Forming complex dextrous manipulations from task primitives *

Paul Michelman Peter Allen
Center for Research in Intelligent Systems, Columbia University
New York, New York 10027

Abstract

This paper discusses the implementation of complex manipulation tasks with a dextrous hand. The approach used is to build a set of primitive manipulation functions and combine them to form complex tasks. Only fingertip, or precision, manipulations are considered. Each function performs a simple two-dimensional translation or rotation that can be generalized to work with objects of different sizes and using different grasping forces. Complex tasks are sequential combinations of the primitive functions. They are formed by analyzing the workspaces of the individual tasks and controlled by finite state machines. We present a number of examples, including a complex manipulation—removing the top of a child-proof medicine bottle—that incorporates different hybrid position/force specifications of the primitive functions of which it is composed. The work has been implemented with a robot hand system using a Utah-MIT hand.

1 Introduction

Demonstrations of the manipulation capabilities of dextrous robot hands have lagged behind the building of research hands. Several significant experiments have shown particular capabilities of robot hands. Okada [11] built a three-fingered hand and demonstrated several elementary manipulations with his hand (putting a nut on a bolt). Fearing [4] used a Stanford/JPL hand to twirl a small block. Starr [16] and Paetsch and von Wichert [13] have used Cartesian object stiffness control for parts alignment and peg-in-hole tasks.

In our work, a functional set of manipulation primitives is defined. The tasks can be generalized to perform different functions in complex tasks by varying parameters. They can also be combined to create complex tasks. Incorporated into each task is task partitioning of the fingers' roles using position and force specifications. The partitioning aids in maintaining grasp stability during manipulation and robustness for grasping objects of unknown geometry. There are several justifications for using elementary functions. First, task planning is simplified with predefined modules. Once a planner decides that a motion is required (for example, "rotate object around a particular axis"), a list of available functions is consulted. The function is particularized for the task by instantiating a set of parameters for position, size and timing. The initial grasp configuration and approximate finger motions are part of the primitive manipulation function. A second reason for using primitive functions is that in reality there is a finite set of motions that a hand must perform. The biologist Bernard Campbell [2] has written that the human hand has 58 basic motions. John Russell Napier [10], who studied the evolution of the human hand, wrote: "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) The two basic patterns are those in which the primary object motion is caused by arm motion ("power grasps") and those in which the object motion is caused by finger motion ("precision grasps"). Cutkosky [3] further developed a classification of grasps used by machinists and an expert system that could select a start grasp given a set of task criteria.

In this paper, we discuss a set of primitive functions for *precision manipulation* with the Utah-MIT hand and show how those task primitives can be combined into more complex tasks. Section 2 briefly describes the set of manipulations for rotating and translating grasped objects. Section 3 presents an example task—removing the top of a child-proof medicine bottle—which combines two elementary tasks. A description of its implementation on a system with a Utah-MIT hand is included. Section 4 discusses the experimental results and suggestions for extensions of the work.

2 Manipulation task primitives

An elementary set of primitive functions includes those cooperative finger motions that are required to translate and

*This work was supported in part by DARPA contract DACA-76-92-C-007, NSF grants IRI-86-57151, CDA-90-24735, CDA-93-09749 North American Philips Laboratories, Siemens Corporation and Rockwell International.

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 require the rotation of 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, (5) the overall goals of the tasks, and (6) possibly the direction of rotation.

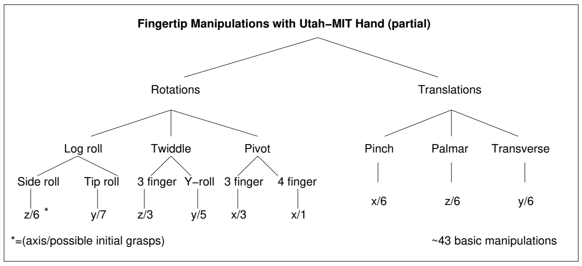

Figure 1: Simple fingertip manipulation with Utah-MIT hand (The leaf nodes of the tree contain the number of possible start grasps for a motion in a particular direction. For the seven possible start grasp configurations for the y-axis log-rolling manipulation, the notation is "y/7".)

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 1 is a tree illustrating the basic functions for the Utah-MIT hand. To perform the basic repertoire of manipulations, there are at least 43 initial grasp configurations[1] possible, even for this limited set. For example, to rotate an object with a circular cross section around the y-axis (using a log-rolling strategy), seven initial grasp configurations are possible. Although there are often many start grasps possible for the manipulations, the basic strategy is the same for each of them and, fortunately, the methods of choosing the finger trajectories and maintaining stability are the same for each. The strategies incorporate several features: (1) approximate finger trajectories based on an estimate of object size; (2) task partitioning to maintain grasp stability during motion; (3) workspace analysis to calculate the maximum path length of the object motion and the initial grasp positions of the fingers and the hand that allow the manipulation to take place; and (3) a set of parameters (as described above).

