PROGRAMMING DEXTEROUS MANIPULATION BY DEMONSTRATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Michael Leo Turner

June 2001

READING COMMITTEE SIGNATURE PAGE

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

Dr. Mark Cutkosky, Principal Advisor

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

Dr. Bernard Roth

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

Dr. Oussama Khatib

Approved for the University Committee on Graduate Studies

_____

ABSTRACT

Programming Dexterous Manipulation by Demonstration

by

Michael Leo Turner

Doctor of Philosophy in Mechanical Engineering

Stanford University

Advisor: Dr. Mark Cutkosky

Current robot manipulation systems lack the flexibility and intelligence inherent in human manipulation of objects. Research in the field of dexterous manipulation has resulted in the development of multi-finger mechanical hands which can stably manipulate and regrasp objects without releasing. In addition, sensors have been added to the robot fingers which emulate the human sense of touch, including force, vibration and pressure distribution. Despite such progress, a robot is still a poor substitute in situations where it is too dangerous or expensive to place a human. The fundamental difference is the ability to adapt to new situations and respond to unexpected occurrences. This thesis examines three different approaches to this problem: autonomous manipulation, telemanipulation and programming by demonstration.

In autonomous manipulation, a robot hand is programmed to perform a specific task. A state diagram is used to set the control law for each particular phase of the task. The sensors on the fingers are used to detect events requiring a transition from one state to another. An extended example of a two-fingered robot hand manipulating an unknown object is used to display the capabilities and limitations of this approach.

In telemanipulation, a robot is directly controlled by a human user. In the current implementation, the user wears an instrumented glove which measures the joint angles of the hand. User hand motions are mapped into robot hand motions, allowing the user to remotely perform a task. In addition, the user may wear a haptic feedback device which can display the forces sensed by the robot. The performance results and analysis of human subjects performing prototypical tasks are presented.

In programming by demonstration, a layer of separation is created between the user and the robot hand. The user, wearing the instrumented glove, interacts with a virtual world through a virtual human hand. The motions of the virtual objects are tracked for the purpose of generating a control law for the robot to perform the same or similar motion with a physical object. The added layer of abstraction allows a more intuitive human interface and optimized robot motion while still maintaining human decision making. The issues associated with implementation of tracking intended human manipulation of a virtual object are discussed.

ACKNOWLEDGEMENTS

They said "Go West!" so west I rode
to see the world and start my quest.
I learnt my trade among the best,
and near the end with head unbowed.

But thanks I give to those whose aid
has kept my feet upon the road.
From friends who helped to share the load
to folks whose faith has never frayed.

To Mom and Dad who did not doubt
and sisters strong stood by my side.
For Amy kept my vision wide
while Donna never shut me out.

To Tone who shared my worst and best,
and Phil the cook who kept me fed.
For all the laughs I shared with Ted.
To Jorge, Sean, Beth and the rest.

To Jonathan, the bearded one,
for thoughts on facts and faith and fun.

To Wes with hair of silken gold,
a sailor's mouth and tiger's soul.

To Chris, my friend so Strait and strong,
I cry Sha-bu-ya loud and long.

To Allie-O, who led the way,
plays hard yet smiles on every day.

To Costa, friend and fellow freak,
So cool and smooth and still a geek.
I owe you more than it may seem,
for field and lab, we are a team.

And Ryan came into our fold
to tackle tasks which stopped me cold
for hours long and days on end.
A faithful minion and a friend.

To Mark, the MAN, with mind so rare,
as guide and aide beyond compare.
Profs Khatib and Roth who read this book,
my thanks for your insightful look.

Table of Contents

CHAPTER 1 Introduction 1

1.1 Robots and Humans 1
1.2 Thesis Outline 3
1.3 Contributions 4

CHAPTER 2 Previous Work 6

2.1 Dexterous Manipulation 6
2.1.1 Design of Dexterous Robot Manipulators 6
2.1.2 Contact Kinematics 7
2.1.3 Grasp Selection 8
2.1.4 Control of Robot Manipulation 9
2.2 Human Hand Modeling 12
2.2.1 Human Hand Kinematics 12
2.2.2 Human Grasp Analysis 12
2.3 Calibration 13
2.4 Haptic Feedback 14
2.4.1 Human Perception 15
2.4.2 Design of Feedback Mechanisms 15
2.4.3 Effect on Human Performance 17
2.5 Telemanipulation 18
2.6 Virtual Manipulation 20
2.7 High-Level Programming 21

CHAPTER 3 Dexterous Manipulation 24

3.1 Haptic Exploration 24
3.2 Manipulation 25
3.2.1 Mechanical Setup 25
3.3 Exploratory Procedure Algorithm 26
3.3.1 States and Transitions 26
3.3.2 Rolling and Sliding: Motion Planning and Control 27
3.3.3 Simulation 28
3.3.4 Experiments 29

CHAPTER 4 Human Hand Model 31

4.1 Kinematic Model 31
4.2 Hand Calibration 33
4.2.1 Two Fingered Calibration 33
4.2.1.1 Angular Offset Calibration 34
4.2.1.2 Including Unknown Sensor Gains 36
4.2.1.3 Limitation of Parameter Deviation 38
4.2.1.4 Performance Metrics and Experimental Results 39

4.2.2 Four Fingered Calibration . . . 43
4.2.2.1 Modeling Middle or Ring Finger . . . 44
4.2.2.2 Abduction . . . 44
4.2.2.3 Thumb variation . . . 45
4.2.2.4 Simultaneous Four Finger Calibration . . . 46
4.3 Inverse Kinematics . . . 47
4.3.1 Index, Middle and Ring Fingers . . . 47
4.3.1.1 Abduction joint . . . 48
4.3.1.2 Parallel joints . . . 48

CHAPTER 5 Haptic Feedback in Dexterous Telemanipulation . . . 51

5.1 Haptic Perception with Arm-Grounded Feedback . . . 51
5.1.1 Experimental Setup . . . 53
5.1.2 Experiments . . . 55
5.1.2.1 Object Size Discrimination . . . 55
5.1.2.2 Force Regulation . . . 57
5.1.2.3 Object Stiffness Discrimination . . . 59
5.1.3 Conclusion . . . 60
5.2 Haptic Feedback In Dexterous Telemanipulation . . . 61
5.2.1 Human interface . . . 61
5.2.1.1 Instrumented Glove . . . 62
5.2.1.2 Arm-grounded Force Feedback . . . 62
5.2.1.3 6-DOF Kinematic Linkage . . . 62
5.2.2 Robot System . . . 63
5.2.2.1 Dexterous Robot Hand . . . 63
5.2.2.2 Industrial Robot Arm . . . 63
5.2.3 Communication . . . 64
5.2.4 Testing . . . 64
5.2.4.1 Block Stacking . . . 65
5.2.4.2 Knob Turning . . . 67
5.2.5 Results . . . 69

CHAPTER 6 Tracking Human Manipulation . . . 71

6.1 System Overview . . . 71
6.2 Virtual Environment . . . 72
6.3 Tracking Manipulation . . . 73
6.3.1 Basic Approach . . . 73
6.3.2 Contact Acquisition and Release . . . 76
6.3.3 Sliding . . . 77
6.3.4 Rolling Contact . . . 79
6.3.5 Integrated Algorithm . . . 83
6.4 Testing Manipulation Tracking . . . 84
6.4.1 Testing Setup . . . 84
6.4.2 Three Fingered Grasp on 50 mm object . . . 85

CHAPTER 7 Conclusions and Future Work . . . 88

7.1 Conclusions 88
7.1.1 Autonomous Dexterous Manipulation 88
7.1.2 Hand Model 88
7.1.3 Telemanipulation 89
7.1.4 Tracking Human Manipulation 90
7.2 Future Work 90
7.2.1 Autonomous Programming 90
7.2.2 Grasp Selection 92
7.3 Final Thoughts 93

Bibliography 94

APPENDIX A Kinematic Hand Model Details A-1

A.1 Thumb A-2
A.1.1 Transformation Matrices A-2
A.1.2 Jacobians A-4
1.2 Index Finger A-5
A.2.1 Transformation Matrices A-5
A.2.2 Jacobian A-7

APPENDIX B Hand Calibration Details B-1

List of Figures

CHAPTER 1 Introduction 1

CHAPTER 2 Previous Work 6

CHAPTER 3 Dexterous Manipulation 24

Figure 3-1. Diagram of States and Transitions During Exploratory Manipulation 26

CHAPTER 4 Human Hand Model 31

Figure 4-1. Kinematic Model of Human Hand 32
Figure 4-2. Comparison of Actual and Calculated Finger Separation of a Calibrated Subject 41
Figure 4-3. Calibrated Thumb Parameters for Different Methods 45
Figure 4-4. Inverse Kinematics of Parallel Joints on Finger 48

CHAPTER 5 Haptic Feedback in Dexterous Telemanipulation 51

Figure 5-1. CyberGrasp Force Tracking with Flexed and Unflexed Cable 54
Figure 5-2. Size Discrimination through CyberGrasp 56
Figure 5-3. Force Regulation Results with CyberGrasp (left) and with Direct Contact (right) 58
Figure 5-4. Block Stacking 65
Figure 5-5. Completion Time Ratio and Failure Occurrences for Block Stacking 66
Figure 5-6. Knob Turning 67
Figure 5-7. Completion Time Ratio and Failure Occurrences for Knob Turning 68

CHAPTER 6 Tracking Human Manipulation 71

Figure 6-1. Graphical Display of a Virtual Environment 72
Figure 6-2. Controlled Object Motion from Measured Fingertip Position 75
Figure 6-3. Contact Acquisition and Release with Intentional Hysteresis to Avoid “Chatter” 77
Figure 6-4. Sliding and Non-sliding Contact Forces Based on Friction Cone 78
Figure 6-5. Determination of New Finger Location Subject to Rolling Constraint 80
Figure 6-6. Representation of Complete Algorithm for Tracking Intended Human Manipulation 83
Figure 6-7. Comparison of Actual Object Motion and Simulated Object Motion 86

CHAPTER 7 Conclusions and Future Work 88

Figure 7-1. Proposed Information Loop for Programming by Demonstration 92

APPENDIX A Kinematic Hand Model Details A-1

Figure A-1. Human Hand Model with Link Length and Reference Frames Defined A-1
Figure A-2. Hand Model with All Joint Angles at Zero A-2

APPENDIX B Hand Calibration Details B-1

1 Introduction

> ...[He] flexed and extended his fingers gently; the two pairs of waldoes in the screen followed in exact, simultaneous parallelism. ...He then started hand movements of definite pattern; the waldoes at the power tool reached up, switched on the power, and began gently, gracefully to continue the machining of the casting. A mechanical hand reached down, adjusted a vernier, while the other increased the flow of oil cooling the cutting edge. ...[He] continued with unhurried skill, his finger motions with his waldoes exerting pressure which would need to be measured in fractions of ounces, but the two sets of waldoes, paralleled to him thousands of miles below, followed his motions accurately and with force appropriate to the heavy work at hand.
>
> - Waldo, Robert Heinlein 1940

1.1 Robots and Humans

Sixty years ago, science fiction writer Robert Heinlein envisioned mechanical hands performing delicate manufacturing tasks under remote supervision and control of a human operator. As research has advanced toward this prescient vision, the operation of robot hands has come to be called “robotic dexterous manipulation” and, when controlled by a human hand, “telemanipulation.” The work presented here, on programming dexterous manipulation by demonstration, develops a procedure intended to further this research and ease the implementation of the technology into practical applications.

Robots are a common sight today assembling electronic equipment, welding car bodies and machining parts. The assembly line robot is superior to its human counterpart when it comes to qualities such as positional accuracy, speed of performance, strength and endurance. However, robots have limitations which humans do not experience. These limitations frustrate the efforts to utilize robots outside of structured and controlled environments such as a factory floor. Even on the factory floor, the manufacturing process today is a hybrid of human labor and automation.

One important limitation is the ability to deal with new or changing situations. When a robot encounters a situation for which it was not expressly programmed, its behavior soon becomes unacceptable. For example, consider an automobile assembly line, where both robots and people are used to spot weld the frame of the car together. When a new version of the car is introduced, the experienced human welder can draw upon previous knowledge and similarities between the old version and the new to anticipate the necessary work. A few verbal instructions might be given, such as “Make sure you place a weld at this spot.” or “Keep the welds tighter together along this frame.” Within a short time, the skilled worker has learned the new routine. The robot, on the other hand, must be reprogrammed by an experienced programmer. Each weld is reassigned, and nothing can be taken for granted. Such adjustments can take days or even weeks.

Complexity in programming robots is one reason that technical developments in robotics are slow to be implemented onto the factory floor. Visual feedback and force control are relatively recent introductions, while other ideas such as dexterous manipulation are still to be introduced. Each increase in robot complexity requires a corresponding increase in the expertise and effort in programming.

One solution to the programming difficulty is teleoperation. In teleoperation, a human user controls the motion of the robot directly. The intelligence, learning and experience of the human can be combined with the strength, endurance and speed of the robot.

The quote from Heinlein describes a person performing a specific type of teleoperation called telemanipulation. One type of telemanipulation entails tracking human finger motions to control robotic finger motions, with the intention of providing human-like versatility and complexity in robot hand performance. In the fictional example, the operator is able to command the mechanical hand to perform various tasks such as turning on a power switch, adjusting oil flow and setting a vernier. A representation of the forces experienced by the mechanical hand (called a waldo in the story) is displayed back to the operator.

A significant advantage of such a system is that the operator can be physically distant from the robot. The robot may be in a location where it is difficult or dangerous to place a human, such as a hazardous waste spill or in the deep sea, but where the unstructured environment makes a human’s intelligence and adaptability advantageous. However, teleoperation is

less suitable for controlling robots performing repetitive tasks, because it requires the full time attention of a human operator.

This thesis presents research in dexterous manipulation and telemanipulation which develops a framework by which a person can demonstrate a task for the robot. Demonstration takes advantage of many intuitive and natural skills which a human utilizes and it offers an opportunity to facilitate implementation of dexterous manipulation. The computer tracks the human motion, and determines the desired effect upon objects within the local environment. In the future, the robot will then able to mimic the human performance to repeat the desired motion of the objects. The procedure is called programming by demonstration.

1.2 Thesis Outline

Chapter 2 provides an overview of work performed in the fields of dexterous manipulation, telemanipulation and robotic programming.

Chapter 3 develops an extended example where a robot performs dexterous manipulation. A two-fingered robot manipulates an approximately known object to explore the surface of the object for features and shape modeling. This example gives insight into the state of the art for autonomous dexterous manipulation.

Chapter 4 presents a kinematic model of the human hand. This model approximates the human hand as a series of links with pure hinge joints. This model was developed for use with the CyberGlove, an instrumented glove that measures the angles of 22 joints on the human hand. The forward kinematics, inverse kinematics, Jacobian and a calibration routine to customize the model to a particular human hand are also developed. This model is used for telemanipulation (Chapter 5) and for tracking intended motion of a virtual object (Chapter 6).

Chapter 5 describes experiments in telemanipulation with and without utilizing haptic feedback. Robot finger motions track the finger motions of a human subject wearing the CyberGlove. The output of force sensors on the robot are displayed to the user through the CyberGrasp, a mechanism worn on the back of the hand which can apply forces to the human fingers. In the first set of experiments, the subjects used the robot to grasp unseen objects. They were asked to discriminate between objects of different sizes, to discriminate

between objects of different stiffness and to regulate the force of application. In the second set of experiments, the subjects were asked to perform several prototypical tasks, such as stacking blocks, rotating an object and extracting a constrained object.

Chapter 6 presents a means of tracking human manipulation. A person wearing the CyberGlove interacts with a virtual object. The motion of the object is inferred from the motion of the human fingers while subject to the contact constraints of rolling and sliding. The set of virtual object poses and positions are recorded for use in programming a dexterous robot to manipulate an analogous real object.

Chapter 8 summarizes the impacts of the research presented here and describes the planned methodology of converting a desired object motion from Chapter 6 into a programmed manipulation by a robot. The robot controller will use the human demonstration to determine grasp selection and commanded object motions, subject to the particular kinematic constraints of the particular robot.

1.3 Contributions

The two most significant contributions in this thesis are presented in Chapter 4 and Chapter 6. Chapter 4 develops a model of the human hand based on biomechanical research as a kinematic linkage with pin joints. Particular emphasis is placed on modeling the carpometacarpal joint at the base of the thumb to adequately capture the full range of motion, while maintaining measurability. The forward kinematics, Jacobians and inverse kinematics of the model are presented. The development of the model is supported by a closed loop calibration routine expanded from similar routines used for calibrating robotic mechanisms. The routine can calibrate up to four fingers for bone length, sensor offset, sensor gain and sensor cross coupling. Chapter 6 develops an algorithm for tracking manipulation of a virtual object, which is consistent with natural human manipulation. Most of the previously developed virtual manipulation environments are based on a dynamic response to contact forces which can lead to unintended object motion, and do not maintain rolling contact constraints between the fingers and the object, which is important for performing fine manipulations. The process proposed in this thesis matches the measured motion of the human fingertips to the best fit object motion, allowing for sliding contact, rolling contact and finger release/regrasp.

These contributions are supported by the work presented in the remaining chapters. Chapter 3 presents a method for autonomously manipulating an unknown or partially known object for the purpose of exploration. Chapter 5 tests the efficacy of hand grounded force feedback for displaying internally and externally grounded forces during dexterous telemanipulation. Chapter 7 utilizes the desired object motion gathered in Chapter 6 and discusses how this can be used to autonomously program a dexterous robot hand to perform the same object motion.

2 Previous Work

2.1 Dexterous Manipulation

2.1.1 Design of Dexterous Robot Manipulators

The fundamental characteristic of dexterous robot hands is the ability to change a grasped object's position or orientation with respect to the hand without releasing it. The general goal is to have a robot hand with some of the versatility of the human hand. The hand may be an approximation of the human hand, or it may be specialized for particular tasks. It can include various sensors to detect forces, vibrations or pressure distributions.

Grupen, Henderson et al. [37] survey the technologies which support general-purpose manipulation. They present a variety of mechanical manipulators and discuss constraints on manipulation strategy selection.

The Utah/MIT hand is one of the first anthropomorphic hands. It has four fingers with four degrees of freedom each, with a "thumb" finger in opposition to the others [57]. The Robonaut hand [80] was designed for space based operations. The five finger hand combined with its integrated wrist and forearm has fourteen independent degrees of freedom. It approximates the kinematics and strength of an astronaut's hand inside a pressurized space suit glove.

Planar dexterous hands are a fairly common choice in dexterous manipulation. The fewer degrees of freedom, with respect to the anthropomorphic hands, enables a significant reduction in development costs. They are useful for testing manipulation strategies [30], sensor development [54] or performing primarily planar tasks [123].

Placing the robot hand on the end of larger arm, such as Nagai and Yoshikawa [93] describe, produces a redundant macro-micro manipulator for grasping and manipulation. The hand is used to add compliance and fine motion due to small inertia, while the arm increases the overall range of motion.

2.1.2 Contact Kinematics

An understanding of the nature of the contact between the robot finger and the object is a fundamental part of dexterous manipulation. The dexterous finger is not required to maintain a single contact location and orientation with respect to the object. It may roll or slide along the object surface, or even release and reacquire contact.

When the object is manipulated, it is common for the fingertip to roll on the surface of the object. Rolling is defined by a velocity constraint, namely that the relative velocity between the two surfaces at the point of contact must be zero. If the system is planar, this can be integrated and used as a position constraint. However, for spatial motions the constraint is non-holonomic and can only be solved instantaneously in the velocity domain.

The generalized formulation of the rolling constraint has been expressed in multiple ways. Kerr and Roth [63] describe the motion of the object with pure rolling as a set of differential equations. Montana [91] derives a matrix formulation of the motion of a point of contact over the surfaces of two rolling surfaces. In [92], Montana derives the kinematics of the fingers-object system in a configuration-space description. The contact kinematics are formulated as a 'virtual' kinematic chain, with one large closed kinematic chain composed of smaller chains.

The control of whether or not a finger will slide along the surface of an object requires a model of the contact friction. Most of the grasp analyses assume a Coulomb friction model and compensate for errors by using a conservative friction coefficient. Howe, Kao et al. [50] point out that most robot fingers are not in point contact. They examined experimental measurements of the initiation of sliding as a function of load. As a result, they propose a model of friction with a linear function of torsion and shear magnitudes to predict the onset of slip.

Cai and Roth [18] study spatial motions combining rolling and sliding for point contact between bodies. The formulations are based on instantaneous time-based kinematics and assume knowledge of the relative motion at the point of contact. In [19], Cai and Roth expand the investigation to include bodies maintaining line contact.

2.1.3 Grasp Selection

One of the primary challenges of dexterous manipulation is the seemingly straightforward task of picking up the object. The task is not trivial for an arbitrarily shaped object and a dexterous hand with a high number of degrees of freedom.

Mason and Salisbury [84] introduce the grasp matrix for a multi-finger hand holding an object, which relates the forces and torques applied by the fingers to the net force and torque applied to the object. This is somewhat analogous to the robot Jacobian. An analysis of the grasp matrix can determine if a particular grasp can resist arbitrary disturbances, a grasp state called form (or force) closure. (Mason and Salisbury define force closure less strictly than form closure. Most later works, and this thesis, use them interchangeably according to the stricter definition.)

For a given hand and object, there are usually a large number of grasps which will satisfy the form closure criteria. To select a "good" or an "optimal" grasp, it is necessary to develop measures of grasp quality and a means of searching the space of possible grasps.

One measure of grasp quality is the manipulability, or ability of the robot hand to impart motions to the object. In [64], Kerr and Roth examine cases where a robot grasp is over-constrained or underconstrained, based on examination of the grasp matrix, and formulate what manipulations can be performed by a given grasp.

Park and Starr [101] develop two metrics for grasp quality. The uncertainty grasp index measures the decrease in grasp stability for small finger position errors. The task compatibility grasp index is a representation of how well a particular grasp can perform a particular task.

The grasp requiring minimum grasping energy while maintain grasp stability is defined as the optimal grasp by Buss, Hashimoto et al. [15]. It is calculated by using a linearly constrained semidefinite programming problem for which there are known globally exponentially convergent solutions via gradient flows. In [16], Buss, Hashimoto et al. further formulate the task of selecting grasping force as an optimization problem on the smooth manifold of linearly constrained positive definite matrices for which there are known globally exponentially convergent solutions via gradient flows. Han, Trinkle et al. [39] use the

Buss optimization above to formulate the basic grasp analysis problems as a set of convex optimization problems involving linear matrix inequalities.

Miller and Allen [90] develop a grasp analysis system which can determine the types of contact and quality of grasp for a given a 3D object, hand, and pose. In addition, the system can search a subset of possible hand configurations for a highest quality grasp.

The ability to do a grasp analysis in real time is critical for many robotic applications. Allen, Timcenko et al. [4] built a system to address problems in robotic hand-eye coordination for grasping moving objects to enable a robot to track and grasp a moving object. Allen, Miller et al. [5] further develop the robotic hand system to utilize joint position and force sensing with tactile sensors and vision modules to determine finger contacts and applied forces for grasping tasks. Li, Qin et al. [78] propose a modular hierarchical system capable of tracking the motion of a grasped object, optimizing the grasp during manipulation and adjusting grasping forces to maintain contact and react to external forces.

2.1.4 Control of Robot Manipulation

Once an object has been grasped, it is possible to take advantage of the dexterous hand by performing in-hand manipulation. A dexterous robot hand can change the orientation of the object with respect to the hand without releasing, as well as change the grasp configuration by rolling, sliding or releasing fingers. Stable control of the object in the hand requires accurate control of the forces applied to the object and proper formulation of the contact constraints.

Allen, Michelman et al. [2] developed a comprehensive grasping environment capable of performing tasks such as locating moving objects and picking them up, manipulating man-made objects as tools, and recognizing unknown objects through touch. An integrated programming environment was designed to program dextrous hands, vision sensors, and multiple-degree-of-freedom manipulators utilizing grasping and grasping primitives. Allen, Michelman et al. [3] improved the system for programming and controlling a multisensor robotic hand (Utah-MIT Hand). The hand was programmed to autonomously pick up and pour a pitcher, unscrew a light bulb and explore a surface; tasks which combine hand-arm actuation with force, position, and tactile sensing.

The finger forces applied to the object can be subdivided into “external” forces which produce a net force and moment on the object and “internal” forces which produce no net force nor torque on the object (used for maintaining a secure grasp), according to Yoshikawa and Nagai [128]. The external forces are the minimum finger forces necessary to provide the desired external force. Based on the external forces, it is possible to determine the additional internal forces required to maintain stability of the grasp. In [129], Yoshikawa and Nagai define the manipulating force as a force which produces a specified resultant force, is not inverse to the grasping (internal) force and is orthogonal to the grasping force component. The grasping and manipulating forces can be synthesized for a given manipulation task. Nakamura, Yoshikawa et al [95] use the resulting force to maintain dynamic equilibrium and as a restoring force.

The successful implementation of rolling and sliding contact control can significantly enhance the performance of the robot hand. Trinkle and Paul [120] exploit sliding contact between the object and the robot hand to gain a secure, enveloping grasp of a two dimensional object. Yoshikawa, Yokokohji et al. [130] use controlled slip to modify the grasp and increase manipulation range for a three fingered robot hand to perform a quasi-static object manipulation.

Rolling contacts are investigated by Paljug, Yun et al. [99] for a redundant multi-arm manipulation system. A nonlinear feedback scheme is used for simultaneously controlling motion and contact conditions. Sarkar, Yun et al. [110] enhance the system with a dynamic model of the system developed from the velocity and acceleration equations for three-dimensional rolling contact.

A feedback loop to the rolling manipulation is proposed by Jiang, Choi et al [59], in which the contact velocities are determined by tactile sensor feedback. By coordinating the motion generation, the grasp quality during dexterous manipulation can be controlled.

The combination of rolling and sliding control is developed by Yun, Kumar et al. [131]. Nagashima, Seki et al. [94] analyze the kinematic relationship between object and finger-joint motions for the cases when the contact is constrained by pure rolling, twist rolling or slide rolling. In addition, manipulation kinematics equations are developed using incremental variations of object motion and finger motion.

During manipulation, it is often useful to have different control strategies for different stages of the process. For example, fingers out of contact might be controlled by position control while fingers in contact are controlled by force control. Michelman and Allen [88] break complex manipulation tasks into a set of primitive manipulation functions. Complex tasks are sequential combinations of the primitive functions. The combination of primitive functions is determined by analysis of the overall task, and under the control of finite state machines. This work is extended by Michelman [89] with an experimental demonstration of robot hand dexterity.

The transition from motion with a free finger to contact with the object can result in finger-tip oscillations, particularly if there is also a transition to force control. Hyde and Cutkosky [53][54] minimize these oscillations by "pre-shaping" the input commands to initialize the constraints based on the current state of the system. In [55], Hyde and Cutkosky develop a mid-level control framework which incorporates phases, events, and transitions.

In an interesting application of dexterous manipulation, Hilhorst and Tanie [45] develop an adaptive control scheme to allow a robot to manipulate an object with unknown mass and inertia by a robotic hand. Using knowledge of the robot hand dynamics, an initial estimate of object inertia is corrected during the course of manipulation to improve precision and control.

Dexterous Manipulation Summary:

Shimoga [112] surveys existing research of autonomous multifingered robotic hands, and the progress made in achieving dexterity, equilibrium, stability, and dynamic behavior. These properties are necessary to autonomously perform complex tasks in a way similar to human hands. Okamura, Smaby et al. [98] provide a more recent overview of current research in dexterous manipulation.

At this time, dexterous manipulation can be considered a mature field of research. The current robot hands can match the human hand in degrees-of-freedom, and exceed the human hand in positional resolution. The understanding of the kinematics of contact is extensive, though new methods which can analyze and control the system more efficiently are still being developed. The principal limitation of autonomous robot manipulation is robot intel-

ligence. It is necessary for a robot programmer to anticipate likely events and instruct the robot a priori how to respond.

2.2 Human Hand Modeling

2.2.1 Human Hand Kinematics

An, Chao et al. [6] developed a three dimensional model of the human hand from examining the anatomical structure of ten “normal” human hands. An analysis of relative bone lengths and tendon connections is used to calculate force and motion capabilities of normal and pathological hands.

Cooney, Lucca et al. [26] note that the carpometacarpal joint at the base of the thumb has two degrees of freedom, but has motion in three anatomic planes. Hollister, Buford et al. [48] determined that the motion of the CMC joint results because the two axes of rotation are non-orthogonal and non-intersecting. Guirintano, Hollister et al. [38] build a five link kinematic model of the thumb using the two skew axes of the CMC joint.

Kramer [73] patented a kinematic model of the human thumb, designed for use with the CyberGlove, which represents each of the joints as a pin joint. The thumb metacarpal is modeled as two intersecting and orthogonal joints, measured by two external goniometers. The third degree of motion is modeled such that the volar face of the thumb is oriented towards a unique “control point”, nominally located 1 cm out from the palm at the base of the ring finger.

2.2.2 Human Grasp Analysis

Napier [96] performed the seminal work in analyzing the grasping movements of the human hand. The set of human grasps are divided into two primary categories. Precision grasps are those where the object is usually held by the fingertips and where manipulability is more important than the ability to resist large external forces. Power grasps are those where the object is usually constrained by the palm and both the proximal and distal surface of the fingers and where force closure is more important than manipulability.

Cutkosky [27] constructed a taxonomy of grasps based on a study of machinists in a small batch manufacturing operation. Analysis of the grasps used enabled a predictive system to

anticipate which grasp would be used in a certain circumstance based on issues such as task, required force and object size. In [28], Cutkosky and Howe apply the taxonomy to the robot hand design and grasp selection.

Kao, Cutkosky et al. [62] use data obtained from observing human grasping behaviors in grasping tasks to analyze several methods of stiffness calibration. For small displacements, there is a linear relationship between force and displacement in human grasps. Using robotic grasp analysis measures, it is possible to anticipate the reaction of human grasps to externally applied loads.

MacKenzie and Iberall [81] discuss the nature of the human hand and how the brain controls it. An extensive model of grasping is developed based on behavioral, computational and biological evidence. Further effort is made to unify the issues and terminology of human grasp analysis with robot grasp analysis.

Human Hand Modeling Summary:

The study of the human hand has generally come from two approaches, the medical and the mechanical. The medical approach has focused on the interactions of the bones and tendons for the purpose of treatment and repair. The mechanical approach has focused on the external properties of the hand as a closed system. The mechanical approximations are often crude, particularly when it comes to capturing the motion of the thumb. The analysis of grasp and use in both approaches does allow us to develop and understanding of human manipulation strategies.

2.3 Calibration

Hollerbach and Wampler [47] provide an excellent overview of kinematic calibration procedures. They categorize the methods into open loop, closed loop and screw-axis measurement procedures. They also examine several solutions to issues of calibration such as multiple closed loops, parameter weighting and observability.

The closed loop method, which entails moving the mechanism through its kinematic redundancy without moving the end effector, is particularly useful, since it can be used without an external measurement system. Wampler and Hollerbach [126] present a unified formulation to calibrate kinematic mechanisms using a closed loop method. Joint measurement

errors are eliminated through a least squares iteration. Bennett and Hollerbach [7] use the closed loop calibration to calibrate the fingers of the Utah/MIT hand. One finger is rigidly connected to the thumb to form a closed loop chain. To prevent degenerate solutions, the length of one link is not calibrated, but is constrained to a known length.

The calibration of the human hand is a bit more challenging than for a robotic mechanism. The joint motions are harder to measure and the kinematics are not as well described. For many applications, a visual representation that "looks like a human hand" is sufficient. Kuch and Huang [74] developed a vision based human hand calibration method using three views of the hand. The software which comes with the CyberGlove allows the user to modify a number of parameters until the modeled hand is visually acceptable [29]. These models are not intended for manipulation, but to provide a display of human hand motions and gestures for presentation.

Performing fingertip manipulations requires greater fingertip accuracy. Rohling and Hollerbach [105] calibrate the human index finger by using a singular value decomposition to match the modeled position of the finger based on joint measurements to the tip position measured by an external sensor. Finger parameters calculated include bone lengths and angular joint offsets. In [106], Rohling and Hollerbach minimize the condition of the decomposition through parameter scaling, model reduction and pose set selection to achieve more accurate and reliable results.

Calibration Summary:

The closed loop kinematic chain calibration procedures are well developed for artificial mechanical structures, particularly when the first order approximation of the values are close to the actual values. Using such a calibration method on the human hand for the purpose of fine finger manipulation has not adequately been explored, particularly in calibrating the complexities of the human thumb.

2.4 Haptic Feedback

The creation of mechanical systems which stimulate the human sense of touch (provide haptic feedback) has opened a new channel of information which a computer can display to a user. This is particularly useful for interacting with virtual or remote worlds.

