Shared autonomy in a robot hand teleoperation system

Paul Michelman
michelma@cs.columbia.edu

Peter Allen
allen@cs.columbia.edu

Center for Research in Intelligent Systems, Columbia University
New York, New York 10027

Abstract

This paper considers adding autonomy to robot hands used in teleoperation systems. Currently, the finger positions of robot hands in teleoperation systems are controlled via a robot master using a Dataglove or exoskeleton. There are several difficulties with this approach: accurate calibration is hard to achieve; robot hands have different capabilities from human hands; and complex force reflection is difficult. In this paper, we propose a model of hand teleoperation in which the input device commands the motions of a grasped object rather than the joint displacements of the fingers. To achieve this goal, the hand requires greater autonomy and the capability to perform high-level functions with minimal external input. Therefore, a set of general, primitive manipulation functions that can be performed automatically is defined. These elementary functions control simple rotations and translations of the grasped objects. They are incorporated into a teleoperation system by using a simple input device as a control signal. Preliminary implementations with a Utah/MIT are discussed.

1 Introduction

This paper considers the use of robot hands in teleoperation systems. Teleoperation has been cited as a means of controlling robot hands in industry and for prosthetics [7, 6]. The traditional means of teleoperating a robot hand has been using a Dataglove or exoskeleton master: there is a direct mapping from the human hand to the robot hand. Generally, the finger positions of the human master are translated to the robot and visual or force feedback are returned from the robot to the master. (See Speeter *et al.* [21, 18], Burdea *et al.* [4], Hong and Tan [8]). There are several difficulties with this approach:

1. Calibration: It is difficult to find a direct mapping from the human hand master to the robot. For example, Hong and Tan [8] developed a complex three-step process each user must repeat.

2. The capabilities of robot hands are different from those of human hands. For example, the human hand can translate objects along only a single axis while a hand such as the Utah/MIT [10] hand has the ability to translate objects in three Cartesian directions. Using a Dataglove thus reduces the manipulatory capabilities of the robot hand.

3. Controlling the many degrees of freedom of these hands (e.g., 16 degrees of freedom for the Utah/MIT hand) requires high-bandwidth communication.

4. Autonomy: Traditionally, robot commands are displacements rather than functions. Without a high-level function, it is difficult to enhance a robot's autonomy. Furthermore, in situations in which there are long communication delays between the master and the robot (greater than 1 second), it is useful for the robot to perform certain functions autonomously (Bejczy and Kim [2]).

5. High degree-of-freedom force feedback is still experimental and expensive. Burdea *et al.* [4], however, have developed a hand master that returns the grasping forces in three directions to the user.

The traditional means of control is "manual control," in which the interface between the robot and the user primarily transfers data and performs coordinate transformations. We propose to increase the autonomy of the robot hand by shifting the control space from the joint positions to the space of the grasped object. In Sheridan's [20] hierarchy of control modes, this is termed "supervisory control": the control of robot motion is shared between the operator and the robot. Rather than translate the motions of the master's fingers directly to motions of the robot hand, a simple, low degree-of-freedom input device, such as a joystick, controls the motions of the *manipulated object* directly. The motions of the manipulated object normally involve a single degree of freedom; they are two-dimensional translations and rotations. The system in turn computes the required finger trajectories to achieve the motion. The use of a low degree-of-freedom input device is appealing because it alleviates the problems cited above related to difficulty of calibration, the high input and sensor bandwidths required for full telemanipulation. In addition, increasing the autonomy of the robot allows it to perform tasks such as maintaining grasp forces and resisting external disturbances automatically. Low DOF input devices can include voice recognition systems, trackballs, Spaceballs[1], myoelectric signals, and other devices used in industry and rehabilitation.[2] Figure 1 illustrates the main system components as applied to a screwdriving task. The important point to note is that the input device controls the position of the screwdriver rather than the joint angles of the fingers. The feedback to the operator is also in the object space in the form of actual position of the screwdriver and the resistance torque applied to the screwdriver by the environment.

