

## An Experimental Study of Hierarchical Control Laws for Grasping and Manipulation using a Two-Fingered Planar Hand

Karin Hollerbach  
Department of EECS  
University of California  
Berkeley, CA 94720

Richard M. Murray  
Department of Mechanical Engineering  
California Institute of Technology  
Pasadena, CA 91125

S. Shankar Sastry  
Department of EECS  
University of California  
Berkeley, CA 94720

### Abstract

*We compare the performance of hierarchical and single-level controllers in a grasping context, and we conclude that for rapid, planar grasping motions of heavy objects the performance of a hierarchical control structure is superior to that of the two single-level controllers tested. Although the theory discussed here applies to grasping problems of arbitrary complexity, we focus on planar, two-fingered grasping for the sake of clarity and to simplify implementation and experimental testing of the proposed control algorithms.*

### 1 Introduction

Interest in complex, multi-fingered robotic hands has seen an increase in the last few years, as advanced designs and a rigorous theory used to describe them have been developed. Early research on multi-fingered hands focused on the description of hand kinematics [8] and on the generation of stable grasps [13]. Other researchers have developed simple algorithms used in grasping control of single hands [7, 10] as well as control of associated groups of robots, cooperating to perform a single task [1, 12].

Advances in multi-fingered hand design are readily apparent when perusing the literature: the more well-known designs include the Utah-MIT hand [4], the Stanford/JPL hand [14], the NYU hand [2]. However, the problem of overcoming the computational burden associated with control of some of the more complicated hand designs has not been adequately addressed. In response to the computational difficulties surrounding control of complicated robotic hands we seek an approach to multi-fingered hand control that significantly reduces the computational burden placed on the controller while improving the grasping performance of the hand, and that is essentially design-independent. A beginning in the development of such an approach was made in [9], and the experimental results presented here may be considered an implementation of the basic philosophy of hierarchical robot control as laid out there.

In this paper we compare the performance in a planar grasping task of a hierarchical control algorithm with that of single-level controllers containing no hierarchical structure. We then describe the methods and the hardware used in the experimental comparison of the controllers' performance and subsequently present the experimental evidence demonstrating the superior performance in rapid, planar movements of heavy objects of the hierarchical controller.

### 2 Hierarchical Control of Grasp Dynamics

Following the results of [11], the equations of motion for a multi-fingered hand grasping an object can be written as

$$\tilde{M}(q)\ddot{x} + \tilde{C}(q, \dot{q})\dot{x} + \tilde{N}(q, \dot{q}) = F_e + GJ_h^{-T}\tau \quad (1)$$

where  $q = (\theta, x)^T$ ,  $\theta$  represents the finger joint angles,  $x$  is the configuration of the object, and  $F_e$  is the external force applied to the object. We also define the hand Jacobian matrix,  $J_h$ , the grasp map,  $G$ , and the set of applied joint torques,  $\tau$ . In the object frame of reference,  $\tilde{M}$  is the matrix of the effective mass of the system, and  $\tilde{C}$  is the effective Coriolis and centrifugal matrix. These matrices include the dynamics of the fingers, which are being used to actually control the motion of the object. However the details of the finger kinematics and dynamics are effectively hidden in the definition of  $\tilde{M}$  and  $\tilde{C}$ .

We note, from the equations of motion (1), that, in the absence of external forces, if the fingertip force,  $J_h^{-T}\tau$ , is in the null space of  $G$  then the net force in the object's frame of reference is zero and causes no net motion of the object. This force acts against the constraint and is generally termed an *internal* or *constraint* force. We can use this internal force to satisfy other conditions, such as keeping the contact forces inside the friction cone.

To illustrate the control of robot systems, we review an important controller that has appeared in the

robotics literature, a variant of the proportional-plus-derivative controller (PD) [5] and apply it to systems of the form described in equation (1).

In PD controllers the desired stiffness (and potentially damping) of the end effector is specified. Consider the control law

$$F = M(q)\ddot{x}_d + C(q, \dot{q})\dot{x}_d + N(q, \dot{q}) + K_v\dot{e} + K_p e \quad (2)$$

where  $F$  is the force required to move the object along the desired trajectory,  $x_d$ ,  $e$  is the trajectory position error, and  $K_v$  and  $K_p$  are symmetric positive definite gain matrices. Using a Liapunov stability argument, it can be shown that the actual trajectory of the robot converges to the desired trajectory asymptotically [5].

A multi-fingered robot hand can be modeled as a set of robots that are connected to an object by a set of constraints. In [9], a system was proposed for building hierarchical control laws for complex interconnected robotic systems. We apply that formulation here. In designing hierarchical controllers such as the one used here, a key issue is how to properly model a robot that has a controller attached to it. For controllers that are very fast relative to higher levels, it is often a good approximation to model the robot as an ideal force generator, with no mass, implying that controllers at higher levels can ignore the dynamic properties of the robot, since these properties are being compensated for at a lower level.

In order to demonstrate stability of the hierarchical control scheme proposed here, we model the scheme as a two-step hierarchy: a low level PD at the motor level and a high level PD-type control for the "hand". Standard singular perturbation arguments [6] may be used to show that the scheme is exponentially convergent, provided that the low level controller is fast enough, that is, provided the sample period is small enough.