2.4.1 Human Perception

Humans rely significantly on the sense of touch to explore and interact with the world [70]. It is second only to the sense of sight in the amount of brain power devoted to it. Proper utilization of haptic feedback should improve human-computer interaction

It is clearly unnecessary to display information which the human is unable to detect. The first step, therefore, is to determine the limits of the human perception. Srinivasen and Chen [116] performed experiments to determine the just noticeable difference (JND) for contact force perception in humans. Across a range of forces and contact conditions, the JND was about 7%. Tan, Pang et al. [117] experimentally tested the JND of human perception for length, force and compliance. Length discrimination was within 1.0 mm for a 10 mm span, increasing to 2.4 mm for a reference span of 80 mm. The JND for force was about 5-10% of the reference force over a variety of forces, displacements and spans. Human subjects could detect a 5-15% difference in compliance from a reference compliance. In [118], Tan, Durlach et al. found that the JND of compliance deteriorates to as much as 99% when force and/or work cues are reduced or eliminated. In [119], Tan, Durlach et al. continue to examine the effect of work and force cues in detecting compliance by examining active pinch grasps.

2.4.2 Design of Feedback Mechanisms

The display of haptic feedback can take many forms. Some of the devices are designed to be worn by the user, while others are intended to mounted on a table or the ground. Most of the devices display forces to the user, but vibration, temperature or contact area displays are also possible.

Burdea and Zhuang [13] outline human factors influencing the design of dexterous masters, including hand geometry, force perception and particularly the effective impedance of the man-machine system. In [12], Burdea and Zhuang examine the differences in a control strategy for a dextrous master in contrast to the general case of bilateral teleoperation due to the need to model the human hand in more detail. They also review existing prototypes of dextrous masters with force feedback. An extensive overview of the state of the art of haptic interfaces in 1996 can be found in Burdea [14].

An example of a haptic feedback device which is worn by the user is the Rutgers Master II, developed by Gomez, Burdea et al. [34]. The system reads hand gestures and displays forces to four fingers in real time by means of pneumatic cylinders connected to the finger tips and grounded at the heel of the palm. Another example, which also uses pneumatics for force generation, is the exoskeleton by Hurmuzlu, Ephanov et al. [52] which provides forces to the shoulder and arm.

An example of a table mounted device is the PHANToM haptic interface designed by Massie and Salisbury [85]. The user holds a stylus on the end of motorized linkage. The device measures the stylus position and exerts a force at the stylus tip. The low inertia, low friction device can produce high fidelity forces in three dimensions.

The addition of tactile feedback, stimulation of the skin corpuscles at the fingertip, is incorporated by Shimoga, Murray et al. [113] into a VPL data glove. Tactile feedback is achieved through commercially available microactuators, called tactors, on the operator's finger tips. Touch display in dexterous telemanipulation is useful for perception of surface properties and contact constraint control. Kontarinas, Son et al. [72] present a means of displaying tactile contact information to an operator. The measured pressure distribution from a robot hand in contact with an object is measured with tactile array sensors, and a corresponding shape is created on the master manipulator tactile display.

In addition to the physical display device, it is also important to properly model the interactions with a virtual world. Maekawa and Hollerbach [82] developed a haptic display which allows the operator to receive sensations of contact with a virtual object. The operator can trace the surface, grasp and manipulate the object. The haptic display device is mounted on the Sarcos Dexterous Arm Master to allow user arm motions.

Hashimoto, Kunii et al [43] built a Dynamic Force Simulator which models object dynamics, contact model and friction characteristics of a virtual representation of the human hand interacting with objects. The simulator includes a derivation of the kinematics and forces between the hand and object, and a calculation of appropriate force feedback displayed to the user through a sensor glove.

2.4.3 Effect on Human Performance

Regarding human performance, the most important question is, “Is there a benefit to the addition of haptic feedback?” The exact answer depends on the method of feedback as well as whether the user is interacting with a virtual world or a remote location. In general, the addition of feedback improves the user’s ability to perform delicate tasks.

The use of haptic feedback in a remote manipulation task can be complicated by the presence of time delays between the master and slave. Hannaford [41] presents a linear analysis of the stability problems when using kinesthetic force feedback in remote manipulation. The results suggest a trade-off between manipulator stability and telemanipulation fidelity.

But, when Hannaford, Wood et al. [42] developed a stable teleoperation system with a dissimilar master and slave, utilizing force feedback, they asked expert operators to perform a series of generic and application-driven tasks with varying levels of force feedback and shared control. All of the performance measures improved as capability was added.

Howe [51] realized that conveying small forces and contact sensations to the operator requires careful modeling of the contact between the human finger and the master device. The addition of vibrational feedback improved the sensation of contact and allowed the display of frictional information measured by the slave. It is also possible to display the texture of a surface, as shown by Siira and Pai [114] using a stochastic modeling approach.

Several of the haptic feedback devices currently available only provide hand-grounded force feedback to the user. In these devices, when a force is displayed to the user, the reaction force is internal to the body and not applied externally to the ground. Richard and Cutkosky [103] tested whether there was a performance difference between these devices and world-grounded devices. Human subjects performed comparably well in boundary detection and size discrimination tasks using grounded and ungrounded force feedback.

Adams, Moreyra et al. [1] present an approach to guarantee that a haptic interface remains stable, while maximizing performance. The key element in ensuring stability is the virtual coupling network, an artificial link modeled as a spring and damper, between the haptic display and the virtual environment.

Haptic Feedback Summary:

There are a wide variety of haptic feedback devices which have been developed. The most common channel of haptic feedback is through the display of force, since small and efficient motors are readily available. The addition of vibrational feedback has been found to be a straightforward way of displaying transient information. The use of cutaneous feedback is limited by the inability to get a sufficient number of sufficiently strong actuators on the surface of the human fingertip. Most studies indicate that the additional channel of information improves performance in human-computer interaction.

2.5 Telemanipulation

Telemanipulation is the control of a dexterous robot hand by a human, usually by tracking human finger motions. Salisbury [109] examines several issues critical to performing telemanipulation which are not seen in traditional teleoperation. It is necessary to control redundant degrees of freedom, maintain a stable grasp and map to non-anthropomorphic mechanisms. Human control of the remote process may be performed at a variety of functional levels.

Mapping from human hand motions to robot motions is the one of the first steps in designing a telemanipulation system. Wright and Stanisic [127] compare two approaches for mapping from an EXOS hand master to a Utah/MIT dexterous hand. In the first case, the joint angles of the master are directly tracked by the robot hand. In the second case, a kinematic model of the human hand is used to calculate the finger position, and the robot finger is commanded to the same position. Rohling and Hollerbach [104] developed a process for mapping human finger positions measured with a SARCOS hand master to a Utah/MIT dexterous hand. Fischer, van der Smagt et al. [33] calibrated a data glove using a novel neural network technique. The data glove calibration leads to a mapping between the human and artificial hand workspaces for the purpose of telemanipulation. Each of these mappings takes advantage of the anthropomorphic robot hand to match fingertip positions and orientations, though some accommodation is made for differences between human and robot workspaces.

Rather than map positions, Jau [58] measures human hand forces using a sixteen degree of freedom exoskeleton controller that map directly to a sixteen degree of freedom anthropomorphic four finger robot hand. Utilizing an active compliance system, operators were able to perform tool grasping and analogous tasks.

Turki and Coiffet [121] developed hand transformations between a 14 joint dexterous hand master and a mechanical hand as a slave for telemanipulation tasks. The control of the slave is synthesized by two different methods.

The quality of the information which the user receives from the slave device should have an effect on performance, and determining the degree of effect is important for designing future systems. Massimono and Sheridan [86] performed an experimental study where human subjects performed several 'peg-in-hole'-type telemanipulation tasks with various forms of visual and force feedback. The video medium did not significantly affect task times relative to direct viewing. Buttolo, Kung et al. [17] decoupled the individual effect of the component parts of a telemanipulation system by having a group of subjects perform the same set of tasks directly on a physical setup, on a virtual implementation capable of providing visual and force feedback through an haptic display, and remotely on the real setup using a telemanipulation system.

Since it may not be possible to display all applicable information to the user, or distance may create unreasonable time delays, it is useful for the robot hand to exercise some level of intelligence to augment performance. Hirzinger, Brunner et al. [46] developed a framework to use a multisensor robot on board the space shuttle which can be controlled autonomously or be teleoperated by astronauts or by ground control. The ground control modes include a task-level oriented programming technique involving ‘learning by showing' concepts in a virtual environment. Under these control systems, the robot was able to assemble a truss structure, connect/disconnect an electrical plug and grasp free-floating objects. Li, Cox et al. [77] developed a variety of shared control techniques and optimized algorithms to enable an operator to control the slave through the master's telepresence and virtual reality equipment. The slave is a dual-arm, dual-hand robot with a stereo camera to provide an operator-centered perspective of the remote environment. The slave is capable of performing a variety of grasping and manipulating tasks.

Telemanipulation Summary:

Telemanipulation takes advantage of the human intelligence and robot capabilities. It is a useful way of increasing the capabilities of a mechanical manipulator. Many issues of time delay stability and mapping have been examined, though future improvements seem likely. For example, most of the current mappings from human to robot hand assume an anthropomorphic hand. One limitation of a direct telemanipulation system is that it requires the full time attention of a human operator, which may be inefficient if a task is largely repetitive.

2.6 Virtual Manipulation

In the application of virtual manipulation, the motion of user's fingers are tracked (perhaps by a data glove) for the purpose interacting with objects in a virtual environment. To classify it as manipulation, it must be feasible for the user to perform more than static grasping motions.

The interactions between the human fingers and a virtual object have been modeled in several ways. Kijima and Hirose [69] develop a model which allows both a dynamic interaction and a quasi-static interaction between the fingers and the object. The transitions between the types of interaction are also developed. Boulic, Rezzonico et al. [8] design a method for allowing a person wearing a digital glove to manipulate a virtual object based on the interpretation of instantaneous hand posture variations. A state controller keeps track of whether the object is free, partially constrained or fully grasped. Penetration of the commanded fingertips in a grasped object corresponds to an applied force on the object, resulting in manipulation.

Tzafestas and Coiffet [124] compute the forces applied to each individual finger of a human hand from interactions with static and dynamic objects in a virtual environment. External forces and moments are mapped through the grasp matrix to the space of the contact forces. Interaction forces are used to allow humans to grasp and manipulate objects in the virtual environment.

Pere, Gomez et al. [102] use the Rutgers Hand Master from [34] to allow users to interact with a 3-D virtual environment. Multiple users are able to interact with a single environment through an ethernet connection.

Virtual Manipulation Summary:

The tracking of human manipulation in a virtual environment has largely used a model of the forces applied by the fingers to the object to affect the object state based on dynamics. This method is useful for certain types of hand object interactions, but for fine object manipulations can lead to undesired motion. The process of in-hand manipulation by a human is quasi-static in most cases, and can be examined entirely from a kinematics angle.

2.7 High-Level Programming

In between the extremes of entirely autonomous robot hands and the direct telemanipulation there are a variety of alternatives. The basic idea is that the human can specify the desired robot hand motion at a higher level of abstraction without having to deal with some of the mapping and communication issues of direct telemanipulation.

Brunner, Arbter et al. [9] have created a shared autonomy approach that distributes intelligence between man and machine. The robot is graphically simulated within its environment, with an emulation of embedded sensor function in a task driven robot programming approach.

The tele-autonomous system developed by Conway, Volz et al. [24] is closest to autonomy on the spectrum. The robot acts autonomously, but the human is able to make adjustments during run time if the situation warrants. In [25], Conway, Volz et al. expand the framework to eliminate problems from time delays inherent in remote operation. In [35], Graves and Volz further improve the framework to receive and integrate decisions from multiple sources (such as a human and an autonomous controller). The weight placed upon each decision source can be modified dynamically to many modes of control, such as shared, teleoperative, and supervisory.

Michelman and Allen [87] develop a set of general, primitive manipulation functions which allow the robot to automatically perform the commanded object motion. The robot is programmed by defining the desired sequence of primitives.

A higher level of abstraction utilizes “virtual tools.” In the interface developed by Cannon, Thomas et al. [20], an operator specifies object positions and destinations by pointing and using verbal directives. The trajectory planning is performed by the by the robot. In [21], Cannon and Thomas expand the virtual tools framework to include the potential for supervisory and cooperative human-robot interaction. Similarly, Voyles and Khosla [125] developed Gesture-Based Programming. A robot has a repertoire of basic skills which it uses to construct a task procedure based on a human demonstration. The matching from the human to the robot is performed by an agent based program.

Pao and Speeter [100] use algebraic information of human hand positions to teach corresponding positions in a target domain, in this case a four fingered robot hand.

In many cases, the most important information to be conveyed to the robot is the desired position of an object. By utilizing a vision system, Kang and Ikeuchi [60] teach a robot by having a human demonstrate the task. The demonstrated grasp can be mapped to the robot by equating human fingers to functionally equivalent robot fingers, producing a kinematically feasible manipulator grasp and adjusting the grasp locally. In [61], Kang and Ikeuchi expand the grasp mapping procedure to consider factors from the functional and physical level. Mapping of power grasps as well as precision grasps is implemented.

In the absence of a vision system, or a local copy of the robot’s environment, it is possible to have the user demonstrate the task in a virtual setting. Kunii and Hashimoto [75] use the Dynamic Force Simulator from [43] for allowing a human to interact with a virtual object. A robot at a remote site learns the task from the human motion. Learning schemes involving neural nets and radial basis functions are used to compensate for small errors.

Kheddar, Tzafestas et al. [66] develop a high level teleoperation system where the teleoperator achieves tasks manually in a natural way within a virtual environment. The virtual tasks are being reproduced by any slave robot within the remote site, using a customized mapping. In [67][68], Kheddar, Tzafestas et al. demonstrate the high level teleoperation system to simultaneously control four kinematically different robots spread around the globe. The operator performed the task within a simulated environment, and a bi-lateral mapping for each robot was used to have each robot perform the same task.

Bruns [10][11] combines the virtual manipulation with the physical demonstration. This enables users wearing an instrumented glove to interact with a concrete object which has a corresponding virtual object in a virtual environment. The virtual objects often have a higher level meaning associated with them, such as conveyors and objects in a manufacturing plant, or stages in a process planning diagram, while the physical objects provide cutaneous feedback to the user.

High Level Programming Summary:

The use of high level programming techniques allows a system to take advantage of some human control while still utilizing robot capabilities. The particular method will depend on the intended application. Methods which use a gesture recognition program will require some training of users, while methods which use neural net learning may take a several demonstrations and a long processing time to reach a solution. The best solution would seem to be one which allows the user to make natural human motions that can be mapped intelligently to robot commands.

3 Dexterous Manipulation

The purpose of research on dexterous manipulation is to develop robot hands which approximate the versatility and sensitivity of the human hand. The research includes various aspects such as the design of robot hands with multi-degree of freedom fingers capable of applying independent forces to a grasped object and the construction of sensors on the robot fingers to provide object information.

This chapter presents research in the area of stably grasping and manipulating objects. An exploratory procedure is developed which allows a robot hand to stably manipulate an object for the purpose of detecting unknown surface features. The presentation treats haptic exploration (Section 3.1), manipulation (Section 3.2) and a discussion of experimental implementation (Section 3.3).

3.1 Haptic Exploration

Haptic exploration is the primary mechanism by which humans learn about the surface properties of objects. We use vision to discern overall shape and appearance of objects, but rely on touch to tell if objects are rough, wet, slippery, or warm. Although humans and animals use touch sensing in this way, the use of touch for robots has been quite limited, especially in comparison to developments in computer vision. The absence of touch sensing is not a serious problem in applications where object properties are predictable, but as robots start to be used in unstructured environments (for example, to explore remote planetary surfaces) the robots must identify and adapt to surface properties of the objects encountered.

A distinguishing characteristic of haptic object exploration is that it is coupled with manipulation. Haptic sensing provides information such as object weight and friction needed for stable manipulation, and manipulation allows exploration of the entire surface by the fingertips. In addition, control of contact force, position and orientation is required, so precise manipulation control is a prerequisite for tactile exploration.

3.2 Manipulation

The exploration described here proceeds as a sequence of phases in which a subset of the fingertips is used to stabilize and reorient the object while the remaining fingertips roll or slide over the surface in search of suitable contact locations for manipulating the object in the next phase. Sensors on the fingers can determine surface properties and locate features such as grooves or ridges. Studies of human exploration show a similar approach when manipulating and examining an object through the sense of touch: "...part of the hand typically stabilizes and part explores. For example, the fingers may hold the object while it is surveyed by the thumb" [70].

3.2.1 Mechanical Setup

The dexterous robot hand used in these experiments has two co-planar fingers each with three degrees of freedom. Each degree of freedom is directly controlled by a dedicated DC servomotor. The workspace of the robot hand, for manipulation purposes, is approximately 100 mm x 100 mm.

The robot fingertip positions are measured with encoders, geared down to provide a position resolution of approximately 0.05 mm. At the end of each finger is a two-axis force sensor with a sensitivity of approximately 0.1 N. One finger has a hemi-cylindrical soft fingertip covered with a textured rubber skin. The other finger has a flat, hard fingertip equipped with an 8x8 tactile array sensor.

The robot is commanded by an impedance control law, which drives the fingertips to a specified position. The closed-loop servo rate is 1000 Hz, which enables smooth motion and force control.

The table surface between the two fingers is covered with a rubber pad, and can act as an non-actuated "palm." An object can be held securely between one finger and the palm, allowing the other finger to move freely. This minimal configuration is sufficient for exploring problems associated with ensuring robust and smooth exploration of arbitrary objects, while maintaining a modest set of states and transition results.

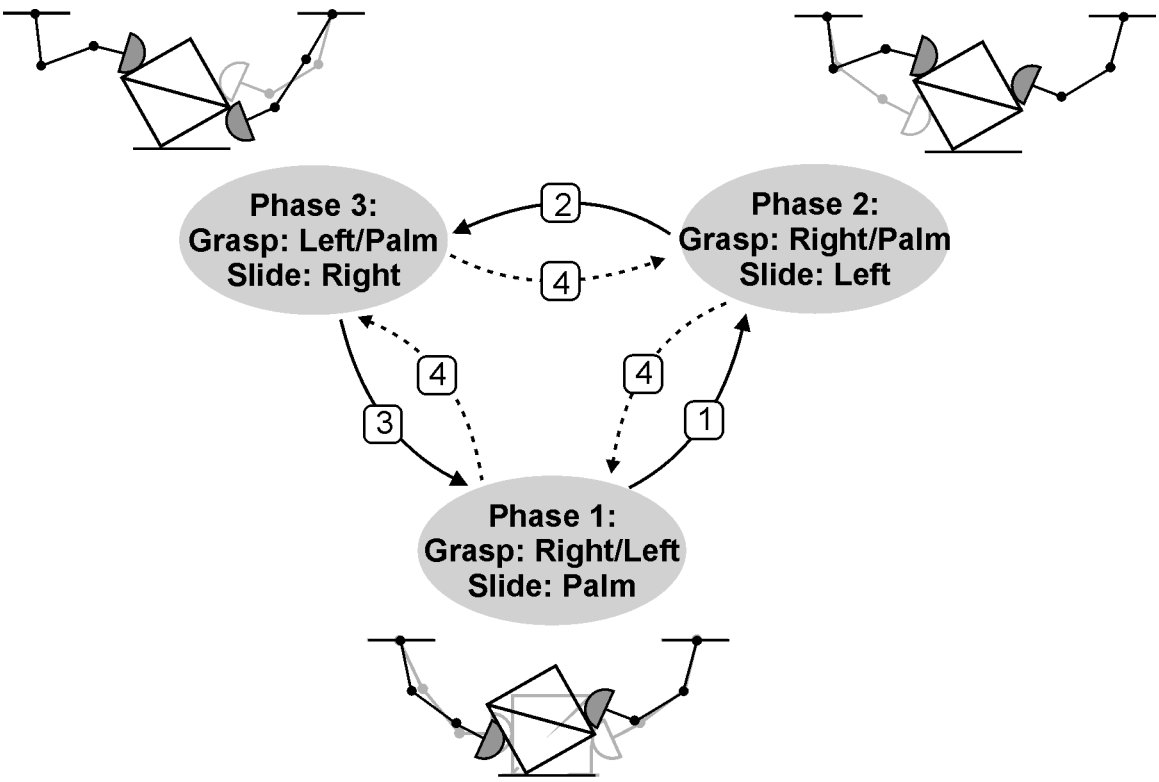

Figure 3-1. Diagram of States and Transitions During Exploratory Manipulation

3.3 Exploratory Procedure Algorithm

The description of the exploratory procedure treats state transitions (Section 3.3.1), motion planning and control (Section 3.3.2), computer simulation (Section 3.3.3) and experimental implementation (Section 3.3.4).

3.3.1 States and Transitions

The object exploration proceeds as a repeated series of phases in which some fingers manipulate the object while others traverse the object surface. The goal of the manipulation is to ensure complete and smooth traversal of a wide range of object shapes.

With the minimal testbed described above, the state transition diagram can be represented as a three phase cycle. (Figure 3-1). For the purposes of this experiment, we begin in Phase 1 and rotate the object clockwise. In practice, the cycle can begin with any phase that admits a stable grasp and proceed in either direction.

In the first phase, the object is grasped by the two fingers and rotated clockwise while optionally maintaining contact with the palm (in which case the palm could be equipped with sensors to garner object information.) The objectives are to reorient the object and to bring the left finger into a suitable location for holding the object.

In the second phase, the left finger holds the object while the right finger rolls and/or slides over the surface to a location that will allow it to hold the object in the next phase. The third phase is similar to the second; the object is held between the palm and the right finger while the left finger rolls and/or slides to a location suitable for stable two fingered manipulation.

The usual ending condition of each phase is when a finger reaches a workspace limitation. However, a phase will also terminate if the grasp is starting to become unstable. Grasp stability is computed using the method of Nakamura, Yoshikawa et al. [95], which states that a two fingered grasp is stable if the line of internal force between the two points of contact lies within both friction cones. Depending on the size and shape of the object, a phase may reach a workspace limitation before reaching a stable grasp for the next phase. In this case, it becomes necessary to modify the sequence and bypass the nominal next phase in an effort to reorient the part and obtain better contact locations. This is shown as transitions (4) in Figure 3-1.

3.3.2 Rolling and Sliding: Motion Planning and Control

During phase 1, the control variables are the object position and orientation, the distance that each contact moves over the object surface, and the internal grasp force. The object is rotated as far as possible, subject to workspace and grasp stability limits. Trajectory planning is done assuming pure rolling. This is an idealization since the fingertips are soft and have a distributed contact patch. However, as shown in Chang and Cutkosky [22], the deviations in rolling distances are negligible if contact forces are light.

During phases 2 and 3, the control variables for the moving finger include the trajectory of the contact, the orientation of the fingertip and the normal force. The specification of the fingertip orientation determines the amount of sliding (relative to rolling) that takes place. At one extreme, the orientation can be made consistent with pure rolling, and at the other the fingertip orientation can be kept constant relative to the object.

In practice, the duration of each phase is mainly a function of the workspace of the fingers. Therefore, the fingertip roll/slide ratio is planned using a simple heuristic that attempts to keep the fingers within their workspaces for as long as possible. The cartesian workspace of each finger is divided into four regions, each of which has a “preferred” fingertip orientation - one that maximizes the local configuration space. As each phase is planned, the approximate final position of the finger is mapped to one of the four work space regions and the corresponding preferred orientation is found. The orientation is then interpolated between the initial value and this final value.

The planning is done dynamically at the start of each phase, using a current estimate of the object curvature and surface orientation. When the exploration task is just beginning this estimate may be poor, in which case the phase will end quickly as the finger reaches the edge of its workspace. If the fingertip has not moved enough to grasp the object stably in the next phase, the algorithm reverts to the previous phase (transitions (4) in Figure 2). If a stable grasp within the finger workspace still cannot be found, the algorithm terminates.

3.3.3 Simulation

The experimental procedure was first simulated numerically to determine how well the algorithm would traverse a range of object shapes, including round and square objects and to test the sensitivity of the approach to workspace limits. The simulation modeled the grasp kinematics and included forces and friction coefficients in testing the grasp stability, but did not include inertial terms.

Figure 3-1 shows three phases of the simulation during a clockwise manipulation of a square object. The grey lines show the finger positions at the start of each phase and the solid lines show final positions. The final position of one phase is the starting position for the next phase. The coefficient of friction between the object and fingertips was assumed to be 1.0, a typical value for the rubber-coated fingertips used on the actual robot.

In general, the robot transitions from one stable configuration into the next. In some cases, rotation about a sharp corner on an object would drive the right finger outside of its workspace in phase 2 before it could reach stable configuration for phase 3. In several of these cases, the robot was able to recover by skipping phase 3 and rotating the object with two fingers again.

The simulation revealed that finger workspaces were the most serious limitation and led to the development of the heuristic, mentioned in the previous section, for specifying the fingertip orientation at the end of each phase. Even so, the algorithm usually could not handle long thin objects.

The simulation did not account for the effects of noisy sensors and imperfect control of the fingertip trajectories and forces. These effects were evaluated in experiments on the actual robot.

3.3.4 Experiments

As described previously, the experimental testbed is a two-fingered planar robot hand with a passive palm. The robot was controlled using dynamic object impedance control and the phase event transition framework of Hyde and Cutkosky [55].

Utilizing the exploratory procedure, the robot was able to successfully circumnavigate an approximately known round object. Sensor information from the fingers tracking the surface of the object was used to be able to detect a ridge surface feature on the object.

However, the manipulation was not performed successfully every time. The finger which is rolling and sliding along the object surface is applying a normal and a tangential force to the object during motion. For the purpose of analyzing the grasp between the two grasping fingers, these forces are considered to be external disturbance forces. If the internal force between the grasping fingers is insufficient, the external forces can induce slip at the grasp surface.

The normal and tangential forces of the moving finger are regulated by measuring force sensors on the fingertip with the intent of not applying too large of a disturbance force. However, noise in the sensors or differences between the assumed and actual shape of the object can lead to larger than intended contact forces.

Once unintended slip occurs, it is usually fatal to the manipulation process. The object may be released from the grasp, and the minimalist manipulation state system in this procedure does not have a provision for a lost grasp. While the state system could be augmented with additional states for slip conditions and dropped objects and the robot fingers could be

enhanced with slip detectors, the problem is indicative of a larger issue in dexterous manipulation.

The larger issue is that the robot is only as intelligent and versatile as the program which controls it. The program can only deal with situations that the programmer anticipated and which the robot can reliably detect. In comparison to humans, this is a grave deficiency. Humans have a wealth of experience and flexibility in manipulation "programming." The next chapters discuss research which combines the intelligence of a human with the capabilities of a dexterous robot hand, through both direct telemanipulation (Chapter 5) and demonstration (Chapter 6).

4 Human Hand Model

A kinematic human hand model is essential to both telemanipulation and programming by demonstration as explained in later chapters. This chapter develops a kinematic model of the human hand where the joints are approximated as simple hinges (Section 4.1), and a method for calibrating the model to a particular user wearing an instrumented glove (Section 4.2). It also describes an inverse kinematic method for determining joint positions based on tip position (Section 4.3).

4.1 Kinematic Model

The human hand is a remarkably complex mechanism, and researchers have made various approximations when modeling it, depending on the application. This section develops a kinematic hand model that allows joint sensor readings to be mapped to coordinate frames and joint angles. For the application here, modeling of the tendons or external appearance is not necessary. An overview of research into modeling the human hand is presented in Section 2.2.

This section develops a sufficiently accurate kinematic model of the human hand for dexterous manipulation. Using the models developed by Rohling and Hollerbach [105] and Kramer [73], observations by Cooney, Lucca et al. [26] and Hollister, Buford et al. [48], and empirical examinations, we have developed a kinematic model suited for measuring and displaying fine fingertip manipulations. In this model, the human hand is converted to a mechanical linkage, with finger bones (as the links) connected by pin joints. The model does not take into account effects such as soft tissue deformation or bone-on-bone sliding, because these effects are not observable by the glove and are assumed to cause little error in the estimated tip position.

For convenience, the base coordinate system shown in Figure 4-1 is located in the hand at the point where the thumb and the index metacarpal meet. (In the figure, the X_0, Y_0, Z_0 system is displaced from this point for clarity.) The base frame x-axis points along the index metacarpal bone, the y-axis is directed outward from a flat open palm, and the z-axis is defined by the right hand rule.

The index finger is defined similarly to that presented in Rohling and Hollerbach[106]. The index metacarpophalangeal joint has two orthogonal collocated degrees of freedom, abduction (I_{ABD}) and flexion (I_{MPJ}). The I_{MPJ}, I_{PIJ} and I_{DIJ} joints are all defined such that the axes of rotation are parallel.

The middle, ring and pinky fingers are kinematically identical to the index finger, with the bases of the fingers offset along the z-axis. The model does not currently include palmar arching, the rounding of the back of the hand which allows a human to bring the pinky into opposition contact with the thumb. The absence of palmar arch is primarily due to the fact that the CyberGlove does not measure the value reliably.

Modeling the thumb is more challenging. Cooney, Lucca et al. [26] show that even though the metacarpal bone has three modes of motion (flexion/extension, abduction/adduction

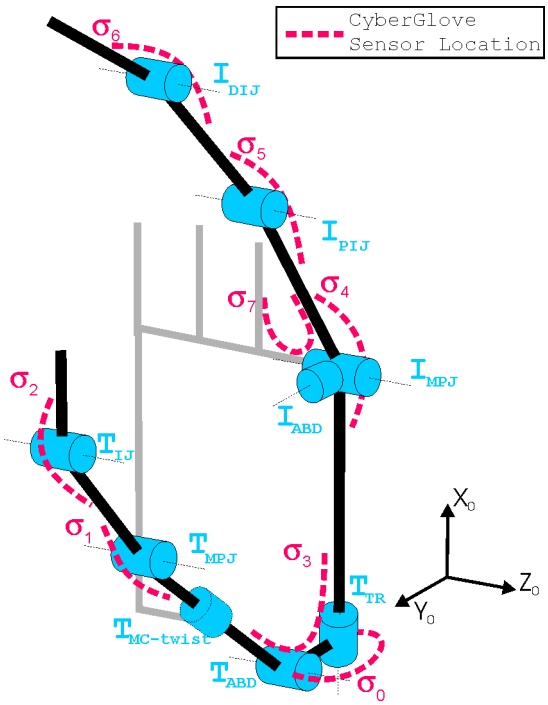

Figure 4-1. Kinematic Model of Human Hand

and pronation/supination), it only has two degrees of freedom. In other words, for a given flexion (rotation about the palm) and abduction (angle between the thumb and palm), the thumb has a unique pronation (twist). Hollister, Buford et al. [48] demonstrate that this is because the thumb rotation (T_{TR}) axis and the thumb abduction (T_{ABD}) axis are non-orthogonal and non-intersecting.

The model in Figure 4-1 maintains orthogonality for computation purposes, and to better match the sensor positioning of the CyberGlove. The T_{TR} joint is located at the base of the thumb with the axis of rotation along the index metacarpal. The T_{ABD} axis is offset from and orthogonal to the T_{TR} axis. In order to account for the thumb pronation, an unsensed axis is placed along the thumb metacarpal ("$T_{MC\ twist}$"). The angle of this "joint" is a linear function of the abduction and flexion.

Some hand models [74] place a spherical joint at the metacarpophalangeal joint (T_{MPJ}). One of the axes of motion, however, exhibits little motion without being externally forced. The "principal" T_{MPJ} axis is parallel to the T_{ABD} axis for a 0° $T_{MC\ twist}$. The interphalangeal joint (T_{IJ}) is parallel to the T_{MPJ}.

A complete listing of the forward kinematics transformations and Jacobians can be found in Appendix A.

4.2 Hand Calibration

This section describes a two fingered calibration (Section 4.2.1) and four fingered calibration (Section 4.2.2). The four fingered calibration uses the two fingered calibration on a sequence of finger pairs.

4.2.1 Two Fingered Calibration

A reliable and expedient means of calibrating the user's hand for the CyberGlove has been developed. A computer routine has been developed which can be performed quickly and accurately each time the device is used, so a new user can begin to work in a matter of minutes.

The current calibration routine is called a "zero-hardware solution," in which the user places the thumb and index finger tips together and maintains rolling contact while moving the fingers. The computer records 80 distinct data points with the CyberGlove sensor values

over 40 seconds. Because the fingers remain in contact, the hand can be approximated as a closed rigid-body kinematic chain with the fingertips in contact at a fixed point. This is only an approximation because of several factors, including rolling motion and soft tissue deformation. This closed kinematic chain has one unsensed joint at the finger-thumb contact point, with three rotational degrees of freedom.

The error at each data point is the calculated separation between the two fingers in the nominal kinematic model. A least squares regression iteration is used to modify the nominal model to an error-minimizing model for the particular user. The major advantage of the zero hardware solution is the lack of any external calibration sensor, such as a vision system, which would increase complexity and cost.

The motivation for this approach follows Rohling and Hollerbach [105], who calibrate the human index finger using a least squares regression with a vision system to measure the tip position and close the kinematic chain, and Bennet and Hollerbach [7] who calibrate the Utah/MIT hand by rigidly connecting the finger to the thumb to form a closed kinematic chain.

4.2.1.1 Angular Offset Calibration

In a calibration routine, each additional parameter increases the complexity and the computation time. The goal is to find the calibration routine with the fewest parameters which captures the behaviour of the system. As a first order approach, a calibration is developed which only optimizes the angular offset parameters for each joint. The equation for joint angle ϕ_i is

$$\phi_i = g_i\sigma_i + \theta_i \quad \textbf{(4-1)}$$

where σ_i is the raw sensor value, g_i is the gain and θ_i is the offset value. The θ_i value is calibrated to "zero" the angle appropriately.

