29%	-3%	—

time, or in other words, given a subject's baseline performance his/her performance on one ID performance tasks with other IDs may be inferred with some degree of confidence.

Also, it is interesting to compare the IP determined here with results from other past experiments. The range of IPs reported in this experiment for the four assistance modes is 2.01 - 2.56. Fitts' original paper reports IP for the human hand in the range of 8.9-12.9. MacKenzie et al. studied computer input devices and found IPs of 3.3 (Trackball), 4.5 (mouse) and 4.9 (tablet) [73].

Subjectively, from observations of subjects, this comparison seems about right. Compared to moving objects with the hand, rate of task performance was on the order of one fourth or one sixth. Compared to clicking items on screen, the rate of task performance was on the order of half. Although, 2-D mouse clicking is much easier than 6-D peg-block transfer.

7.7.2 The "Teleport Effect"

The first surprising aspect of the data is that the assistance functions do not demonstrate a resounding improvement in grasp time. Grasp time is the measure of task completion, i.e., from the time the block is picked up from its source location to the time it is placed at its final destination. This is especially surprising, since subjective observation of subjects shows them moving quickly between the pick-up to placement locations.

Segmentation of the data into the 'move' phase and 'place' phase reveals an interesting phenomenon, which can be described as the "teleport effect".

Using Jetstream and Attractive Cylinders assistance modes users show improvement in ballistic

movement time of -11% and -16% respectively compared to unassissted operation. Meanwhile the placement time with Attractive Cylinders assistance was +13% (slower) and for Jetstream by +2%. For grasp time these two modes show +1% and -2% change from no assistance, respectively and these are not statistically significant. This illustrates that marked improvement in the ballistic movement phase is counteracted by increased time in the placement phase, resulting no overall gains.

Similar results are observed in the path length data. While Attractive Cylinders and Jetstream modes evidenced essentially identical ballistic move phase path lengths (+2%, P=.14 and -2%, P=.11 respectively), they each showed markedly longer placement phase paths (+20%, $<$.001 and +8%, P=.034). Again illustrating that despite a faster time-to-target, the user experienced difficulty in finishing the task once the goal area was reached.

Why? The data from these experiments clearly show a trade-off between ballistic movement performance and placement performance. For future assistance function design, it's important to look at why the effect occurs, and what mitigating strategies can be adopted.

The Attractive Cylinders function is an attractive sinusoidal force-field. The haptic rendering function (see Section 7.3.2) includes only the attractive force, but no velocity-dependent damping factor. Therefore the fixture is adding energy to the system that must be damped out by the user and natural mechanical damping. This is a likely contributor to the teleport effect in that case.

Sources of error from the Jetstream function on the other hand, are harder to identify. However, the transition from fast ballistic movement into the dead-zone around the target is a likely source of confusion or disorientation, since automatic motion ends abruptly upon entering that zone. The user is required to quickly start a new movement: identify the new location after a fast move, decide a new path to the peg, begin the necessary movement to the peg, and correct for errors.

A similar result is illustrated in Figure 7.17 from [134]. There, a mismatch in system state between the autonomous and manual bi-lateral controller causes discontinuities in force. During object manipulation this could lead to objects being dropped, obstacles being hit, or other catastrophic events.

The teleport effect is the performance detriment incurred when switching from computer assisted control to pure manual control. This leads to the following problem formulation for computer

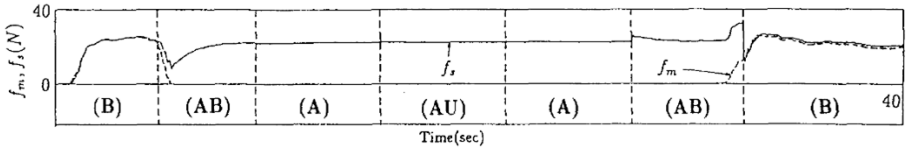


Figure 7: Experimental result: Changing operation modes (Experiment 1)

Figure 7.17: This figure from [134] shows the force profile of a ramp-up and hold force-control task. Control is exchanged and combined between bilateral control (B), autonomous (A), and unilateral (U) control. Note the sudden force discontinuities during (AB) control-exchange phases.

assistance functions: there appears to be a need for state estimation and matching before transitioning between manual and assisted operation.

In addition, there is a human factors consideration that cannot be overlooked. Each of the assistance function geometries is defined relative to the vertically-oriented pegs. However, with the 2-D video at an angle of 35° down from horizontal, users tend to follow a trajectory that moves along this diagonal. Following this curved path, rather than a straight line, allows the user to visually align the block with its intended peg *during* the movement to target. As a result, the user has already made path adjustments necessary to place the peg, and does not need to recalculate a new path and perform new visual corrections to make the block-peg alignment. Use of 3D vision may help with this problem, but nonetheless it illustrates the need for the assistance functions to take human factors into account.

Therefore the following guideline should be followed when designing computer assistance functions: assistance functions, to the extent possible, should be designed to cooperate with the natural ergonomics of human task performance.

As an example, further performance gains might be made by aligning the axes of the assistance function geometric constraints with the user's line of sight, instead of the more obvious vertical arrangement based purely on task geometry. There may be some trade-off between task requirements and human factors requirements, i.e., the Golf Course mode may provide a safer path by requiring an approach from above, while a diagonal approach might be easier and more efficient. However, to the extent possible the functions should be designed to accommodate the way humans naturally

operate.

Finally, computer assisted manipulation is not something most people do in their day-to-day life. Therefore, it's possible that experience with the AF would increase users' ability to leverage the assistance they afford. This question is left for future work.

7.7.3 Excursions and Safety

In some operations, acting outside of a target area could incur damage, and therefore "excursions" can be said to inversely correlate to safety.

By this measure, Golf Course assistance shows strong improvement over unassisted operation with 33% fewer excursions and 29% less distance traveled outside the target area (sum over all trials). This is a reasonable conclusion since the Golf Course is designed to restrict users' motion to intended areas of operation.

The Golf Course is not a "hard" virtual fixture in the sense that the robot will follow users who overcome the force of the haptic assistance function, and thus excursions are still permitted. Whether or not users should be allowed this freedom is a design choice. From the data available in this experiment it is impossible to say whether the excursions that occurred with the Golf Course assistance were accidental or purposeful and necessary to complete the operation.

Jetstream assistance led to slightly decreased excursions (14% fewer and 3% decreased path length), probably due to the fact that it brings the user directly to the destination. On the other hand, disorientation due to the teleport effect and parallax error in viewing angle are likely responsible for the excursions.

The Attractive Cylinders mode showed worse performance in excursions. This mode entailed 11% more excursions and a 55% increase in the path length traveled outside of the target region. This is most likely due to the absence of damping in the haptic rendering function. The Attractive Cylinders assistance function could possibly be improved by applying control theoretic principles to optimize the assistance offered in reaching a goal. In some sense, the assistance function has as a goal the controlled movement of the robot end effector to the target area. To achieve this the function applies control effort of force on the human operator's hand.

This leads to the following problem formulation: what optimality criteria can be applied to

maximize task performance? How can traditional control theory be applied in assisting human movement?

For example, excursions can be minimized by optimizing virtual fixture parameters for maximum overshoot; with a dynamical model of the master, slave, and human hand dynamics (under simplifying assumptions) an optimal controller could be designed to move the end effector towards a goal. It's unclear how this approach would interact with human decision making.