Related work is described by Farry *et al.* [7] at Rice University, who have proposed using myoelectric control for a complex robot hand. Electrodes collect myoelectric signals from a human subject. The signals are low-pass filtered and the frequency spectra analyzed. The signal processing shows the potential of disambiguating between several different grasps–the chuck and key grasps by noting individual signatures in the spectra of the grasps. Once the desired grasp is understood, the Utah/MIT hand is to be programmed to produce it. Here again there is a mapping from a low-bandwidth input signal—one of two potential hand grasps—to a complex hand function.

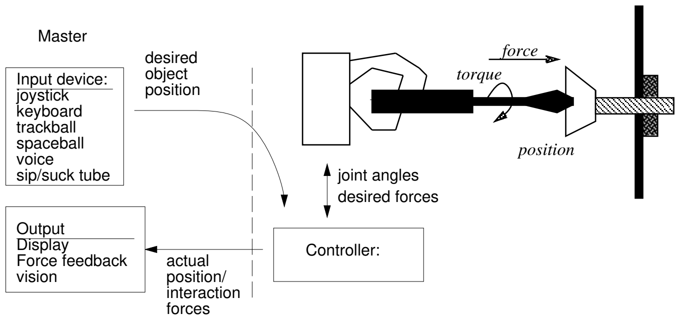

Figure 1: Components of shared autonomy system

Our work deals with *manipulation* of grasped objects rather than pure grasping. First, while the hand is still at a distance from the object to be grasped, the hand is preshaped. Second, the input device—a Spaceball, for example—is used to position the arm. Third, the hand grasps the object. The grasping function is controlled by the input device. Ideally, it should be possible to control the arm position *while* adjusting the grasp. This could be done using a number of position and orientation sensing devices, such as a Polhemus sensor. Furthermore, it is useful to feed back finger force information from the fingers to adjust the grasp strength. Finally, the object is manipulated using the input device. In the case of single-axis manipulations, such as translation in the x-direction, the signal from the input device is filtered to ignore off-axis commands. Notice that in each phase of the manipulation, the input device serves a different control function.

The rest of this paper presents an approach to solving the problem of mapping a low degree-of-freedom input device to a complex robot with the Utah/MIT hand as an example. An important part is finding hand primitives for the particular manipulation tasks. Once these functions are identified, the proposed idea of shared autonomy becomes more tractable. Section 2 describes a set of elementary manipulation functions for the Utah/MIT hand. Each robot hand has its own set of functions based on the manipulatory capabilities of the hand; that is, based on its kinematics, its workspace, its sensory capabilities, and so on. Section 3 discusses the different elements in a complete task and experimental work at Columbia University with the Utah/MIT hand.

[1] The Spaceball is a multi-function input device that senses forces and moments applied to it. It also as an array of software programmable buttons that can be assigned functions during a task.

[2] See Webster *et al.* [22] for a discussion of the range of input devices used for people with disabilities.

2 Elementary functions

The complexity of robot hands makes direct telemanipulation difficult. A typical hand has three or four fingers, each with up to four joints. Sensing a hand master's positions accurately and relaying them to a robot hand is a complex and error-prone task. In addition, the pure relaying of joint displacements to the hand is a low-level action with no sense of functionality. The hand cannot perform autonomously because its higher-level goals are defined. Low-level tasks such as maintaining grasp stability and resisting external disturbances cannot be built into a teleoperated hand that relies solely upon manual control.

Our approach to enhancing the hand's functionality has been to design a set of basic functions that are a toolkit for manipulation.[3] Complex tasks are composed of these simpler building blocks (or "units of action," as motor control researchers call them [11]). These manipulations (described in Michelman [15]), which the hand performs autonomously, are translations and rotations of objects with circular and rectangular cross sections. It is assumed that all finger contacts are on the fingertips and that motions are performed slowly enough to allow quasistatic analysis of forces to be used. To incorporate them into a teleoperation system, the control input is transferred from within the controller to the input device.