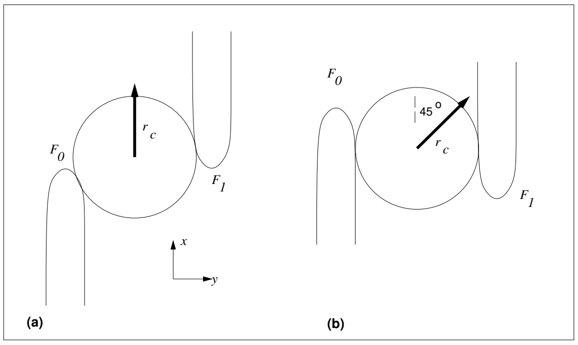

Figure 2: Log rolling strategy

Task partitioning was discussed in Michelman and Allen [9]. The individual fingers in a manipulation are given specific roles in a manipulation. The roles can often be described using C-surface specifications (Mason [6]) and implemented with a hybrid position/force controller (Raibert and Craig [15]). For example, suppose the task is to rotate a cylinder with two fingers. The technique used for rotating cylindrical objects is the so-called "log-rolling" strategy, shown in Figure 2. (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 top is not known and must be learned from sensing. It also 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 [5]) 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: $f_{0y} = f_{1y}$, 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 [7], and the selection of internal grasping forces is an active area of research (for example, see Park and Starr [14]). By holding the position of one finger fixed in the grasping direction, it is possible to control the grasping forces by modulating the opposing finger's contact force. We call this the "principle of the fixed surface," and it is a useful way to specify position and force directions for manipulation tasks. 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. Using the sim-

[1] A "grasp configuration" is considered to be the set of fingers performing the grasp and the approximate location on the fingertip. For example, the log-rolling strategy pictured in Figure 2 uses a grasp configuration consisting of fingers 0 and 1, with side contacts on each finger.

ple Coulomb model, to avoid slip, the ratio of tangential to normal forces, $\frac{f_t}{f_n} < \mu$, 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	p	p

Figure 3: Task partitioning: Hybrid position/force specifications

3 A complex task: removing a childproof bottle top

In a real application, the elementary tasks are often combined sequentially to form complex tasks. The technique we have used is to build a finite state machine to sequence through the tasks. Tests, such as guarded moves, are included at the ends of certain states. An important issue in combining elementary tasks is determining which tasks can be combined and what is the workspace for the combined task. This is described in Michelman [8]. This section discusses an example of a common task that uses complex specifications: removing the top of a childproof medicine bottle. The task uses two elementary tasks: log-roll to rotate the top followed by a translation in the z-direction to lift it.

Figure 4: Example task: Removing top of childproof bottle

Figure 4 is a photograph of the Utah-MIT hand grasping the bottle. The axis of rotation is in the z-direction. It is assumed that at the start of the task, the bottle is sealed such that simple rotation will not be sufficient to remove the lid. It must first be pressed down (in the $-z$-direction) and rotated counterclockwise approximately a quarter turn. (In the photograph, the fingers are exerting the downward pressure on the top.) After the initial rotation, the lid can only continue to be unscrewed by applying an upward force (in the $+z$-direction) while rotating. After each counterclockwise rotation, the hand tests to see if the top is free of the threads. If it is, it lifts the top from the bottle. If not, the fingers regrasp the lid and turn it again. A finite state machine for this task is shown in Figure 5(a). The numbers in the figure refer to the following stages: (1) Grasp the top. (2) Apply downward force while rotating counterclockwise. (3) Regrasp the top. (4) Apply upward force while rotating counterclockwise. (5) If top is not free of threads, go to (3). Lift the top from the bottle.