It is assumed in this approach that the sensor gains are fairly constant from user to user. The CyberGlove manual claims "A unique property of CyberGlove's proprietary sensor design provides that as long as the sensor completely covers the arc of the joint between adjacent bone segments, the sensor will provide an output proportional to the angle between the bones." [29]

Also implicit in this approach is the assumption that while human hand sizes vary, the relative length of the bones is approximately constant. An, Chao et al. [6] examined a variety of "normal" human hands and found the ratio of bone lengths to the index metacarpal bone length to only vary by 1 to 2%.

Initially, there is a "default" model which represents an average human hand typical sensor gains and offsets. The relative length of each bone is taken from published biomechanics data [6]. The initial sensor gains and offsets are found by using a graphical display of the hand model and manually adjusting the parameters to produce a visually acceptable representation of the human hand across a variety of users.

After storing N poses with the thumb and index finger in a closed loop, position error vectors $\Delta\mathbf{d} \in \Re^3$ are generated for each data point n from the calculated index position $\mathbf{d}_I$ to the corresponding calculated thumb position $\mathbf{d}_T$, using the forward kinematics of the hand model.

$$\Delta\mathbf{d}_n = \mathbf{d}_{T_n} - \mathbf{d}_{I_n} \tag{4-2}$$

The hand model kinematics also provide Jacobians, $\mathbf{J}_\phi$, relating joint angle velocity to Cartesian space linear velocity; these Jacobians are functions of the joint angles, $\boldsymbol{\phi}$.

$$\dot{d}_n = \mathbf{J}_\phi(\boldsymbol{\phi}_n) \cdot \dot{\phi}_n \tag{4-3}$$

The Jacobian relation can be rewritten for infinitesimal motions and vertically concatenated from 1 to N to get

$$\begin{bmatrix} \Delta\mathbf{d}_1 \\ \cdots \\ \Delta\mathbf{d}_n \\ \cdots \\ \Delta\mathbf{d}_N \end{bmatrix} = \begin{bmatrix} \mathbf{J}_\phi(\boldsymbol{\phi}_1) \\ \cdots \\ \mathbf{J}_\phi(\boldsymbol{\phi}_n) \\ \cdots \\ \mathbf{J}_\phi(\boldsymbol{\phi}_N) \end{bmatrix} [\Delta\boldsymbol{\theta}] \tag{4-4}$$

or, more compactly, $\Delta\mathbf{d} = C\Delta\boldsymbol{\theta}$, where $\boldsymbol{\phi}_n$ is the set of joint angles for pose n, $\boldsymbol{\theta}$ is a vector of angular offsets applied to every set of $\boldsymbol{\phi}_n$, and C is the matrix of concatenated Jacobians. The least squares solution to Eq. 4-4 is the value of $\Delta\boldsymbol{\theta}$ which minimizes

$$\frac{1}{2} \cdot (\Delta\mathbf{d} - \mathbf{C}\Delta\boldsymbol{\theta})^T \cdot (\Delta\mathbf{d} - \mathbf{C}\Delta\boldsymbol{\theta}) \tag{4-5}$$

Since **C** is not square, there is not a unique solution such that the equation is equal to zero. To find the minimum, we take the partial derivative with respect to $\Delta\boldsymbol{\theta}$ and set it equal to zero.

$$\frac{\delta}{\delta\Delta\boldsymbol{\theta}}(\Delta\mathbf{d} - \mathbf{C}\Delta\boldsymbol{\theta})^T \cdot (\Delta\mathbf{d} - \mathbf{C}\Delta\boldsymbol{\theta}) = 0 \quad \textbf{(4-6)}$$

$$\frac{\delta}{\delta\Delta\boldsymbol{\theta}}(\Delta\mathbf{d} - \mathbf{C}\Delta\boldsymbol{\theta})^T \cdot \Delta\mathbf{d} - (\Delta\mathbf{d} - \mathbf{C}\Delta\boldsymbol{\theta})^T \cdot \mathbf{C}\Delta\boldsymbol{\theta} = 0 \quad \textbf{(4-7)}$$

$$(\Delta\mathbf{d} - \mathbf{C}\Delta\boldsymbol{\theta})^T \cdot \mathbf{C} = 0 \quad \textbf{(4-8)}$$

$$\Delta\boldsymbol{\theta} = (\mathbf{C}^T \cdot \mathbf{C})^{-1} \cdot \mathbf{C} \cdot \Delta\mathbf{d} \quad \textbf{(4-9)}$$

We can evaluate a least-squares solution, $\Delta\boldsymbol{\theta}$, to this linearized system using the left pseudo-inverse of **C**, based on the SVD. At each iteration step, $\boldsymbol{\theta}$ is modified by $\Delta\boldsymbol{\theta}$. The values of $\Delta\mathbf{d}$ and **C** are recalculated, using the new $\boldsymbol{\theta}$. The iteration continues until $\Delta\boldsymbol{\theta} \rightarrow 0$. An extensive discussion on the numerical stability of this convergence is presented by Rohling and Hollerbach [105].

4.2.1.2 Including Unknown Sensor Gains

Implementation of the angular offset calibration resulted in poor accuracy of the modeled hand. Because the CyberGlove attaches to the soft tissue of the hand, and due to the nature of the sensors, the conversion gain from sensor value to angular quantity is variable from user to user, unlike in an exoskeleton-type hand master. In addition, some glove sensors are physically cross-coupled, i.e., sensor values may change due to the movement of more than one joint.

The solution is to expand the calibration to optimize the values of the sensor gains, using the relation

$$\phi_i = g_i\sigma_i + \theta_i \quad \textbf{(4-10)}$$

where σ_i is the raw sensor value, and g_i is the gain for sensor i. A new Jacobian is formed by adding columns for the new gain parameters:

$$\Delta d_n = \hat{\mathbf{J}}_n \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_\phi & \mathbf{J}_\phi \cdot \mathrm{Diag}(\boldsymbol{\sigma}_n) \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{g} \end{bmatrix} \quad \textbf{(4-11)}$$

Note that the individual $\hat{\mathbf{J}}_n$ has linearly dependent columns, however the concatenated Jacobian matrix $\hat{\mathbf{C}}$ will not, because each set of sensor readings $\boldsymbol{\sigma}_n$ is distinct.

The sensors on the glove are not physically isolated from each other, since they are all stitched to the fabric of the glove. Movement of the hand about one axis can cause a change in the reading of more than one sensor. For instance, motion by the thumb about the T_{ABD} axis will cause some change in the bending (and therefore the signal) of the T_{TR} sensor. Cross-coupling effects are modeled by including a cross gain parameter. In general, the angle of a joint can be written as offset plus the sum of a gain times the signal for every sensor.

$$\phi_i = \theta_i + \sum_{j=1}^{S} g_i^j \cdot \sigma_j \qquad \textbf{(4-12)}$$

Where S is the number of sensors, and g_i^j is the gain from sensor j for joint i. Fortunately, $g_i^j = 0$ where $i \neq j$ for most cases. (Flexing the interphalangeal joints on the index should not noticeably affect sensor readings on the thumb.) The Jacobian matrix $\hat{\mathbf{J}}$ in Eq. 4-11 is augmented with columns of the form:

$$J_{\phi_i} \cdot \sigma_j \qquad \textbf{(4-13)}$$

for each cross gain g_i^j added to the parameters vector, where J_{ϕ_i} is the column of J associated with ϕ_i. At this point, only four cross coupling terms are used $g_{T_{TR}}^{T_{ABD}}$, $g_{T_{ABD}}^{T_{TR}}$, $g_{T_{MCtwist}}^{T_{ABD}}$ and $g_{T_{MCtwist}}^{T_{TR}}$.

$\hat{\mathbf{J}}_n$ is further expanded to include relations between bone lengths and fingertip positions. The Jacobian for the bone length $\mathbf{J}_L$ is calculated as if a prismatic joint was located on the bone.

Bone lengths, gains, and offsets are then be incorporated into a generalized parameter vector p. In summary we are now calibrating for:

nine constant offset parameters of the model: θ_i

eight bone lengths, L_i

eight sensor gains, g_i

and four cross-coupling terms, g_j^i

The combined Jacobian equation for each data point is:

$$\Delta d_n = \hat{\mathbf{J}}_n \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{g} \\ \Delta\mathbf{L} \\ \Delta\mathbf{g}_{cross} \end{bmatrix} \tag{4-14}$$

or in more detail:

$$\Delta\mathbf{d}_n = \begin{bmatrix} \mathbf{J}_\phi & \mathbf{J}_\phi \cdot \mathrm{Diag}(\boldsymbol{\sigma}_n)\, \mathbf{J}_L \end{bmatrix} \begin{bmatrix} J_{\phi_{T_{ABD}}} \cdot \sigma_{T_{TR}} & J_{\phi_{T_{TR}}} \cdot \sigma_{T_{ABD}} & J_{\phi_{T_{MCtwist}}} \cdot \sigma_{T_{TR}} & J_{\phi_{T_{MCtwist}}} \cdot \sigma_{T_{ABD}} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{g} \\ \Delta\mathbf{L} \\ \begin{bmatrix} \Delta g^{T_{TR}}_{T_{ABD}} \\ \Delta g^{T_{ABD}}_{T_{TR}} \\ \Delta g^{T_{TR}}_{T_{MCtwist}} \\ \Delta g^{T_{ABD}}_{T_{MCtwist}} \end{bmatrix} \end{bmatrix} \tag{4-15}$$

When this revised calibration was performed, the hand parameters converged quickly to trivial solutions. In one case, all the gains converge to zero, eliminating all finger motion. The other parameters converge to a configuration where the finger offset is exactly zero. Because the fingers do not move, the error is zero for all sensor readings. In a second case, all bone lengths converge to zero, similarly eliminating all motion.

To ensure a convergence to a biologically feasible model, it is necessary to place constraints on the acceptable values of each parameter.

4.2.1.3 Limitation of Parameter Deviation

As mentioned previously, a graphical display is used to determine "default" values of angular offset $\boldsymbol{\theta}$ and gain $\mathbf{g}$. This type of manual calibration, however, produces spatial accuracy which is not sufficient for fine telemanipulation tasks. These values are used as a first approximation of the physical values, and are used to generate a nominal "biologically feasible" parameter set from which we can limit parameter variance.

To complement our generalized parameter vector $\mathbf{p}$, we introduce two new vectors, $\mathbf{p_0}$ and $\boldsymbol{\rho}$, which are generalized vectors of nominal values, and acceptable variance of each param-

eter, respectively. In this way, the nominal value and acceptable deviation of each parameter can be controlled independently. The equation

$$[\mathbf{p_0} - \mathbf{p}] = \mathbf{I}\Delta\mathbf{p} \tag{4-16}$$

has the effect of driving $\mathbf{p}$ toward $\mathbf{p_0}$, if iterated and $\Delta\mathbf{p}$ is applied to $\mathbf{p}$. The rate of convergence can be modified by left multiplying both sides by any nonsingular matrix $\mathbf{V}$. We choose $\mathbf{V}$ to be a diagonal square matrix with elements

$$V_{ii} = -\sqrt{N}\frac{\partial}{\partial p_i}\left[\left(\frac{p_{0i} - p_i}{\rho_i}\right)^m\right] \tag{4-17}$$

where m is an integer. This $\mathbf{V}$ matrix restricts each parameter to a virtual potential well about the nominal value when incorporated into the least squares fit, equation (7).

$$\begin{bmatrix} \Delta\mathbf{d}_1 \\ \cdots \\ \Delta\mathbf{d}_N \\ \mathbf{V}\cdot(\mathbf{p_0} - \mathbf{p}) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{J}}(\phi_1, \mathbf{p}) \\ \cdots \\ \hat{\mathbf{J}}(\phi_N, \mathbf{p}) \\ \mathbf{V} \end{bmatrix} [\Delta\mathbf{p}] \tag{4-18}$$

Eq. 4-18 limits the variance of our parameters to pre-defined ranges, and weights the variance equally with the position error, $\Delta\mathbf{d}$. The potential well function can also be adjusted by the choice of m; at m = 10, when the difference between a parameter, p, and its default value, $\mathbf{p_0}$, is $0.9 \cdot \rho$, the corrective $\Delta\mathbf{p}$ is equivalent to a positional error of 0.34mm, but if the difference is $1.5 \cdot \rho$, the corrective $\Delta\mathbf{p}$ is equivalent to a positional error of 57.0mm. It is important for numerical stability that each p_i not stray too much outside the variance. This is avoided by reducing the step size on the optimization iteration adaptively.

A more detailed mathematical representation of the calibration process can be found in Appendix B.

4.2.1.4 Performance Metrics and Experimental Results

An initial measure of calibration quality can be obtained by examining the final calibration set of separation vectors $\Delta\mathbf{d}$ from index tip to thumb tip. The RMS calibration error, e, is defined as the RMS of the set of N distances between the fingertips, one for each pose. The separation between fingertips, however, cannot be measured exactly because of two important effects. The first limitation is the resolution of the CyberGlove, e.g., the I_{MPJ} joint resolution is approximately ±0.85°, which corresponds to a spatial position precision no better

than 1mm in some hand configurations. Secondly, when actually in contact, the fingertips deform depending on the pinch force exerted. From empirical measurements, we estimate that this incurs an RMS separation uncertainty of approximately 2 mm, with peak uncertainty of approximately 3 mm. Therefore, because of these two effects we expect an ideal calibration technique to produce an error on the order of 2-4 mm.

The ten users in the experiments described in Section 5.2 were calibrated using the full calibration technique. For these users, the average error was 5.26 mm, with a standard deviation across users of 1.40 mm and worst-case value of 7.8 mm. In contrast, when calibrating with angular offsets only, error values range from 11 mm up to 25 mm in the worst case. The difference between the two calibration methods was readily apparent by visual inspection of the graphical hand model display. Use of the offsets-only technique resulted in visually incongruous hand configurations.

To further compare the two methods, a second performance criterion is introduced, with telemanipulation in mind. The goal of the calibration technique is to provide accurate information about the relative positioning of the user's fingertips, in particular the separation distance between them. To this end, a test was performed to compare the calculated separation to the actual separation of the user's fingers.

While the user manipulated a thin rod of known length between the index and thumb fingertips, a set of poses were recorded. Rods of length 12.5, 25.5, 40.5, 50, and 64 mm were used. The 12.5 mm rod was difficult to manipulate and therefore prone to error. Figure 4-2 demonstrates the relationship between actual and calculated separation for both calibration techniques on a single subject. The error bars correspond to one standard deviation in a set of 300 poses across the workspace of their hand. The linear regression fit is constrained to pass through the origin.

The full calibration technique shows substantially better linearity than the offsets-only technique. Linearity is more important than exact size, because the scale of the user's hand is unknown. Good linearity directly corresponds to a better measurement of the size of a virtual object grasped by the hand.

The analyses above only show how well the calibration procedure reduces the tip error. It does not demonstrate that the model has converged to an accurate representation of the

human hand, nor does it necessarily demonstrate that the ultimate configuration is the global minimum and not a local minimum.

To test whether the calibration converged to an accurate and global minimum, it was necessary to generate a “perfect” set of poses with a known minimizing solution. Using the kinematic model of the hand described in 4.1 and known parameter values, a data set of 80 poses was generated such that there was zero error in relative tip position at each pose. If the calibration procedure is accurate, it should converge to this known configuration using the perfect pose set since there is no difference between the measured and calibrated kinematic model, zero sensor noise and an existing exact solution. The known solution was placed in the center of the parameter deviation potential well, so that the parameter deviation error is zero as well as the tip separation error.

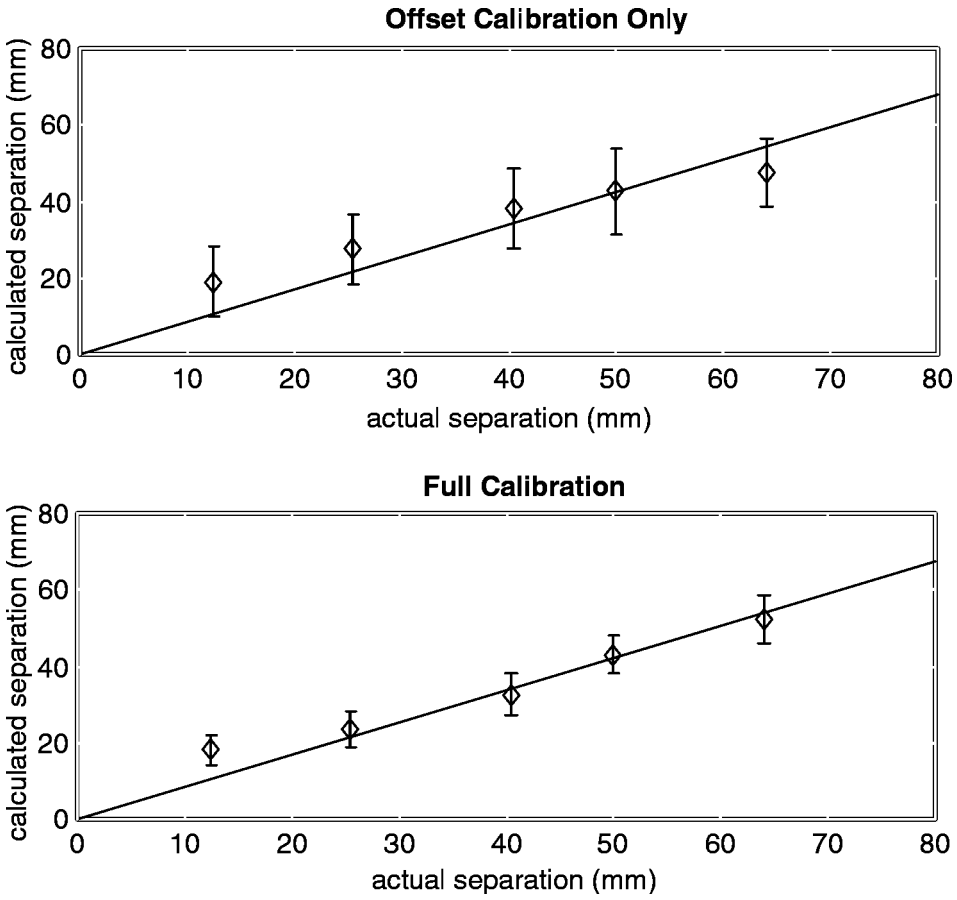

Figure 4-2. Comparison of Actual and Calculated Finger Separation of a Calibrated Subject

When tested, the calibration routine did not converge to the known solution. Even if the initial values of the model were set to those of the known solution, such that the initial tip error and deviation error were both zero, the model would diverge and settle at a configuration with a non-zero RMS error. The calibration is still dominated by a tendency toward the trivial solution where all the link lengths are set to zero, and settles on a solution which balances this tendency with the parameter deviation constraint.

To understand this phenomenon, it is easiest to visualize the calibration in a single dimension. Assume for the moment that all of the link lengths are represented by a single value L. The RMS tip error is essentially zero at both the default value of L_0 and at L = 0. The reason the calibration does not converge to the local minimum at L_0 is that the least squares minimization (Eq. 4-5) is dependent on the Jacobian as well as the tip error.

At L = 0, the RMS tip error is exactly zero to the limits of machine precision, so the change in each parameter is exactly zero. However, at $L = L_0$, the RMS tip error is zero only to the precision of the input data (typically on the order of 10^{-8}). As result, the change in each parameter is not zero. The value of Eq. 4-5 is not exactly zero at $L = L_0$. Therefore, the least squares solution tends toward the trivial solution.

The addition of the parameter deviation constraint prevents the solution from converging all the way to the trivial solution. But at $L = L_0$, the equation for the parameter deviation is zero to the tenth derivative. So, the calibration moves to a solution which balances the trivial solution and the parameter deviation, instead of the “correct” solution.

Rohling and Hollerbach prevent closed-loop trivial solutions by fixing a gain and a link length [7]. Fixing a link length to a non-zero value eliminates the solution where all of the link lengths converge to zero, and fixing a gain eliminates the solution where all of the gains converge to zero. For the purpose of examining convergence on the perfect data set, it is largely arbitrary which gain and link are set. Intelligent selection of the fixed gain and link on real data will be discussed later.

Once a link length and gain have been fixed, the behaviour of the calibration routine becomes dependent upon the nature of the errors in the model. If the only errors between the initial model estimate and the correct answer for the perfect data set are in the link length parameters, the model converges to the correct solution. However, if there are errors

in the gain or angular offset parameters, the model will oscillate about the correct solution. The lack of convergence is caused by the difference in scale of the parameters. A single unit change in link length (1 mm) has a smaller effect than a unit change in angular offset (1 radian) or gain (1 radian/volt). However, if the difference in effect is not explicitly accounted for, the left inverse of the Jacobian will result in a preferential change in link length values.

For example, if the only error between the initial model and the correct model is a 10 mm error in the length of the index proximal phalange, in the first iteration the change in L_1 is nearly 10 mm and the other changes are essentially zero. Within a few iterations, the models match exactly. However, if the only error between the initial model and the correct model is a 0.1 radian error in the angular offset of the I_{MPJ} joint, in the first iteration the change in θ_{IMPJ} is much smaller than 0.1 radians and the changes in index finger link lengths are on the order of 1 to 2 mm. The result is that the model does not converge exactly.

It is necessary to place a weighting upon the changes in each parameter, so that relative changes in parameter values are proportional to their impact upon the error. We represent this by minimizing

$$\frac{1}{2} \cdot \Delta \mathbf{p}^T \cdot \mathbf{W} \cdot \Delta \mathbf{p} \qquad \textbf{(4-19)}$$

Where $\mathbf{W}$ is a square matrix such that $W_{ii} = \left(\frac{1}{w_i}\right)^2$ and w_i is the change in parameter i which will cause a 1 mm change in tip position. Taking the partial derivative with respect to $\Delta\mathbf{p}$ yields

$$\Delta \mathbf{p}^T \cdot \mathbf{W} = 0 \qquad \textbf{(4-20)}$$

We add this constraint to the least squares equation from Eq. 4-8.

$$(\Delta \mathbf{d} - \mathbf{C}\Delta \mathbf{p})^T \cdot \mathbf{C} + \Delta \mathbf{p}^T \cdot \mathbf{W} = 0 \qquad \textbf{(4-21)}$$

$$\Delta \mathbf{p} = (\mathbf{C}^T \cdot \mathbf{C} - \mathbf{W}^T)^{-1} \cdot \mathbf{C}^T \cdot \Delta \mathbf{d} \qquad \textbf{(4-22)}$$

With the addition of the weighting matrix, the model converges to the correct solution.

4.2.2 Four Fingered Calibration

A method was also developed for calibrating four fingers of the hand by using pairs of fingers and the two fingered calibration approach. The four fingers are the thumb, index,

middle and ring fingers. The pinky finger is excluded due to the lack of a palmar arch sensor and an absolute abduction sensor. Palmar arch has a much greater effect on the orientation of the pinky finger MPJ joint than on the MPJ joint of the other fingers. In addition, the lack of an absolute abduction sensor means that a movement of the fingers which kept the same relative abduction between them would not be measured by the glove. With the absence of the pinky as an active finger, the relative abduction between the ring and pinky fingers is treated as an absolute measurement.

Clearly, it is not possible to calibrate all four fingers as a single kinematic loop. It also not possible to bring all four finger tips together to form a single linkage with multiple loops. It is difficult to bring all four fingers to a position where their tips can approximated as a single point. Furthermore, a hand in such a configuration has such a limited range of motion that achieving a sufficient diversity of data points impossible.

The remaining solution, without resorting to an external measurement system, is to perform the calibration on three closed loop "chains." One data set is gathered with the thumb and index finger tips together as described above. A second data set is gathered with the thumb and the middle finger tips together. The fingers move through the null space while keeping the fingers together. The sensor values are recorded, identical to the procedure for the index and thumb. The third and final set has the thumb and ring finger tips together.

4.2.2.1 Modeling Middle or Ring Finger

Adapting the two fingered calibration routine to be performed with the middle or ring finger is straightforward. With the assumption of no palmar arch stated in 4.1, the modeled middle and ring fingers are kinematically parallel to the index finger for which the routine was developed. The only additional parameter which needs to be added to the list is a z-axis offset from the XY plane to the MPJ of the finger. (The offset is 0, by definition, for the index finger).

4.2.2.2 Abduction

As mentioned previously, the CyberGlove does not have an absolute abduction sensor. For the two fingered calibration, the relative abduction sensor reading between the index and middle finger was assumed to be an absolute measurement of index finger abduction. This

assumption is reasonable, provided the user keeps the middle finger fairly stationary during calibration and use.

In the four fingered calibration, the relative abduction sensor reading between the ring and the pinky finger is assumed to be an absolute measurement of the ring finger abduction. The absolute abduction of the middle finger is assumed to be the sum of the ring finger abduction and the relative abduction between the ring and middle. Similarly, the index finger abduction is assumed to be the sum of the middle finger abduction and the relative abduction between the middle and index.

This is not an ideal assumption, however it is a necessary one given the equipment. An effort to place an absolute sensor, such as along the outside of the pinky finger, would be too cross coupled with the MPJ flexion motion to be useful. Again, provided the pinky finger remains fairly static during calibration and use, the assumption is adequate.

4.2.2.3 Thumb variation

When a two fingered calibration is performed on each data set, the thumb does not converge to identical parameter values when calibrated with the index as it does with the middle or ring. Figure 4-3 shows a sample of the 17 thumb parameters for the three different methods.

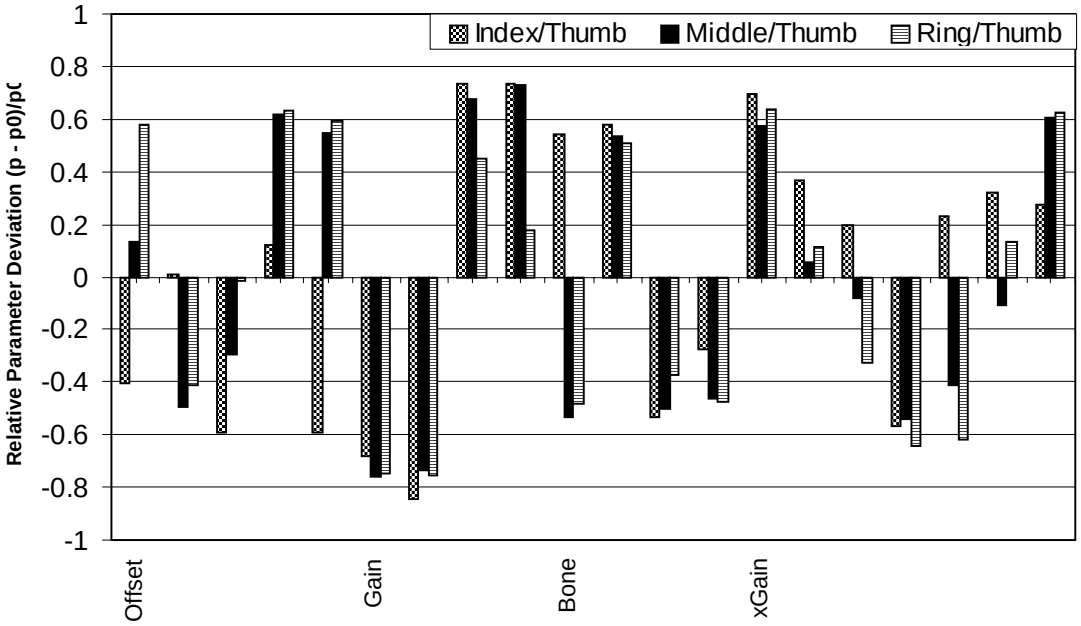

Figure 4-3. Calibrated Thumb Parameters for Different Methods

In general, the bone lengths of the thumb increase and the T_{MC_Twist} parameters increase in the thumb/ring finger calibration to extend further across the palm.

One method is to calibrate the thumb to a configuration which minimizes the error in all three data sets after the individual closed loop calibrations are performed. The parameters of the non-thumb fingers are held fixed, and the calibration is performed, iterating only on the values of the thumb parameters. Essentially, we are using our measurements of the other fingers as an external measurement of the thumb position.

The result of this method was a typical RMS error close to 20 mm and occasional thumb configurations which did not visually appear to be accurate. The high error and poor configuration are a result of needing the calibration of the thumb to compensate for the errors in the finger calibrations.

4.2.2.4 Simultaneous Four Finger Calibration

To improve the overall accuracy, it is necessary to calibrate all the fingers simultaneously. This is done by concatenating all three closed loop data sets. It is essentially solving three series of Eq. 4-18 together and looks like

$$\begin{bmatrix} \Delta\mathbf{d}_1 \\ \Delta\mathbf{d}_2 \\ \Delta\mathbf{d}_3 \\ \mathbf{V}_I \cdot (\mathbf{p}_{I,0} - \mathbf{p}_I) \\ \mathbf{V}_M \cdot (\mathbf{p}_{M,0} - \mathbf{p}_M) \\ \mathbf{V}_R \cdot (\mathbf{p}_{R,0} - \mathbf{p}_R) \\ \mathbf{V}_T \cdot (\mathbf{p}_{T,0} - \mathbf{p}_T) \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{J}}(\phi_1, \mathbf{p})_I & 0 & 0 & \hat{\mathbf{J}}(\phi_1, \mathbf{p})_T \\ 0 & -\hat{\mathbf{J}}(\phi_2, \mathbf{p})_M & 0 & \hat{\mathbf{J}}(\phi_2, \mathbf{p})_T \\ 0 & 0 & -\hat{\mathbf{J}}(\phi_3, \mathbf{p})_R & \hat{\mathbf{J}}(\phi_3, \mathbf{p})_T \\ \mathbf{V}_I & 0 & 0 & 0 \\ 0 & \mathbf{V}_M & 0 & 0 \\ 0 & 0 & \mathbf{V}_R & 0 \\ 0 & 0 & 0 & \mathbf{V}_T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{p}_I \\ \Delta\mathbf{p}_M \\ \Delta\mathbf{p}_R \\ \Delta\mathbf{p}_T \end{bmatrix} \quad \textbf{(4-23)}$$

where $\Delta\mathbf{d}_j$ and ϕ_j are the tip error and joint angles, respectively, for the index-thumb data set when j = 1, the middle-thumb data set when j = 2 and the ring-thumb data set when j = 3. $\mathbf{V}_f$, $J(\phi, \mathbf{p})_f$ and $\Delta\mathbf{p}_f$ are the parameter deviation matrix, Jacobian and change in parameters, respectively, associated with the parameters for the index (f = I), middle (f = M), ring (f = R) and thumb (f = T). Note that the Jacobians for the parallel fingers are negative, because the error vector is $\Delta\mathbf{d} = \mathbf{d}_T - \mathbf{d}_f$.

The result is a significant improvement in RMS error at the tip positions (~12 mm instead of ~20 mm), and a more visually acceptable model of the hand.

The effectiveness of this procedure relies upon the assumption that the correlation between sensor value and angle is a linear equation. This appears to be a reasonable assumption, with the exception the T_{TR} sensor (the base joint of the thumb). The sensor does not appear to be linear across its range, and as a result tends to calibrate to a higher gain. The problem can be seen on the graphical display, as small movements of the user's thumb about the TR axis result in noticeably large movements of the modeled thumb. We overcome this by taking advantage of the need to fix one of the sensor gains. By observation, a sensor gain value for the TR sensor is selected which gives the best fit across the range of motion, and then fixed during calibration. The resulting RMS tip errors are still on the order of 12 mm, but the 'hyperactivity" of the thumb is notably decreased.

4.3 Inverse Kinematics

In Chapter 6, we describe a virtual representation of the human hand interacting with a virtual object with rolling and sliding. In order to display the hand, it is necessary to have an inverse kinematics method to determine the joint angles of a finger based on a desired tip position.