The elementary set of functions includes the cooperative finger motions required to translate and rotate objects in a desired way. There seem to be an endless number of ways to manipulate objects. By isolating the basic strategies for manipulation, it is possible develop a set of functions that are parametrized for a wide range of tasks. For example, the same motions used to turn the top of a jar may be used to turn a screwdriver. The similarities between these two tasks are that both rotate a cylindrical object about its axis and both require estimating and compensating for torques resisting rotation. What are the differences between these two primitives? Among them are: (1) the sizes of the grasped objects, (2) the amount of torque to exert, (3) the directions of the exerted forces (with the screwdriver, it is necessary to exert a force along screwdriver shaft, for example), (4) the amount of time required to perform the tasks, and (5) possibly the direction of rotation. These parameters can be intuited during the task by sensing the finger contact points and inferred from the input device.

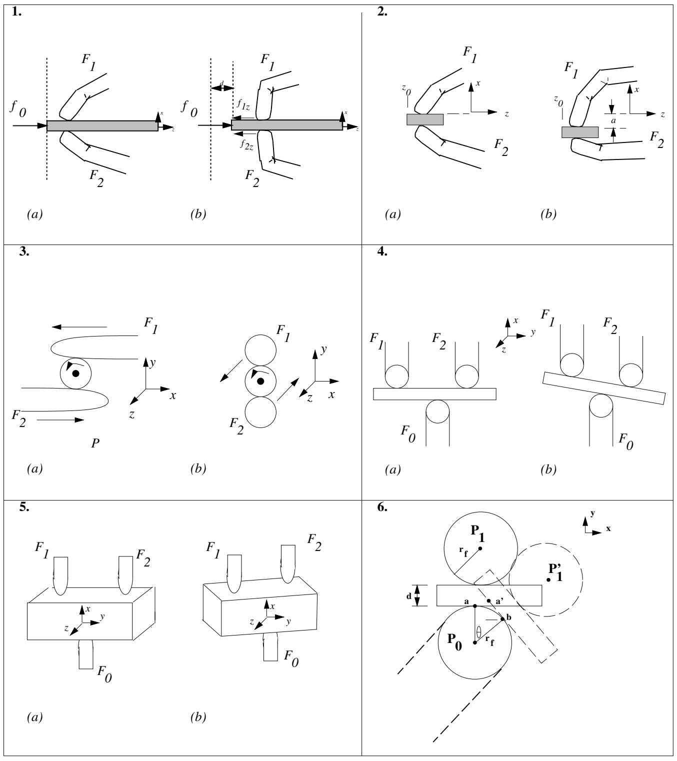

Figure 2: The basic manipulations: (1)-(2) translations toward and parallel to palm; (3) Log-rolling; (4) Twiddle; (5) Pivot; (6) 'Y'-roll

Space does not permit the elaboration of the complete set of primitive functions defined for the Utah/MIT hand. The set includes three basic translations and three rotation strategies. Figure 2 illustrates the manipulations. The translational strategies are similar. Figure 2(1)-(2) show the motion of a rectangular object toward and par-

[3]Our work takes its inspiration from human manipulation studies. There seems to be a limited number of the *types* of manipulations that a person can perform. Napier [17] has written of human hands: "Considering the enormous variety of activities that the hand is called upon to perform, it might be supposed that prehensile movements would be too numerous for simple analysis. However the diversity of movements is more apparent than real; it is not so much that there is a profusion of *actions* concerned in day-to-day activities as that there is a multiplicity of objects involved—switches, doorknobs, latches, cutlery, cups, glasses, pens, pencils, erasers, buttons and coins. In fact, there are only two main patterns and two subsidiary patterns (p. 75)." Other researchers have taken claimed more than these few patterns in manipulation. The biologist Bernard Campbell [5] stated that there are fifty-eight basic manipulation motions. The take-off point is that there is a limited set.

allel to the palm of a hand. Figure 2(3) presents the so-called "log-rolling" strategies, which are used to rotate objects with circular cross sections. Side rolling is shown in Figure 2(3a), and finger tip rolling in Figure 2(3b). In side rolling, the contact point moves along the cylindrical side portion of the fingertip, while in the tip rolling it moves around the spherical end of the tip. Figure 2(4) illustrates a common strategy used to rotate rectangular objects with the fingertips, the "twiddle" manipulation. Figure 2(5) is the "pivot" rotation, an example of using controlled slip (Brock [3]). A final rolling strategy is shown in Figure 2(6), in which the contacts cause a rectangular object to rotate around the circumference of a finger, a variant of the twiddle rotation.