7.8 Results: Telerobotic FLS

7.8.1 Separate Task Components

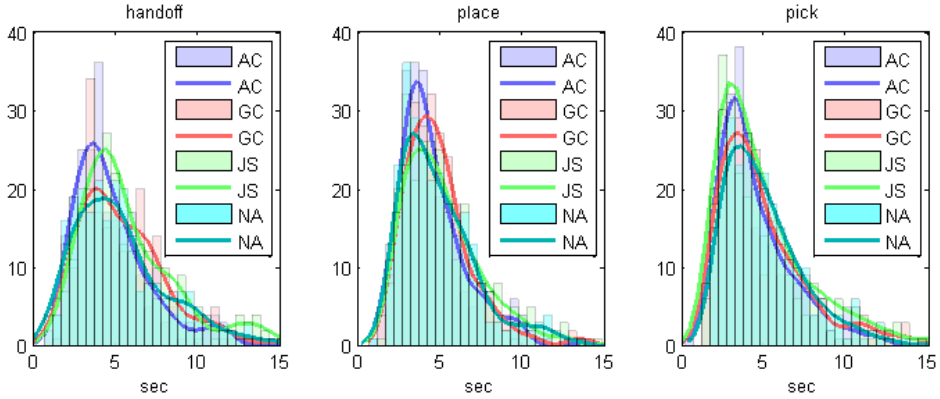
The TFLS task performance is reported here in terms of block transfer time and block transfer path length for each of the three task components: pick-up, handoff and placement.

Figure 7.18a shows histograms of phase times for each of the three phases. Tables 7.8a and 7.8b report mean and standard deviation of movement time and path length respectively. Attractive cylinders assistance shows lower mean movement times, while Jetstream assistance entailed longer time on average.

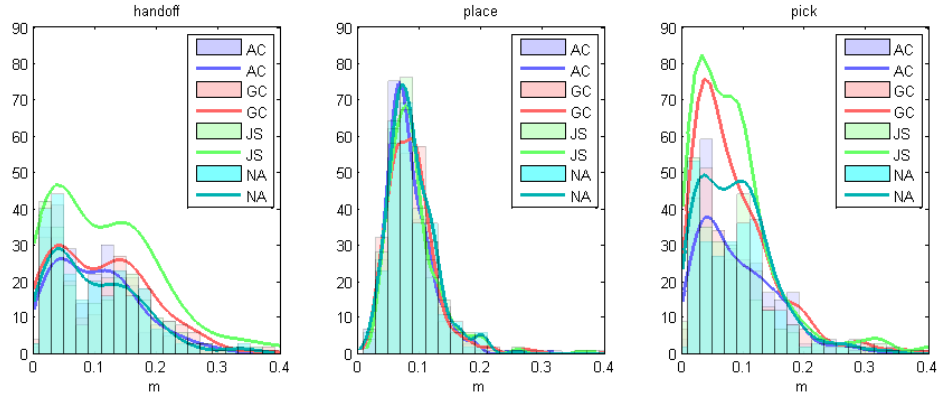
The significance of the means is reported in Table 7.8c and 7.8d. The most notable effect is with Attractive Cylinders assistance, and in all phases this mode appears to have the effect of improving task completion time. P-values for statstical significance of each phase are pick-up: $p<.1$, handoff: $p<.1$, place $p<.005$. The first two reflect significance at the 0.1 level, which is not a strong indicator, while the third is significant at the 0.01 level which is quite a strong indicator.

7.8.2 Total Task Completion

Here, performance measures are reported for the entire task, i.e. all twelve block transfers together. Mean and standard deviation of total time and path length taken for the task are reported in Table 7.9. Again we see that AC was faster by an average of 18.8 sec, which is significant at $p<.05$. Also, the Jetstream mode was slower by about 18.9 sec, also significant at the $p<.05$ level.



(a) Time



(b) Path Length

Figure 7.18: Histogram of phase completion times for each of the three action phases of the FLS task.

	Pick	Handoff	Place
AC	4.93/2.87	4.60/2.11	4.57/2.54
GC	6.02/4.15	4.99/2.43	5.21/3.83
JS	6.57/4.96	5.32/2.72	5.50/4.80
NA	5.54/3.48	4.98/2.43	5.21/3.19

(a) Mean / StDev of Times (s)

	Pick	Handoff	Place
AC	0.1050/0.072	0.084/0.037	0.091/0.068
GC	0.1169/0.093	0.089/0.044	0.089/0.086
JS	0.1291/0.127	0.091/0.046	0.092/0.110
NA	0.1008/0.073	0.090/0.048	0.090/0.075

(b) Mean / StDev of Path Length (m)

Pick

	Estimate	e^{Est}	%	p
AC	-0.090	0.9	-9%	0.073
GC	0.079	1.0	8%	0.115
JS	0.147	1.1	16%	0.003

Handoff

	Estimate	e^{Est}	%	
AC	-0.066	0.9	-6%	0.08
GC	0.012	1.0	1%	0.74
JS	0.064	1.0	7%	0.093

Place

	Estimate	e^{Est}	%	p
AC	-0.132	0.8	-12%	0.004
GC	-0.043	0.9	-4%	0.351
JS	-0.025	0.9	-3%	0.580

(c) ANOVA of Times

Pick

	Estimate	e^{Est}	%	p
AC	0.068	1.07	7%	0.406
GC	0.106	1.11	11%	0.194
JS	0.121	1.13	13%	0.118

Handoff

	Estimate	e^{Est}	%	p
AC	-0.062	0.94	-6%	0.101
GC	-0.0069	0.99	-1%	0.855
JS	0.0168	1.03	2%	0.655

Place

	Estimate	e^{Est}	%	p
AC	0.028	1.03	3%	0.702
GC	-0.049	0.95	-5%	0.507
JS	-0.11	0.90	-10%	0.139

(d) ANOVA of Path Lengths

Table 7.8: (a,b) Mean/Standard deviations of each phase for each assistance function. (c,d) ANOVA results comparing assistance modes for each phase of the TFLS block transfer task. Each assistance function is compared to the no assistance case. Columns are estimated difference of log-transformed means, exponentiated estimate, percent difference of means, P-value significance.

		AC	GC	JS	NA
Time	Mean / StDev (sec)	164.4 / 37.9	188.7 / 60.8	202.2 / 57.5	183.3 / 53.9
	Estimate (sec)	-18.8	5.3	18.9	—
	p-value	0.046	0.565	0.045	—
Path length	Mean / StDev (m)	3.266 / 0.421	3.440 / 0.589	3.625 / 1.081	3.286 / 0.640
	Estimate (m)	-0.019	0.153	0.339	—
	p-value	0.905	0.359	0.045	—

Table 7.9: Time and path length metrics of the entire TFLS task (twelve blocks), n=22. The mean and standard deviation are reported, also the ANOVA result giving the estimated difference in means, and the significance (P-value).

7.9 Conclusions: TFLS

Overall, the TFLS task performance was improved by attractive cylinder assistance, and hindered by Jetstream assistance.

Interestingly, whereas the Jetstream assistance contributed the most help in the FL task, it was actually a detriment in the TFLS task. Jetstream assistance increased time required per subtask by an average of 7%, and overall task performance by about 19 seconds. There was also a significant increase in total path length, an average of 33cm ($p<.05$).

This change in effect is probably due to the relatively short ballistic movements required by the TFLS task (2cm) compared to the FL task (3-12 cm). With the shorter ballistic paths, it's possible that the teleport effect causes greater disorientation. In addition, the Jetstream mode imposes greater motion restrictions, since the automation feature automatically moves to a target, even if the user tries to move away. Therefore, with a block pushed out of alignment by an awkward grasp, for example, the user is forced to compensate by changing orientation instead of repositioning. This additional rotational requirement takes time.

In contrast, attractive cylinder assistance improved task completion by around 10% on individual movement phases, and about 19 seconds overall. Path length metrics, however, were not so clearcut,

so it appears the AC mode helps by allowing the user to move faster, rather than with more efficient paths.