4.3.1 Index, Middle and Ring Fingers

As mentioned previously, the index, middle and ring fingers are kinematically similar. The following equations utilize the subscript 'I' for the index finger, though the same method is used for solving the inverse kinematics for the middle and ring. Each finger has four joints, though it does not have four independent degrees of freedom. It is very difficult to flex the I_{DIJ} without flexing the I_{PIJ}. When the finger is not constrained by external forces, it is a reasonable approximation [6] to state that:

$$\phi_{I_{DIJ}} = \frac{2}{3}\phi_{I_{PIJ}} \qquad \textbf{(4-24)}$$

With this assumption, the finger is modeled as a three degree of freedom mechanism. We now develop a means of determining a unique joint configuration,

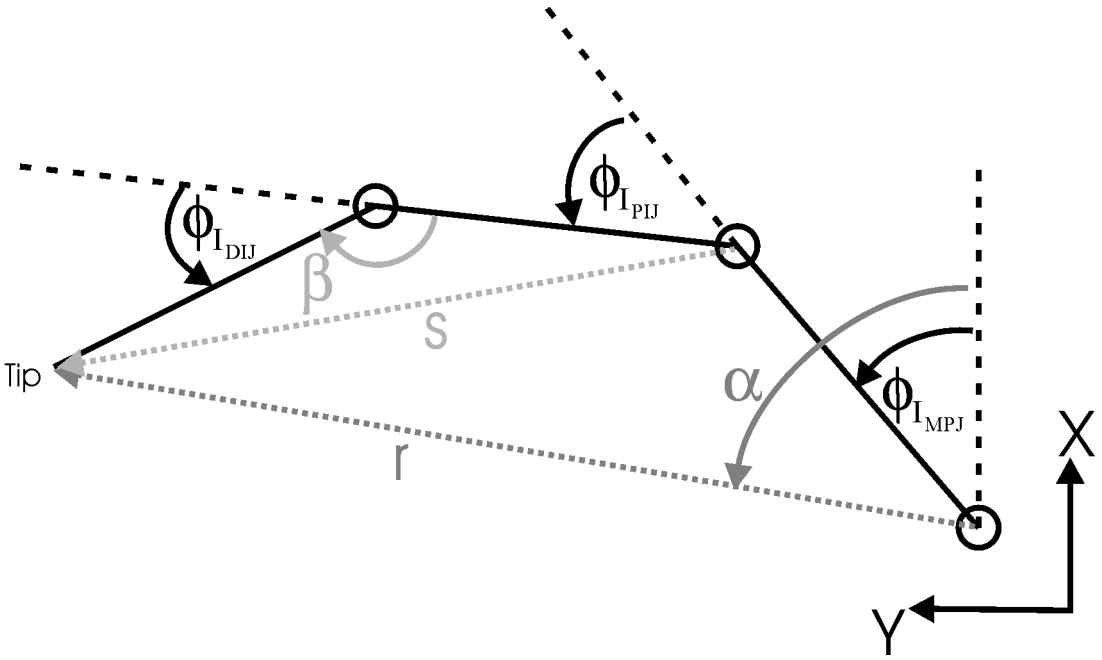

Figure 4-4. Inverse Kinematics of Parallel Joints on Finger

$\boldsymbol{\phi}_I = \left[\phi_{I_{ABD}}\ \phi_{I_{MPJ}}\ \phi_{I_{PIJ}}\ \phi_{I_{DIJ}}\right]^T$ to achieve a tip position ${}^0\mathbf{d}_{I_{tip}} = \left[x_{tip}\ y_{tip}\ z_{tip}\right]^T$ in the hand coordinate frame from Figure 4-1.

4.3.1.1 Abduction joint

The first joint to be determined is the finger abduction joint. The other three joints have parallel axes of rotation, and thus move the finger in a single plane. The abduction joint is set such that the desired tip location lies within the plane of action of the three parallel joints.

$$\phi_{I_{ABD}} = \operatorname{atan}\left(\frac{z_{tip} - z_{ABD}}{x_{tip} - x_{ABD}}\right) \qquad \textbf{(4-25)}$$

where ${}^0\mathbf{d}_{I_{ABD}} = \left[x_{ABD}\ y_{ABD}\ z_{ABD}\right]^T$ is the location of the finger's abduction joint, or the base of the finger in the hand coordinate frame. This will, in general, return a unique answer in the range of $|\phi_{I_{ABD}}| \le 90°$. In the degenerate case of $z_{tip} - z_{ABD} = 0$ and $x_{tip} - x_{ABD} = 0$, $\phi_{I_{ABD}}$ is indeterminate. In practice, this situation is overcome by assuming it to be equal to its last known "good" value.

4.3.1.2 Parallel joints

The finger can now be modeled as a three degree of freedom mechanism in a plane as shown in Figure 4-4. For convenience, the location of the tip is converted into cylindrical coordinates in the plane of action of the finger.

$$r = \left\| {}^{0}\mathbf{d}_{I_{tip}} - {}^{0}\mathbf{d}_{I_{ABD}} \right\| \tag{4-26}$$

$$\alpha = \operatorname{atan2}\left(y_{tip} - y_{ABD}, \frac{x_{tip} - x_{ABD}}{\cos(\phi_{I_{ABD}})} \right) \tag{4-27}$$

We begin with the standard equation for the vector sum of three links.

$$r\,e^{i\alpha} = L_1 e^{i\phi_{I_{MPJ}}} + L_2 e^{i\phi_{I_{PIJ}}} \cdot e^{i\phi_{I_{MPJ}}} + L_3 e^{i\phi_{I_{DIJ}}} e^{i\phi_{I_{PIJ}}} e^{i\phi_{I_{MPJ}}} \tag{4-28}$$

We then isolate $\phi_{I_{MPJ}}$

$$e^{i\phi_{I_{MPJ}}} = \frac{r\,e^{i\alpha}}{L_1 + L_2 e^{i\phi_{I_{PIJ}}} + L_3 e^{i\phi_{I_{DIJ}}} e^{i\phi_{I_{PIJ}}}} \tag{4-29}$$

Eq. 4-29 is multiplied by it's complex conjugate and simplified to an equation only in $\phi_{I_{PIJ}}$ and $\phi_{I_{DIJ}}$.

$$\begin{aligned} r^2 = {} & L_1^2 + L_2^2 + L_3^2 + L_1 L_2 e^{-i\phi_{I_{PIJ}}} + L_1 L_2 e^{-i\phi_{I_{PIJ}}} + \\ & L_1 L_3 e^{-i(\phi_{I_{PIJ}} + \phi_{I_{DIJ}})} + L_1 L_3 e^{i(\phi_{I_{PIJ}} + \phi_{I_{DIJ}})} + \\ & L_2 L_3 e^{-i\phi_{I_{DIJ}}} + L_2 L_3 e^{i\phi_{I_{DIJ}}} \end{aligned} \tag{4-30}$$

Using Eq. 4-24, we substitute in $\gamma = \frac{1}{3}\phi_{I_{PIJ}} = \frac{1}{2}\phi_{I_{DIJ}}$, and use the Euler formula to get an equation in a single variable.

$$r^2 = L_1^2 + L_2^2 + L_3^2 + 2L_1 L_2 \cos(3\gamma) + 2L_1 L_3 \cos(5\gamma) + 2L_2 L_3 \cos(2\gamma) \tag{4-31}$$

which, by expanding the cosine terms becomes:

$$\begin{aligned} r^2 = {} & L_1^2 + L_2^2 + L_3^2 + 8L_1 L_2 \cos(\gamma)^3 - 6L_1 L_2 \cos(\gamma) + \\ & 32L_1 L_3 \cos(\gamma)^5 - 40L_1 L_3 \cos(\gamma)^3 + 10L_1 L_3 \cos(\gamma) + \\ & 4L_2 L_3 \cos(\gamma) - 2L_2 L_3 \end{aligned} \tag{4-32}$$

If we substitute $\cos(\gamma) = (1 - g^2)/(1 + g^2)$, where $g = \tan(\gamma/2)$, the result is a tenth order polynomial in g.

$$\begin{aligned} 0 = {} & (L_1^2 + L_2^2 + L_3^2 - r^2 - 2L_1 L_2 - 2L_1 L_3 + 2L_2 L_3) \cdot g^{10} + \\ & (5L_1^2 + 5L_2^2 + 5L_3^2 - 5r^2 + 26L_1 L_2 + 90L_1 L_3 - 6L_2 L_3) \cdot g^8 + \\ & (10L_1^2 + 10L_2^2 + 10L_3^2 - 10r^2 + 28L_1 L_2 - 420L_1 L_3 - 28L_2 L_3) \cdot g^6 + \\ & (10L_1^2 + 10L_2^2 + 10L_3^2 - 10r^2 - 28L_1 L_2 + 420L_1 L_3 - 28L_2 L_3) \cdot g^4 + \\ & (5L_1^2 + L5_2^2 + L5_3^2 - 5r^2 - 26L_1 L_2 - 90L_1 L_3 - 6L_2 L_3) \cdot g^2 + \\ & L_1^2 + L_2^2 + L_3^2 - r^2 + 2L_1 L_2 + 2L_1 L_3 + 2L_2 L_3 \end{aligned} \tag{4-33}$$

For a reachable location, ${}^0\mathbf{d}_{I_{tip}}$, there will be at least two real solutions. One corresponds to the configuration shown in Figure 4-4, and one to the "mirror" configuration created by inverting the shown configuration about the line r. The mirror configuration is not a valid configuration as it requires hyperextension of the PIJ. Additional real roots may occur for $\phi_{I_{PIJ}} \geq 180°$, which are also not valid configurations. The only valid configuration for the human hand is the minimum, positive real root.

Having solved for g, we can determine $\phi_{I_{PIJ}}$ and $\phi_{I_{DIJ}}$.

$$\begin{aligned} \gamma &= 2\,\mathrm{atan}(g) \\ \phi_{I_{PIJ}} &= 3\gamma \\ \phi_{I_{DIJ}} &= 2\gamma \end{aligned} \qquad \textbf{(4-34)}$$

Now it remains to solve for $\phi_{I_{MPJ}}$. First, we solve for the length of vector **s** by the law of cosines.

$$s = \sqrt{L_2^2 + L_3^2 - 2L_2L_3\cos(\beta)} \qquad \textbf{(4-35)}$$

where $\beta = \pi - \phi_{I_{DIJ}}$.

The angle between L_1 and r can be determined by the law of cosines and is equal to $\alpha - \phi_{I_{MPJ}}$. Therefore,

$$\phi_{I_{MPJ}} = \alpha - \mathrm{acos}\left(\frac{s^2 - r^2 - L_1^2}{2rL_1}\right) \qquad \textbf{(4-36)}$$

For all values of ${}^0\mathbf{d}_{I_{tip}}$ inside the workspace of the finger, the unique joint configuration $\boldsymbol{\phi}_I$ can be determined by solving Eq. 4-25, Eq. 4-32, Eq. 4-34 and Eq. 4-36.

5 Haptic Feedback in Dexterous Telemanipulation

Using the hand model and calibration routine from Chapter 4, we can incorporate a human level of learning and intelligence to a robotic system by allowing a human to directly control the robot. In the field of dexterous manipulation, it is desirable to control a robot hand by tracking compatible motions by the human hand. Furthermore, the ability to feel forces experienced by the robot should improve the human's capability in directing and controlling the robot.

This chapter presents two separate experiments which add force feedback to the user during telemanipulation. The first experiment (Section 5.1) examines the perceptual resolution of a human to information concerning object size, object stiffness and force when controlling a robot with arm-grounded force feedback. These tests show that haptic feedback from a hand mounted force feedback system can provide useful information during telemanipulation.The second experiment (Section 5.2) evaluates the benefits and drawbacks of the arm-grounded force feedback device when subjects perform prototypical manipulation tasks. These test show that the force feedback system does not increase speed of performance although it may benefit the learning process. How intuitive and natural the telemanipulation process seemed to the user was largely a function of how well a particular user's hand was calibrated and mapped to the robot hand.

5.1 Haptic Perception with Arm-Grounded Feedback

Force feedback has long been recognized as an important capability for teleoperation of dexterous arms and hands. Much previous work on fingertip force feedback has used "desktop" devices (e.g. [51][43]), in which reaction forces are directed, or "grounded," through the base to a static reference frame (commonly a desktop or floor). These devices are capable of accurate fingertip force application, but they often have a limited workspace because

they are mounted to a stationary platform. Devices that can apply forces to the operator's hand, while allowing free arm motions, are less common and, typically, quite bulky, such as the SARCOS dexterous master [82] and the SMU PHI [52].

Recent developments advance the opportunity to provide high fidelity forces to the fingers of a person's hand using a portable system. Examples from academic labs include the Rutgers Master II [34] and the LPR Hand Master [124]. Another promising instantiation of this technology is the CyberGrasp haptic interface produced by Virtual Technologies, as an add-on to the instrumented CyberGlove.

Devices such as the CyberGrasp are examples of "portable hand masters," which can apply forces to the fingertips. They are not grounded in the traditional sense. In this thesis we use the term "arm grounded" to describe devices where the reaction forces are applied locally to the user's hand or forearm, rather than a stationary platform. As a consequence, it is possible to create realistic forces that are internal to the hand, such as grasp forces, but it is difficult to create external forces, such as those that arise from contacting a surface in the environment, because they can give rise to unrealistic feedback. When the robot is touching a hard surface, the contact force at the fingertip is sensed by the user, and increases as the user continues to move into the surface. However, since this force is only grounded with respect to the user's arm, there is no physical impediment to further motion of the arm. This phenomenon has been described by several investigators including Srinivasan and Salisbury [115] and Gomez and Burdea [34]. Richard and Cutkosky [103] have examined the effect of using ungrounded haptic interfaces while contacting objects in a virtual environment. There has been little work done to quantify the performance of an ungrounded device in a telemanipulation environment.

Given the potential advantages in terms of reduced complexity and weight, and increased workspace of an ungrounded hand master as compared to a grounded system, we are interested in determining how well subjects can perform common tasks with an ungrounded system. This section presents results obtained while teleoperating a two-fingered dexterous robot with the CyberGrasp force feedback system. Subjects were asked to perform a series of object size and stiffness discrimination tests, and force-regulation tests. They com-

manded robot motions through the instrumented glove and received force feedback through the cable-driven exoskeleton.

5.1.1 Experimental Setup

The CyberGlove and CyberGrasp system was connected to the CyberGrasp system controller running a real-time operating system with an update rate of 1000 Hz. The robot was controlled by a separate PC, again running at 1000 Hz.

Motions of the human fingers are measured by an instrumented glove. As discussed in Chapter 4, the CyberGlove is a right-handed glove with 22 bend sensors measuring most of the degrees of freedom of the human hand. The resolution on each sensor is 0.2° to 0.8° depending on the range of motion of the particular joint. By reading only appropriate sensors for the thumb and index finger, data collection can be run at 200 Hz. The position information is transmitted to the robot controller over two dedicated D/A channels and updated at 100 Hz.

Force feedback is provided to the user's fingertips through the CyberGrasp mechanism, a cable driven device designed for use with the CyberGlove. A set of motors, worn in a backpack, apply tension to cables in teflon sheaths, which in turn apply forces to each finger. The forces applied to the finger are unipolar, since the cable can only pull along a single axis, and are grounded to the back of the user's hand, so no forces restrain arm motion.

The motors can apply forces up to 12 N and are updated at 1000 Hz to appear smooth and continuous to the user. The system has a resonance in the range of 20 Hz and a cutoff frequency on the order of 40 Hz. The robot controller provides readings from the force sensors on the robot to the CyberGrasp controller over two D/A channels at an update rate of 1000 Hz.

The principal performance constraint is static friction between the tendon and the sheath. Figure 5-1a shows that with an unflexed cable, the output forces track the commanded forces well, with some hysteresis when the tension is decreasing. If the cable is flexed, the contact between tendon and sheath increases in area and force, and results in significant hysteresis for both increasing and decreasing cable tensions. The graph in Figure 5-1b shows the friction property for a cable severely flexed (doubled back upon itself). During

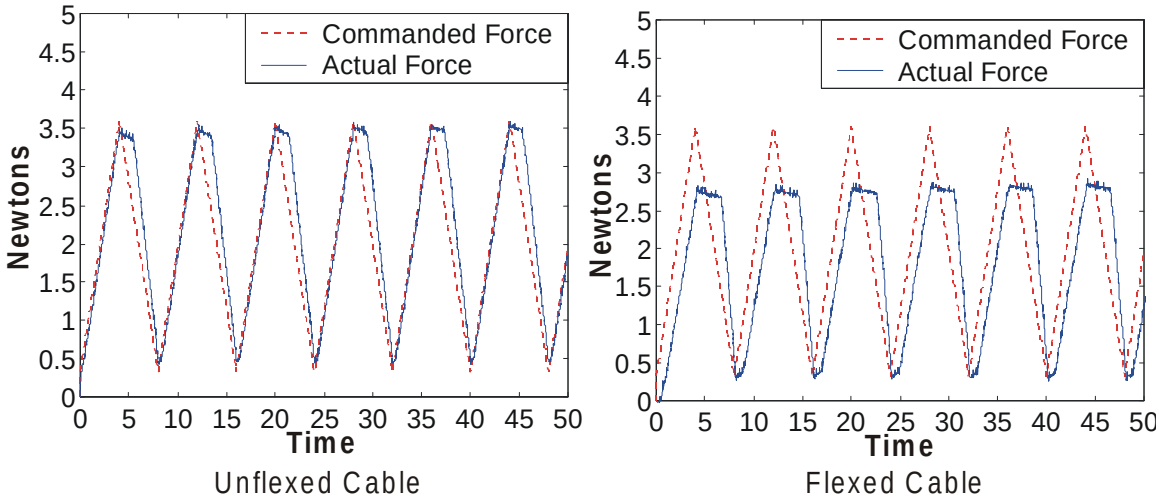

Figure 5-1. CyberGrasp Force Tracking with Flexed and Unflexed Cable

the tasks described below, the cables from the backpack to the user's hand are slightly flexed.

The robot used in the experiments is a two-fingered hand described in Chapter 3. Each finger has three degrees-of-freedom, each controlled by a dedicated DC servomotor. The workspace of the robot hand, for manipulation purposes, is approximately 100 x 100 mm, or slightly larger than the workspace of a human hand when manipulating small objects between the thumb and index finger.

The robot fingertip positions are measured with encoders, geared down to provide a position resolution of approximately 0.05 mm. At the end of each finger is a two-axis force sensor with a sensitivity of approximately 0.1 N, and a hemi-cylindrical fingertip covered with a textured rubber "skin."

The robot is commanded by an impedance control law which drives the fingertips to a specified position. For the purposes of this experiment, the desired positions of each fingertip are controlled independently, taking position commands from the CyberGlove. The closed-loop servo rate is 1000 Hz, which enables smooth motion and force control.

Since the robot is a two-fingered manipulator, only two fingers of the glove/feedback apparatus are required. The middle finger and thumb were used, because they best approximate the kinematics of the planar robot manipulator. The force applied to the thumb and middle

finger oppose each other directly while pinching, whereas the index and thumb forces, for example, are more skewed.

In order to test a simple pinch grasp, the motion of the robot is constrained to a horizontal line in the workplane. The commanded positions of the two robot fingers are based upon the positions of the tips of the operator's middle finger and thumb. This set of experiments predates the hand model and calibration routine described in Chapter 4. Instead, the system is recalibrated for each user so that the robot fingers are in contact when the user's middle finger and thumb are in contact, and are fully spread when the user's middle finger and thumb are fully spread. Since the robot fingers are symmetrical, while the human hand is not, mapping from the middle finger to the robot is not identical to the thumb mapping. However, the difference is minor, and can be neglected when a consistent pinching motion is used.

5.1.2 Experiments

Twelve untrained operators were recruited for the experiments. Each operator's right hand was fitted with a CyberGlove/CyberGrasp mechanism. A brief explanation of what the experiment entailed was provided, and each operator was allowed to practice grasping objects between the robot fingers to gain familiarity with the system. The operators were then asked to perform the object size discrimination test, followed by the force regulation test and finally the object stiffness discrimination test.

5.1.2.1 Object Size Discrimination

The object size discrimination test is used to examine how accurately the telemanipulation system conveys object size and contact occurrence. If the mechanism has too much compliance, or if it does not repeatably produce force at the same position, objects of slightly different sizes are indistinguishable.

For the object size test, the subject was asked to discern which of two unseen objects was larger. One of two objects was placed between the fingertips of the manipulator for the subject to grasp, while the contact forces were relayed back. When the subject was ready, the second object was placed between the robot fingers. The subject could ask for the objects to be exchanged as many times as desired in order to help make the decision.

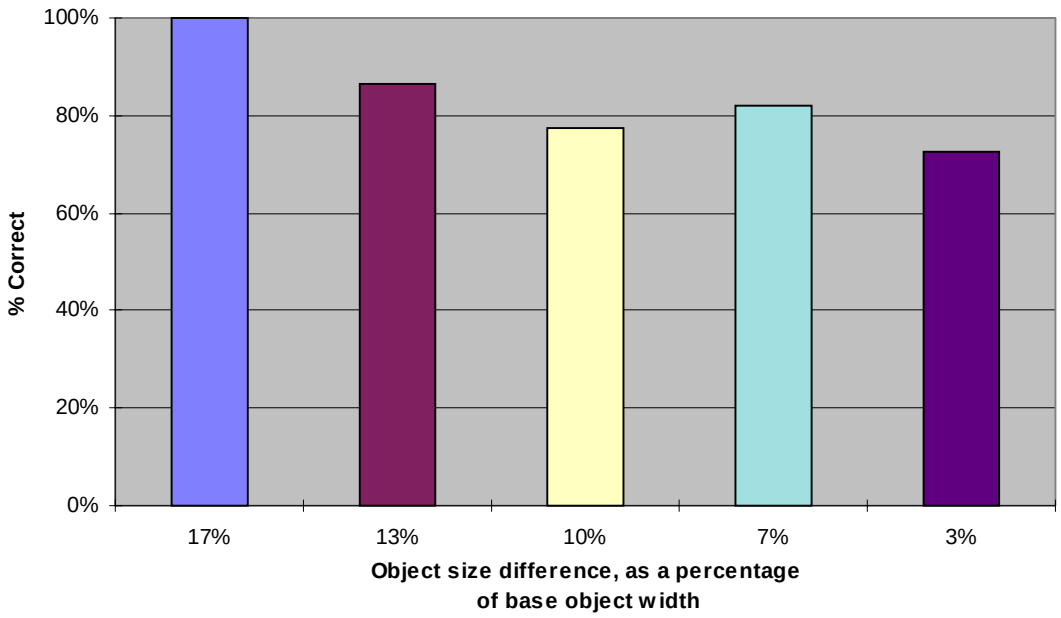

Figure 5-2. Size Discrimination through CyberGrasp

Five different pairwise comparisons were used, and each was used twice. Each set was formed with a 75 mm block and one of five other blocks from 77.5 mm to 87.5 mm in increments of 2.5 mm. For example, a 17% difference was displayed using a 75 mm block and an 87.5 mm block. The size differentials were 3%, 7%, 10%, 13% and 17%. On the first run, the tests were given in increasing order of difficulty (i.e. 17% first and 3% last) to help the user develop a strategy. (Though whether the large block was shown first or last was randomized in each test, and the user was not told that the tests were getting more difficult.) On the second run, the difficulty order was randomized. No significant difference in the success rate was apparent between the two runs.

While the size difference in the robot grasp was fixed, the corresponding size difference in the user's hand varied from subject to subject. This was primarily a function of the size of the user's hand. However, since the robot fingers were calibrated to be in contact when the user's fingers were in contact, the percentage difference in size was constant. In general, the perceived object size was smaller than the physical object.

As can be seen in Figure 5-2, there was a 100% success rate in the 17% difference, tending to decrease with each size difference to a 75% success rate at the 3% difference. From a qualitative standpoint, the 3% test seemed to be near the limit of perception. Subjects

tended to be noticeably less confident and took longer to make a decision than in the higher value tests. No measurements were recorded of how long the subject took or how often the subject switched between objects.

For comparison, Tan et. al [117] determined that the just-noticeable difference in the separation of the thumb and the index finger, when grasping parallel plates, is about 3% on average when the fingers are nominally 80 mm apart. Therefore, in our most challenging differentiation, it is possible that the users were limited by their own proprioception, in addition to whatever challenges the overall system may have provided. In an effort to separate these effects, users were asked to repeat the 3% test using their fingers directly on the object, and each performed the test successfully

The results obtained with CyberGrasp compare favorably to those reported by Richard and Cutkosky [103], where subjects were asked to compare wall positions simulated with a linear motor and requiring whole arm motions to move the finger. Their subjects had an 87% success rate for 5 mm (10%) differentials. The fact that the ungrounded apparatus appears to perform as well as a direct-drive linear motor, in this particular test, is an indication of good positional resolution.

In summary, the positional fidelity of the telemanipulation system seems to approach the limits of human proprioception when grasping objects. However due to the limited stiffness of the robot grasp, even a solid block will appear to have some compliance to the user.

5.1.2.2 Force Regulation

The force regulation test was used to determine how well a user can apply a consistent force using the telemanipulation setup. If the system cannot measure, apply and display forces in a consistent manner, it will not be possible to maintain a steady grasp force.

For the force regulation test, an acrylic block was placed within the workspace of the robot. A real-time measurement of the force applied to the object was displayed on the monitor of the robot controller. The user was asked to apply a constant force of approximately 2.5 N. The visual display was updated at a rate of 10 Hz. Each test was run for a minimum of 20 seconds.

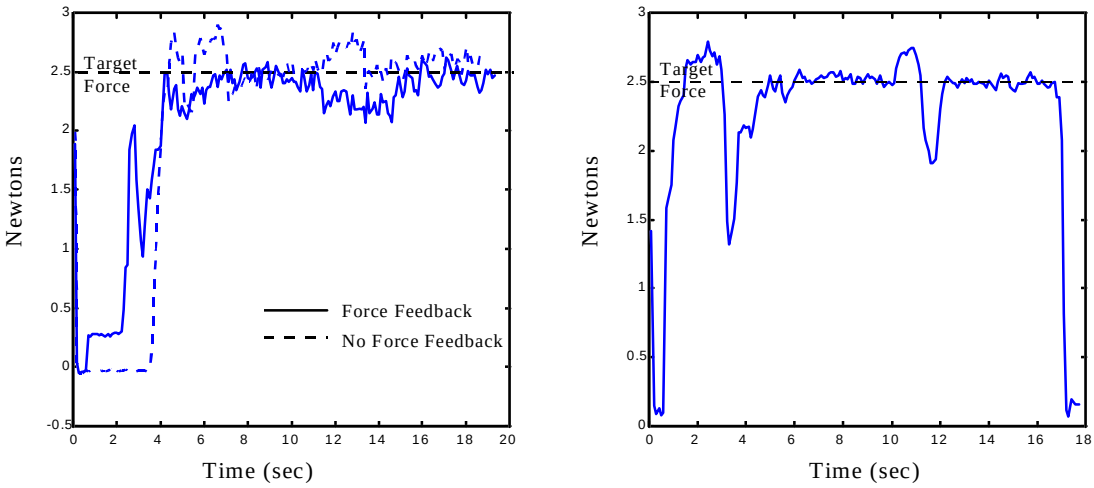

Figure 5-3. Force Regulation Results with CyberGrasp (left) and with Direct Contact (right)

The value of 2.5 N was selected as a representative force within the comfortable range of typical human pinch forces. The use of a single force level limits the generality of the conclusions that can be reached from this examination.

A typical example of the force results for a subject are shown by the solid line in Figure 5-3a. It is apparent that the setup is subject to some noise. Each user's results were subjected to the same apparent noise which has a frequency of about 3 Hz. The source of the noise is believed to be the A/D conversion of the commanded robot position signal on the robot controller.

The dashed line in Figure 5-3a shows the results from performing the same test without any force feedback. The user could still use the force measurement on the monitor to regulate the commanded position of the robot, which allows control of the resulting force through the impedance control law.

In order to provide a point of comparison, the user's ability to apply a desired force directly to an object was tested. Figure 5-3b shows the results from a person applying force between the middle finger and thumb directly to one of the robot force sensors. To minimize variations between the two types of tests, the robot was commanded to maintain a static position, and the user was wearing the CyberGlove, though no positional input was recorded.

In order to eliminate any data from force ramp-up or ramp-down stages or possible distractions during the test, statistical analysis was performed on the user's most consistent five second span.

For the force feedback test, the best standard deviation was 0.06 N, with 0.10 N being the average. Subjects settled around values with 0.2 N of the target force. Without force feedback, their performance was not notably diminished. This is indicative of the coupling between force and position in the impedance control law. Force errors invariably produced significant (2 mm/N) position errors that became visible on the monitor. In the tests in which the user directly compressed the robot force sensor, their control of the magnitude of the force is greatly increased, settling to within 0.05 N and an average standard deviation of 0.05 N. In addition, the 3 Hz servo noise was greatly diminished.

In related experiments in which subjects pressed a force sensor while viewing a display on a monitor, Srinivasan and Chen [116] determined an average absolute error in force control of 0.04 N for a force of 1.5 N, with a standard deviation of 0.006 N. These results suggest that if the system noise can be reduced the force regulation ability of users in telemanipulation may approach the ability in direct manipulation, when visual feedback is present.

5.1.2.3 Object Stiffness Discrimination

The object stiffness test was used to test how smoothly the system ramps up forces. If the force ramp-up was irregular or discontinuous, it would not be possible to discern stiffness characteristics.

For the object stiffness test, the subject was asked to discern, without looking, which of two objects of similar size was stiffer. Since the tip of each robot finger can be modeled as a spring based on the impedance control law, using two fingers to squeeze an object would effectively be three springs in series. For this reason, the users controlled a single robot finger with their middle finger and compressed the object against a hard wall.

Note that in this case, the lack of grounded forces is more significant because an internal grasp force is not applied. In the first two tasks, subjects could feel a "reaction" force in the opposite finger when pinching an object. In this task, they only feel a force applied to their middle finger but nothing restricts the motion of their arm. This is the most "unrealistic" force representation of the experiment.

Two objects of significantly different stiffness were used. One object was an acrylic block with a stiffness much greater than the commanded stiffness of the robot finger (set at 525 N/m). The second object was a block of soft packing foam with a spring constant of approximately 150 N/m, or 30% of the impedance of the robot control. The sizes of the two objects in an uncompressed state were equal, and both edges of the soft block were covered with a piece of acrylic to make the surface properties similar.

The subjects were fairly successful at determining which of the objects was stiffer, choosing correctly 75% of the time. However, many subjects described the process as the most difficult of the tests. Many said that while the two objects felt decidedly different, they were not always sure which one was stiffer.

There are two principal reasons why this test is more difficult than the size discrimination test. First, the impedance control of the robot makes it an inherently compliant system with an effective positional stiffness of 525 N/m. The user is essentially trying to tell whether there is an extra spring in series with the system. Second, the maximum force applied to the user's fingers was limited to 12 N to prevent injury. This level was low enough that users were able to move their fingers forward, even at maximum force. Effectively, the subject was trying to sense the difference between a force that ramped up to 12 N in 2.3 mm (acrylic block) or one that ramped up to 12 N in 8 mm.

That such a significant difference in stiffness is nearly undetectable shows the largest deficiency of the setup. For comparison, Tan and Srinivasan [117] found that when subjects pushed against an elastic beam with stiffnesses ranging from 15300 N/m to 41500 N/m, they could detect stiffness differences of 20%. It is likely that the current system can be improved by reducing system noise and friction. The performance in this test could also be improved by increasing the robot stiffness, at a cost to the force regulation ability, and by increasing the maximum force applied to the hand, at a cost of more user discomfort

5.1.3 Conclusion

It is clear from these tests that haptic feedback from a hand-mounted force-feedback system can provide useful grasping information during telemanipulation. We also observe that for the simple tasks that were explored, time lags, limited stiffness, and effects such as servo noise and friction are more serious drawbacks than the lack of grounded forces.

However, we should also acknowledge some difficulties. The results presented here represent a second set of experiments performed with the CyberGlove/CyberGrasp setup and the dexterous robot. In the original experiments (not discussed here), the positional information between the glove computer and the robot control computer was sent over a serial line. In addition, the impedance control of the robot was optimized for smooth motions rather than minimized dynamic response. The result was a delay of 50 to 100 msec to position commands, which made the tasks more difficult and instabilities more likely. In the original experiments, the users achieved comparable results in the object size test, but had significantly more difficulty in the force-regulation and object-stiffness tests.

It is clear that with the system described here we are still not able to take full advantage of the proprioception and sensitivity of the human user. There are still physical lags in a telemanipulation system that do not allow one to display forces with the crispness and resolution that nature does.

5.2 Haptic Feedback In Dexterous Telemanipulation

Utilizing the knowledge gained on human perception with the CyberGrasp force feedback device, a more advanced telemanipulation system was investigated. In this system, the user is able to command in-hand manipulations and arm motions of the robot.

A new planar dexterous two-fingered robot hand was developed which can be affixed to the end of a larger Adept SCARA industrial robot as an end effector. The industrial robot arm tracks the motions of the human arm as measured by a 6 DOF linkage attached to the user’s wrist. Incorporating the industrial robot substantially increases the user's workspace, allowing the hand to be continually positioned and oriented to best advantage. In addition, object motion is not restricted to a single plane.

5.2.1 Human interface

The human interfaces consists of the instrumented glove, the arm grounded feedback mechanism and the six-degree-of-freedom kinematic linkage for measuring arm movement.

5.2.1.1 Instrumented Glove

The user's finger positions are measured with CyberGlove described previously. A predictive algorithm is used so the data appear to be continuous at 1000 Hz, and can be used in robot control.

The glove was calibrated for each user by using a least-squares regression to determine such parameters as finger length, sensor gains and sensor offsets. The resulting fingertip position accuracy was approximately 5 mm for manipulation of small objects. A more detailed explanation can be found in Chapter 4.

In this experiment, the glove was used to track only the motions of the index finger and thumb. The index finger was used (instead of the middle as in the previous experiment) because of the greater experience a human has in manipulating with the thumb and index fingers.

5.2.1.2 Arm-grounded Force Feedback

Force feedback is again supplied to the fingers by the CyberGrasp mechanism. The force applied to the finger acts to straighten the finger. The exact line of force action is configuration dependent, but in general has a positive projection onto the axis of pinch force between the index and thumb. For telemanipulation, the full magnitude of the pinch force is displayed to the finger, rather than a vector projection, to allow the user to regulate applied forces while manipulating an object. This results in somewhat inaccurate force direction information displayed to the user.