As noted above, each of these strategies can be used in different start configurations. The following table (Figure 2) outlines the forces monitored for all strategies. (The coordinate frame used is shown in Figure 5.) In all cases, once a grasp is attained, it is maintained automatically during the manipulation. The force feedback, $[f_x, f_y, f_z, m_x, m_y, m_z]$, represents the estimated force and moments resisting the motion of the *object* in a particular direction (rather than the contact forces). The resisting forces are estimated from sensed contact force information.

Strategy	Control parameter	Force feedback
Palmer Translation	X	f_x
Pinch Translation	Z	f_z
Transverse Translation	Y	f_y
General Translation	X, Y, Z	f_x, f_y, f_z
Side-roll rotation	R_y	m_y
Tip-roll rotation	R_z	m_z
Twiddle rotation	R_y or R_z	m_y or m_z

Task partitioning is discussed in Michelman and Allen [16]. The individual fingers in a manipulation are given specific roles in a manipulation. The roles can often be described using C-surface specifications (Mason [13]) and implemented with a hybrid position/force controller (Raibert and Craig [19]). For example, suppose the task is to rotate a cylinder with two fingers. The technique used for rotating cylindrical objects is the "log-rolling" strategy, shown in Figure 2(3). (With the Utah/MIT hand, it is possible to perform this manipulation with the sides of the thumb and index finger.) To achieve a rotation of angle θ, finger, F_0, moves $r_c\theta$ in the x direction and F_1 moves the same distance in the $-x$ direction. In general, the precise geometry of the cylinder is not known or it may be desired to vary the grasping force (to increase the amount of torque applied to the top, for example).

Task partitioning is used to control the grasping force easily. Normally, one finger (or virtual finger, Iberall [9]) remains rigid in the direction of the grasp and the opposing finger modulates the grasping force. Here the internal grasping forces are seen clearly from the equilibrium condition: $f_{0y} + f_{1y} = 0$, where f_{0y} is the component of the thumb's contact force in the y direction. There are an infinite number of solutions for the internal grasping forces (Salisbury [14]), and the selection of internal grasping forces is an active area of research. For the simple two-contact case, in which the forces are diametrically opposed to each other, the grasping forces are always equal and opposite. By holding the position of one finger fixed in the grasping direction, it is possible to control the grasping forces by controlling the force of the other finger. We call this the "principle of the fixed surface," and it is a useful way to specify position and force directions for manipulation tasks. In addition, if both fingers obey pure force control, the position of the object is not stable in the grasping directions. Figure 3 summarizes the position- and force-controlled directions for this task. In general, the grasping forces are set so as to avoid slippage. For each finger contact force, the tangential and normal forces are f_t and f_n. Using the Coulomb model, to avoid slip, the ratio $\frac{f_t}{f_n} < \mu$, where μ is the coefficient of friction between the object object and the fingers.

Finger	x	y	z
F_0	p	f_{0y}	p
F_1	p	f_{1y}	p

Figure 3: Task partitioning: Hybrid position/force specifications

Consider an elementary function for translating a block in the x-direction of the Utah/MIT hand (which is parallel to the palm moving toward and away from the thumb). The manipulation is shown in Figure 4. People have very little range in performing this motion—if they can do it at all—due to kinematic limits.