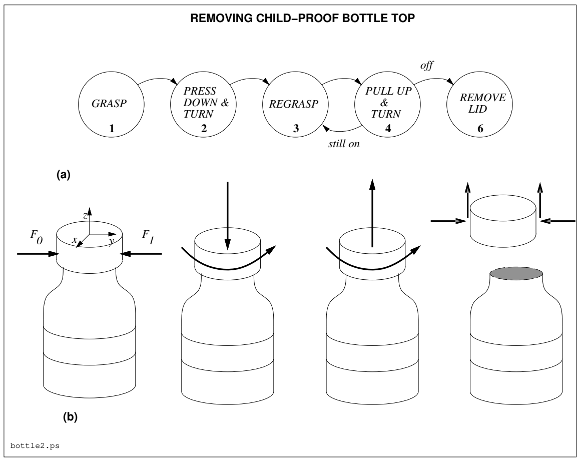

Figure 5: Finite state machine: rotation plus translation

Each part of the task has a different task specification. Figure 6 shows the four specifications for the task. In the figures, a "p" represents position control and a "f" represents "force" control. It is assumed that two fingers are used for the manipulation. There is nothing in the task to preclude the use of a three- or four-fingered grasp. The two-fingered manipulation is presented because it is the most the straightforward and the technique used in the implementation (see below).

In part (1), the grasp on the bottletop is acquired. As discussed above, one of the fingers uses force control to modulate the grip strength, while the opposing finger uses position control to remain rigid, the "fixed surface." Throughout the remaining portions of this task, the grasp is maintained by this task partitioning. Initially, the fingers close until the required force, $-f_{ygrasp}$, is attained for finger F_1, while finger F_0 remains stiff.

In part (2) of the task, the bottletop is pressed down and turned. To turn the top, F_0 uses a position trajectory in the $-x$-direction while F_1 follows a $+x$-trajectory. The motions of the fingers in the opposite direction cause the rotation of the top. Simultaneously, the fingertips press down on the bottletop. If it is tight, it can only be removed by applying a $-z$-force while rotating it clockwise approximately one quarter turn. To apply the downward force, *both* fingers press down at the same time. In Figure 6(2), the downward forces are shown as $-f_z$.

1. Grasp/5. Lift	x	y	z
F_0	p	p	p
F_1	p	$-f_{ygrasp}$	p

2. Press/turn	x	y	z
F_0	p	p	$-f_z$
F_1	p	$-f_{ygrasp}$	$-f_z$

4. Pull/turn	x	y	z
F_0	p	p	$+f_z$
F_1	p	$-f_{ygrasp}$	$+f_z$

Figure 6: Task specifications for removing bottle lid

Part (3), regrasping, is similar to part (1) and performed repeatedly. The fingers release the top, return to their start positions, and regrasp. Part (4) of the task involves turning and lifting the top while maintaining the grasping forces. The specification is similar to part (2), except a force in the $+z$-direction must be applied. The particular type of bottle chosen requires that the top be lifted and turned simultaneously. If not, the ridges on the bottletop catch on the bottle and the top does not rotate. The upward motion is specified in Figure 6(4) with $+f_z$ commands for F_0 and F_1.

Finally, when the top has been sufficiently loosened and is free, the fingers lift it. To decide whether the top is free, the top is lifted slightly (pulled a small amount in the $+z$ direction). If it moves freely, it is loose. If it encounters resistance, it is still being held by the threads and must be turned more.

The specifications denote the directions of position and force control, but not the values of force applied or the actual position trajectories. Deciding on these values is an important part of task learning. The grasping force is chosen to prevent slip during the rotation. As with the choice of most force values, two criteria are used: effectiveness and minimum force. The value must be strong enough to maintain a stable grasp. If it is too strong, however, there are two risks. First, there is the possibility of damaging the grasped object. Second, in practice grasp stability decreases with increased grasping force.[2] (In our experiments, the values were determined by trial and error.) The finger trajectories are dependent on the type of initial grasp used. For a two-fingered side roll manipulation, the fingers follow straight-line trajectories in opposite directions as described in the previous section.

Grasping strategy With each regrasping, it is likely that one finger makes contact before the others. It is important that when one finger contact is achieved, that finger stop moving until the other fingers make contact. If it does not, the first finger will displace the object to be grasped. Therefore, the contact threshold for the finger making first contact is much lower than for the second finger.[3] The strategy then can be outlined as follows (for two fingers): (1) Approximately center fingers around object to be grasped. (2) Move F_0 and F_1 toward the object to grasp. (3) Continue motion until either the contact force on F_0 *or* F_1 exceeds threshold force, $f_{thresh1}$. (4) Stop the motion of the finger that is in contact and continue the motion of the opposing finger until it reaches threshold $f_{thresh2}$. This strategy assumes that the object size is not known precisely in advance and is learned via the contacts. $f_{thresh1}$ is set to the lowest value that can be reliably detected. This type of strategy is used for acquiring grasps for any manipulation.

3.1 Hand dependencies for example 1

The specifications given in Figure 6 are independent of implementation to a certain degree. The specifications do not describe: (1) the type of grasp used; (2) the fingers used to implement the grasp; (3) the type of contact motions used; (4) the geometry of the bottle; (5) a number of physical properties of the task: the amount of force needed to rotate the top, the coefficients of friction between the fingers and top, and the mass of the top.