5.2.1.3 6-DOF Kinematic Linkage

The motion of the user's arm is measured by a six-degree-of-freedom linkage attached to the back of the CyberGrasp on the user's hand. Each joint is measured with an encoder, allowing computation of the hand position and orientation to within 0.5 mm and 0.2° respectively. The workspace of a reference point on the sixth link of the linkage is approximately a curved horizontal cylinder 200 cm long with a cross sectional diameter of 30 cm. Roll, pitch and yaw motions of the sixth link all extend to ±90°, though there is a kinematic singularity when the pitch is ±90°. The device is counterweighted to ease the load on the user's arm, though extended use can grow tiring.

The device does not actively apply forces to the user's hand, but is a prototype for a world grounded force feedback addition to the CyberGrasp.

5.2.2 Robot System

The robot system consists of the dexterous robot hand and the industrial robot arm.

5.2.2.1 Dexterous Robot Hand

"Dexter" is a two-fingered robotic hand, with two degrees of freedom per finger. Each degree of freedom is powered by a low friction, low inertia DC servomotor. The motor is connected to the link through a cable/drum drive similar to those found in haptic feedback devices such as the PHANToM by Sensable Devices. As a result, the hand has very low friction and is backdrivable. The motors are fairly small, due to weight and space limitations, but are still capable of providing enough force at the fingertips to pick up a 250g object, such as a softball, which more than suffices for the purpose of these experiments.

The links are 100 mm long, and each has over 120° of motion. The workspace of the hand is about 400 mm by 150 mm, with a positional resolution of 0.08 mm. This workspace is sized to best manipulate objects from about one to three inches wide.

Two-axis strain gage force sensors have been incorporated into the robot fingertips to read the forces applied by the robot to the object. The force sensors have good linearity and are accurate to ± 0.1 N up to 5 N. The fingertips have a foam core and a compliant, textured rubber skin in order to minimize contact instability and decrease object slipping.

5.2.2.2 Industrial Robot Arm

The robot hand is placed on the end of an Adept 1, a 4 degree of freedom SCARA industrial robot arm. The Adept has a positional resolution of 0.04 mm and 0.05°. The workspace is approximately 1100 mm long by 350 mm wide by 175 mm high. The robot trajectory is controlled by the Adept controller, which requests new commanded positions as needed.

The speed of the Adept robot is limited for the safety of the robot hand. The speed limitation and the lack of direct control of the robot motors results in a noticeable lag between human arm motions and robot arm motions.

5.2.3 Communication

Communication lag between the controller and the actuator can lead to instabilities, particularly in a force control system. To avoid this problem, almost all of the computations are performed on a single computer.

The principal controller is a Pentium 233 MHz machine running the QNX real time operating system. Using a real time OS allows us to run multiple processes simultaneously at different rates and different priorities, with near instantaneous communication between processes. For example, robot motor torques are updated at 1000 Hz with high priority even though the hand kinematics (mass matrix, Jacobian) are updated at 200 Hz and the graphic interface is updated at 7 Hz. The various processes effectively run in parallel rather than in series.

The CyberGlove is connected to the controller through the PC's serial port. The CyberGrasp is controlled by a Servo2Go card connected to the motor amps. The dexterous robot hand is controlled by a separate Servo2Go card connected to linear current motor amps which also reads the encoder values and force sensors. The kinematic linkage has a separate controller which communicates the joint angles with the main computer over a digital I/O.

The Adept robot controller generates a smooth and stable trajectory for the Adept robot. The Adept controller requests and receives new positions over an ethernet connection from the principal controller whenever the trajectory path can be amended. The lag between the human arm motions and the Adept arm motions is not a function of the communication, but is due to the speed and acceleration constraints on the Adept, which means it will not reach the latest position for a finite amount of time. This lag could lead to accidental collisions between the end-effector robot and the world, but users seemed to quickly adjust to this problem.

5.2.4 Testing

A diverse set of ten subjects were asked to perform two prototypical manipulation tasks in order to evaluate the performance of the system. Each subject had the CyberGlove calibrated to their hand, and the mapping to the robot hand customized to best allow the robot fingers to follow the motions of the subject's fingers.[36]

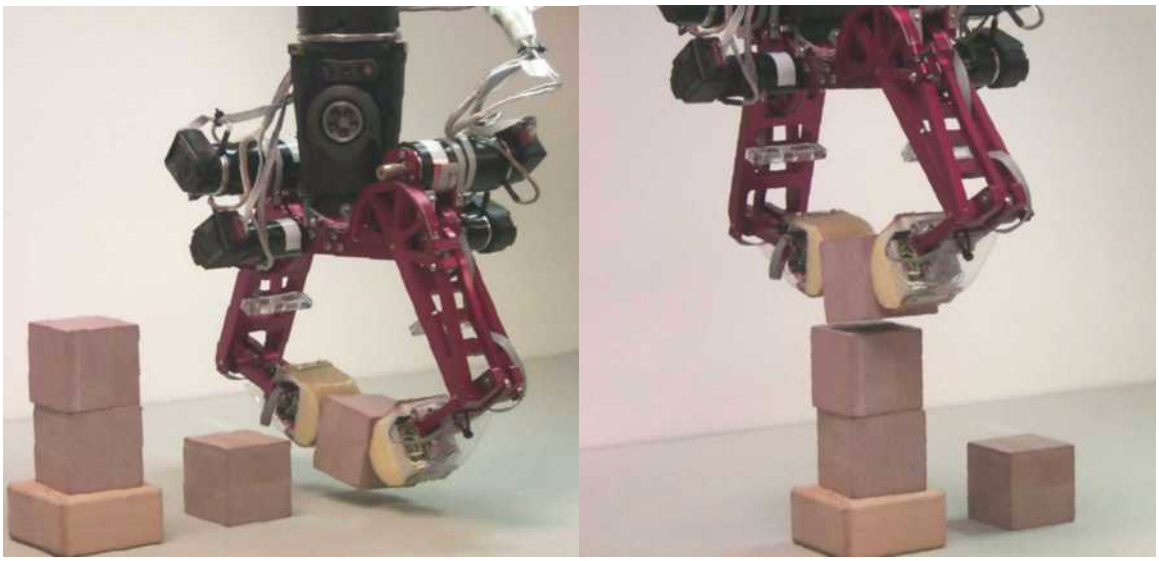

Figure 5-4. Block Stacking

Subjects performed each task twelve times, six with force feedback from the CyberGrasp and six without it in a pattern of AAAA-BBBB-AA-BB, where A is one force mode (on/ off) and B is the other. The pattern was selected in an effort to minimize the effect of learning. Each subject has a trial with each force mode near the end of the task set, and time to completion analysis was performed on each subject's best run under each condition. In order to eliminate order-sensitive issues, a random determination was used to select whether a subject began with force feedback on or off.

The time to completion of each trial was measured, as well as the subject's success in completing the task.

5.2.4.1 Block Stacking

The subjects were asked to construct a tower of four blocks at a target location, as seen in Figure 5-4. This test examines whether the presence of the internal forces displayed by CyberGrasp affects performance of a primarily pick-and-place task.

The task began with the four 50 mm blocks in preset locations and the robot hand and arm in the designated "safe" area. The time measurement started when the robot arm left the safe area. The subject controlled the robot arm to guide the robot hand over the block and then grasped the block. The block is carried to the target spot and placed. The grasp and place motion was repeated on three more blocks to construct the tower. In order to place

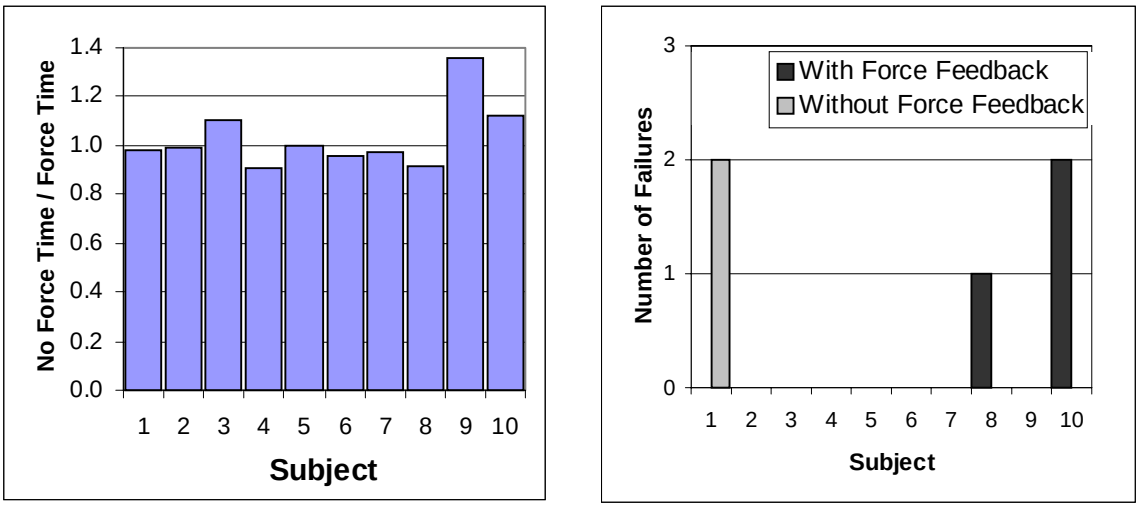

Figure 5-5. Completion Time Ratio and Failure Occurrences for Block Stacking

the final block on the tower, it was usually necessary to move the block upward within Dexter's workspace.

If a subject dropped a block within the workspace of the robot, they could pick it back up and continue. If the subject knocked over the tower or dropped a block beyond the workspace of the robot the task was considered unsuccessful.

The quantitative results do not show a significant difference in time to completion or number of failures. Figure 5-5a shows the ratio of time to completion without force feedback to the time to completion with force feedback of each subjects best run for each condition. The average time ratio for subjects was 1.03, meaning that the best run with force feedback was typically only 3% faster than the best run without force feedback. Similarly, Figure 5-5b shows that the total number of unsuccessful trials was nearly identical for the two conditions.

Qualitatively, most of the subjects preferred the force feedback mode. During the run, comments such as "That feels nice," and "The forces tell you where you are and what is going on," were common. Upon completion of all the tasks, all subjects said they felt more comfortable with the forces on in this task. Some subjects pointed out that the forces could still mislead you, however. One subject mentioned that touching the table felt exactly the same as touching the object, since the CyberGrasp can only pull in one direction.

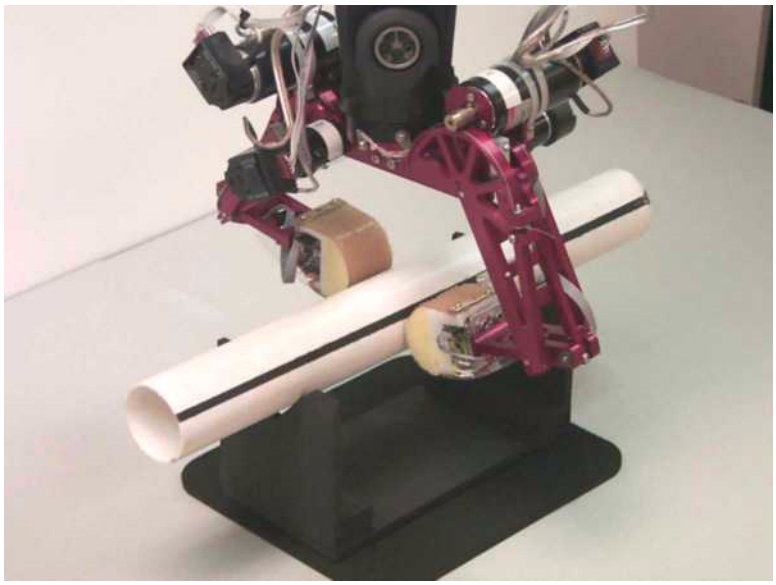

Figure 5-6. Knob Turning

One interesting subject comment is that it seemed easier to learn the task with force feedback, but once a subject was comfortable it didn't seem to make much difference. The data collected seem to support this. Subjects who started the task with force feedback on performed noticeably faster (on average, 15%) on their first trial than those that did not have forces on their first trial. Unfortunately, due to the small subject pool we can not generalize this statement with any statistical confidence.

A likely reason for the overall lack of time difference between the two modes, besides human muscle memory and proprioception, is that a significant portion of the task involved moving the larger and slower industrial robot arm. Conditions for moving the arm were identical in the two force modes and likely contributed significantly to the overall time.

5.2.4.2 Knob Turning

The subjects were asked to use the robot fingers to roll a cylinder through a full 360° rotation, as seen in Figure 5-6. The purpose of this task was to examine the effect of the CyberGrasp force feedback on performing two-fingered manipulation.

The task began with the 400 mm long by 50 mm diameter cylinder resting in two V-shaped notched supports near the ends of the cylinder. The robot hand was located above the cylinder. The subject was instructed to lower the hand and rotate the object through a full 360°

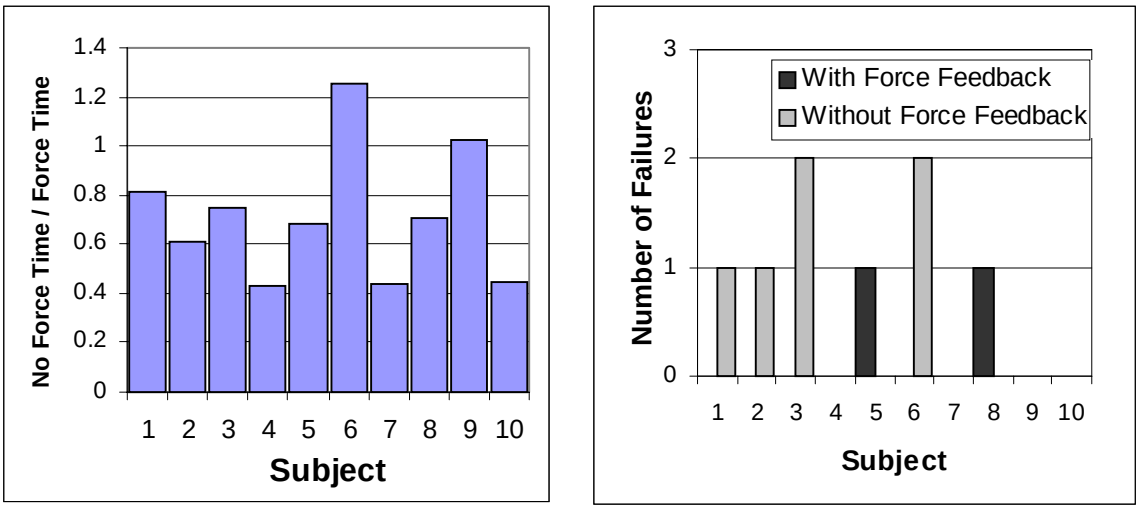

Figure 5-7. Completion Time Ratio and Failure Occurrences for Knob Turning

clockwise rotation. If the subject knocked the cylinder off of the stand, the attempt was considered to be unsuccessful.

No instructions were given to the subjects about the method of rotating. Most subjects did primarily use two fingered manipulation, though some subjects rotated the cylinder by pushing it with a single finger.

The subject-to-subject variability was high in this task, probably due to varying levels of mapping quality from the human to the robot. One subject rotated the object with nearly flawless two-fingered manipulation on his first attempt, with a time lower than many subjects' best performance. Other subjects struggled to coordinate their finger motions even after several trials.

It was fairly clear from watching the subjects that the addition of force feedback actually impeded their ability to roll the object quickly. Figure 5-7a shows the ratio of best run without force feedback over best run with force feedback for each subject. The average ratio across subjects was 0.72 (a ratio of 1.0 would indicate no difference in performance due to force feedback, while a ratio less than one indicates faster completion time without force feedback). Thus, on average, the subjects' best run without force feedback was 28% faster than their best run with force feedback. Using a bootstrap t-test to resample our data, we can state with 95% confidence that the mean population ratio would be 0.72±0.16, illustrat-

ing that force feedback has a negative effect on subject performance with respect to this task.

Conversely, Figure 5-7b displays that our subjects were three times more likely to have a trial failure without force feedback as compared to with force feedback (a 10% failure rate compared to a 3.3% failure rate). Analyzing the data with a bootstrap t-test to examine whether the population is more likely to have a failure without force feedback, we get an α value of 20%. This is not significant enough to state strongly, but it is a promising indication and should be examined further.

These results are confirmed by the qualitative comments made by the subjects. One subject pointed out that the system does not provide a sensation of curvature or rolling. Another subject said, "The forces don't match what I see, so I don't know how to adjust my hand". The difficulty of the task may be due in part to the uncommon motion. One subject pointed out that she really wanted to grasp the cylinder and turn it with her wrist.

Despite the fact that overall performance was better without force feedback, we continue to see indications that it was easier to learn how to perform the task with force feedback than without. Subjects who started the task with force feedback on performed noticeably faster (on average, 17%) on their first trial than those that did not have forces on their first trial. Unfortunately, due to the small subject pool we can not generalize this statement with any statistical confidence.

These results indicate that the single axis of force representation provided by the CyberGrasp does not sufficiently represent the expected forces for rolling an object, and the potentially misleading information slows the user more than having no force information. Rolling of an object uses regulation of the ratio of normal to tangential forces, which cannot be displayed by this system. However, the presence of contact force information does seem to improve a user's ability to maintain a stable grasp and aid the task learning process.

5.2.5 Results

The results indicate that the CyberGrasp force feedback system does not increase speed of performance for simple telemanipulation tasks such as block stacking and object turning.

This is due in part to human skills in learning and muscle memory, as well as the imperfect force transparency due to the single degree of actuation.

Conversely, there is some evidence that force feedback improves manipulation stability and may benefit the task learning process. Subject responses indicate that the benefits of force feedback may be more apparent in tasks which require delicacy or precision. One subject pointed out “It seemed a little more difficult with the forces, because you are taking care where your fingers are.... Otherwise, you just pinch as hard as possible.” Another subject said, “Without forces you aren’t worried about damaging the machine or forcing things”.

6 Tracking Human Manipulation

The first steps in programming dexterous manipulation by demonstration are to track the human hand motion and to infer the intended object trajectory. Interpretation of human intent is not easy when the person is interacting with a virtual representation of an object; the hand may make motions which are inconsistent with manipulation constraints, such as object rigidity, rolling contact or grasp stability. This chapter introduces a procedure to determine the intended object trajectory. The resulting algorithm attempts to create the best approximation of the human motion, while maintaining manipulation and contact constraints such as object rigidity, rolling/sliding contact and finger release and regrasp.

6.1 System Overview

The human operator wears the instrumented glove, CyberGlove, as described previously. Sensor values from the glove are measured and adjusted according to calibration parameters from the routine described in Chapter 4. Due to limitations of the glove, sensor values from four of the five fingers are used. The small “pinky” finger is not measured, but this is not a serious limitation because it is not very useful in fine manipulation, though it is often used in maintaining grasp stability [28].

A separate computer has a graphical display of the human hand and the virtual object, in a MATLAB environment. The display computer communicates with the glove measurement computer via ethernet. The display computer receives calibrated values of the user’s bone lengths and current joint values, and maintains the model of the virtual environment.

A limitation of the current set-up is the update rate of the graphical display. The graphics are currently being performed in MATLAB, because it is relatively easy to generate an accurate display of the mathematical model of the virtual environment. However, the display is updated only two to three times per second, which does not provide for smooth con-

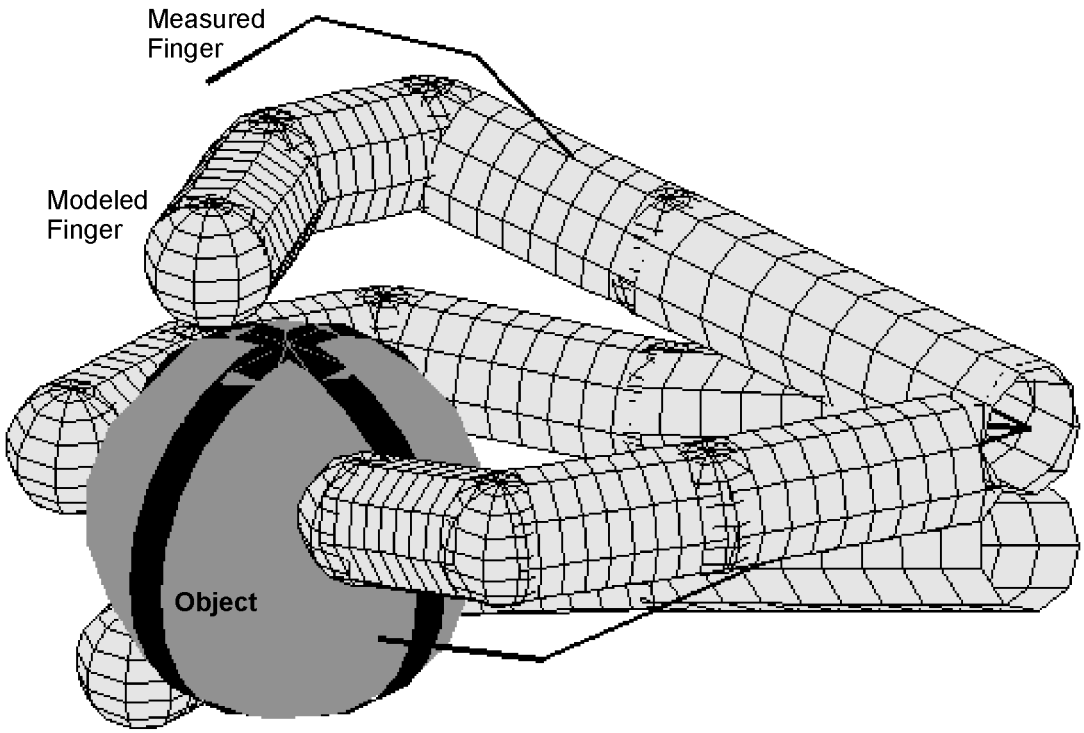

Figure 6-1. Graphical Display of a Virtual Environment

tinuous motion. Fortunately, the update rate does allow for effective evaluation of the manipulation approach.

It should be noted that this work is performed without the use of the CyberGrasp force feedback device described previously. The reasons for this are twofold. First, the manipulation algorithm presented below is entirely kinematics based. The user can fully control the environment by controlling the positions of the fingers. The addition of force feedback would enhance the experience, and perhaps improve performance, though it is not necessary. Second, with the update of the virtual environment currently occurring only two or three times per second, the forces displayed would not feel at all natural or helpful.

6.2 Virtual Environment

Figure 6-1 shows a graphical display of the virtual environment with which the user interacts through the CyberGlove. The three key aspects of the environment are the commanded hand configuration, the virtual hand model and the virtual object.

The solid black lines in Figure 6-1 shows the actual configuration of the user's hand as measured by the instrumented glove and fit to the calibrated hand model. The commanded

hand model does not directly interact with the virtual environment, but this information is displayed as an aide to the user. As the object is manipulated, the commanded configuration may diverge somewhat from the hand model. It is important that the user be able to see exactly what is being commanded as well as the response of the virtual system.

The virtual hand model interacts with the virtual environment while following the motions of the commanded hand. The virtual hand model uses kinematics developed in Chapter 4. Each link of the hand is represented as a cylinder with a spherical cap on each end (sometimes referred to as a cylsphere). This approximation of the finger's external shape is visually satisfactory, graphically efficient and simplifies the rolling calculations.

Currently, the virtual object is the only aspect of the environment with which the virtual hand can interact. The object is defined by a mathematical representation of the object surface and a transformation from the hand frame to a frame connected to the object.

6.3 Tracking Manipulation

There has been considerable research on various aspects of virtual reality. Systems have been developed which allow a person to push, poke, prod and even feel objects whose only reality is a mathematical representation on a computer. Many systems are stylus-based, where the user can only interact with the world through the end of a probe [84]. Some systems allow the person to use a virtual hand to grasp and move objects [124][8]. However, the hand based virtual systems do not include the ability to use many human skills in manipulation, such as rolling, contact sliding and regrasping. The approach developed here combines contact acquisition and release (6.3.2), sliding (6.3.3) and rolling contact (6.3.4) into a single integrated algorithm (6.3.5).

6.3.1 Basic Approach

It is common practice in virtual reality interaction to have the user control a desired position of the virtual probe. The actual position of the virtual probe is connected to the desired position through a virtual spring and damper. For example, when the virtual probe is in contact with a wall the user can command a position a few millimeters inside the wall. The commanded position "pulls" on the probe proportionally to the separation, which can be considered the force applied by the probe to the wall.

This virtual spring and damper are the basis of the interaction between our virtual hand and the virtual object. The tip of each finger in the desired (commanded) hand configuration is connected to the tip of the corresponding finger at the current position of the virtual hand model by a virtual spring. Any offset between the commanded configuration and the current virtual hand's configuration generates a virtual force upon the virtual finger and, if in contact, the object.

One drawback to this approach is the potential buildup of large forces, which can lead to unexpected reaction and instability. For example, consider a virtual block held between the thumb and index finger, where the commanded positions of the index tip and thumb tip are each two cm inside of the block surface, with a high spring value connecting the commanded tips to the virtual hand. In theory, this system should be stable, with the fingers holding the object (albeit with a large internal force). However, in the discretized computer model, the system may become unstable. A small change in the commanded thumb position leads to a non-zero force, and the object may oscillate about its balanced position. If the change is large enough, and the virtual mass of the object is small enough, the object may move (in a single time step) out of the grasp of the fingers. The result is somewhat like a watermelon seed squeezed between the fingers and suddenly launching across the room.

A slightly different approach has been implemented here in order to avoid the difficulty of an unstable dynamic response. Rather than using the forces from the fingertip to determine the dynamics of the grasped object (using f = ma), the space of possible object configurations is searched for a stable configuration, namely where the sum of the applied forces and moments is zero. The object is then moved to the new stable configuration.

Figure 6-2 shows an example of object manipulation. Frame A shows an object grasped between the thumb and index fingers. At the next time step, the measured fingertip location has changed, as shown in frame B. The separation between the measured index tip and the modeled index tip creates a force $\mathbf{F}_I$ on the modeled index fingertip, based on the virtual spring concept described above. Similarly, a force $\mathbf{F}_T$ is applied to the thumb fingertip. The finger forces sum to form a net force and moment on the object ($\mathbf{F}_{obj}$ and $\mathbf{M}_{obj}$). If a dynamics based model is used, the object will accelerate in the direction of the forces. Assuming the response is stable, the object will overshoot and eventually converge to the unique loca-

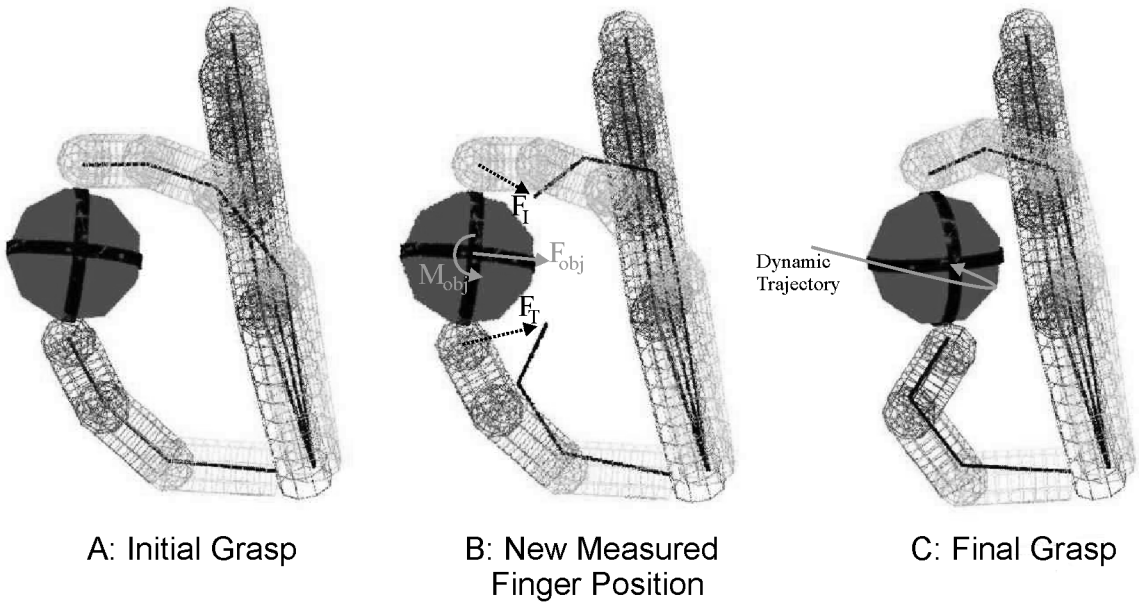

Figure 6-2. Controlled Object Motion from Measured Fingertip Position

tion where the sum of the forces and moments on the object are zero. A potential trajectory of the center of the object under dynamic control is represented in frame C. The methodology proposed here assumes that the user does not intend to apply an oscillatory motion to the object. Instead, the object is moved directly to the stable location (shown in frame C) in a single step.

It is precisely this method that allows us to perform this algorithm without calculating the finger dynamics or specifying the object mass. It is also because of this approach that the update rate between two and three Hertz is acceptable. The object moves directly to the stable configuration, even for large changes in commanded fingertip positions from one step to the next. The stable configuration is found by an interactive search that converges on a minimum potential position of the system.

The effective force and moment on the object are calculated based on the separation between the commanded and current tip positions. We use:

$$\mathbf{F}_{obj} = \sum_{f=\text{thumb}}^{\text{...g}} G_f \cdot k(\mathbf{d}_{comm_f} - \mathbf{d}_{curr_f}) \qquad \textbf{(6-1)}$$

$$\mathbf{M}_{obj} = \sum_{f=thumb}^{...} G_f \cdot k(\mathbf{d}_{comm_f} - \mathbf{d}_{curr_f}) \times \mathbf{d}_{curr_f} \quad \textbf{(6-2)}$$

where the commanded tip position $\mathbf{d}_{comm_f}$ and the virtual hand tip position $\mathbf{d}_{curr_f}$ for each finger f are both calculated in the object frame. $G_f = 1$ if finger f is in contact with the object, and 0 otherwise, since a finger cannot apply a force to the object when not in contact. k is the spring constant between the commanded and current positions. $\mathbf{F}_{obj}$ is the vector sum of the forces on the object, and $\mathbf{M}_{obj}$ is the vector sum of the moments due to the finger forces about each axis of the object frame.

The object position is modified based on the average calculated force and moment. It is translated by $\frac{\mathbf{F}_{obj}}{2 \cdot k \cdot \sum_f G_f}$. It is rotated by $\frac{\|\mathbf{M}_{obj}\|}{2 \cdot \mathbf{r}_{ave}^2 \cdot k \cdot \sum_f G_f}$, about the unit axis $\frac{\mathbf{M}_{obj}}{\|\mathbf{M}_{obj}\|}$ where $\mathbf{r}_{ave}$ is the average magnitude of $\mathbf{d}_{curr_f}$ for fingers in contact with the object. This motion of the object is in the gradient direction of lower potential energy of the virtual springs. The process is then repeated using the new object position to iterate to a position where the sums of the forces and moments on the object are zero.

When there are fewer than three fingers in contact, the orientation of the object is not fully determined. For example, if an object is held between the index and thumb, the rotation of the object about the line between the contact points will not affect the virtual spring energy. In these cases, the rotation of the object about such an axis is set to zero.

Only when the final object position has been calculated is the new object position displayed to the user.

6.3.2 Contact Acquisition and Release

Once a new static position has been calculated for the object, it is necessary to check if the grasp configuration has changed. When manipulating objects, it is common for humans to release contact with some fingers and then to place these fingers in a new location. This allows for a greater range of object motion while maintaining grasp stability.

Using the new object position, the commanded fingertip positions are examined. If the commanded tip position of a finger not in contact with the object is now inside the surface of the object or within one fingertip radius of the surface, its state is changed to “in contact.”

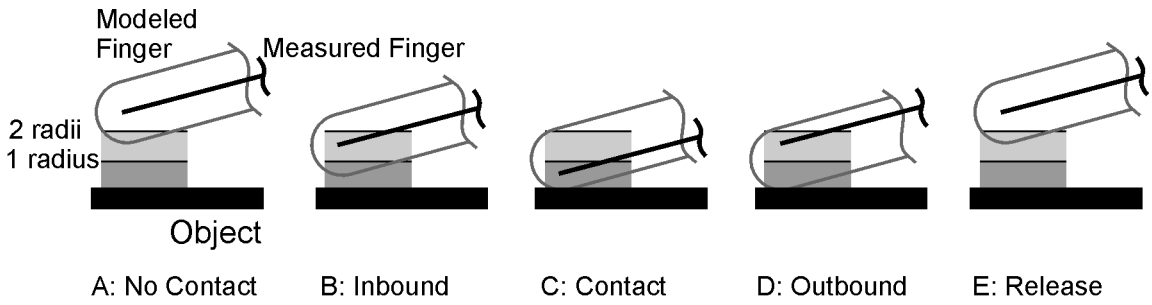

Figure 6-3. Contact Acquisition and Release with Intentional Hysteresis to Avoid "Chatter"

Conversely, if the separation between a contacted fingertip and the object surface is greater than twice the fingertip radius, its state is changed to "not in contact."

There is deliberately some overlap between the two states to avoid spurious transitions. A finger with a separation from the object surface greater than the finger radius but less than twice the finger radius will remain in its current contact state. Without this built in hysteresis, a finger could oscillate between contact states from one update to another.