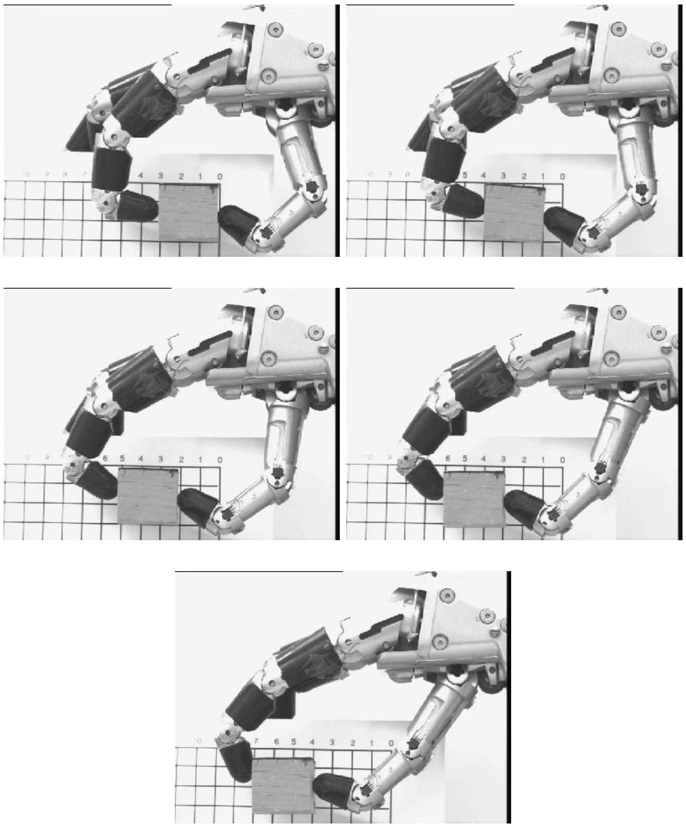

Figure 4: Palmar translation: (Utah/MIT hand photographs)

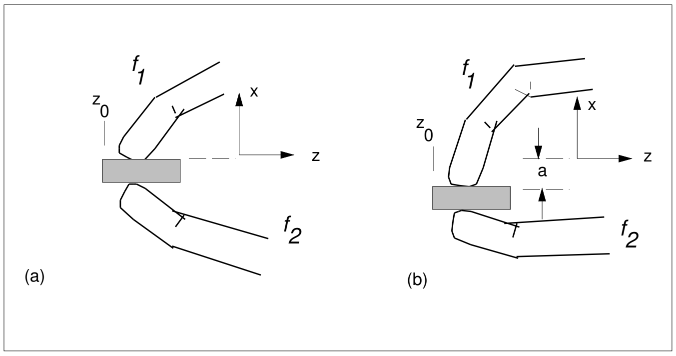

Figure 5: Palmar translation

Palmar translation partitions the fingers' roles into leader and follower (or, equivalently, master/slave). The finger behind the object pushes, while the finger in front complies with the motion. The motion of the object is in the direction of contact normals. In Figure 5, F_1 functions as the master and F_2 the slave. The control of the pushing finger is based on position error, while the slave finger control is based on force errors. Previous work on the coordination of two-arm manipulation has discussed master/slave control for quasistatic manipulation (Kopf and Tetsura [12]). The selection vectors for the master and slave fingers, S_m and S_s, are: $S_m = [0, 0, 0]^T$ and $S_s = [1, 0, 0]^T$. In master/slave control, there is no communication between the two fingers. As the master finger pushes, the slave finger reacts solely to the sensed motion of the grasped object.

Hybrid force/position control enables the hand to maintain grasp stability during manipulation. In the complete teleoperation system, once a manipulation function begins, it maintains the grasp throughout its operation. This section has discussed the elementary manipulation functions and the key components of parametrization and grasp maintenance. The following section outlines how they can be used in a complete teleoperation system.

3 Manipulation system

To perform telemanipulation requires that the grasping and manipulation functions be combined. At each stage in an operation, the input device acquires a different significance. Consider using a joystick with a pushbutton to control a translation function. At the beginning, the joystick is used to command the arm to approach the object to manipulate. Once the arm is nearby, a single degree of freedom of motion is used to control the closing function of the hand. Once the hand has grasped the object, the joystick controls the position of the grasped object. For most operations, the sequence "approach, grasp, manipulate" is fixed and can be represented as a finite state machine. For each manipulation, sensory events need to be included. A general finite state machine is shown in Figure 6.