Some of these aspects of the task must be decided upon by the robot programmer, others are learned during the task. The use of force control in grasping eliminates the need to know the size of the top before the task begins. Once the top is grasped, however, the robot can easily calculate the top's diameter from knowledge of the positions of the grasp points. Other properties, such as the amount of force needed to turn the top, can also be learned during the manipulation. The type of grasp, the fingers to use, and the finger trajectories must of course be decided in advance and are dependent on the type of robot hand used. For example, the strategy used in the implementation described below may not be kinematically possible with a different robot hand. The steps in the task would be identical, however.

[2]The reason grasp stability decreases with increasing grasping force is practical. If two opposing fingers with circular cross sections are grasping a cylindrical object, and the grasping forces are perfectly diametrically opposed on the object, the grasping forces can theoretically be increased infinitely without losing stability. If, on the other hand, the normals of the grasping forces do not line up precisely or are not centered along the diameter, increasing them does cause instability either by creating a moment or by creating tangential grasping forces which will eventually cause slippage. In practice, the normal components of the grasping force are rarely perfectly aligned. People use this "heuristic" all the time.

[3]Sensitive tactile sensing allows people to detect minute contact forces. At this stage, tactile sensors for fingertips with good dynamic range that can detect both brushing contacts and higher magnitude contacts have not been developed. Usually, either surface mount sensors or internal force sensors are used.

Grasp choice With the Utah-MIT hand, there are a number of possible grasps that can be used to perform the bottletop rotation. Three possible start grasps for this task are (a) a two-fingered tip grasp; (b) the two-fingered side grasp (shown above); and (c) is a three-fingered tip grasp. The two- and three-fingered grasps can be selected with a choice of different fingers. For each grasp, the figure shows the start grasp and the grasp one rotation cycle. The initial grasp is chosen with several questions in mind: (1) will the grasp be stable and allow easy manipulation? (2) will the grasp for the log-rolling task allow the translation to be performed after the top is loose?

A two-fingered grasp using the sides of the thumb and index finger was chosen because it is the simplest grasp that allows stability. The choice of this grasp illustrates an important principle of compliant motion tasks: the presence of an external motion constraint reduces the complexity of the grasp required to perform the task. The bottletop is a cylinder. The lateral surfaces of the fingers making contact with the bottletop produce essentially point contacts with friction. Therefore, grasping the top on both sides limits only five degrees of freedom of the top: it is free to rotate around the line connecting the two contact points. The bottle's threads, however, provide a planar contact that eliminates this degree of freedom of the top and only two finger contacts are necessary. To provide force closure in a free-space manipulation would require *three* point contacts with friction. Thus, a two-fingered grasp is seen to be adequate for grasp stability during the first part of the task.

3.2 Implementation results

This task has been implemented with the Utah-MIT hand. The system used in described in Allen *et al.* [1]. For this experiment, the hand was not connected to a robot arm: only fingertip manipulations were used to turn and lift the bottletop. The basic manipulations rely on task partitioning: the ability of the hand to control the applied finger forces in some directions and the position of the finger in other directions. 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.

To develop the force controller, the fingers are treated as springs. When an external force is applied to a finger, it is displaced from its position and the position controllers attempt to maintain the commanded position. Increasing the displacement requires a greater force. The controller makes use of the inherent compliance of the fingers. The force, f_{des}, required to displace a finger from its commanded position, X_0, is

$$f_{des} = k(X - X_0) = k\Delta X \tag{1}$$

where X is the current position. Therefore, to maintain a constant applied force, the displacement, ΔX, must be held constant, which is done by adjusting the commanded position. For example, consider the case in which the controller is commanded to exert zero force in a particular direction. When an external force is applied to the fingers and it moves away from its commanded position, $\Delta X \neq 0$. Setting $f_{des} = 0.0$ in Eq. 1, the new commanded position, X_0, is X: the finger follows the motion imparted to the finger by the external force. For $f_{des} \neq 0$, the difference between X and X_0 is similarly maintained throughout. Thus, "force" commands are proportional to position errors and are specified in units of centimeters. For example, a position error of 1 cm in the x direction has been measured to be approximately 0.5 pounds.

A two-fingered grasp using the side portions of the thumb and index finger was used. The hand robustly removes the top of the childproof bottle. The use of the hybrid position/force commands makes the task generalizeable with respect to the size of the bottle top. During the rotation of the top, one finger is used to maintain the grasp by exerting a force normal to the contact direction. Both fingers either push the top down or pull it up depending on the state of the task. The joint torque controller is used rather than the Cartesian force controller to increase the servo rate.