Figure 6-3 shows a finger moving into and out of contact with an object. In frame A, the tip of the measured finger is more than two tip radii from the object surface, and the finger is defined as not in contact. In frame B, the tip of the measured finger has moved into the transition region between two radii and one radius from the object surface. In this zone, the finger stays in its current state, in this case "no contact." In frame C, the measured tip is within one radius of the object surface so the finger is defined as "in contact." Frame D shows the measured tip moving back into the transition zone, and the finger remains in contact. Finally, in frame E, the measured tip has moved more than 2 radii from the object surface and contact is released.

6.3.3 Sliding

Another means of adjusting finger position is to slide a finger along the surface of the object. Sometimes this is performed in order to explore or track the object surface, as discussed in Chapter 3. Other times, it is performed in order achieve a new desired grasp configuration.

Sliding occurs when the lateral force applied between the finger and the object exceeds the maximum friction force of the contact. We model friction using the Coulomb approxima-

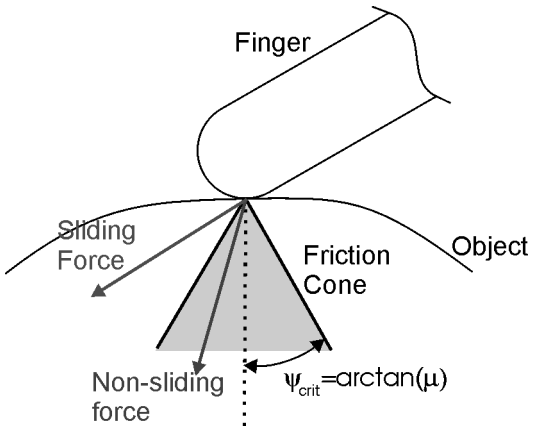

Figure 6-4. Sliding and Non-sliding Contact Forces Based on Friction Cone

tion where the friction force is linearly proportional to the normal force of contact $f_{frict} = \mu \cdot f_{norm}$.

Nakamura, Yoshikawa et al. [95] show how this can be modeled graphically. The locus of all non-sliding contact forces form a cone around the contact normal with an apex angle of $\arctan(\mu)$. A two dimensional representation is shown in Figure 6-4.

Since the effective contact force for each fingertip extends from the current fingertip to the commanded fingertip position, the angle ψ_f between the contact force and the contact normal $\mathbf{n}_{c_f}$ for finger f is calculated by;

$$\psi_f = \arccos\left(\frac{(\mathbf{d}_{comm_f} - \mathbf{d}_{curr_f}) \bullet \mathbf{n}_{c_f}}{\|\mathbf{d}_{comm_f} - \mathbf{d}_{curr_f}\| \cdot \|\mathbf{n}_{c_f}\|}\right) \quad \textbf{(6-3)}$$

If $\psi_f > \arctan(\mu)$, then finger f is in a position where sliding can occur. However, in practice, it is difficult to slide more than a single finger at a time. Therefore, in the case of multiple fingers in a possible sliding configuration, sliding is restricted to the finger with a contact force outside the friction cone and the largest tangential force component. Tangential force is calculated by:

$$\mathbf{f}_{tang_f} = k \cdot \|\mathbf{d}_{comm_f} - \mathbf{d}_{curr_f}\| \cdot \cos(\psi_f) \quad \textbf{(6-4)}$$

If sliding occurs, the position of the fingertip is moved to the nearest location where the finger is touching the object and $\psi_f = 0$.

6.3.4 Rolling Contact

For fine manipulation, humans rely extensively upon rolling between the finger and the object to control the position of the object. Rolling becomes increasingly important as the size of the object decreases. Most of the existing virtual reality interfaces for manipulation do not account for rolling, which would limit accuracy in small object manipulation such as rolling a pencil between two fingers.

The rolling constraint is a velocity constraint. For pure rolling (i.e., no sliding) to occur, the instantaneous velocity of the contact point on the finger must be the same as the instantaneous velocity of the contact point on the object. If there is a non-zero relative velocity between the finger contact point and the object contact point, then some amount of sliding is occurring. The rolling constraint is non-holonomic if the relative rotation between the finger and the object can occur about more than one axis.

Tracking of rolling contact requires either a numerical integration at small time steps [83], or an assumption about the path which the contact point travels. Since this simulation does not have sufficiently small time steps, it is necessary to make an assumption about the path of the contact location. The first assumption made is that the object traveled the shortest distance from the original location to the new location. By assuming that the object traveled the shortest distance consistent with rolling, it can also be assumed that the contact point on the object and the contact point on the finger each travel along a direct path on their respective surfaces during this time step. (A direct path is defined as a curve along the surface of the object projects as a line in the plane tangent to the surface at the original point of contact. For a spherical surface, this is a great circle.) Finally, it is assumed that the values of local curvature are known for all points on the surface of the object and finger.

If pure rolling has occurred with the direct path assumption, the change in contact position on the surface of the fingertip will be the same magnitude and direction as the change in contact position on the surface of the object. Since the human fingers are not six degree of freedom mechanisms, the orientation of the finger tip cannot be controlled independently of the finger tip position. In fact, the parallel fingers each have only three degrees-of-freedom, so for a given finger tip position there is a single finger tip orientation. For rolling, this means that for a given initial contact location and a given object motion there is a

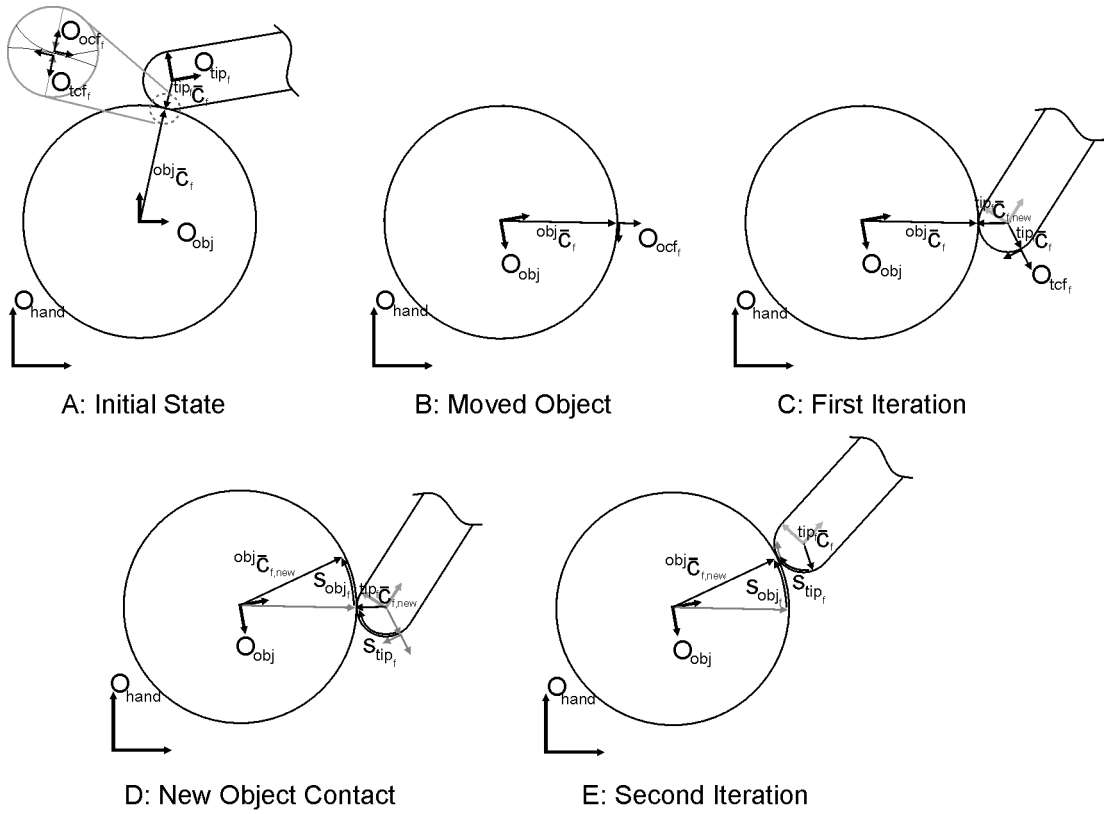

Figure 6-5. Determination of New Finger Location Subject to Rolling Constraint

unique "new" contact location on the object which meets the rolling constraint and the finger inverse kinematics defined in [4.3]. The following method finds the correct new contact location for each finger in contact and not sliding on the object.

Before the object is moved, the contact point is defined in the object frame and in the fingertip frame such that:

$$ {}^{0}_{obj}\mathbf{D} \cdot {}^{obj}\bar{\mathbf{c}} = {}^{0}_{tip_f}\mathbf{D} \cdot {}^{tip_f}\bar{\mathbf{c}} \qquad \textbf{(6-5)} $$

where ${}^{0}_{obj}\mathbf{D}$ is the homogenous transformation matrix from the hand frame to the object frame, ${}^{obj}\bar{\mathbf{c}}_f$ is the homogeneous position vector $\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ of the contact point in the object frame, ${}^{0}_{tip_f}\mathbf{D}$ is the homogenous transformation matrix from the hand frame to the fingertip frame and ${}^{tip_f}\bar{\mathbf{c}}_f$ is the homogenous position vector of the contact point in the fingertip frame. Figure 6-5A shows the initial configuration for a finger in contact with a spherical object. The image is planar for clarity and ease of visualization, though the procedure is general to non-planar rolling.

A contact frame is defined in the object frame with the origin at the contact point and the z-axis pointed along the object outward normal. Similarly, a contact frame is defined in the finger frame with the origin at the contact point and the z-axis pointed along the fingertip's outward normal. The orientation of the x and y axes is somewhat arbitrary, though it is common to place the x axis parallel to the surface arc with the smallest radius of curvature at the contact point. The homogenous transformation ${}^{ocf_f}_{tcf_f}\mathbf{D}$ from the fingertip contact frame O_{tcf_f} to the object contact frame O_{ocf_f} for finger f represents a reversal of the z-axis and a rotation about the z-axis:

$$ {}^{ocf_f}_{tcf_f}\mathbf{D} = \begin{bmatrix} \cos(\gamma_f) & -\sin(\gamma_f) & 0 & 0 \\ \sin(\gamma_f) & \cos(\gamma_f) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(6-6)} $$

where γ_f is the relative angle between the x axes.

In Figure 6-5B, the object position has changed as described in 6.3.1. It is now necessary to determine the new contact location of the non-sliding fingertips on the surface of the object. The procedure is an iterative search over the surface of the object for the location which satisfies the rolling constraint and the hand kinematics.

The first iteration, shown in Figure 6-5C, is to assume that the contact location on the object did not change. The location of the fingertip in the object frame would not change, though its orientation would, due to the limited degrees of freedom of the finger. As can be seen in the figure, at this first iteration, there is a new point of contact on the finger ${}^{tip_f}\bar{\mathbf{c}}_{f,\,new}$.

Using the inverse kinematics discussed in Chapter 4, the transformation matrix from the hand frame to the new fingertip frame ${}^{0}_{tip_f}\mathbf{D}_{new}$ is calculated. The contact location ${}^{tip_f}\bar{\mathbf{c}}_{f,\,new}$ in the fingertip frame is determined by

$$ {}^{tip_f}\bar{\mathbf{c}}_{f,\,new} = \left({}^{0}_{tip_f}\mathbf{D}_{new}\right)^{-1} \cdot {}^{0}_{obj}\mathbf{D}_{new} \cdot {}^{obj}\bar{\mathbf{c}}_{f} \quad \textbf{(6-7)} $$

where ${}^{0}_{obj}\mathbf{D}_{new}$ is the transformation to the moved object frame.

The new contact location ${}^{tcf_f}\bar{\mathbf{c}}_{f,\,new}$ in the tip contact frame at the initial contact location O_{tcf} is

$$ {}^{tcf_f}\bar{\mathbf{c}}_{f,\,new} = {}^{tcf_f}_{tip_f}\mathbf{D} \cdot {}^{tip_f}\bar{\mathbf{c}}_{f,\,new} \quad \textbf{(6-8)} $$

where ${}^{tcf_f}_{tip_f}\mathbf{D}$ is the transformation from the fingertip frame to the initial tip contact frame.

The length and direction of the minimum-length path along the surface of the finger from the origin of the tip contact frame to the new tip position can now be calculated. If the local curvature of the fingertip is assumed to be spherical, these values can be determined by

$$s_{tip_f} = r_k \cdot \text{acos}\left(\frac{1 - {}^{tcf_f}\bar{\mathbf{c}}_{f,new}[z]}{r_k}\right) \quad \textbf{(6-9)}$$

$$\alpha_{tip_f} = \text{atan2}({}^{tcf_f}\bar{\mathbf{c}}_{f,new}[y], {}^{tcf_f}\bar{\mathbf{c}}_{f,new}[x]) \quad \textbf{(6-10)}$$

where s_{tip_f} is the path length along the fingertip surface, α_{tip_f} is the angle of the path in XY plane of the tip contact frame, r_k is the local radius of curvature and ${}^{tcf_f}\bar{\mathbf{c}}_{f,new}[\ldots]$ are the x, y and z components of the location of the new contact point in the initial tip contact frame ${}^{tcf_f}\bar{\mathbf{c}}_{f,new}$. Figure 6-5D shows s_{tip_f} for a particular case in which α_{tip_f} (rotation about the z-axis of O_{tcf_f}) is zero.

Under rolling constraints, the length and path of the travel of the contact point along the surface of the object must be same as along the surface of the fingertip. This is not the case for the first iteration fingertip model in Figure 6-5C, since the contact point does not move on the object but moves by s_{tip_f} on the fingertip. To iterate toward the correct solution, a new contact point on the object is calculated to match the contact point path of s_{tip_f} and α_{tip_f}.

$$s_{obj_f} = s_{tip_f} \quad \textbf{(6-11)}$$

$$\alpha_{obj_f} = -\alpha_{tip_f} + \gamma_f \quad \textbf{(6-12)}$$

where γ_f is the relative rotation between the contact frames for finger f, as shown in Equation 6-6.

The new contact position for finger f on the object is found by following a path of length s_{obj_f} along the surface of the object in the direction α_{obj_f} in the object contact frame. If the local curvature of the object is also modeled as being a sphere, the new contact position is calculated by

$${}^{ocf_f}\bar{\mathbf{c}}_{f,new} = \begin{bmatrix} \cos(\alpha_{obj_f}) \cdot \sin(\omega_{obj_f}) \cdot r_{k,obj} \\ \sin(\alpha_{obj_f}) \cdot \sin(\omega_{obj_f}) \cdot r_{k,obj} \\ (\cos(\omega_{obj_f}) - 1) \cdot r_{k,obj} \end{bmatrix} \quad \textbf{(6-13)}$$

where $r_{k,obj}$ is local radius of curvature of the object, and $\omega_{obj_f} = s_{obj_f} / r_{k,obj}$ is the angle sub-tended by the arc at the center of curvature.

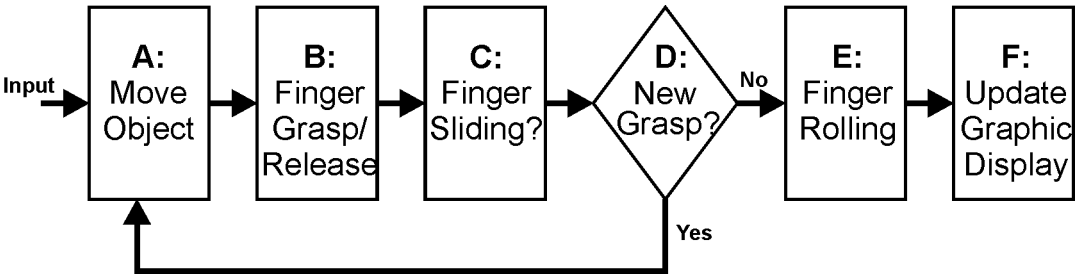

Figure 6-6. Representation of Complete Algorithm for Tracking Intended Human Manipulation

The new contact location on the object in the object frame is

$$ {}^{obj}\bar{\mathbf{c}}_{f,\,new} = \left({}^{ocf_f}_{obj}\mathbf{D}\right)^{-1} \cdot {}^{ocf_f}\bar{\mathbf{c}}_{f,\,new} \qquad \textbf{(6-14)} $$

where ${}^{ocf_f}_{obj}\mathbf{D}$ is the transformation from the object frame to the object contact frame for finger f. Figure 6-5D shows the calculated new contact location based on s_{tip_f} and α_{tip_f} for the sample case.

This is not the final answer, as shown in Figure 6-5E, because if the contact location is at ${}^{obj}\bar{\mathbf{c}}_{f,\,new}$, the contact point on the finger is no longer at ${}^{tip_f}\bar{\mathbf{c}}_{f,\,new}$. The values of s_{tip_f} and s_{obj_f} are closer than in the first iteration, but are not equal. To find the location where they are equal requires further iteration. The value of ${}^{0}_{tip_f}\mathbf{D}_{new}$ is recalculated for a finger in contact at ${}^{obj}\bar{\mathbf{c}}_{f,\,new}$ in the moved object frame. Equation 6-7 through Equation 6-14 are repeated, using ${}^{obj}\bar{\mathbf{c}}_{f,\,new}$ in place of ${}^{obj}\bar{\mathbf{c}}_{f}$ in Equation 6-7. The process is repeated until the change in contact position on the object from one iteration to the next is sufficiently small.

6.3.5 Integrated Algorithm

The integration of object motion, finger grasp and release, sliding and rolling into a single process is shown in Figure 6-6. The input to the algorithm is the current state of the hand, object and grasp as well as the commanded fingertip positions as measured from the user. In step A, the new location of the object is calculated to minimize the difference between the fingertips in contact and the commanded fingertip positions as described in 6.3.1. In step B, the commanded tip positions are compared to the surface of the new object location to determine if any fingers have gained or lost contact as described in 6.3.2. In step C, the

effective contact force of each finger in contact with the object is compared to the friction cone to determine which, if any, finger should slide as described in 6.3.3.

Step D is a branch point in the decision path based on changes in the grasp configuration. The new object position calculated in Step A is based on the subset of fingers which are in contact in the input configuration. However, if the subset of fingers in contact changes in step B, the calculated object position is no longer the minimized location for the grasp state. Similarly, when a finger slides on the surface of the object, the grasp configuration can change significantly. If there is any change in the grasp configuration, either from finger grasp/release or sliding, the process returns to step A to recalculate the object position using the new configuration. If there is no change in the grasp configuration, the process continues to step E.

In step E, fingers which are in contact at the beginning of the process and have not lost contact or slid have their new contact locations calculated according to the rolling constraint as described in 6.3.4. Finally, in step F, the new state of the hand-object system is displayed to the user through the graphical interface.

6.4 Testing Manipulation Tracking

As stated previously, the purpose of this framework is to allow a user to manipulate a simulated object in an intuitive and natural manner. To demonstrate this, a user held a real object in between his fingers while wearing the CyberGlove and interacting with a virtual object of identical shape and size. The motion of the physical object was tracked by a video camera and compared to the motion of the simulated object.

6.4.1 Testing Setup

The user's hand was calibrated using the four fingered calibration routine described in 4.2.2. An accurate calibration is particularly critical to this experiment, since it is a direct comparison of the actual hand and the modeled hand. In the telemanipulation tasks described in 5.2, the user could compensate for small errors in calibration through the use of visual feedback. The need for an accurate calibration in this testing was the motivation for several of the developments in the calibration procedure, such as the weighting matrix and constraining the TR sensor gain.

The user's arm and wrist were constrained to minimize motion, since the absolute motion of the hand is not currently measured. The object was grasped by the user and held against a clear acrylic sheet with a grid imprinted on it.

A video camera placed on the opposite side of the acrylic sheet recorded the motion of the object. After manipulation, the video images were digitized and imported into the computer. The relative motion of the object from its starting location is measured by tracking fiducial marks on the object.

In the simulation, the model hand fingers are placed into contact with the virtual object in the same grasp configuration as the user's hand on the physical object. As the simulation tracks the input from the CyberGlove, the resulting position and orientation of the object is recorded at each time step.

6.4.2 Three Fingered Grasp on 50 mm object

The user held a spherical object with a diameter of 50 mm between the thumb, index and middle fingers. With the wrist constrained and the object held against the acrylic, the motion of the object was nominally constrained to lie a plane parallel to the XZ plane (parallel to the flat palm).

The user manipulated the object for thirty seconds, applying rotations in both the positive and negative directions about the Y axis and translating in the X and Z directions. A comparison of the trajectories of the geometric center and orientation of the actual and simulated objects is shown in Figure 6-7.

The simulated object center motion tracks the actual object center well along the x-axis (which points from the base of the thumb to the base of the index finger). The direction and shape of the two trajectories are visually similar, with a 3.9 mm RMS tracking error.

The rotation of the object about the y-axis shows a greater deviation in the tracking. The same general shape and direction can be seen in the motion of the simulated object as with the actual object, but the simulated object rotates noticeably more. The RMS deviation between the two values is 0.20 radians (11.7º).

The deviation in object orientation can be explained by looking at the comparison of the trajectories along the z-axis. The visual similarity between actual and simulated trajectories

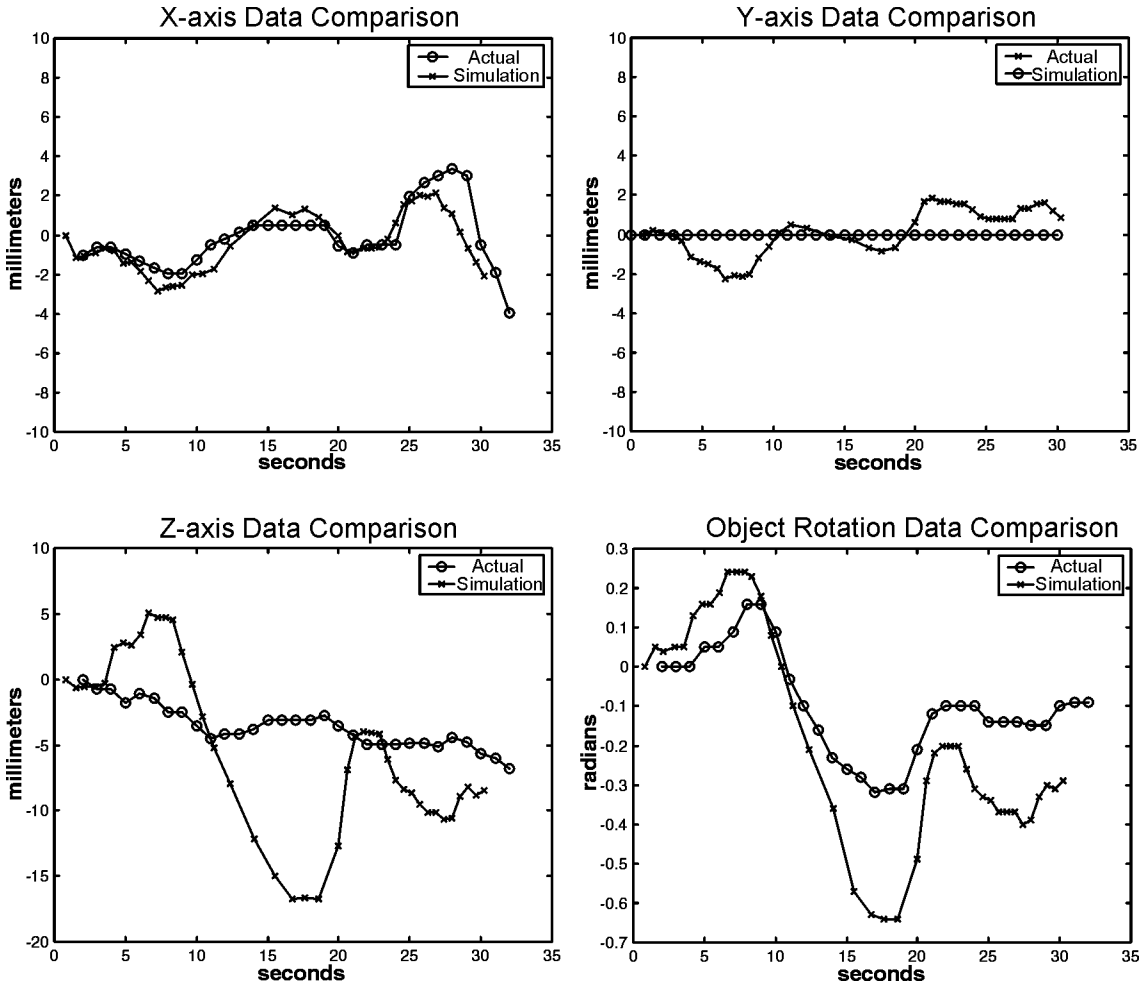

Figure 6-7. Comparison of Actual Object Motion and Simulated Object Motion

disappears, and the RMS error is 7.7 mm. The cause of this disparity is the gain on the thumb TR sensor. As discussed in 4.2.2, this sensor is significantly non-linear, and tends to calibrate to a high value. For this testing, the gain was fixed at a nominal value. The large motions of the simulated object along the z axis indicate that this nominal value may still be too large for the range of thumb motion used in this manipulation.

The assumption that the motion of the object is nominally constrained to a plane parallel to XZ plane is shown to be reasonable by examining the motion of the center of the simulated object along y-axis. The simulated object has a maximum deviation in the y direction of 1.8 mm, with an RMS deviation of 1.2 mm. In addition, the rotation of the simulated object about the y-axis was typically four to five times larger than the rotation about the x or z-axes.

In summary, the simulation does a fairly good job of capturing the actual intention of the user in performing object manipulation. The primary limitation to better natural manipulation remains the difficulty of accurately calibrating the glove sensors for a particular human hand.

7 Conclusions and Future Work

7.1 Conclusions

This thesis has examined three different approaches for controlling robotic manipulation, with the intention of improving the flexibility of robot performance and the implementation of new tasks.

7.1.1 Autonomous Dexterous Manipulation

Chapter 3 describes the development of a manipulation procedure for the exploration of an unknown object by an autonomous dexterous robot hand. The robot was able to successfully grasp and manipulate an unknown object such that the robot fingers could circumnavigate the surface in the plane of finger actuation. Sensors on the robot fingers were able to detect both events, such as contact, and features, such as a ridge on the object surface.

However, the robot was not successful on many attempts. The fundamental limitation in robot manipulation is in the control structure rather than in the hardware. The robot can only respond and recover in situations for which it has been explicitly programmed. To attempt to construct a state diagram which comprehensively addresses every anticipated difficulty would be cumbersome even for a simple task.

7.1.2 Hand Model

Chapter 4 introduces the kinematic model and calibration of the human hand, the use of which allows for more direct human control of the robot manipulation. The model is based on biological knowledge of the hand, though it is modified for kinematic simplicity and use with the CyberGlove. The calibration routine is designed to match the kinematic model to a particular user's hand without the need for a visual tracking system.

The model successfully captures the degrees of freedom and range of motion of the first four fingers of the human hand. It models each of the joints as a simple hinge joint, which

has been shown by Rohling and Hollerbach [106] to introduce errors on the order of only 1%. It does not model the palmar arch motion of the hand, which is not accurately measured by the CyberGlove and predominantly affects the motion of the pinky finger. The complex motion about the two skew axes at the carpometacarpal joint of the thumb is modeled as motion about three non-skew axes with the motion of one axis being a linear function of the other two axes. This has the effect of capturing the motion of the thumb while maintaining kinematic simplicity.

The calibration procedure models the motion of the thumb and finger touching tip to tip as a closed kinematic chain. This procedure was able to successfully calibrate the CyberGlove and model to a new user for the purpose of dexterous manipulation. However, it was necessary to account for several factors to produce a non-trivial, biologically acceptable solution. First, the closed kinematic chain does not have an absolute measurement of chain length, and tended to converge to a zero-length trivial solution. Second, calibrating for a large number of parameters can result in biologically unacceptable local minima such as hyperflexed joint angles. Third, simultaneously calibrating for parameters of different units (such as radians and millimeters) can hinder convergence due to scaling issues. These factors can be compensated for by including a deviation matrix which encourages biologically feasible solutions and a scaling matrix that equalizes the effect of each parameter.

7.1.3 Telemanipulation

The hand model and calibration routine were used to allow a user to control a robot by wearing the CyberGlove and CyberGrasp force feedback device. The users performed two prototypical manipulation tasks, both with force feedback and without.

The calibration and hand model were good enough for users to successfully perform the tasks. The model matched the desired relative fingertip motions of the user, such that the robot moved in a manner which was controllable and predictable even for a novice user.

The addition of force feedback to the user's hand during manipulation enabled a gentler and generally more stable performance, though limitations in the CyberGrasp mechanism and the speed of the robot arm meant that force feedback did not improve task completion time.

The most significant variation from subject to subject was the quality of the mapping from the human hand model to the robot kinematics. This mapping was manually customized for each user, and lacked a quantitative estimate of quality.

7.1.4 Tracking Human Manipulation

In order to bypass the difficulties of mapping to an non-anthropomorphic robot hand, Chapter 6 develops a method for tracking human manipulation of a virtual representation of the object. In the virtual environment, the user is controlling a four-fingered virtual hand using the developed model and calibration. Since the virtual hand is anthropomorphic, the user can use natural manipulation strategies such as rolling, sliding and regrasping.

The virtual environment assumes quasi-static manipulation with point contact between the fingers and the object. Fingers are allowed to acquire and release contact as well as slide along the surface. Fingers which maintain non-sliding contact will roll on the object surface.

The process was tested by having a user hold a real object in their hand, and comparing the motion of the real object with the motion of the virtual object. In general, the motion of the virtual object captured the essence of the real object motion. Some motions, however, were accentuated due to non-linearities in the CyberGlove measurements.

7.2 Future Work

Using the algorithm presented in the Chapter 6, a user is able to demonstrate a desired object manipulation. The next step is to convert the recorded object trajectory into a program to allow the robot to perform the same or similar manipulation upon a real object.

7.2.1 Autonomous Programming

Many researchers have examined the issue of autonomously generating a program for a dexterous robot hand to perform a specific manipulation. The challenge is to be able to determine a satisfactory motion of the robot fingers which will result in the desired change in object configuration while maintaining grasp stability.

The control of the instantaneous motion of an object held with a specific grasp has been understood for several years. Mason and Salisbury [84] develop the grasp matrix relating

the fingertip forces to the resulting force on the object. Montana [91] and Cai and Roth [19] provide useful mathematical descriptions of the rolling constraint between two surfaces for the purpose of manipulation.

The manipulation of an object to a desired configuration is likely to exceed the finite workspace of one or more of the robot fingers. In this case, to continue the motion of the object it is necessary to change the grasp configuration through sliding or finger release. Hong, Lafferriere et al. [49] proposed using this method of finger gaiting to change from one "good" grasp to another, which is analogous to the procedure used by humans [70]. The work presented in Chapter 3, as well as Montana's [92] work on a robot twirling a baton are specific implementations of this strategy.

Recent work has combined these ideas into a general manipulation planner which will generate motions of the robot fingers to achieve an arbitrary object configuration. Cherif and Gupta [23] present an algorithm for reorienting 3-D smooth object with rolling and sliding by combining a global planner to move toward the desired position and a local planner to avoid restricted motions. Han and Trinkle [40] build a planner which includes finger contact release as well as the specification of the desired final grasp state.

The commanded trajectory realized by the virtual tracking in Chapter 6 can be recreated by the robot by incorporating a manipulation planner. The proposed process overview is shown in Figure 7-1. One benefit of this process is that the transfer from the object motion in the virtual world to the robot path planner can take place with arbitrary delays. For example, the robot may only trail the human hand by a fraction of a second or by minutes or hours as appropriate for the task and communication capabilities.

The dotted line of feedback between the actions in the real world and the human is the most significant benefit to this system over a robot strictly running a path planning program. If the motion of the object is different than intended, or if unexpected difficulties arise, the human is still "in control" and can utilize the adaptability and experience which are difficult to incorporate into a robot. Perhaps the best "intelligence" we can ask for from the robot is

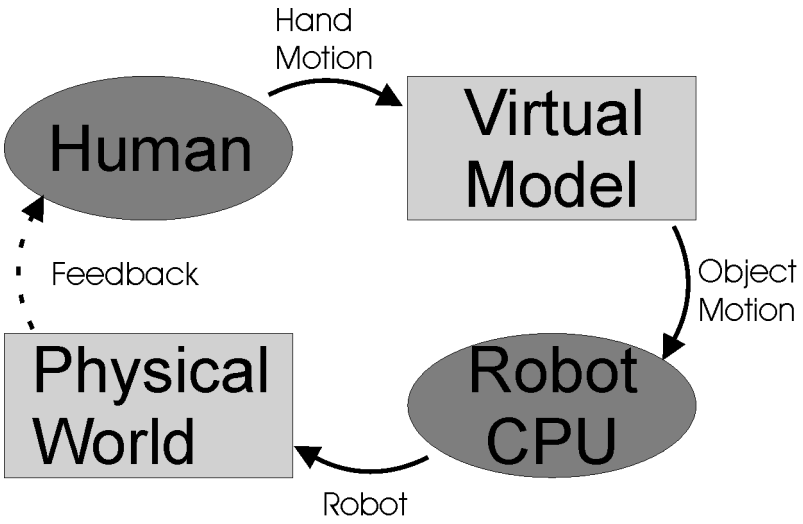

Figure 7-1. Proposed Information Loop for Programming by Demonstration

to utilize its sensors to prevent unintended dropping of the object and to recognize events which are beyond its scope and request human intervention.