Once again Golf Course mode showed slightly longer performance time and path length. Unfortunately due to short movement times and bi-manual operation, it was not possible to segment the users' task performance into shorter ballistic and targeting phase movements, and examine the excursion (and thereby safety) performance of the Golf Course.

7.10 Summary and Future Work

This chapter has examined the effect of a few types of computer assistance functions on human controlled teleoperation performance of two different tasks. From these example cases, the following conclusions are drawn:

Teleport and Handoff There appears to be a need for state estimation and matching before transitioning between manual and assisted operation. Specifically, when trading from a condition with a strong effect to a pure manual mode, a performance detriment is observed in terms of time taken to complete a subsequent action. Some method for ameliorating this impact is necessary.

Human Centered Assistance Function Design Computer assistance functions should take into account human task performance factors. From the results of the present experiment, it is hypothesized that greater improvements may be made by aligning the AF geometries with the humans' line of sight, rather purely aligning them with the task geometry. This is somewhat counterintuitive, since a resultant path might be sub-optimal.

Design of Optimal Assistance Functions Force feedback virtual fixtures can be viewed as position controllers to control the human hand. A control theoretic analysis of the problem might lead to improvements in time and accuracy. However, this will entail a trade-off between controlling the user, and allowing the user to be in control, and if a task is that well defined it might be more suitable for automation.

Integration With More Realistic Tasks TFLS results showed that the AC assistance mode improved task performance. Therefore, further testing should be performed to illustrate how this as-

sistance could be added into existing teleoperation systems applications.

This work explores a wide but shallow portion of the complete AF design space. There are a great many alternative ways the presented features might be changed. A simple example is that the spring, damper, and acceleration parameters could be tuned for optimal performance, possibly using an online estimator that adapts fixture parameters to each user. As a further example, further assumptions about "safe operation" areas would entail different partitioning of the automation workspace of the Jetstream function, allowing faster motion and slowing the user down as the robot approached a target.

The AF presented here are pre-programmed and human controlled. This is sufficient to test the features in the lab, but for application to real-world scenarios the next step should be to use sensing such as computer vision and automated task analysis to automatically create the AF. Furthermore, some kind of human intention recognition is necessary to make sure the assistance functions are cooperating with the human's objectives.

Ultimately telerobotics allows us to use robots to perform tasks we wouldn't otherwise be capable of, e.g. dexterous minimally invasive surgery, explosive ordinance disposal, or deep sea operation. The computer assistance methods described here show ways of improving our capability to do these high-value tasks, and their implementation will have immediate benefit to a variety of critical applications.

Chapter 8

CONCLUSIONS AND FUTURE WORK

This thesis has presented new work in teleoperation, directly and indirectly contributing to advances in human-machine cooperative control of teleoperation robotics.

Cooperative control augments human capabilities with robotic ones, with a goal of reaching parity with human operation or possibly even super-human levels of task performance and safety. Fulfilling that promise will dramatically effect existing applications and provide benefit to a wide variety of tasks being performed by humans using robots today.

This work has focused, where possible, on applications in telesurgery where increased performance can directly improve quality of life in patients; for example, better nerve sparing in radical prostatectomy will reduce rates of incontinence and sexual dysfunction. At the same time, telerobotics is used in many other high-value applications from outer space to under-sea, all of which will benefit from improved computer-assisted operation.

A major part of this thesis has discussed new platforms for telerobotics research: standards for interconnecting telerobotics master and slave systems, and an open-source platform for telesurgery research. These are fundamental enabling technologies for developing telerobotics technology in the future, towards which this thesis contributes.

8.1 Internet Standards For Teleoperation

Teleoperation over the Internet will bring about ubiquitous telepresence robotics. As robots proliferate, a standard like the Interoperable Telerobotics Protocol (ITP) in Chapter 5 will enable that to happen. The work presented here proves by demonstration that a stateless protocol using Cartesian space position and orientation commands is an effective and easy way to interconnect heterogeneous master-slave systems, paving the way for a robust standard implementation.

The ITP is a simple data interface and as such is only part of the necessary protocol stack for Internet teleoperation. An additional protocol layer is necessary for presence, session management,

and negotiation of capabilities. A likely candidate is the Session Initiation Protocol that was explored in depth in [55]. Other likely candidates are Jabber/XMPP or Skype. Also, securing the telerobotics communications is absolutely vital, but, again, this requires work in general Internet networking security which is outside the scope of this thesis.

Work in Chapter 5 demonstrated new paradigms for telerobotics control and training. The fastest block transfer times were performed using an exoskeleton master- a human interface technology not typically found in telesurgery. Future engineering in telerobotics should use the concept of interoperability to experiment with the wide variety of existing master robots as controllers for telesurgery slave systems.

Open Platforms for Telerobotics and Telesurgery Research The Raven II robot is driving openness and innovation in surgical robotics research, and is a new standard for surgical teleoperation research. As an open-source telesurgery research platform, Raven II gives researchers a testbed for new techniques and ideas.

Continued integration of man-machine control and vision guided operation will transform Raven II from a teleoperation system to a mixed or shared autonomy robot. As these technologies mature, so should the safety and reliability of the robot. Eventually Raven II should be reliable enough to move from laboratory trials to animal and human trials, and finally to human therapeutic application.

A major question regards software reliability and validation. Currently, Raven II makes extensive use of the open source Robot Operating System (ROS) middleware. However, ROS provides no guarantees regarding reliability, and bringing ROS up to a level of safety necessary for FDA approval, for example, would require extensive work.

An alternative path from laboratory results to clinical application is: development and proof on the Raven II, and clinical application on an FDA approved robot. This would require the significant step of porting a potentially large, complex code-base to another robot.

Unpublished work with the Raven II has required additional sensors; encoders co-located with the joints rather than the motors, computer vision sensors, tool mounted force sensors, and tool-carriage mounted actuators. Future development of an open robotic research platform might include power and data connections for these add-ons.

Fundamentally, the Raven II is built on a conventional minimally invasive surgery (MIS) paradigm

of long, stiff tool shafts inserted through the abdominal wall with dexterous wrists. Future work will incorporate high degree of freedom tools providing a wider range of motion, possibly with redundant actuation for increased range of motion and access. More specialized tools, as are already available for the daVinci, will target individual procedures, reducing the complexity of some operations. Finally, new robotic designs will facilitate single-port surgery while maintaining existing manipulation capabilities.

The robotic revolution in surgery has begun, and the Raven II is part of the proliferation of technology and expertise.

Virtual Fixtures This work has illustrated the use of virtual fixtures (VF) for object handling and manipulation. Results in Chapter 7 illustrated when and how virtual fixture assistance can help or hinder manual operation, and offered several guidelines for improving VF design and use.

The study used a few VF designs, a guidance and a forbidden region design, to tease out principles of VF engineering. However, the wide design space for virtual fixtures was only sparsely explored. The guidance virtual fixture (Attractive Cylinders), for example, used an attractive force region to guide a user towards a goal. Another approach would be to calculate an optimal path to the goal, and constrain robot motion to that path. An extensive study is necessary to find the most effective designs. Results from the present work will inform that process.

One thing that would help considerably, is a virtual fixture rendering library. This would certainly be similar to existing haptics libraries like Chai3D [24] and OpenHaptics [48], but with added capabilities for sensing and environment modeling, and integration with a slave robot.

Significant work in VFs will involve human intention recognition. The presented work is controlled entirely by pre-programming and in real-time by the experimenter. Human intention recognition is, in itself, an active area of research, particularly in the realm of human and robots cooperatively acting in the same space [18, 45]. In telerobotics, this capability will be even more aptly applied, as the human's motion gives the machine many clues about what predictions to make. With robotic recognition of human intention for, say, grasping a block, the machine intelligence will be able to greatly assist with the task.

Finally, real-time environment sensing is necessary to generate and update virtual fixtures during operation. Future work in this direction is discussed previously in Chapter 4.