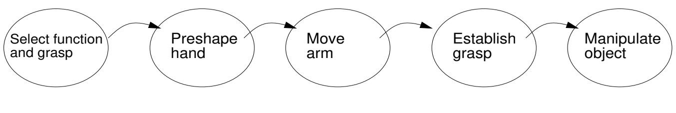

Figure 6: General finite state machine

The system at Columbia University currently uses a keyboard and monitor to control the hand system and relay information to the master. The system is described in Allen *et al.* [1]. The elementary manipulation functions outlined in the preceding section have been implemented and work autonomously. hand to control the applied finger forces in some directions and the The Utah/MIT hand includes an analog position controller, but no method to control applied finger forces. To verify the use of task partitioning, a force controller was developed. Two force-control strategies have been developed. One controls the computed fingertip force, where the force is measured in the Cartesian hand frame. The second controls the joint torque. With the Cartesian controller, arbitrary force directions can easily be specified. With the joint controller, individual joint torques are controlled separately. The joint torque controller runs between five and ten times faster than the Cartesian controller because it does not have to perform kinematic and inverse kinematic transforms in every cycle.

For each manipulation function, there are several possible start grasps. For example, each translation function can be performed with two or more fingers. For the Utah/MIT hand using purely fingertip grasps, there are three possible two-finger grasps, three possible three-finger grasps, and one possible four-finger grasp, or seven possible start configurations. Therefore, before preshaping, the start grasp is selected. After the hand is positioned near the object to be grasped, the input device is used to control the grasp. Most grasps rely on simply flexing the fingers in a predefined way. During the flexing motion, the contact forces are monitored and relayed to the master, who determines when the object is grasped sufficiently. Currently, it is possible to set a contact limit threshold so that motion continues until the limit is reached; after that point, the hand no longer responds to input device.

After the grasp is established, the input device controls object manipulation purely. Since the manipulation primitives control only a single rotational or translational degree of freedom of a grasped object, the mapping from the input device is straightforward. In [15], the autonomous functions are described. For teleoperation, the internal command signals are replaced by the external master control of the teleoperator. Thus far, the palmar translation and twiddle manipulation have been operated using keyboard command signals and feeding back external force information to the user.

4 Conclusions

Elementary palmar translation and log-rolling have been implemented using the system as described. As a teleoperation system, its ultimate utility depends on the ease with which the interface can be used. The input device used in this preliminary work was a keyboard, and force feedback was relayed numerically to a monitor. Other interfaces present new challenges. With a keyboard, the magnitude of the object displacement for each motion command and the motion velocity are fixed. Tracking the motion of a joystick will require varying both the velocity and movement distance of the grasped object. The usefulness or failure of a teleoperation system is centrally tied to the ease of use of the interface. Full hand master teleoperated control is complex kinematically, but a straightforward user interface: the robot follows the master hand. The low degree-of-freedom input device is simple kinematically, but requires a more complex interface to enable the user to switch between primitive functions and portions within the task. Interface development for this type of system is an area of research at Columbia.

Teleoperated tasks are completed 50% more quickly when force sensations are used than when they are absent [20]. Currently, feedback information during grasp is relayed numerically to the user. Single-axis force reflection is a desirable enhancement to the system and could possibly be achieved with a joystick.

Anthropomorphic, dextrous robot hands have been developed to be used with teleoperation control systems. Several factors have made this task daunting; particularly the complexity of the calibration, the differing kinematic capabilities of robot and human hands, and the lack of functional understanding of manipulation have slowed progress in this area. In space systems, for example, where bandwidth is an expensive commodity, every reduction of the complexity of communication between the operator and the robot is important. In addition, in situations in which there may be long time delays ($>$ 1 second) in communication, it is imperative that the robot perform some of its time-critical functions automatically. We have proposed increasing the autonomy of robot hands in teleoperation tasks as a way to improve their functionality for both industrial and prosthetic applications. Augmenting the hand's autonomy allows system designers to use vastly simpler and less expensive input and force-reflecting devices. Hands such as the Stanford/JPL hand and the Utah/MIT hand have the kinematic have the kinematic and sensory capability to perform precision manipulation tasks. It is hoped that augmenting their autonomous functioning will likewise increase their practicality.