The future implementation of this process is still non-trivial. The previously described planners do not incorporate knowledge of the robot kinematics into the path planning. The robot may have a more restricted range of motion compared to the human hand (such as the two fingered robot used in Chapter 5) which will prevent perfect trajectory matching. If the path planner includes the robot kinematics, it can develop an acceptable alternative trajectory.

The interesting question raised is “What is the best alternative trajectory?” The smallest integral of the positional error may be sufficient for many applications, but in some cases certain axes may take precedence over others. For example, if the robot is manipulating an open topped reservoir such as a cup of coffee, maintaining the vertical orientation is more important than tracking the position of the object center. How can the human convey that priority information to robot in an intuitive manner?

7.2.2 Grasp Selection

One challenging aspect of robot manipulation is in grasp selection. What qualifies as a good grasp? How much better is one grasp over another? What is the best grasp for this object, this robot hand and this particular task?

The most important aspect of a specific grasp is the presence of force closure, or the ability of the grasp to resist an arbitrary disturbance wrench. This can be inferred from the grasp matrix [84]. The challenge lies in searching the space of all possible grasps and in optimizing for the selection criteria.

As discussed in Chapter 2, there are a number of choices for a grasp metric. The uncertainty grasp index is a measure of how robust the grasp stability is for small finger position errors [101]. Another option is to find the grasp with the minimum grasping energy [15]. The minimization of the effect of gravitational and inertial forces can be achieved by placing the object's center of mass in the geometric center of the grasp points [32].

Perhaps the most useful grasp metric would be manipulability, or the ability to accommodate object motion [63]. This takes into account the kinematics of the fingers in contact. (A grasp which securely holds an object, but does not allow object motion would not be very effective for the proposed system.)

One opportunity of this system is to have the robot grasp selection use information from a human demonstration. Humans can draw on their experience to intuit a suitable grasp for a particular object for their intended manipulation. By observing the grasp choice the user applied on the virtual object, we gain a reasonable starting point for the robot grasp. If the number of robot fingers is lower than the number of fingers used by the human user, the "virtual" finger approximation introduced by Iberall and MacKenzie [56] can be used.

7.3 Final Thoughts

This thesis has examined three different methods for controlling dexterous robot manipulation, with varying levels of human input and robot autonomy. The ultimate choice of method will depend on the desired task for a particular implementation. This research has provided a natural and intuitive means for humans to control a robot by wearing an instrumented glove and manipulating a virtual object that corresponds to a robot's real object.

Bibliography

[1] Adams, R. J., M. R. Moreyra, et al. (1998). "Stability and performance of haptic displays: Theory and experiments." 1998 ASME International Mechanical Engineering Congress and Exposition, Anaheim, CA, USA, 227-234.

[2] Allen, P., P. Michelman, et al. (1989). “Current research in robotics and automation - an intelligent grasping system.” Computer **22**(3): 50-52.

[3] Allen, P. K., P. Michelman, et al. (1990). “A system for programming and controlling a multisensor robotic hand.” IEEE Transactions on Systems, Man and Cybernetics **20**(6): 1450-1456.

[4] Allen, P. K., A. Timcenko, et al. (1993). “Automated tracking and grasping of a moving object with a robotic hand-eye system.” IEEE Transactions on Robotics and Automation **9**(2): 152-164.

[5] Allen, P. K., A. T. Miller, et al. (1997). "Using tactile and visual sensing with a robotic hand." 1997 IEEE International Conference on Robotics and Automation, ICRA. Part 1 (of 4), Albuquerque, NM, USA, 676-681.

[6] An, K. N., E. Y. Chao, et al. (1979). “Normative model of the human hand for biomechanical analysis.” Journal of Biomechanics **12**: 775-788.

[7] Bennet, D. J. and J. M. Hollerbach (1990). Closed-loop kinematic calibration of the Utah-MIT hand. Experimental Robotics I: The First International Symposium. V. Hayward and O. Khatib. New York, Springer-Verlag**:** 539-552.

[8] Boulic, R., S. Rezzonico, et al. (1996). "Multi-finger manipulation of virtual object." ACM Symposium on Virtual Reality, Software and Technology, 67-74.

[9] Brunner, B., K. Arbter, et al. (1994). "Task directed programming of sensor based robots." IEEE/RSJ/GI International Conference on Intelligent Robots and Systems. Part 2 (of 3), Munich, Ger, 1080-1087.

[10] Bruns, F. W. (1998). "Integrated Real and Virtual Prototyping." IECON '98,, Aachen, Germany.

[11] Bruns, W. F. (1999). "Complex construction kits for coupled real and virtual engineering workspaces." CoBuild '99, Pittsburgh, PA.

[12] Burdea, G. and J. Zhuang (1991). “Dextrous telerobotics with force feedback. An overview Part 2. Control and implementation.” Robotica **9**(3): 291-298.

[13] Burdea, G. and J. Zhuang (1991). “Dextrous telerobotics with force feedback - an overview. Part 1. Human factors.” Robotica **9**(2): 171-178.

[14] Burdea, G. (1996). Force and touch feedback for virtual reality. New York, John Wiley & Sons.

[15] Buss, M., H. Hashimoto, et al. (1995). "Grasping force optimization for multi-fingered robot hands." 1995 IEEE International Conference on Robotics and Automation. Part 1 (of 3), Nagoya, Jpn, 1034-1039.

[16] Buss, M., H. Hashimoto, et al. (1996). “Dextrous hand grasping force optimization.” IEEE Transactions on Robotics and Automation **12**(3): 406-418.

[17] Buttolo, P., D. Kung, et al. (1995). "Manipulation in real, virtual and remote environments." 1995 IEEE International Conference on Systems, Man and Cybernetics. Part 5 (of 5), Vancouver, BC, Can, 4656-4661.

[18] Cai, C. and B. Roth (1987). "On the spatial motion of a rigid body with point contact." IEEE International Conference on Robotics and Automation, Raleigh, NC, 686-695.

[19] Cai, C. and B. Roth (1988). "On the spatial motion of a rigid body with line contact." IEEE International Conference on Robotics and Automation., Philadelphia, PA, 1036-1041.

[20] Cannon, D. J., G. Thomas, et al. (1994). “Virtual reality based point-and-direct robotic system with instrumented glove.” International Journal of Industrial Engineering - Applications and Practice **1**(2): 139-148.

[21] Cannon, D. and G. Thomas (1997). “Virtual tools for supervisory and collaborative control of robots.” Presence **6**(1): 1-28.

[22] Chang, D. C., M. R. Cutkosky (1995). "Rolling with deformable fingertips." Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Part 2 (of 3) Pittsburgh, PA, 194-199.

[23] Cherif, M., and K.K. Gupta (1999) "Planning quasi-static fingertip manipulations for reconfiguring objects." IEEE Transactions on Robotics and Automation, Vol. 15, No. 5, 837-848.

[24] Conway, L., R. Volz, et al. (1987). "Tele-autonomous systems: methods and architectures for intermingling autonomous and telerobotic technology." IEEE International Conference on Robotics and Automation, Raleigh, NC, 1121-1130.

[25] Conway, L., R. A. Volz, et al. (1990). “Teleautonomous systems: Projecting and coordinating intelligent action at a distance.” IEEE Transactions on Robotics and Automation: 146-158.

[26] Cooney, W. P., M. J. Lucca, et al. (1981). “The kinesiology of the thumb trapeziometacarpal joint.” Journal of Bone Joint Surgery **63A**: 1371-1381.

[27] Cutkosky, M. R. (1989). “On grasp choice, grasp models, and the design of hands for manufacturing tasks.” IEEE Transactions on Robotics and Automation **5**(3): 269-279.

[28] Cutkosky, M. R. and R. D. Howe (1990). Human grasp choice and robotic grasp analysis. Dextrous Robot Hands. S. T. Venkataraman and T. Iberall. New York, Springer-Verlag**:** 5-31.

[29] CyberGlove Reference Manual. Virtual Technologies Inc. 1998

[30] Demmel, J. G. Lafferriere, et al. (1988) "Theoretical and experimental studies using a multifinger planar manipulator." IEEE International Conference on Robotics and Automation. Philadelphia, PA. 390-395

[31] Denavit, J. and R. S. Hartenberg (1955). “A kinematic notation for lower-pair mechanisms based on matrices.” Journal of Applied Mechanics(June 1955): 215.

[32] Ding, D., Y. -H. Liu et al. (2000). "Computing 3-D optimal form-closure grasps." IEEE International Conference on Robotics and Automation. San Francisco, CA. 3573-3578.

[33] Fischer, M., P. van der Smagt, et al. (1998). "Learning techniques in a dataglove based telemanipulation system for the DLR hand." 1998 IEEE International Conference on Robotics and Automation. Part 2 (of 4), Leuven, Belgium, 1603-1608.

[34] Gomez, D., G. Burdea, et al. (1995). "Modeling of the _Rutgers Master II' haptic display." 1995 ASME International Mechanical Engineering Congress and Exposition. Part 2 (of 2), San Francisco, CA, USA, 727-734.

[35] Graves, S. and R. Volz (1995). "Action selection in teleautonomous systems." 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Part 3 (of 3), Pittsburgh, PA, USA, 14-19.

[36] Griffin, W. B., R. P. Findley, et al. (2000). “Calibration and mapping of a human hand for dexterous manipulation.” 2000 ASME IMECE Symposium on Haptic Interfaces.

[37] Grupen, R. A., T. C. Henderson, et al. (1989). “Survey of general-purpose manipulation.” International Journal of Robotics Research **8**(1): 38-62.

[38] Guirintano, D. J., A. M. Hollister, et al. (1995). “A virtual five-link model of the thumb.” Medical Engineering and Physics **17**(4): 297-303.

[39] Han, L., J. C. Trinkle, et al. (1999). "Grasp analysis as linear matrix inequality problems." IEEE International Conference on Robotics and Automation, Detroit, MI, 1261-1268.

[40] Han, L., J. C. Trinkle (1998). "Dextrous manipulation by rolling and finger gaiting." IEEE International Conference on Robotics and Automation, Leuven, Belgium. 730-735.

[41] Hannaford, B. (1989). “Stability and performance tradeoffs in bi-lateral telemanipulation.” **III**(of 3). Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. Available from IEEE Service Cent (cat): 1764-1767.

[42] Hannaford, B., L. Wood, et al. (1991). “Performance evaluation of a six-axis generalized force-reflecting teleoperator.” IEEE Transactions on Systems, Man and Cybernetics 21(3): 620-633.

[43] Hashimoto, H., Y. Kunii, et al. (1993). "Dynamic force simulator for force feedback human-machine interaction." 1993 IEEE Annual Virtual Reality International Symposium, Seattle, WA, USA, 209-210.

[44] Hayati, S. and M. Mirmirani (1985). “Improving the absolute positioning accuracy of robot manipulators.” Journal of Robotic Systems **2**(4): 397-413.

[45] Hilhorst, R. A. and K. Tanie (1994). "Dexterous manipulation of objects with unknown parameters by robot hands." 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 3098-3103.

[46] Hirzinger, G., B. Brunner, et al. (1993). “Sensor-based space robotics - ROTEX and its telerobotic features.” IEEE Transactions on Robotics and Automation **9**(5): 649-661.

[47] Hollerbach, J. M. and C. W. Wampler (1996). “The calibration index and taxonomy for robot kinematic calibration methods.” International Journal on Robotics Research **15**(6): .573-591.

[48] Hollister, A., W. L. Buford, et al. (1992). “The axes of rotation of the thumb carpometacarpal joint.” Journal of Orthopaedic Research **10**: 454-460.

[49] Hong, J.W., G. Lafferriere et al. (1990). "Fine manipulation with multifinger hands." IEEE International Conference on Robotics and Automation, Cincinnati, OH, 1568-1573

[50] Howe, R. D., I. Kao, et al. (1988). "Sliding of robot fingers under combined torsion and shear loading." IEEE International Conference on Robotics and Automation, Cincinnati, OH, 103-105.

[51] Howe, R. D. (1992). "A force-reflecting teleoperated hand system for the study of tactile sensing in precision manipulation." IEEE International Conference on Robotics and Automation, Nice, France, 1321-1326.

[52] Hurmuzlu, Y., A. Ephanov, et al. (1998). “Effect of a pneumatically driven haptic interface on the perceptual capabilities of human operators.” Presence **7**(3).

[53] Hyde, J. M. and M. R. Cutkosky (1993). "Contact transition control: An experimental study." IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 363-368.

[54] Hyde, J. M. and M. R. Cutkosky (1994). “Controlling contact transition.” IEEE Control Systems Magazine **14**(1): 25-30.

[55] Hyde, J. M. and M. R. Cutkosky (1998). “Phase management framework for event-driven dextrous manipulation.” IEEE Transactions on Robotics and Automation **14**(6): 978-985.

[56] Iberall, T. and C. L. MacKenzie (1990). Opposition space and human prehension. Dextrous Robot Hands. S. T. Venkataraman and T. Iberall. New York, Springer-Verlag**:** 32-54.

[57] Jacobsen, S. C., E. K. Iversen et al. (1986) "Design of the Utah/MIT Dextrous Hand." IEEE International Conference on Robotics and Automation, San Francisco, CA. 1520-1532.

[58] Jau, B. M. (1995). "Dexterous telemanipulation with four fingered hand system." 1995 IEEE International Conference on Robotics and Automation. Part 1 (of 3), Nagoya, Jpn, 338-343.

[59] Jiang, S. L., K. K. Choi, et al. (1999). "Coordinated motion generation for multifingered manipulation using tactile feedback." IEEE International Conference on Robotics and Automation, Detroit, MI, 3032-3037.

[60] Kang, S. B. and K. Ikeuchi (1994). "Robot task programming by human demonstration: Mapping human grasps to manipulator grasps." IEEE/RSJ/GI International Conference on Intelligent Robots and Systems. Part 1 (of 3), Munich, Ger, 97-104.

[61] Kang, S. B. and K. Ikeuchi (1997). “Toward automatic robot instruction from perception - mapping human grasps to manipulator grasps.” IEEE Transactions on Robotics and Automation **13**(1): 81-95.

[62] Kao, I., M. R. Cutkosky, et al. (1997). “Robotic stiffness control and calibration as applied to human grasping tasks.” IEEE Transactions on Robotics and Automation **13**(4): 557-566.

[63] Kerr, J. and B. Roth (1986). “Analysis of Multifingered Hands.” International Journal of Robotics Research **4**(4): 3-17.

[64] Kerr, J. and B. Roth (1986). "Special grasping configurations with dexterous hands." IEEE International Conference on Robotics and Automation, San Francisco, CA, 1361-1367.

[65] Khatib, O. (1987). “Unified approach for motion and force control of robot manipulators: the operational space formulation.” IEEE Journal of Robotics and Automation **3**(1): 43-53.

[66] Kheddar, A., C. Tzafestas, et al. (1997). "Hidden robot concept - high level abstraction teleoperation." 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Part 3 (of 3), Grenoble, Fr, 1818-1824.

[67] Kheddar, A., C. Tzafestas, et al. (1997). "Parallel multi-robots long distance teleoperation." 1997 8th International Conference on Advanced Robotics, ICAR'97, Monterey, CA, USA, 1007-1012.

[68] Kheddar, A., C. Tzafestas, et al. (1998). “Multi-robot teleoperation using direct human hand actions.” Advanced Robotics **11**(8): 799-825.

[69] Kijima, R. and M. Hirose (1995). Fine object manipulation in virtual environments. Virtual Environments '95. M. Gobel. New York, Springer.

[70] Klatzky, R. L. and S. Lederman (1990). Intelligent exploration by the human hand. Dexterous Robot Hands. S. T. Venkataraman and T. Iberall, Springer-Verlag.

[71] Klatzky, R. L., S. J. Lederman, et al. (1991). “Task-driven extraction of object contour by human haptics. Part 1.” Robotica **9**(1): 43-51.

[72] Kontarinis, D. A., J. S. Son, et al. (1995). "Tactile shape sensing and display system for teleoperated manipulation." 1995 IEEE International Conference on Robotics and Automation. Part 1 (of 3), Nagoya, Jpn, 641-646.

[73] Kramer, J. F. (1996). Determination of thumb position using measurements of abduction and rotation. United States Patent #5482056.

[74] Kuch, J. J. and T. S. Huang (1994). "Human computer interaction via the human hand: a hand model." 28th Asilomar Conference on Signals, Systems & Computers. Part 2 (of 2), Pacific Grove, CA, USA, 1252-1256.

[75] Kunii, Y. and H. Hashimoto (1997). "Tele-teaching by human demonstration in virtual environment for robotic network system." 1997 IEEE International Conference on Robotics and Automation, ICRA. Part 1 (of 4), Albuquerque, NM, USA, 405-410.

[76] Lederman, S. J., R. L. Klatzky, et al. (1991). “Task-driven extraction of object contour by human haptics. Part 2.” Robotica **9**(2): 179-188.

[77] Li, L., B. Cox, et al. (1996). "Development of a telepresence controlled ambidextrous robot for space applications." 1996 13th IEEE International Conference on Robotics and Automation. Part 1 (of 4), Minneapolis, MN, USA, 58-63.

[78] Li, Z. X., Z. Qin, et al. (1998). "Coordinated motion generation and real-time grasping force control for multifingered manipulation." 1998 IEEE International Conference on Robotics and Automation. Part 4 (of 4), Leuven, Belgium, 3631-3638.

[79] Lloyd, J.E., J.S. Beis et al. (1999). "Programming contact tasks using a reality-based virtual environment integrated with vision." IEEE Transactions on Robotics and Automation 15 (3): 423-434.

[80] Lovchik, C. S. and M. A. Diftler (1999). "Robonaut hand: a dextrous robot hand for space." IEEE International Conference on Robotics and Automation, Detroit, MI, 907-912.

[81] MacKenzie, C. L. and T. Iberall (1994). The grasping hand. Amsterdam ; New York, North-Holland.

[82] Maekawa, H. and J. M. Hollerbach (1998). "Haptic display for object grasping and manipulating in virtual environment." 1998 IEEE International Conference on Robotics and Automation. Part 3 (of 4), Leuven, Belgium, 2566-2573.

[83] Maekawa, H., Tanie K. et al. (1995). "Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact." 1995 IEEE International Conference on Robotics and Automation, Nagoya, Jpn, 743-750

[84] Mason, M. T. and J. K. Salisbury (1985). Robot hands and the mechanics of manipulation. Cambridge, Mass., MIT Press.

[85] Massie, T. H. and K. J. Salisbury (1994). "PHANToM haptic interface: a device for probing virtual objects." 1994 International Mechanical Engineering Congress and Exposition, Chicago, IL, USA, 295-299.

[86] Massimino, M. J. and T. B. Sheridan (1994). “Teleoperator performance with varying force and visual feedback.” Human Factors **36**(1): 145-157.

[87] Michelman, P. and P. Allen (1994). "Shared autonomy in a robot hand teleoperation system." IEEE/RSJ/GI International Conference on Intelligent Robots and Systems. Part 1 (of 3), Munich, Ger, 253-259.

[88] Michelman, P. and P. Allen (1994). "Forming complex dextrous manipulations from task primitives." 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 3383-3388.

[89] Michelman, P. (1998). “Precision object manipulation with a multifingered robot hand.” IEEE Transactions on Robotics and Automation **14**(1): 105-113.

[90] Miller, A. T. and P. K. Allen (1999). "Examples of 3D grasp quality computations." IEEE International Conference on Robotics and Automation, Detroit, MI, 1240-1246.

[91] Montana, D. J. (1988). “Kinematics of Contact and Grasp.” International Journal of Robotics Research **7**(3): 17-32.

[92] Montana, D. J. (1995). “Kinematics of multi-fingered manipulation.” IEEE Transactions on Robotics and Automation **11**(4): 491-503.

[93] Nagai, K. and T. Yoshikawa (1995). "Grasping and manipulation by arm/multifingered-hand mechanisms." 1995 IEEE International Conference on Robotics and Automation. Part 1 (of 3), Nagoya, Jpn, 1040-1047.

[94] Nagashima, T., H. Seki, et al. (1997). “Analysis and simulation of grasping/manipulation by multi-fingersurface.” Mechanical Machine Theory **32**(2): 175-191.

[95] Nakamura, Y., T. Yoshikawa, et al. (1989). “Dynamics and stability in coordination of multiple robotic mechanisms.” International Journal of Robotics Research **8**(2): 44-61.

[96] Napier, J. R. (1956). “The prehensile movements of the human hand.” Journal of Bone and Joint Surgery **38**(4): 902-913.

[97] Okamura, A. M., M. L. Turner, et al. (1997). "Haptic exploration of objects with rolling and sliding." 1997 IEEE International Conference on Robotics and Automation, ICRA. Part 3 (of 4), Albuquerque, NM, USA, 2485-2490.

[98] Okamura, A. M., N. Smaby, et al. (2000). “An overview of dexterous manipulation.” IEEE International Conference on Robotics and Automation. San Francisco, CA. 255-262

[99] Paljug, E., X. Yun, et al. (1994). “Control of rolling contacts in multi-arm manipulation.” IEEE Transactions on Robotics and Automation **10**(4): 441-452.

[100] Pao, L. and T. H. Speeter (1989). “Transformation of human hand positions for robotic hand control.” **III**(of 3). Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. Available from IEEE Service Cent (cat): 1758-1763.

[101] Park, Y. C. and G. P. Starr (1990). "Optimal grasping using a multifingered robot hand." 1990 IEEE International Conference on Robotics and Automation, Cincinnati, OH.

[102] Pere, E., D. Gomez, et al. (1996). "PC-based virtual reality system with dextrous force feedback." 1996 ASME International Mechanical Engineering Congress and Exposition, Atlanta, GA, USA, 495-502.

[103] Richard, C. and M. R. Cutkosky (1997). "Contact force perception with an ungrounded haptic interface." 1997 ASME International Mechanical Engineering Congress and Exposition, Dallas, TX, USA, 181-187.

[104] Rohling, R. N. and J. M. Hollerbach (1993). "Optimized fingertip mapping for teleoperation of Dextrous robot Hands." IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 769-775.

[105] Rohling, R. N. and J. M. Hollerbach (1993). “Calibrating the human hand for haptic interfaces.” Presence **2**(4): 281-296.

[106] Rohling, R. N. and J. M. Hollerbach (1994). "Modeling and parameter estimation of the human index finger." 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 223-230.

[107] Rovetta, A., F. Cosmi, et al. (1995). “Teleoperator response in a touch task with different display conditions.” IEEE Transactions on Systems, Man and Cybernetics **25**(5): 878-881.

[108] Rus, D. (1999). “In-hand dexterous manipulation of piecewise-smooth 3-D objects.” International Journal of Robotics Research **18**(4): 355-381.

[109] Salisbury, K. (1988). “Issues in human/computer control of dexterous remote hands.” IEEE Transactions on Aerospace and Electronic Systems **24**(5): 591-596.

[110] Sarkar, N., X. Yun, et al. (1997). “Dynamic control of 3-D rolling contacts in two-arm manipulation.” IEEE Transactions on Robotics and Automation **13**(3): 364-376.

[111] Shastri, S. V., T. Iberall, et al. (1990). Dextrous robot hands. New York, Springer-Verlag.

[112] Shimoga, K. B. (1996). “Robot grasp synthesis algorithms: a survey.” International Journal of Robotics Research **15**(3): 230-266.

[113] Shimoga, K. B., A. M. Murray, et al. (1996). "Touch display system for interaction with remote and virtual environments." 1996 ASME International Mechanical Engineering Congress and Exposition, Atlanta, GA, USA, 523-529.

[114] Siira, J. and D. K. Pai (1996). "Haptic texturing - a stochastic approach." IEEE International Conference on Robotics and Automation. Minneapolis, MN, v.1 557-562

[115] Srinivasan, M. A. and K. Salisbury (1992). "Virtual Environment Technology fo Training (VETT)", BBN Report No. 7661, 1992, Cambridge, MA.

[116] Srinivasan, M. A. and J.-s. Chen (1993). "Human performance in controlling normal forces of contact with rigid objects." 1993 ASME Winter Annual Meeting, New Orleans, LA, USA, 119-125.

[117] Tan, H. Z., X. D. Pang, et al. (1992). "Manual resolution of length, force, and compliance." Winter Annual Meeting of the American Society of Mechanical Engineering, Anaheim, CA, 13-18.

[118] Tan, H. Z., N. I. Durlach, et al. (1993). "Manual resolution of compliance when work and force cues are minimized." 1993 ASME Winter Annual Meeting, New Orleans, LA, USA, 99-104.

[119] Tan, H., N. Durlach, et al. (1995). “Manual discrimination of compliance using active pinch grasp: The roles of force and work cues.” Perception and Psychophysics **57**: 495-510.

[120] Trinkle, J. C. and R. P. Paul (1990). “Planning for dexterous manipulation with sliding contacts.” International Journal of Robotics Research **9**(3): 24-48.

[121] Turki, L. and P. Coiffet (1995). "On grasp synthesis and planning of multifingered robot hands for a telemanipulation task." 1995 4th IEEE International Workshop on Robot and Human Communication, RO-MAN, Tokyo, Jpn.

[122] Turner, M. L., D. H. Gomez, et al. (1998). "Preliminary tests of an arm-grounded haptic feedback device in telemanipulation." 1998 ASME International Mechanical Engineering Congress and Exposition, Anaheim, CA, USA, 145-149.

[123] Turner, M. L., R. P. Findley, et al. (2000). “Development and testing of a telemanipulation system with arm and hand motion.” ASME IMECE Symposium on Haptics Interfaces.

[124] Tzafestas, C. and P. Coiffet (1997). "Computing optimal forces for generalized kinesthetic feedback on the human hand during virtual grasping and manipulation." 1997 IEEE International Conference on Robotics and Automation, ICRA. Part 1 (of 4), Albuquerque, NM, USA, 118-123.

[125] Voyles, R. M. and P. K. Khosla (1999). "Gesture-based programming: a preliminary demonstration." IEEE International Conference on Robotics and Automation, Detroit, MI, 708-713.

[126] Wampler, C. W., J. M. Hollerbach, et al. (1995). “Implicit loop method for kinematic calibration and its application to closed-chain mechanisms.” IEEE Transactions on Robotics and Automation **11**(5): 710-724.

[127] Wright, A. K. and M. M. Stanisic (1990). "Kinematic mapping between the EXOS Handmaster exoskeleton and the Utah/MIT dextrous hand." IEEE International Conference on Systems Engineering, Pittsburgh, PA, 101-104.

[128] Yoshikawa, T. and K. Nagai (1988). "Evaluation and Determination of Grasping Forces For Multi-Fingered Hands." IEEE International Conference on Robotics and Automation, Philadelphia, PA, 245-248.

[129] Yoshikawa, T. and K. Nagai (1991). “Manipulating and grasping forces in manipulation by multifingered robot hands.” IEEE Transactions on Robotics and Automation **7**(1): 67-77.

[130] Yoshikawa, T., Y. Yokokohji, et al. (1993). "Object handling by three-fingered hands using slip motion." IEEE/RSJ International Conference on Intelligent Robots and Systems, Yokohama, Jpn, 99-105.

[131] Yun, X., V. Kumar, et al. (1992). "Control of multiple arms with rolling constraints." IEEE International Conference on Robotics and Automation, Nice, Fr., 2193-2198.

Appendix A: Kinematic Hand Model Details

This appendix contains a more complete mathematical description of the human hand model developed for this research. Figure A-1 shows the model for the index and thumb, with the link lengths designated. The base frame is located at the base of the thumb at the intersection with the index metacarpal. The x-axis of the base frame points toward the metacarpal joint of the index finger, the y-axis is perpendicular to the flat palm and the z-axis is defined by the right hand rule.

The homogenous transforms are denoted such that

$$ {}^{B}\bar{\mathbf{d}} = {}^{B}_{A}\mathbf{D} \cdot {}^{A}\bar{\mathbf{d}} \qquad \textbf{(A-1)} $$

where ${}^{A}\bar{\mathbf{d}}$ is the homogenous position vector of a point with respect to frame A, ${}^{B}\bar{\mathbf{d}}$ is the homogenous position vector of the same point with respect to frame B, and ${}^{B}_{A}\mathbf{D}$ is the homogeneous transformation from frame A to frame B.