At the end of the day, pure-manual operation has significant drawbacks in terms or proprioception, sensing, and actuation. These drawbacks must be overcome to allow us to perform the types of tasks we want to do. Virtual fixtures are an important way this can be accomplished.

Mixed Autonomous Telerobotics This thesis has introduced a new type of mixed autonomy assistance based on a spatially varying autonomy function. In a way, this is the logical evolution of virtual fixture, i.e., when the computer is capable of assisting the human operator with a task, it may be able to perform the same task semi-autonomously. Furthermore, just as virtual fixtures are inherently geometric, the Jetstream is also based on the geometry of a virtual environment.

The new mixed autonomy function introduced in Chapter 6, the "Lendvay Jetstream", is demonstrated with one type of spatial function for determining a desired velocity: distance from environmental contact. Another likely velocity function is based on distance from the target. Also in these experiments, the environment surface was planar. More sophisticated geometry might be sensed using a device like the Kinect or modeled from prior information such as a CAD model.

One drawback of the implemented Jetstream feature was limitation to positioning assistance. The addition of orientation assistance might also be beneficial. Orientation was not included in the experimental implementation and tests, since the orientation of the cylindrical block was not known. With some additional sensing and object recognition, significant new orientation capabilities may be added.

Many aspects of VF design and development also apply to mixed autonomy telerobotics. Improved environment sensing, intention recognition, and assistance function design are necessary for effective application of mixed autonomous telerobotics.

In addition, a very important consideration in shared telerobotic control is matching system state between the autonomous and human controller when transitioning between individual and shared control. Methods must be found to match the system state (velocity, acceleration, force, and virtual potential energy) when switching between control modes (manual, assisted, autonomous).

New Dimensions in Human-Machine Collaborative Telerobotics Virtual fixtures and shared autonomy represent promising technologies that have been proven in this and other lab-based scenarios to help with basic task performance. Now, however, the technology is almost ready to make a

quantum leap into practical use. This will require several technical advances: accurate environment modeling, task modeling, and human intention recognition.

The work presented here has demonstrated new capabilities for telerobotics. This research points the way towards the development, deployment and usefulness of new robots, and demonstrates improvements on existing robotic capabilities that are currently used in critical, real-world applications.

BIBLIOGRAPHY

[1] Joint Architecture for Unmanned Systems (JAUS) Reference Architecture, Version 3.0, http://www.jauswg.org/.

[2] Microsoft Robotics Studio. http://msdn.microsoft.com/robotics.

[3] http://en.wikipedia.org/wiki/Statistical_power.

[4] Daniel Aarno, Staffan Ekvall, and Danica Kragic. Adaptive virtual fixtures for machine-assisted teleoperation tasks. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1139–1144. IEEE, 2005.

[5] Daniel Aarno and Danica Kragic. Layered hmm for motion intention recognition. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5130–5135. IEEE, 2006.

[6] Jake Abbott, Panadda Marayong, and Allison Okamura. Haptic virtual fixtures for robot-assisted manipulation. In *Robotics Research*, volume 28 of *Springer Tracts in Advanced Robotics*, pages 49–64. Springer, 2007.

[7] Jake J Abbott. *Virtual fixtures for bilateral telemanipulation*. PhD thesis, Johns Hopkins University, 2006.

[8] Jake J Abbott and Allison M Okamura. Virtual fixture architectures for telemanipulation. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 2798–2805. IEEE, 2003.

[9] J. Accot and S. Zhai. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 295–302. ACM, 1997.

[10] J.R. Adler Jr, SD Chang, MJ Murphy, J. Doty, P. Geis, and SL Hancock. The Cyberknife: a frameless robotic system for radiosurgery. *Stereotactic and functional neurosurgery*, 69(1-4 Pt 2):124, 1997.

[11] Mehdi Ammi and Antoine Ferreira. Multimodal virtual fixtures and metaphors for dexterous telemicromanipulation. In *Proceedings of EuroHaptics*, pages 433–439, 2006.

[12] Mehran Anvari. Establishment of the worlds first telerobotic remote surgical service. *Annals of Surgery*, 241(3), March 2005.

[13] Mehran Anvari. Remote telepresence surgery: the canadian experience. *Surgical Endoscopy*, 21(4):537 – 541, 04 2007.

[14] Antonia Pérez Arias, Henning P Eberhardt, Florian Pfaff, and Uwe D Hanebeck. The plen-haptic guidance function for intuitive navigation in extended range telepresence scenarios. In *World Haptics Conference (WHC), 2011 IEEE*, pages 475–480. IEEE, 2011.

[15] W.L. Bargar, A. Bauer, and M. Börner. Primary and revision total hip replacement using the Robodoc (R) system. *Clinical orthopaedics and related research*, 354:82, 1998.

[16] Karlin Bark, William McMahan, Austin Remington, Jamie Gewirtz, Alexei Wedmid, David I Lee, and Katherine J Kuchenbecker. In vivo validation of a system for haptic feedback of tool vibrations in robotic surgery. *Surgical endoscopy*, 27(2):656–664, 2013.

[17] C. Basdogan and M.A. Srinivasan. Haptic rendering in virtual environments. *Handbook of virtual environments*, pages 117–134, 2002.

[18] Andrea Bauer, Dirk Wollherr, and Martin Buss. Human–robot collaboration: a survey. *International Journal of Humanoid Robotics*, 5(01):47–66, 2008.

[19] Benjamin Bayart, Abdelhamid Drif, Abderrahmane Kheddar, and Jean-Yves Didier. Visuo-haptic blending applied to a tele-touch-diagnosis application. In *Virtual Reality*, pages 617–626. Springer, 2007.

[20] B.C. Becker, R.A. MacLachlan, L.A. Lobes, G.D. Hager, and C.N. Riviere. Vision-based control of a handheld surgical micromanipulator with virtual fixtures. *Robotics, IEEE Transactions on*, 29(3):674–683, 2013.

[21] Alessandro Bettini, Panadda Marayong, Samuel Lang, Allison M Okamura, and Gregory D Hager. Vision-assisted control for manipulation using virtual fixtures. *Robotics, IEEE Transactions on*, 20(6):953–966, 2004.

[22] K. Bumm, J. Wurm, J. Rachinger, T. Dannenmann, C. Bohr, R. Fahlbusch, H. Iro, and C. Nimsky. An automated robotic approach with redundant navigation for minimal invasive extended transsphenoidal skull base surgery. *Minimally Invasive Neurosurgery*, 48(3):159–164, 2005.

[23] Raúl A Castillo-Cruces and Jürgen Wahrburg. Virtual fixtures with autonomous error compensation for human–robot cooperative tasks. *Robotica*, 28(2):267, 2010.

[24] F Conti, F Barbagli, D Morris, and C. Sewell. "chai 3d: An open-source library for the rapid development of haptic scenes. In *World Haptics Conference (WHC), 2005 IEEE*. IEEE, 2005.

[25] C.R. Doarn, M. Anvari, T. Low, and T.J. Broderick. Evaluation of teleoperated surgical robots in an enclosed undersea environment. *Telemedicine and e-Health*, 15(4):325–335, 2009.

[26] J. Dosher. Detection thresholds and performance gains for small haptic effects. Master's thesis, Citeseer, 2001.

[27] J. Dosher and B. Hannaford. Human interaction with small haptic effects. *PRESENCE*, 14:329–344, June 2005.

[28] W.R. Ferrell and T.B. Sheridan. Supervisory control of remote manipulation. *Spectrum, IEEE*, 4(10):81–88, 1967.

[29] P.M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1954.

[30] Michel Franken, Stefano Stramigioli, Sarthak Misra, Cristian Secchi, and Alessandro Macchelli. Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency. *Robotics, IEEE Transactions on*, 27(4):741–756, 2011.