Acknowledgements This work was supported in part by ARPA contract DACA-76-92-C-007, NSF grants IRI-86-57151, CDA-90-24735, CDA-93-09749, North American Philips Laboratories, Siemens Corporation Rockwell International, and National Science Foundation CISE Postdoctoral Research Associateship in Experimental Science.

References

[1] P. Allen, P. Michelman, and K. Roberts. A system for programming and controlling a multi-sensor robotic hand. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1450–1456, 1990.

[2] A. Bejczy and W. Kim. Predictive displays and shared compliance control for time-delayed telemanipulation. In *IEEE International Workshop on Intelligent Systems and Robots (IROS '90)*, 1990.

[3] D. L. Brock. Enhancing the dexterity of a robot hand using controlled slip. In *1988 IEEE International Conference on Robotics & Automation*, pages 249–251, 1988.

[4] G. Burdea, J. Zhuang, E. Roskos, D. Silver, and N. Langrana. A portable dextrous master with force feedback. *Presence*, 1(1):18–28, Winter 1992.

[5] B. Campbell. *Humankind emerging, Fifth edition*. Scott, Foresman and Company, 1988.

[6] P. Chappell and P. Kyberd. Prehensile control of a hand prosthesis by a microcontroller. *Journal of Biomedical Engineering*, 13:363–369, Sept. 1991.

[7] K. A. Farry and I. D. Walker. Myoelectric teleoperation of a complex robotic hand. In *1993 IEEE International Conference on Robotics & Automation*, volume 3, pages 502–509, 1993.

[8] J. Hong and X. Tan. Calibrating a VPL DataGlove for teleoperating the Utah/MIT hand. In *1989 IEEE International Conference on Robotics & Automation*, pages 1752–1757, 1989.

[9] T. Iberall. The nature of human prehension: three dextrous hands in one. In *1987 IEEE International Conference on Robotics & Automation*, pages 396–401, 1987.

[10] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers. Design of the Utah/MIT dextrous hand. In *1986 IEEE International Conference on Robotics & Automation*, pages 1520–1532, 1986.

[11] J. S. Kelso, editor. *Human Motor Behavior: An Introduction*. Lawrence Erlbaum Associates, 1982.

[12] C. D. Kopf and T. Yabuta. Experimental comparison of master/slave and hybrid two arm position/force control. In *1988 IEEE International Conference on Robotics & Automation*, pages 1633–1637, 1988.

[13] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(6):418–432, 1981.

[14] M. T. Mason and J. K. Salisbury. *Robot hands and the mechanics of manipulation*. MIT Press, 1985.

[15] P. Michelman. *Precision manipulation with a dextrous robot hand*. PhD thesis, Department of Computer Science, Columbia University, September 1993.

[16] P. Michelman and P. Allen. Compliant manipulation with a dextrous robot hand. In *1993 IEEE International Conference on Robotics & Automation*, volume 3, pages 711–716, 1993.

[17] J. R. Napier. *Hands*. Pantheon Books, New York, 1980.

[18] L. Pao and T. H. Speeter. Transformation of human hand positions for robotic hand control. In *1989 IEEE International Conference on Robotics & Automation*, pages 1758–1763, 1989.

[19] M. Raibert and J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 102:126–133, June 1981.

[20] T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. M.I.T. Press, 1992.

[21] T. H. Speeter. Transforming human hand motion for telemanipulation. *Presence*, 1(1):63–78, Winter 1992.

[22] J. G. Webster, A. M. Cook, W. J. Tompkins, and G. C. Vanderheiden, editors. *Electronic devices for Rehabilitation*. John Wiley & Sons, 1985.