The implementation works as follows: (1) preshape thumb and index finger for grasping; (2) grasp top desired grasp force threshold; (3) sense the radius, r, of the bottle top after grasping; (4) calculate the distance to move the fingers for desired rotation angle, θ. The distance, $r\theta$, is equal to arclength around the circumference of the top; (5) initialize compliance controller to maintain grasping force on thumb and to push top down (in the z direction) with both fingers. Joint torque controller is used. Thumb joint 0 exerts the desired grasping force. For top to be pressed down, joint 3 on both fingers is given an extending torque to produce the downward motion; if top is to be pulled up, the joints are given a flexing torque. (6) Move fingers 0 and 1 to produce log-rolling rotation (thumb moves in a straight line in the $+x$ direction and index in the $-x$ direction) to produce the desired rotation angle; (6) lift top by distance, ϵ. If top is still on, go to step (1); (7) move top in the $-z$ direction to lift it off bottle. The values of the force commands have been found by trial and error to avoid slippage. A grasping force, $f_{ygrasp} = 0.25cm$ has been used throughout.

Transition point The two-fingered grasp was used successfully to remove the top repeatedly with a Utah-MIT hand. The success or failure of the task depends on the hand's ability to sense when the top has been sufficiently loosened to be removed. The technique used, as discussed above, was to lift the top slightly after each turn. If the sensed fingertip contact forces were found to increase "substantially," that was an indication that the cap was still constrained by the bottle. The threshold values were determined by averaging the force values (for when the top was free and when it was not) over 100 trials.

Reliability This task was performed 100 times to judge its reliability and the top was removed successfully 98 times. This success was achieved after adjusting the thresholds as discussed aboved. The difficulties in tuning are caused by several factors. (1) One failure was due to a false report by the hand: the top was off and the hand reported that it was on. (2) One failure occurred while checking if the top was still on: the fingers slipped off and the hand did not sense it. This failure is prevented by proper positioning of the hand as a whole at the start of the task. The bottletop is only approximately 1.5 cm in height. Errors larger than several millimeters in positioning the hand in the global coordinate system can cause slippage.

4 Discussion of results

The system has been used to perform a number of experiments; particularly, removing a bottletop, replacing the top, and twirling similar to Fearing's experiments (see [8]). The use of hybrid position/force specifications for each finger with complex tasks allows for a straightforward way to implement tasks. Experiments that use a combination of several primitive functions illustrate several key properties of a system based on task primitives. First, no matter how general the elementary manipulation functions are, they often must be particularized for use in real-world tasks to account for sensing and controlling the interactions between the robot and the environment. In the example above, there are two additions to the elementary log-rolling strategy: (1) the specifications to lift or push down the top while rotating (the commands in the $+$ and $-z$ directions) and (2) terminating conditions. The finite state machine often specifies that a task be executed until a position or force threshold is exceeded.

Second, the planning of the complex tasks must be taken into account. For example, how are the tasks combined? Which tasks are compatible? Although in the example presented here, the planning was not automatic, the planning process suggests possibilities for automation.

Although our work has been achieved with finger motions alone, generally a hand is supported by an arm. The partitioning of motion functions between fingertip manipulations and arm manipulations is another area of planning to be pursued in the future.

As noted by previous researchers, the reliability and autonomy of the system would be increased with improved sensing, particularly tactile sensing to localize the contact points reliably and vision to be able to track object and finger motion during manipulation. We hope to augment the current system with these monitoring sensors.

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

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

[4] R. Fearing. Implementing a force strategy for object reorientation. In *1986 IEEE International Conference on Robotics & Automation*, pages 96–102, 1986.

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

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

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

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

[9] 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.

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

[11] T. Okada. Computer control of multijointed finger system for precise object-handling. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-12(3), May/June 1982.

[12] T. Okada. Object-handling system for manual industry. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-9(2), February 1982.

[13] W. Paetsch and W. von Wichert. Solving insertion tasks with a multifingered gripper by fumbling. In *1993 International Conference on Robotics and Automation*, volume 3, pages 173–179, 1993.

[14] Y. C. Park and G. P. Starr. Finger force computation for manipulation of an object by a multifingered robot hand. In *1989 IEEE International Conference on Robotics & Automation*, pages 930–935, 1989.

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

[16] G. P. Starr. Experiments in assembly using a dexterous hand. *IEEE Transactions on Robotics and Automation*, 6(3):342–347, June 1990.