[31] B. Gerkey, R.T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, 2003.

[32] Ali Ghanbari, Hamid Abdi, Ben Horan, Saeid Nahavandi, Xiaoqi Chen, and Wenhui Wang. Haptic guidance for microrobotic intracellular injection. In *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*, pages 162–167. IEEE, 2010.

[33] Tricia L Gibo, Lawton N Verner, David D Yuh, and Allison M Okamura. Design considerations and human-machine performance of moving virtual fixtures. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 671–676. IEEE, 2009.

[34] Deanna Glassman. Raven surgical robot training in preparation for da vinci use: a randomized prospective trial. In *Medicine Meets Virtual Reality 21*, 2014.

[35] K. Goldberg, S. Gentner, C. Sutter, and J. Wiegley. The mercury project: A feasibility study for internet robots. *IEEE Robotics & Automation Magazine*, 7(1):35–40, 2000.

[36] W.B. Griffin, W.R. Provancher, and M.R. Cutkosky. Feedback strategies for telemanipulation with shared control of object handling forces. *Presence: Teleoperators & Virtual Environments*, 14(6):720–731, 2005.

[37] G. S. Guthart and NS J. K. Salisbury. The intuitive(tm) telesurgery system: overview and application.

[38] B. Hannaford, J. Hewitt, T. Maneewarn, S. Venema, M. Appleby, and R. Ehresman. Telerobotic remote handling of protein crystals. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[39] B. Hannaford and A. Okamura. Haptics. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics (Winner of two 2008 PROSE Awards, American Association of Publishers)*. Springer, Berlin, Heidelberg, 2008.

[40] B. Hannaford, L. Wood, D.A. McAffee, and H. Zak. Performance evaluation of a six-axis generalized force-reflecting teleoperator. *Systems, Man and Cybernetics, IEEE Transactions on*, 21(3):620 –633, 1991.

[41] Blake Hannaford. A design framework for teleoperators with kinesthetic feedback. *Robotics and Automation, IEEE Transactions on*, 5(4):426–434, 1989.

[42] Blake Hannaford, Jacob Rosen, Diana W Friedman, Hawkeye King, Phillip Roan, Lei Cheng, Daniel Glozman, Ji Ma, Sina Nia Kosari, and Lee White. Raven-ii: an open platform for surgical robotics research. *Biomedical Engineering, IEEE Transactions on*, 60(4):954–959, 2013.

[43] Keyvan Hashtrudi-Zaad and Septimiu E Salcudean. Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation. *Robotics and Automation, IEEE Transactions on*, 18(1):108–114, 2002.

[44] Daan Hennekens, Daniela Constantinescu, and Maarten Steinbuch. Continuous impulsive force controller for forbidden-region virtual fixtures. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2890–2895. IEEE, 2008.

[45] Guy Hoffman and Cynthia Breazeal. Collaboration in human-robot teams. In *Proc. of the AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, USA*, 2004.

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

[47] Gilgueng Hwang, Daniel Aarno, and Hideki Hashimoto. Haptic guidant bilateral teleoperation for single-master multi-slave system. In *Proceedings of Eurohaptics*, number 2006.7, 2006.

[48] Brandon Itkowitz, Josh Handley, and Weihang Zhu. The openhapticsł toolkit: a library for adding 3d touchł navigation and haptics to graphics applications. In *Eurohaptics Conference, 2005*, pages 590–591. IEEE, 2005.

[49] M. Jakopec, F. Rodriguez y Baena, S.J. Harris, P. Gomes, J. Cobb, and B.L. Davies. The hands-on orthopaedic robot "acrobot": Early clinical trials of total knee replacement surgery. *Robotics and Automation, IEEE Transactions on*, 19(5):902 911, oct. 2003.

[50] M.Y. Jung, T. Xia, A. Deguet, R. Kumar, R. Taylor, and P. Kazanzides. A surgical assistant workstation (saw) application for teleoperated surgical robot system. *The MIDAS Journal - Systems and Architectures for Computer Assisted Interventions.*, 2009. http://hdl.handle.net/10380/3079.

[51] Ankur Kapoor and Russell H Taylor. A constrained optimization approach to virtual fixtures for multi-handed tasks. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3401–3406. IEEE, 2008.

[52] Ben Kehoe, Gregory Kahn, Jeffrey Mahler, Jonathan Kim, Alex Lee, Anna Lee, Keisuke Nakagawa, Sachin Patil, W. Douglas Boyd, Pieter Abbeel, and Ken Goldberg. Autonomous multilateral debridement with the raven surgical robot. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014.

[53] Ryo Kikuuwe, Naoyuki Takesue, and Hideo Fujimoto. A control framework to generate nonenergy-storing virtual fixtures: use of simulated plasticity. *Robotics, IEEE Transactions on*, 24(4):781–793, 2008.

[54] H. Hawkeye King, Blake Hannaford, Ka-Wai Kwok, Guang-Zhong Yang, Paul Griffiths, Allison Okamura, Ildar Farkhatdinov, Jee-Hwan Ryu, Ganesh Sankaranarayanan, Venkata Arikatla, Kotaro Tadano, Kenji Kawashima, Angelika Peer, Thomas Schau, Martin Buss, Levi Miller, Daniel Glozman, Jacob Rosen, and Thomas Low. Plugfest 2009: Global interoperability in telerobotics and telemedicine. In *International Conference on Robotics and Automation, ICRA 2010*, May 2010.

[55] H.H. King, B. Hannaford, J. Kammerl, and E. Steinbach. Establishing multimodal telepresense sessions using the session initiation protocol (SIP) and advanced haptic codecs. In *Proceedings of North American Haptics Symposium 2010*, March 2010.

[56] H.H. King, T. Low, K. Hufford, and T. Broderick. Acceleration compensation for vehicle based telesurgery on earth or in space. In *Intelligent Robots and Systems, 2008. IROS IEEE/RSJ International Conference on*, pages 1459–1464, Sept. 2008.

[57] D. Kragic, P. Marayong, M. Li, AM Okamura, and GD Hager. Human-machine collaborative systems for microsurgical applications. *The International Journal of Robotics Research*, 24(9):731, 2005.

[58] Alex B Kuang, Shahram Payandeh, Bin Zheng, Frank Henigman, and Christine L MacKenzie. Assembling virtual fixtures for guidance in training environments. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS'04. Proceedings. 12th International Symposium on*, pages 367–374. IEEE, 2004.

[59] Katherine J Kuchenbecker, Jonathan Fiene, and Günter Niemeyer. Improving contact realism through event-based haptic feedback. *Visualization and Computer Graphics, IEEE Transactions on*, 12(2):219–230, 2006.

[60] Katherine J Kuchenbecker, Jamie Gewirtz, William McMahan, Dorsey Standish, Paul Martin, Jonathan Bohren, Pierre J Mendoza, and David I Lee. Verrotouch: high-frequency acceleration feedback for telerobotic surgery. In *Haptics: Generating and Perceiving Tangible Sensations*, pages 189–196. Springer, 2010.

[61] Y.S. Kwoh, J. Hou, E.A. Jonckheere, and S. Hayati. A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *Biomedical Engineering, IEEE Transactions on*, 35(2):153 –160, Feb 1988.

[62] Dale A Lawrence. Designing teleoperator architectures for transparency. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 1406–1411. IEEE, 1992.

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

[64] G. Lee and B. Hannaford. Anisotropies of touch in haptic icon exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, October 2003.

[65] Ming Li, Masaru Ishii, and Russell H Taylor. Spatial motion constraints using virtual fixtures generated by anatomy. *Robotics, IEEE Transactions on*, 23(1):4–19, 2007.