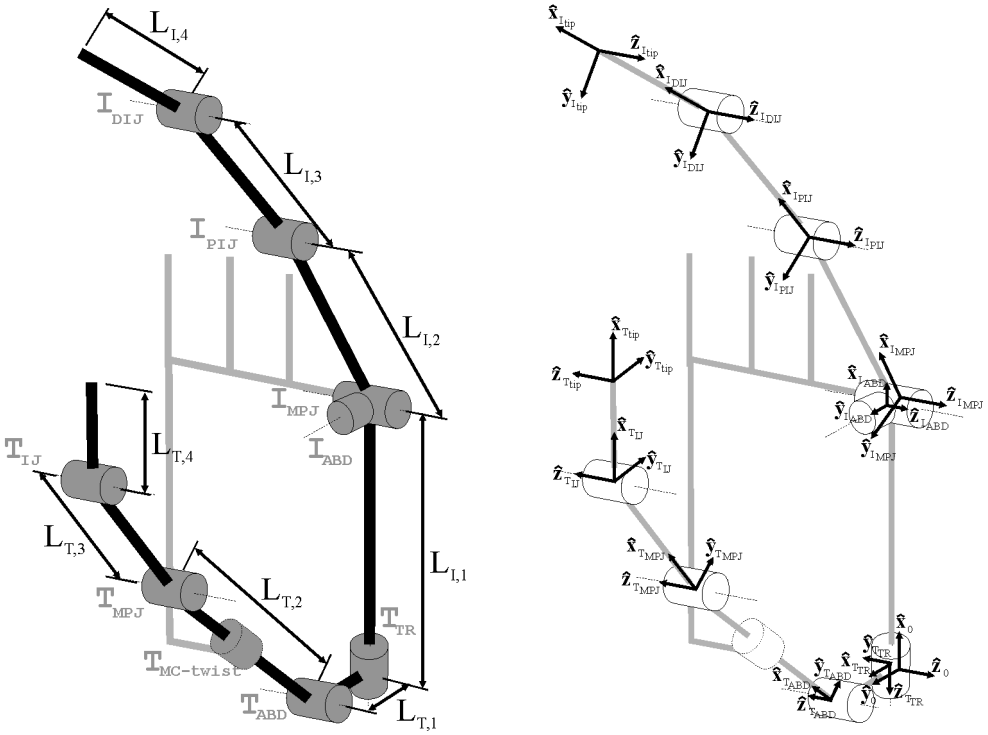

Figure A-1. Human Hand Model with Link Length and Reference Frames Defined

A.1 Thumb

A.1.1 Transformation Matrices

The T_{TR} frame of the thumb has the z-axis pointing along the axis of rotation, the x-axis points toward the T_{ABD} axis along the common normal and the y-axis is defined by the right hand rule. The homogeneous transformation from the T_{TR} frame to the base frame is defined by a change of axes and rotation of $\phi_{T_{TR}}$ about the z-axis. For $\phi_{T_{TR}} = 0$, the x-axis of the T_{TR} frame is coincident with the z-axis of the base frame.

$$ {}^{0}_{T_{TR}}\mathbf{D} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ \sin(\phi_{T_{TR}}) & \cos(\phi_{T_{TR}}) & 0 & 0 \\ \cos(\phi_{T_{TR}}) & -\sin(\phi_{T_{TR}}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(A-2)} $$

The thumb T_{ABD} frame is defined with the z-axis pointing along the abduction axis of rotation, the x-axis pointing toward the T_{MPJ} axis along the common normal (the thumb metacarpal bone) and the y-axis defined by the right hand rule. The transformation from the T_{ABD}

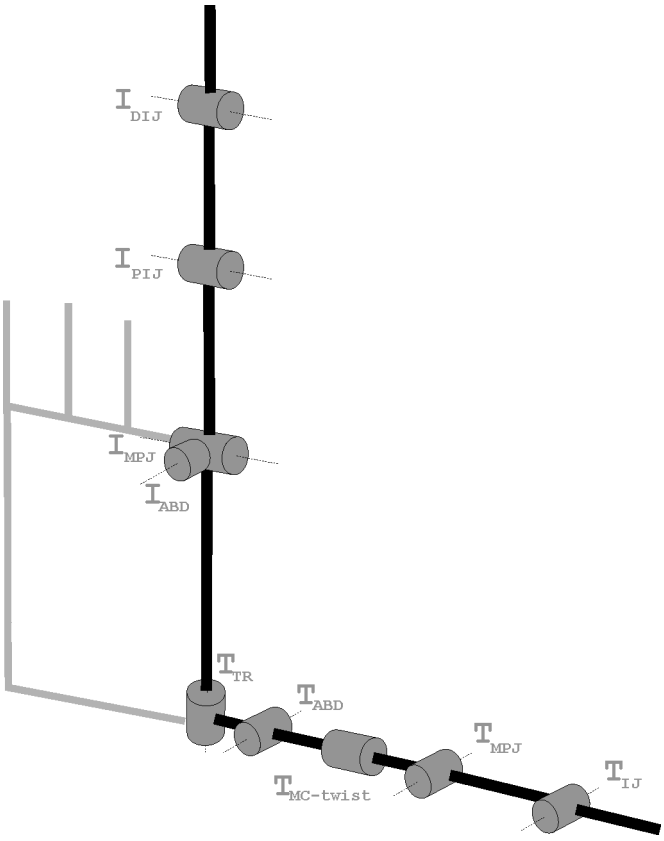

Figure A-2. Hand Model with All Joint Angles at Zero

frame to the T_{TR} frame involves a rotation $\phi_{T_{ABD}}$ about the T_{ABD} z-axis, a change of axes and a translation $L_{T,1}$ along the T_{TR} x-axis. For $\phi_{T_{ABD}} = 0$, the x-axis of the T_{ABD} frame is parallel to the x-axis of the T_{TR} frame.

$$ {}^{T_{TR}}_{T_{ABD}}\mathbf{D} = \begin{bmatrix} \cos(\phi_{T_{ABD}}) & -\sin(\phi_{T_{ABD}}) & 0 & L_{T,1} \\ 0 & 0 & 1 & 0 \\ -\sin(\phi_{T_{ABD}}) & -\cos(\phi_{T_{ABD}}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-3)} $$

The thumb T_{MPJ} frame is defined with the z-axis pointing along the T_{MPJ} axis of rotation, the x-axis pointing toward the T_{IJ} axis of rotation along the common normal (the thumb proximal phalange). The transformation from the T_{MPJ} frame to the T_{ABD} frame involves a rotation $\phi_{T_{MPJ}}$ about the T_{MPJ} z-axis, a translation $L_{T,2}$ along the T_{ABD} x-axis and a rotation $\phi_{T_{MCtwist}}$ about the T_{ABD} x-axis (the unsensed MCtwist motion). For $\phi_{T_{MPJ}} = 0$ and $\phi_{T_{MCtwist}} = 0$, the x-axis of the T_{MPJ} frame is parallel to the x-axis of the T_{ABD} frame.

$$ {}^{T_{ABD}}_{T_{MPJ}}\mathbf{D} = \begin{bmatrix} \cos(\phi_{T_{MPJ}}) & -\sin(\phi_{T_{MPJ}}) & 0 & L_{T,2} \\ \sin(\phi_{T_{MPJ}})\cos(\phi_{T_{MCtwist}}) & \cos(\phi_{T_{MPJ}})\cos(\phi_{T_{MCtwist}}) & -\sin(\phi_{T_{MCtwist}}) & 0 \\ \sin(\phi_{T_{MPJ}})\sin(\phi_{T_{MCtwist}}) & \cos(\phi_{T_{MPJ}})\sin(\phi_{T_{MCtwist}}) & \cos(\phi_{T_{MCtwist}}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-4)} $$

The T_{IJ} frame is defined with the z-axis pointing along the T_{IJ} axis of rotation, the x-axis pointing toward the finger tip and the y-axis defined by the right hand rule. The transformation from the T_{IJ} frame to the T_{MPJ} frame is a rotation $\phi_{T_{IJ}}$ about the T_{IJ} z-axis and a translation $L_{T,3}$ along the T_{MPJ} x-axis. For $\phi_{T_{IJ}} = 0$, the x-axis of the T_{IJ} frame is parallel to the x-axis of the T_{MPJ} frame.

$$ {}^{T_{MPJ}}_{T_{IJ}}\mathbf{D} = \begin{bmatrix} \cos(\phi_{T_{IJ}}) & -\sin(\phi_{T_{IJ}}) & 0 & L_{T,3} \\ \sin(\phi_{T_{IJ}}) & \cos(\phi_{T_{IJ}}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-5)} $$

The finger tip frame is defined with the same orientation as the T_{IJ} frame, located at the tip of the finger. The transformation from the tip frame to the T_{IJ} frame is a pure translation along the T_{IJ} x-axis.

$$ {}^{T_{IJ}}_{T_{tip}}\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & L_{T,4} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-6)} $$

The transformation from each frame to the base frame can be found by matrix multiplication.

$$ {}^{0}_{T_{ABD}}\mathbf{D} = {}^{0}_{T_{TR}}\mathbf{D} \cdot {}^{T_{TR}}_{T_{ABD}}\mathbf{D} \quad \textbf{(A-7)} $$

$$ {}^{0}_{T_{MPJ}}\mathbf{D} = {}^{0}_{T_{ABD}}\mathbf{D} \cdot {}^{T_{ABD}}_{T_{MPJ}}\mathbf{D} \quad \textbf{(A-8)} $$

$$ {}^{0}_{T_{IJ}}\mathbf{D} = {}^{0}_{T_{MPJ}}\mathbf{D} \cdot {}^{T_{MPJ}}_{T_{IJ}}\mathbf{D} \quad \textbf{(A-9)} $$

$$ {}^{0}_{T_{tip}}\mathbf{D} = {}^{0}_{T_{IJ}}\mathbf{D} \cdot {}^{T_{IJ}}_{T_{tip}}\mathbf{D} \quad \textbf{(A-10)} $$

A.1.2 Jacobians

The calibration routine used for the glove relies upon knowing the Jacobians for the fingers, according to the form:

$$ {}^{o}\dot{\mathbf{d}}_{T_{tip}} = \mathbf{J}_{\phi_T}\dot{\boldsymbol{\phi}}_T + \mathbf{J}_{L_T}\dot{\mathbf{L}}_T \quad \textbf{(A-11)} $$

where ${}^{o}\dot{\mathbf{d}}_{T_{tip}}$ is the linear velocity of the thumb tip in the base frame, $\mathbf{J}_{\phi_T}$ is the Jacobian correlating angular velocity of the joints $\dot{\boldsymbol{\phi}}_T$ with the tip velocity and $\mathbf{J}_{L_T}$ is the Jacobian correlating the rate of change in link (bone) lengths $\dot{\mathbf{L}}_T$ with the tip velocity (as if each bone had a prismatic joint in it).

The column of the joint angle Jacobian $\mathbf{J}_{\phi_T}$ associated with the T_{TR} joint is determined by taking the cross product of the T_{TR} axis of rotation (${}^{T_{TR}}\hat{\mathbf{z}}_{T_{TR}}$, the z-axis in the T_{TR} frame) with the (non-homogeneous) position vector of the finger tip in the T_{TR} frame ${}^{T_{TR}}\mathbf{d}_{T_{tip}}$. The resulting vector is expressed in the base frame by left multiplying by the rotational component of the T_{TR} frame to base frame transformation ${}^{0}_{T_{TR}}\mathbf{D}_{rot}$. (The rotational component is contained by rows 1 through 3 and columns 1 through 3.)

$$ \mathbf{J}_{\phi_T}(\text{column } \phi_{T_{TR}}) = {}^{0}_{T_{TR}}\mathbf{D}_{rot} \cdot ({}^{T_{TR}}\hat{\mathbf{z}}_{T_{TR}} \times {}^{T_{TR}}\mathbf{d}_{T_{tip}}) \quad \textbf{(A-12)} $$

The remaining columns of the joint angle Jacobian associated with the other joints are defined similarly.

$$\mathbf{J}_{\phi_T}(\text{column } \phi_{T_{ABD}}) = {}^{0}_{T_{ABD}}\mathbf{D}_{rot} \cdot ({}^{T_{ABD}}\hat{\mathbf{z}}_{T_{ABD}} \times {}^{T_{ABD}}\mathbf{d}_{T_{tip}}) \quad \textbf{(A-13)}$$

$$\mathbf{J}_{\phi_T}(\text{column } \phi_{T_{MCtwist}}) = {}^{0}_{T_{ABD}}\mathbf{D}_{rot} \cdot ({}^{T_{ABD}}\hat{\mathbf{x}}_{T_{ABD}} \times {}^{T_{ABD}}\mathbf{d}_{T_{tip}}) \quad \textbf{(A-14)}$$

$$\mathbf{J}_{\phi_T}(\text{column } \phi_{T_{MPJ}}) = {}^{0}_{T_{MPJ}}\mathbf{D}_{rot} \cdot ({}^{T_{MPJ}}\hat{\mathbf{z}}_{T_{MPJ}} \times {}^{T_{MPJ}}\mathbf{d}_{T_{tip}}) \quad \textbf{(A-15)}$$

$$\mathbf{J}_{\phi_T}(\text{column } \phi_{T_{IJ}}) = {}^{0}_{T_{IJ}}\mathbf{D}_{rot} \cdot ({}^{T_{IJ}}\hat{\mathbf{z}}_{T_{IJ}} \times {}^{T_{IJ}}\mathbf{d}_{T_{tip}}) \quad \textbf{(A-16)}$$

The column of the link length Jacobian $\mathbf{J}_{L_T}$ associated with the length $L_{T,1}$ is the unit vector describing the orientation of the link in the base frame. (A change in the link length of a single kinematic chain will necessarily change the tip position in the direction the link is pointing.) The link points along the x-axis of the T_{TR} frame. The unit vector orientation of the T_{TR} x-axis ${}^{0}\hat{\mathbf{x}}_{T_{TR}}$ is the left column of the rotational component of the T_{TR} to base transformation ${}^{0}_{T_{TR}}\mathbf{D}_{rot}$.

$$\mathbf{J}_{L_T}(\text{column } L_{T,1}) = {}^{0}_{T_{TR}}\mathbf{D}_{rot}(\text{column } 1) \quad \textbf{(A-17)}$$

The columns of the link length Jacobian $\mathbf{J}_{L_T}$ associated with the other links are defined similarly.

$$\mathbf{J}_{L_T}(\text{column } L_{T,2}) = {}^{0}_{T_{ABD}}\mathbf{D}_{rot}(\text{column } 1) \quad \textbf{(A-18)}$$

$$\mathbf{J}_{L_T}(\text{column } L_{T,3}) = {}^{0}_{T_{MPJ}}\mathbf{D}_{rot}(\text{column } 1) \quad \textbf{(A-19)}$$

$$\mathbf{J}_{L_T}(\text{column } L_{T,4}) = {}^{0}_{T_{IJ}}\mathbf{D}_{rot}(\text{column } 1) \quad \textbf{(A-20)}$$

1.2 Index Finger

A.2.1 Transformation Matrices

The I_{ABD} frame is defined with the y-axis pointing along the axis of rotation, the x-axis is parallel to the base frame x-axis for $\phi_{I_{ABD}} = 0$ and the z-axis according to the right hand rule. The transformation from the I_{ABD} frame to the base frame is a rotation $\phi_{I_{ABD}}$ about the y-axis in the I_{ABD} frame and a translation $L_{I,1}$ along the x-axis of the base frame.

$${}^{0}_{I_{ABD}}\mathbf{D} = \begin{bmatrix} \cos(\phi_{I_{ABD}}) & 0 & \sin(\phi_{I_{ABD}})) & L_{I,1} \\ 0 & 1 & 0 & 0 \\ -\sin(\phi_{I_{ABD}}) & 0 & \cos(\phi_{I_{ABD}}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-21)}$$

The I_{MPJ} frame is defined with the z-axis pointing along the axis of rotation, the x-axis pointing toward the I_{PIJ} axis of rotation along the common normal and the y-axis defined by the right hand rule. The transformation from the I_{MPJ} frame to the I_{ABD} frame is a rotation $\phi_{I_{MPJ}}$ about the IMPJ z-axis. For $\phi_{I_{MPJ}} = 0$ the I_{MPJ} frame and I_{ABD} frame are identical.

$$ {}^{I_{ABD}}_{I_{MPJ}}\mathbf{D} = \begin{bmatrix} \cos(\phi_{I_{MPJ}}) & -\sin(\phi_{I_{MPJ}}) & 0 & 0 \\ \sin(\phi_{I_{MPJ}}) & \cos(\phi_{I_{MPJ}}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-22)} $$

The I_{PIJ} frame is defined with the z-axis pointing along the axis of rotation, the x-axis pointing toward the I_{DIJ} axis of rotation along the common normal and the y-axis defined by the right hand rule. The transformation from the I_{PIJ} frame to the I_{MPJ} frame is a rotation $\phi_{I_{PIJ}}$ about the I_{PIJ} z-axis and a translation $L_{I,2}$ along the I_{MPJ} x-axis. For $\phi_{I_{PIJ}} = 0$ the I_{PIJ} frame is oriented parallel to the I_{MPJ} frame.

$$ {}^{I_{MPJ}}_{I_{PIJ}}\mathbf{D} = \begin{bmatrix} \cos(\phi_{I_{PIJ}}) & -\sin(\phi_{I_{PIJ}}) & 0 & L_{I,2} \\ \sin(\phi_{I_{PIJ}}) & \cos(\phi_{I_{PIJ}}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-23)} $$

The I_{DIJ} frame is defined with the z-axis pointing along the axis of rotation, the x-axis pointing toward the index finger tip and the y-axis defined by the right hand rule. The transformation from the I_{DIJ} frame to the I_{PIJ} frame is a rotation $\phi_{I_{DIJ}}$ about the z-axis of the I_{DIJ} frame and a translation $L_{I,3}$ along the x-axis of the I_{PIJ} frame. For $\phi_{I_{DIJ}} = 0$ the orientation of the I_{DIJ} frame is parallel to the I_{PIJ} frame.

$$ {}^{I_{PIJ}}_{I_{DIJ}}\mathbf{D} = \begin{bmatrix} \cos(\phi_{I_{DIJ}}) & -\sin(\phi_{I_{DIJ}}) & 0 & L_{I,3} \\ \sin(\phi_{I_{DIJ}}) & \cos(\phi_{I_{DIJ}}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-24)} $$

The index tip frame is defined with the same orientation as the I_{DIJ} frame located at the finger tip. The transformation from the I_{tip} frame to the I_{DIJ} frame is a translation $L_{I,4}$ along the I_{DIJ} x-axis.

$$ {}^{I_{DIJ}}_{I_{tip}}\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & L_{I,4} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(A-25)} $$

The transformation from each of the frames to the base frame is defined by matrix multiplication.

$$ {}^{0}_{I_{MPJ}}\mathbf{D} = {}^{0}_{I_{ABD}}\mathbf{D} \cdot {}^{I_{ABD}}_{I_{MPJ}}\mathbf{D} \quad \textbf{(A-26)} $$

$$ {}^{0}_{I_{PIJ}}\mathbf{D} = {}^{0}_{I_{MPJ}}\mathbf{D} \cdot {}^{I_{MPJ}}_{I_{PIJ}}\mathbf{D} \quad \textbf{(A-27)} $$

$$ {}^{0}_{I_{DIJ}}\mathbf{D} = {}^{0}_{I_{PIJ}}\mathbf{D} \cdot {}^{I_{PIJ}}_{I_{DIJ}}\mathbf{D} \quad \textbf{(A-28)} $$

$$ {}^{0}_{I_{tip}}\mathbf{D} = {}^{0}_{I_{DIJ}}\mathbf{D} \cdot {}^{I_{DIJ}}_{I_{tip}}\mathbf{D} \quad \textbf{(A-29)} $$

A.2.2 Jacobian

As with the thumb, we develop Jacobians for the index finger, according to the form:

$$ {}^{o}\dot{\mathbf{d}}_{I_{tip}} = \mathbf{J}_{\phi_I}\dot{\boldsymbol{\phi}}_I + \mathbf{J}_{L_I}\dot{\mathbf{L}}_I \quad \textbf{(A-30)} $$

where ${}^{o}\dot{\mathbf{d}}_{I_{tip}}$ is the linear velocity of the index tip in the base frame, $\mathbf{J}_{\phi_I}$ is the Jacobian correlating angular velocity of the joints $\dot{\boldsymbol{\phi}}_I$ with the tip velocity and $\mathbf{J}_{L_I}$ is the Jacobian correlating the rate of change in link (bone) lengths $\dot{\mathbf{L}}_I$ with the tip velocity (as if each bone had a prismatic joint in it).

The column of the joint angle Jacobian $\mathbf{J}_{\phi_I}$ associated with the I_{ABD} joint is determined by taking the cross product of the I_{ABD} axis of rotation (${}^{I_{ABD}}\hat{\mathbf{y}}_{I_{ABD}}$, the y-axis in the I_{ABD} frame) with the (non-homogenous) position of the finger tip in the I_{ABD} frame ${}^{I_{ABD}}\mathbf{d}_{I_{tip}}$. The resulting vector is expressed in the base frame by left multiplying by the rotational component of the I_{ABD} frame to base frame transformation ${}^{0}_{I_{ABD}}\mathbf{D}_{rot}$. (The rotational component is contained by rows 1 through 3 and columns 1 through 3.)

$$ \mathbf{J}_{\phi_I}(\text{column } \phi_{I_{ABD}}) = {}^{0}_{I_{ABD}}\mathbf{D}_{rot} \cdot ({}^{I_{ABD}}\hat{\mathbf{y}}_{I_{ABD}} \times {}^{I_{ABD}}\mathbf{d}_{I_{tip}}) \quad \textbf{(A-31)} $$

The columns of the joint angle Jacobian $\mathbf{J}_{\phi}$ associated with the remaining joint angles are defined similarly.

$$ \mathbf{J}_{\phi_I}(\text{column } \phi_{I_{MPJ}}) = {}^{0}_{I_{MPJ}}\mathbf{D}_{rot} \cdot ({}^{I_{MPJ}}\hat{\mathbf{z}}_{I_{MPJ}} \times {}^{I_{MPJ}}\mathbf{d}_{I_{tip}}) \quad \textbf{(A-32)} $$

$$\mathbf{J}_{\phi_I}(\text{column } \phi_{I_{PIJ}}) = {}^{0}_{I_{PIJ}}\mathbf{D}_{rot} \cdot ({}^{I_{PIJ}}\hat{\mathbf{z}}_{I_{PIJ}} \times {}^{I_{PIJ}}\mathbf{d}_{I_{tip}}) \quad \textbf{(A-33)}$$

$$\mathbf{J}_{\phi_I}(\text{column } \phi_{I_{DIJ}}) = {}^{0}_{I_{DIJ}}\mathbf{D}_{rot} \cdot ({}^{I_{DIJ}}\hat{\mathbf{z}}_{I_{DIJ}} \times {}^{I_{DIJ}}\mathbf{d}_{I_{tip}}) \quad \textbf{(A-34)}$$

The column of the link length Jacobian $\mathbf{J}_{L_I}$ associated with $L_{I,1}$ (nominally the index metacarpal bone) is the unit vector of the orientation of the link in the base frame. Since the metacarpal does not more with respect to the base frame, by definition it is the unit x-axis of the base frame.

$$\mathbf{J}_{L_I}(\text{column } L_{I,1}) = {}^{0}\hat{\mathbf{x}}_0 \quad \textbf{(A-35)}$$

The column of the link length Jacobian $\mathbf{J}_L$ associated with $L_{I,2}$ (nominally the index proximal phalange) is the unit vector orientation of the link. The link is parallel to the x-axis of the I_{MPJ} frame, and the unit vector orientation of the I_{MPJ} x-axis in the base frame ${}^{0}\hat{x}_{I_{MPJ}}$ is the first column of the rotational component of the transformation from the I_{MPJ} frame to the base frame ${}^{0}_{I_{MPJ}}\mathbf{D}_{rot}$.

$$\mathbf{J}_{L_I}(\text{column } L_{I,2}) = {}^{0}_{I_{MPJ}}\mathbf{D}_{rot}(\text{column } 1) \quad \textbf{(A-36)}$$

The columns of $\mathbf{J}_{L_I}$ associated with the remaining link lengths are defined similarly.

$$\mathbf{J}_{L_I}(\text{column } L_{I,3}) = {}^{0}_{I_{PIJ}}\mathbf{D}_{rot}(\text{column } 1) \quad \textbf{(A-37)}$$

$$\mathbf{J}_{L_I}(\text{column } L_{I,4}) = {}^{0}_{I_{DIJ}}\mathbf{D}_{rot}(\text{column } 1) \quad \textbf{(A-38)}$$

Appendix B: Hand Calibration Details

This appendix will expand the particulars of the calibration procedure described in Chapter 4.

The human hand model is described by a vector of parameters $\mathbf{p}$, where

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_T \\ \mathbf{p}_I \\ \mathbf{p}_M \\ \mathbf{p}_R \end{bmatrix} \quad \textbf{(B-1)}$$

and $\mathbf{p}_T$ is a vector of the parameters associated with the thumb, $\mathbf{p}_I$ is a vector of parameters associated with the index finger, $\mathbf{p}_M$ is associated with the middle finger and $\mathbf{p}_R$ is associated with the ring finger.

The parameters associated with the thumb are

(B-2)

$$\mathbf{p}_T = \left[\theta_{T_{TR}}\ \theta_{T_{MPJ}}\ \theta_{T_{IJ}}\ \theta_{T_{ABD}}\ \theta_{T_{MCtw}}\ g_{T_{TR}}\ g_{T_{MPJ}}\ g_{T_{IJ}}\ g_{T_{ABD}}\ L_{T,1}\ L_{T,2}\ L_{T,3}\ L_{T,4}\ g_{T_{TR}}^{T_{ABD}}\ g_{T_{ABD}}^{T_{TR}}\ g_{T_{MCtw}}^{T_{TR}}\ g_{T_{MCtw}}^{T_{ABD}}\right]^T$$

recalling that for joint angle ϕ_i, θ_i is the angular offset, g_i is the gain and g_i^j is the cross gain from sensor j. For example, the thumb abduction joint angle $\phi_{T_{ABD}}$ is

$$\phi_{T_{ABD}} = \theta_{T_{ABD}} + g_{T_{ABD}} \cdot \sigma_{T_{ABD}} + g_{T_{ABD}}^{T_{TR}} \cdot \theta_{T_{TR}} \quad \textbf{(B-3)}$$

where σ are the raw sensor values. Note that there is no $g_{T_{MCtw}}$, since the MCtwist joint is an unsensed axis. L_i are bone lengths.

The parameters for the remaining fingers are similar, though there are currently no cross gains.

$$\mathbf{p}_I = \left[\theta_{I_{MPJ}}\ \theta_{I_{PIJ}}\ \theta_{I_{DIJ}}\ \theta_{I_{ABD}}\ g_{I_{MPJ}}\ g_{I_{PIJ}}\ g_{I_{DIJ}}\ g_{I_{ABD}}\ L_{I,1}\ L_{I,2}\ L_{I,3}\ L_{I,4}\right] \quad \textbf{(B-4)}$$

$$\mathbf{p}_M = \left[\theta_{M_{MPJ}}\ \theta_{M_{PIJ}}\ \theta_{M_{DIJ}}\ \theta_{M_{ABD}}\ g_{M_{MPJ}}\ g_{M_{PIJ}}\ g_{M_{DIJ}}\ g_{M_{ABD}}\ L_{M,1}\ L_{M,2}\ L_{M,3}\ L_{M,4}\right] \quad \textbf{(B-5)}$$

$$\mathbf{p}_R = \left[\theta_{R_{MPJ}}\ \theta_{R_{PIJ}}\ \theta_{R_{DIJ}}\ \theta_{R_{ABD}}\ g_{R_{MPJ}}\ g_{R_{PIJ}}\ g_{R_{DIJ}}\ g_{R_{ABD}}\ L_{R,1}\ L_{R,2}\ L_{R,3}\ L_{R,4}\right] \quad \textbf{(B-6)}$$

The calibration process begins with a default vector of $\mathbf{p}_0$. This default vector has values which make a graphical display of the hand model look "good" for a typical user. Essentially, this means that the relative bone lengths match biological data and the gains and offsets are such that the joint angles are nominally between 0° and 90°. These values could be modified to improve the appearance for a particular user, but the values would not be sufficient for the purpose of fine finger manipulation. For our particular CyberGlove, the default parameters are set at:

(B-7)

$$\mathbf{p}_{T_0} = \left[0.0\ -0.2\ -0.4\ -0.4\ 0.52\ 1.0\ 1.0\ 0.7\ 1.0\ 10.0\ 40.2\ 32.1\ 28.1\ 0.0\ 0.0\ 0.0\ 0.0\right]^T$$

$$\mathbf{p}_{I_0} = \left[0.0\ 0.0\ 0.0\ 0.1\ 0.8\ 1.0\ 0.8\ 1.0\ 88.0\ 43.5\ 24.7\ 19.7\right]^T \quad \textbf{(B-8)}$$

$$\mathbf{p}_{M_0} = \left[0.0\ 0.0\ 0.0\ 0.1\ 0.8\ 1.0\ 0.8\ 1.0\ 88.0\ 45.1\ 29.7\ 26.4\right]^T \quad \textbf{(B-9)}$$

$$\mathbf{p}_{R_0} = \left[0.0\ 0.0\ 0.0\ 0.1\ 0.8\ 1.0\ 0.8\ 1.0\ 88.0\ 43.5\ 24.7\ 19.7\right]^T \quad \textbf{(B-10)}$$

where offsets are in radians, gains are in radians/volt and link lengths are in millimeters.

As described in Chapter 4, the user is asked to form a closed chain between their thumb and one finger. Sensor readings are recorded as the fingers move while keeping the fingers together. This is performed for each of the index, middle and ring fingers with the thumb. For each finger, 80 data points are taken.

The tip positions of the finger and the thumb are calculated according to the forward kinematics in Appendix A using the current estimate of the parameters. For a perfect match of the model, the two positions would be identical. The error in the model for data point n, $\Delta\mathbf{d}_n$, is defined as the difference between the thumb tip position $\mathbf{d}_{T_n}$ and the finger position $\mathbf{d}_{f_n}$.

$$\Delta\mathbf{d}_n = \mathbf{d}_{T_n} - \mathbf{d}_{f_n} \quad \textbf{(B-11)}$$

The Jacobian relating the thumb parameters to the error vector is

(B-12)

$$\Delta \mathbf{d}_n = \left[J_{\phi_T} \; J_{\phi_T} \text{:} \text{Diag}(\boldsymbol{\sigma}_{n_T}) \; J_{L_T} \left[J_{\phi_{T_{ABD}}} \cdot \sigma_{T_{TR}} \; J_{\phi_{T_{TR}}} \cdot \sigma_{T_{ABD}} \; J_{\phi_{T_{MCtwist}}} \cdot \sigma_{T_{TR}} \; J_{\phi_{T_{MCtwist}}} \cdot \sigma_{T_{ABD}} \right] \right] \cdot \Delta \mathbf{p}_T$$

where the angular J_{ϕ_T} and length J_{L_T} Jacobians are as defined in Appendix A.

The Jacobian relating the change in the parallel finger f parameters (where f can be I, M or R for the index, middle and ring respectively) to the error in tip position is

$$\Delta \mathbf{d}_n = -\left[J_{\phi_f} \; J_{\phi_f} \cdot \text{Diag}(\boldsymbol{\sigma}_{n_f}) \; J_{L_f} \right] \cdot \Delta \mathbf{p}_f \qquad \textbf{(B-13)}$$

where the angular J_{ϕ_f} and length J_{L_f} Jacobians are as defined in Appendix A. The negative sign reflects the fact that the index position is subtracted from the thumb position to get $\Delta \mathbf{d}_n$. We condense this notation to read

$$\Delta \mathbf{d}_n = \hat{\mathbf{J}}_T \cdot \Delta \mathbf{p}_T - \hat{\mathbf{J}}_f \cdot \Delta \mathbf{p}_f \qquad \textbf{(B-14)}$$

Just using the basic Jacobian equation to iterate to a solution led to trivial or biologically inappropriate solutions. To constrain the parameters to an acceptable range, the Jacobians are augmented by a potential well function. The equation for constraining the parameters for a finger f is:

$$\mathbf{V}_f \cdot (\mathbf{p}_{0_f} - \mathbf{p}_f) = \mathbf{V}_f \cdot \Delta \mathbf{p}_f \qquad \textbf{(B-15)}$$

where $\mathbf{V}_f$ can be any non-singular matrix. Iterating on this equation independent of the Jacobian has the effect of driving $\mathbf{p}_f$ toward $\mathbf{p}_{0_f}$. The rate of convergence can be adjusted by setting $\mathbf{V}_f$ appropriately. In our implementation, we use a diagonal matrix with a tenth order polynomial on the diagonals.

$$\mathbf{V}_f = \text{Diag}\left(-\sqrt{N} \frac{\partial}{\partial \mathbf{p}_f} \left[\left(\frac{\mathbf{p}_{0_f} - \mathbf{p}_f}{\boldsymbol{\rho}_f} \right)^{10} \right] \right) \qquad \textbf{(B-16)}$$

where N is the number of points in the data collection (so that the proportional effect of the deviation well is independent of the number of points), and $\boldsymbol{\rho}_f$ is a vector of acceptable radii of deviation for the parameters $\mathbf{p}_f$.

The tenth order polynomial is almost flat for the range of $\pm\boldsymbol{\rho}_f$, but grows very quickly outside the range. When the potential well is used with the Jacobians in Eq. B-12 and Eq. B-

13, a difference between a parameter p, and its default value p_o, of $0.9 \cdot \rho$ has a corrective $\Delta\mathbf{p}$ equivalent to a positional error of 0.34mm, but if the difference is $1.5 \cdot \rho$, the corrective $\Delta\mathbf{p}$ is equivalent to a positional error of 57.0mm.

The values of ρ_f used in our implementation are:

$$\rho_T = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 0.2 & 0.2 & 0.2 & 0.2 & 7.0 & 5.0 & 4.0 & 4.0 & 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix} \quad \textbf{(B-17)}$$

$$\rho_I = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 8.0 & 6.0 & 4.0 & 4.0 \end{bmatrix} \quad \textbf{(B-18)}$$

$$\rho_M = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 8.0 & 6.0 & 4.0 & 4.0 \end{bmatrix} \quad \textbf{(B-19)}$$

$$\rho_R = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 8.0 & 6.0 & 4.0 & 4.0 \end{bmatrix} \quad \textbf{(B-20)}$$

When the potential well matrices are combined with the Jacobian equations for four fingered calibration, the resulting equation is:

$$\begin{bmatrix} \Delta\mathbf{d}_1 \\ \Delta\mathbf{d}_2 \\ \Delta\mathbf{d}_3 \\ \mathbf{V}_T \cdot (\mathbf{p}_{0_T} - \mathbf{p}_T) \\ \mathbf{V}_I \cdot (\mathbf{p}_{0_I} - \mathbf{p}_I) \\ \mathbf{V}_M \cdot (\mathbf{p}_{0_M} - \mathbf{p}_M) \\ \mathbf{V}_R \cdot (\mathbf{p}_{0_R} - \mathbf{p}_R) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{J}}_T & -\hat{\mathbf{J}}_I & 0 & 0 \\ \hat{\mathbf{J}}_T & 0 & -\hat{\mathbf{J}}_M & 0 \\ \hat{\mathbf{J}}_T & 0 & 0 & -\hat{\mathbf{J}}_R \\ \mathbf{V}_T & 0 & 0 & 0 \\ 0 & \mathbf{V}_I & 0 & 0 \\ 0 & 0 & \mathbf{V}_M & 0 \\ 0 & 0 & 0 & \mathbf{V}_R \end{bmatrix} \begin{bmatrix} \Delta\mathbf{p}_T \\ \Delta\mathbf{p}_I \\ \Delta\mathbf{p}_M \\ \Delta\mathbf{p}_R \end{bmatrix} \quad \textbf{(B-21)}$$

or more succinctly

$$\mathbf{e} = \hat{J} \cdot \Delta\mathbf{p} \quad \textbf{(B-22)}$$

However, when iterating on Eq. B-22, there is an issue due to the different scale of the units for the parameters. A unit change in an angular offset (± 1 radian) will have a much greater effect on the tip position than a unit change in a link length (± 1 mm). For this reason, the minimization procedure preferentially changes some parameters instead of others,

To normalize the parameter units, we add a minimization weighted for a parameter's effect on the tip position. The minimization equation is

$$\min\left(\frac{1}{2} \cdot \Delta\mathbf{p}^T \cdot \mathbf{W} \cdot \Delta\mathbf{p}\right) \quad \textbf{(B-23)}$$

Where **W** is a diagonal matrix such the diagonal elements are $W_{ii} = \left(\frac{1}{w_i}\right)^2$, where w_i is the maximum effect a unit change in parameter p_i will have upon the tip position.

$$w_i = \max(\|J_{p_i}\|) \quad \textbf{(B-24)}$$

where J_{p_i} is the column of the Jacobian associated with parameter p_i.

Recall that the least squares solution to Eq. B-22 can be written as

$$\min((\mathbf{e} - \hat{J} \cdot \Delta\mathbf{p})^T \cdot (\mathbf{e} - \hat{J} \cdot \Delta\mathbf{p})) \quad \textbf{(B-25)}$$

Adding Eq. B-24 and Eq. B-25, we solve for the minimum of the two equations together. The resulting equation for the change in $\Delta\mathbf{p}$ is

$$\Delta\mathbf{p} = (\hat{J}^T \cdot \hat{J} - \mathbf{W}^T)^{-1} \cdot \hat{J}^T \cdot \mathbf{e} \quad \textbf{(B-26)}$$

Iterating with Eq. B-26 produces reliable convergence to the desired solution. This equation was tested with a "perfect" data set $\boldsymbol{\sigma}_{1\ldots N}$ with a solution where $\Delta\mathbf{d} = 0$ at $\mathbf{p} = \mathbf{p}_0$ to verify that it will indeed converge to $\mathbf{p}_0$.