[66] Ming Li, A. Kapoor, and R.H. Taylor. A constrained optimization approach to virtual fixtures. In *Intelligent Robots and Systems, 2008. IROS IEEE/RSJ International Conference on*, pages 1408 – 1413, aug. 2005.

[67] Zhi Li, Daniel Glozman, Dejan Milutinovic, and Jacob Rosen. Maximizing dexterous workspace and optimal port placement of a multi-arm surgical robot. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3394–3399. IEEE, 2011.

[68] Qingping Lin and Chengi Kuo. Virtual tele-operation of underwater robots. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 2, pages 1022–1027. IEEE, 1997.

[69] M. Lum, D. Friedman, J. Rosen, G. Sankaranarayanan, H. King, K. Fodero, R. Leuschke, M. Sinanan, and B. Hannaford. The RAVEN - design and validation of a telesurgery system. *International Journal of Robotics Research*, 28:1183–1197, September 2009.

[70] M.H. Lum, D.C.W. Friedman, H.H. King, T. Broderick, M.N. Sinanan, J. Rosen, and B. Hannaford. Field operation of a surgical robot via airborne wireless radio link. *IEEE Int. Conf. on Field and Service Robotics*, 2007.

[71] M.J.H. Lum, D.C.W. Friedman, G. Sankaranarayanan, H. King, A. Wright, M. Sinanan, T. Lendvay, J. Rosen, and B. Hannaford. Objective assessment of telesurgical robot systems: Telerobotic FLS. In *Proceedings, Medicine Meets Virtual Reality (MMVR)*, Long Beach, CA, 29-Jan — 1-Feb 2008.

[72] M.J.H. Lum, J. Rosen, T.S. Lendvay, M.N. Sinanan, and B. Hannaford. Effect of time delay on telesurgical performance. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[73] I Scott MacKenzie, Abigail Sellen, and William Buxton. A comparison of input devices in elemental pointing and dragging tasks. In *Proceedings of ACM CHI 91 Conference on Human Factors in Computing Systems*, pages 161–166, 1991.

[74] I.S. MacKenzie. A note on the information-theoretic basis for Fitts law. *Journal of Motor Behavior*, 21(3):323–330, 1989.

[75] M. Mahvash, J. Gwilliam, R. Agarwal, B. Vagvolgyi, L.M. Su, DD Yuh, and AM Okamura. Force-feedback surgical teleoperator: Controller design and palpation experiments. In *Haptic interfaces for virtual environment and teleoperator systems, 2008. haptics 2008. symposium on*, pages 465–471, 2008.

[76] Panadda Marayong, Alessandro Bettini, and Allison Okamura. Effect of virtual fixture compliance on human-machine cooperative manipulation. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 2, pages 1089–1095. IEEE, 2002.

[77] Panadda Marayong, Gregory D Hager, and Allison M Okamura. Effect of hand dynamics on virtual fixtures for compliant human-machine interfaces. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006 14th Symposium on*, pages 109–115. IEEE, 2006.

[78] Panadda Marayong, Ming Li, Allison M Okamura, and Gregory D Hager. Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1954–1959. IEEE, 2003.

[79] Jacques Marescaux, J. Leroy, M. Gagner, F. Rubino, D. Mutter, M. Vix, S. E. Butner, and M. K. Smith. Transatlantic robot-assisted telesurgery. *Nature*, 413(27):379–380, September 2001.

[80] MacFarlane Mark, Jacob Rosen, Blake Hannaford, Carlos Pellegrini, and Mika N. Sinanan. Force feedback grasper helps restore the sense of touch in minimally invasive surgery. *Journal of Gastrointestinal Surgery*, 3(5):278–285, 1999.

[81] Hermann Mayer, Istvan Nagy, Alois Knoll, Eva U Braun, Robert Bauernschmitt, and Rdiger Lange. Haptic feedback in a telepresence system for endoscopic heart surgery. *Presence: Teleoperators and Virtual Environments*, 16(5):459–470, 2007.

[82] William McMahan, Jamie Gewirtz, Dorsey Standish, Paul Martin, Jacquelyn A Kunkel, Magalie Lilavois, Alexei Wedmid, David I Lee, and Katherine J Kuchenbecker. Tool contact acceleration feedback for telerobotic surgery. *Haptics, IEEE Transactions on*, 4(3):210–220, 2011.

[83] Manuel Ricardo Moreyra. Wrist with decoupled motion transmission, November 29 2005. US Patent 6,969,385.

[84] George P Mylonas, Ka-Wai Kwok, David RC James, Daniel Leff, Felipe Orihuela-Espina, Ara Darzi, and Guang-Zhong Yang. Gaze-contingent motor channelling, haptic constraints and associated cognitive demand for robotic mis. *Medical image analysis*, 16(3):612–631, 2012.

[85] Nikhil V Navkar, Zhigang Deng, Dipan J Shah, Kostas E Bekris, and Nikolaos V Tsekos. Visual and force-feedback guidance for robot-assisted interventions in the beating heart with real-time mri. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 689–694. IEEE, 2012.

[86] Sina Nia Kosari. *Haptic Virtual Fixtures for Robotic Surgery*. PhD thesis, 2013.

[87] Verena Nitsch, Carolina Passenberg, Angelika Peer, Martin Buss, and Berthold Färber. Assistance functions for collaborative haptic interaction in virtual environments and their effect on performance and user comfort. In *1st International Conference on Applied Bionics and Biomechanics*, 2010.

[88] Jason T Nolin, Paul M Stemniski, and Allison M Okamura. Activation cues and force scaling methods for virtual fixtures. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*, pages 404–409. IEEE, 2003.

[89] R. Oboe and P. Fiorini. A design and control environment for internet-based telerobotics. *The International Journal of Robotics Research*, 17(4):433, 1998.

[90] Allison M. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 5(6):499–208, 2004.

[91] Nicolas Padoy and Gregory D Hager. Human-machine collaborative surgery using learned models. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5285–5292. IEEE, 2011.

[92] Jun Woo Park, Jaesoon Choi, Yongdoo Park, and Kyung Sun. Haptic virtual fixture for robotic cardiac catheter navigation. *Artificial Organs*, 35(11):1127–1131, 2011.

[93] Shinsuk Park, Robert D Howe, and David F Torchiana. Virtual fixtures for robotic cardiac surgery. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2001*, pages 1419–1420. Springer, 2001.

[94] Carolina Passenberg, Raphaela Groten, Angelika Peer, and Martin Buss. Towards real-time haptic assistance adaptation optimizing task performance and human effort. In *World Haptics Conference (WHC), 2011 IEEE*, pages 155–160. IEEE, 2011.

[95] Carolina Passenberg, Angelika Peer, and Martin Buss. A survey of environment-, operator-, and task-adapted controllers for teleoperation systems. *Mechatronics*, 20(7):787–801, 2010.

[96] A. Peer and M. Buss. A new admittance-type haptic interface for bimanual manipulations. In *IEEE/ASME Transactions on Mechatronics*, number 4, pages 416–428, 2008.

[97] A Perez Arias and Uwe D Hanebeck. Wide-area haptic guidance: Taking the user by the hand. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5824–5829. IEEE, 2010.

[98] J.C. Perry and J. Rosen. Design of a 7 degree-of-freedom upper-limb powered exoskeleton. In *Proceedings of the 2006 BioRob Conference*, Pisa, Italy, February 2006.

[99] J.H. Peters, G.M. Fried, L.L. Swanstrom, N.J. Soper, L.F. Sillin, B. Schirmer, and K. Hoffman. Development and validation of a comprehensive program of education and assessment of the basic fundamentals of laparoscopic surgery. *Surgery*, 135(1):21–27, 2004.

[100] Zachary Pezzementi, Allison M Okamura, and Gregory D Hager. Dynamic guidance with pseudoadmittance virtual fixtures. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1761–1767. IEEE, 2007.

[101] Rodolfo Prada and Shahram Payandeh. A study on design and analysis of virtual fixtures for cutting in training environments. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 375–380. IEEE, 2005.

[102] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 2009.

[103] Jing Ren, Rajni V Patel, Kenneth A McIsaac, Gerard Guiraudon, and Terry M Peters. Dynamic 3-d virtual fixtures for minimally invasive beating heart procedures. *Medical Imaging, IEEE Transactions on*, 27(8):1061–1070, 2008.

[104] J. Rosen, B. Hannaford, and R.M. Satava. *Surgical Robotics: Systems, Applications, and Visions*. Springer Verlag, 2010.

[105] Jacob Rosen, Blake Hannaford, and Richard M Satava. *Surgical Robotics: Systems Applications and Visions*. Springer, 2011.

[106] L.B. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Virtual Reality Annual International Symposium, 1993*, pages 76 –82, sep. 1993.

[107] Diego C Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 345–352. ACM Press/Addison-Wesley Publishing Co., 1997.

[108] F. Ryden, H. J. Chizeck, S. Nia Kosari, H. King, and B. Hannaford. Using kinect and a haptic interface for implementation of real-time virtual fixture. In *Robotics Sciences and Systems, Workshop on RGB-D: Advanced Reasoning with Depth Cameras,*, Los Angeles,, June 2011.

[109] F. Ryden and H.J. Chizeck. A proxy method for real-time 3-dof haptic rendering of streaming point cloud data. *Haptics, IEEE Transactions on*, 6(3):257–267, July 2013.

[110] Fredrik Rydén, Sina Nia Kosari, and Howard Jay Chizeck. Proxy method for fast haptic rendering from time varying point clouds. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2614–2619. IEEE, 2011.

[111] Kenneth Salisbury, Francois Conti, and Federico Barbagli. Haptic rendering: introductory concepts. *Computer Graphics and Applications, IEEE*, 24(2):24–32, 2004.

[112] G. Sankaranarayanan and B. Hannaford. Comparison of performance of virtual coupling schemes for haptic collaboration using real and emulated internet connections. In *Proceedings of the 1st international conference on Robot communication and coordination*, 2007.

[113] G. Sankaranarayanan, L. Potter, and B. Hannaford. Measurement and simulation of time varying packet delay with applications to networked haptic virtual environments. In *Proceedings of 1st international Conference on Robot Communication and Coordination (Robocom 2007)*, pages 1–8, Athens, Greece, October 2007. IEEE Press, Piscataway, NJ.

[114] Ganesh Sankaranarayanan, H. Hawkeye King, Song-Young Ko, Mitch Lum, Diana Friedman, Jacob Rosen, and Blake Hannaford. Portable surgery master station for mobile robotic telesurgery. In *Proc. of ROBOCOMM*, Athens, Greece, 2007.

[115] T. Sato and S. Hirai. Language-aided robotic teleoperation system (larts) for advanced teleoperation. *Robotics and Automation, IEEE Journal of*, 3(5):476 –481, oct. 1987.

[116] Samuel B Schorr, Zhan Fan Quek, Robert Y Romano, Ilana Nisky, William R Provancher, and Allison M Okamura. Sensory substitution via cutaneous skin stretch feedback. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2341–2346. IEEE, 2013.

[117] Thomas B Sheridan. *Telerobotics, automation, and human supervisory control*. MIT press, 1992.

[118] R.W. Soukoreff and I.S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies*, 61(6):751–789, 2004.

[119] B. Stanczyk and M. Buss. Development of a telerobotic system for exploration ofhazardous environments. In *Intelligent Robots and Systems, 2008. IROS IEEE/RSJ International Conference on*, 2008.

[120] Nikolay Stefanov, Angelika Peer, and Martin Buss. Online intention recognition for computer-assisted teleoperation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5334–5339. IEEE, 2010.

[121] M.F. Stoelen and D.L. Akin. Assessment of Fitts Law for Quantifying Combined Rotational and Translational Movements. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 52(1):63, 2010.

[122] Eliza Strickland. Meet the Robots of Fukushima Daiichi, 2014. `spectrum.ieee.org/slideshow/robotics/industrial-robots/meet-the-robots-of-fukushima-daiichi`.

[123] Susumu Tachi. Real-time remote robotics-toward networked telexistence. *IEEE Computer Graphics and Applications*, 18(6):6–9, 1998.

[124] Kotaro Tadano and Kenji Kawashima. Development of a master slave system with force sensing using pneumatic servo system for laparoscopic surgery. *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 947–952, 2007.

[125] Kotaro Tadano and Kenji Kawashima. Development of a pneumatically driven forceps manipulator ibis iv. *ICROS-SICE International Joint Conference*, pages 3815–3818, 2009.

[126] J. Tokuda, G.S. Fischer, X. Papademetris, Z. Yaniv, L. Ibanez, P. Cheng, H. Liu, J. Blevins, J. Arata, A.J. Golby, et al. OpenIGTLink: an open network protocol for image-guided therapy environment. *Int J Med Robot Comput Assist Surg*, 5:423–434, 2009.

[127] Hans Utz, Stefan Sablatng, Stefan Enderle, and Gerhard Kraetzschmar. Miromiddleware for mobile robot applications. *Robotics and Automation, IEEE Transactions on*, 18(4), August 2002.

[128] Jur Van Den Berg, Stephen Miller, Daniel Duckworth, Humphrey Hu, Andrew Wan, Xiao-Yu Fu, Ken Goldberg, and Pieter Abbeel. Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2074–2081. IEEE, 2010.

[129] N.C. von Sternberg, A. Kilicarslan, N.V. Navkar, Zhigang Deng, K. Grigoriadis, and N.V. Tsekos. Implementation of a force-feedback interface for robotic assisted interventions with real-time mri guidance. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4869–4874, 2013.

[130] T. Xia, C. Baird, G. Jallo, K. Hayes, N. Nakajima, N. Hata, and P. Kazanzides. An integrated system for planning, navigation and robotic assistance for skull base surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 4(4):321–330, 2008.

[131] Tian Xia, Ankur Kapoor, Peter Kazanzides, and Russell Taylor. A constrained optimization approach to virtual fixtures for multi-robot collaborative teleoperation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 639–644. IEEE, 2011.

[132] Tian Xia, Simon Léonard, Anton Deguet, Louis Whitcomb, and Peter Kazanzides. Augmented reality environment with virtual fixtures for robotic telemanipulation in space. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5059–5064. IEEE, 2012.

[133] Tomonori Yamamoto, Niki Abolhassani, Sung Jung, Allison M Okamura, and Timothy N Judkins. Augmented reality and haptic interfaces for robot-assisted surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 8(1):45–56, 2012.

[134] Y. Yokokohji, A. Ogawa, H. Hasunuma, and T. Yoshikawa. Operation modes for cooperating with autonomous functions in intelligent teleoperation systems. In *Proc. of IEEE/ICRA*, pages 510 –515 vol.3, may. 1993.

[135] Xiaolong Yu, Howard Jay Chizeck, and Blake Hannaford. Comparison of transient performance in the control of soft tissue grasping. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1809–1814. IEEE, 2007.

[136] Junku Yuh. Design and control of autonomous underwater robots: A survey. *Autonomous Robots*, 8(1):7–24, 2000.

[137] Dongwen Zhang, Lei Wang, Jia Gu, Zhicheng Li, and Ken Chen. Realization of spatial compliant virtual fixture using eigenscrews. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 1506–1509. IEEE, 2012.

[138] Craig B Zilles and J Kenneth Salisbury. A constraint-based god-object method for haptic display. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 146–151. IEEE, 1995.

Appendix A

POWER ANALYSIS OF EXPERIMENTAL RESULTS

Statistical power is the probability of *not* committing a type II error, i.e., that a test will reject the null hypothesis when the null hypothesis is false. Post-hoc power analysis relates the sample size and observed effect size to the statistical power in a study. In this appendix, post-hoc analysis of the experiments in Chapter 7 tests the statistical power of the linear modeling ANOVA methods used.

A.1 Methodology

The following methodology was used to determine expected necessary sample size 'n' based on the results of the experiments carried out in this work.

The following quantities are required: betwen groups variance, within groups variance, number of groups, disired α and desired power. From the literature [3] a desired power of 0.8 or 0.95 was used. Figure A.1 shows the relationship between n and power for the Fitts Law task.

Necessary quantities were calculated in R using the following:

```
moveDat1$lnBT <- log(moveDat1$path)
modelx <- (lnBT ~ mode)
outputx <- lm(modelx, data=moveDat1)
av1 <- anova(outputx)

BGV <- av1["Residuals","Mean␣Sq"]
WGV <- av1["mode","Mean␣Sq"]
grps <- 4
siglev <- 0.05
```

This example is for pathlength data, *moveDat*1$*path*, of the total TFLS task time, but the same pattern is repeated for all metrics. Data are log transformed to achieve a normal distribution (checked elsewhere with Shapiro Wilk and QQ normality plot. See Chapter 7 for details.)

Between groups variance, BGV, and within groups variance, WGV, are taken directly from a simplified one-way analysis of variance. The simplified model looks for variation among *any* of the conditions, whereas the model used in Chapter 7 looks for differences between each assistance mode and unassisted operation. This is sufficient for a broad estimate of the power of this study.

The number of groups is fixed at four (three assistance modes, plus unassisted operation) and the desired α level is set at 0.05.

The power analysis itself is performed in R using the *power.anova.test*(), as follows:

```
pt <- power.anova.test(
        between.var= BGV,
        within.var=  WGV,
        groups=grps,
        n=NULL,
        sig.level=siglev,
        power=0.9
)
```

A.2 Results and Conclusions

Power analysis following the above methods was performed on all data reported in Chapter 7 that were analyzed with ANOVA.

The statistical power of an experiment goes up with increased sample size. This relationship, for the data observed in Chapter 7 is illustrated in Figure A.1. The colored lines represent power vs. sample size for each of the the metrics identified in that test. The vertical black line indicates the actual sample size (n=110) used in the experiment.

Table A.1 presents a power analysis of the Fitts' Law task. The required sample size to achieve power $p > 0.8$ was between 13 and 109 for all six metrics. The actual sample size used was $n = 110$ (samples per group), therefore this was a sufficient sample size to make statistical inference based on the observed effect size and variances.

Table A.2 reports the power analysis results for the Telerobotic FLS study. Power analysis is performed on four metrics from that study. The metrics can be divided into two groups: phase metrics and total metrics. The phase metrics (time, pathlength) represent measurements on each

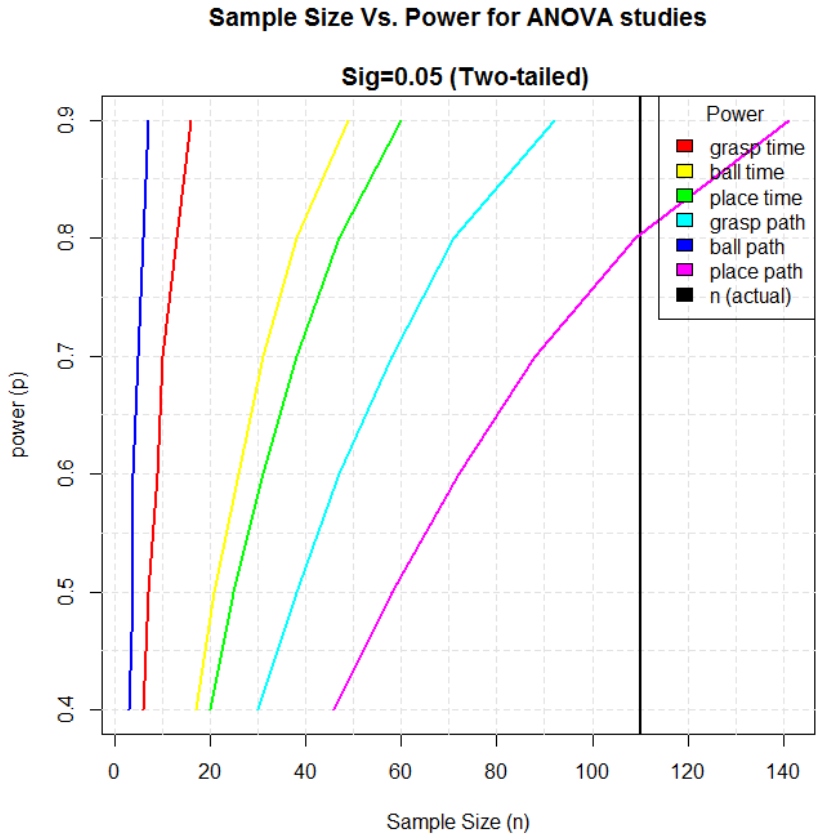


Figure A.1: Sample size vs. Power based on observed effect size and variance for each metric identified in the Fitts' Law experiments of Chapter 7. The sample size used in the experiments was n=110 (shown in black) is sufficient to make statistical inferences with ANOVA.

phase of the task: pick-up, handoff, and place. Data from the three phases are combined into one set to simplify analysis. The total metrics (time, path) are measurements for the entire task, i.e., six blocks moved left-to-right and right-to-left with handoff.

From the foregoing power analysis it is concluded that the experimental design (i.e., the selected sample size) yielded sufficient confidence to reject the null hypothesis when ANOVA showed the null hypothesis to be false. Specifically, the null hypothesis was that teleoperation assistance had no effect on teleoperation performance as measured on the selected tasks by the given metrics.

Metric	groups	between group variance	within group variance	α	power	n (calculated)
grasp time	4	0.0334025	0.1018641	0.05	0.8	13
ball time	4	0.2283528	2.301334	0.05	0.8	38
place time	4	0.4135045	5.158877	0.05	0.8	47
grasp path	4	0.0179578	0.3433491	0.05	0.8	71
ball path	4	0.1184709	0.1314477	0.05	0.8	5
place path	4	0.3382577	9.995755	0.05	0.8	109

Table A.1: Power analysis of Fitts Law Task results.

Metric	groups	between group variance	within group variance	α	power	n (calculated)
phase time	4	0.2679056	1.635347	0.05	0.8	24
phase path	4	0.5840367	0.6344097	0.05	0.8	6
total time	4	0.0868591	0.03376707	0.05	0.95	4
total path	4	0.0379499	0.00558154	0.05	0.95	3

Table A.2: Power analysis of Telerobotic FLS Task results. *Phase metrics represent all phases in one dataset (pick-up, handoff, place), while total time and path represent time and pathlength for an entire TFLS block transfer performance.

VITA

Hawkeye was born in Berkeley California and spent his early years at the Emeryville Artists Cooperative. In 2000 he earned a B.A. in Cognitive Science and Computer Science from Dartmouth College. In 2009 he earned an M.S. in Electrical Engineering from the University of Washington. Between degrees he worked as a web application developer and network engineer at a startup company in Tokyo, Japan.

He was a guest researcher at the Technische Universität München in 2009 and the Tokyo Institute of Technology in 2007. In 2007 he was a guest lecturer at the Advanced Institute of Science and Technology in Seoul, Korea. In 2008 he spent time at SRI International working on zero-gravity telesurgery.

Robots he built with the BioRobotics Laboratory have been featured in Hollywood movies, deployed in undersea and desert environments, and are in use by over fourteen institutions around the world.