### 3 Experimental Setup

The control algorithms presented here have been implemented on a multi-fingered hand, known as Styx [10]. Styx is a two-fingered, planar hand, with each finger consisting of two revolute joints and two links. The distal links are capped by small rubber cylinders that serve as fingertips and as contact "points" between the fingers and the object that is to be manipulated. A diagram of Styx is shown in Figure 1.

The motors used to drive Styx are direct-drive DC motors mounted at the base of each link and are driven with a pulse-width modulated 20 kHz square wave. Each motor contains a quadrature encoder used to sense joint position. The resolution for the proximal

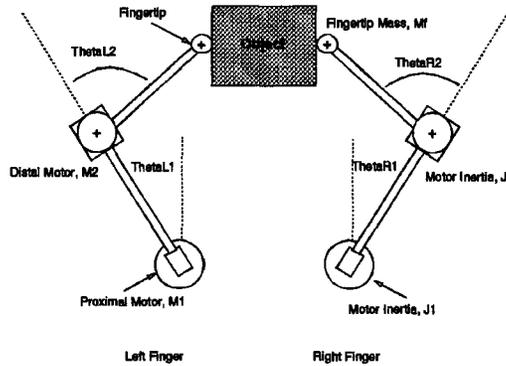


Figure 1: Top View of Styx

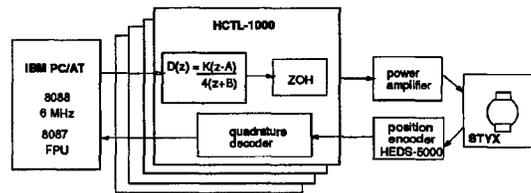


Figure 2: Hardware Supporting Styx

motors is 3600 counts per revolution and for the distal motors 2000 counts per revolution. Styx is connected to an IBM PC/AT running at 6 MHz with an 8087 floating point coprocessor. The motors and encoders are interfaced to the AT using a set of four HP HCTL-1000 motion control chips interfaced to the AT bus. A view of the interconnection of the hardware supporting the Styx system is shown in Figure 2. The parameters associated with Styx kinematics and dynamics are shown in Table 1.

In the discussion of the tracking performance of the hierarchical structure we assumed the existence of two

Link Lengths	$L_1, R_1$	15.3	cm
	$L_2$	12.16	cm
	$R_2$	11.8	cm
Fingertip Radius	$r_f$	1.7	cm
Base Separation	B	20.0	cm
Link Mass	$M_{L1}, M_{R1}$	53	g
	$M_{L2}$	17	g
	$M_{R2}$	20	g
Distal Motor Mass	$M_2$	328	g
Fingertip Mass	$M_f$	3	g
Motor Inertia	$J_1$	18	g cm <sup>2</sup>
	$J_2$	1.74	g cm <sup>2</sup>

Table 1: Styx Parameters

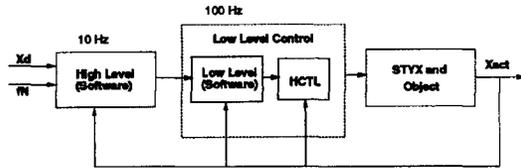


Figure 3: Control hierarchy for Styx, showing control of the desired and actual object trajectories,  $X_d$  and  $X_{act}$ , respectively, and the specified internal force,  $f_N$ .

continuous time controllers, a high level PD plus feed-forward controller in object coordinates as well as a low level PD controller in joint angle coordinates. Here we describe the implementation details of the hierarchical structure, including the points of departure of the implementation from the theory.

Assumptions made to simplify implementation of the control algorithms presented here include: the motor dynamics can be ignored; the fingertips can be modeled as fixed point contacts; the Coriolis and frictional forces are ignored. In order to implement the two-level hierarchical structure, we actually used the three-level structure shown in Figure 3. The upper two levels, consisting of a primary and a secondary control loop, are written in the C programming language, using the Microsoft 5.1 Optimizing C compiler. An assembly language scheduler controls the sample rates of the control loops.

In the left portion of Figure 3 we see the highest control level, the secondary control loop running at 10 Hz. At this level we calculate the inverse kinematics of the object's desired trajectory ( $X_d$  in Figure 3) and perform the high level control functions when we put Styx into the hierarchical control mode. We present results of the high level PD + feedforward controller in object coordinates.

The lower level of the control hierarchy consists of the low level software block shown in Figure 3 in addition to the hardware block to the right of it. The purpose of the primary control loop that represents the low level is to write at a frequency of 100 Hz directly to the motion control hardware the current commanded joint angles received from the secondary loop. In addition, the low level controller is capable of a simple, linear joint angle interpolation that allows the low-level controller to gradually command the joint position to move from the position commanded by the high-level controller at one time step to the position commanded at the following time step.

At the lowest level of the control hierarchy exists the HP HCTL-1000, a digitally sampled, general purpose motor controller. We used the HCTL-1000 in the

	Beam	Box	
Mass	245	33	g
Moment of Inertia	$1.8 \times 10^3$	$1.3 \times 10^3$	$\text{g cm}^2$
Length	12	17	cm

Table 2: Object Parameters

implementation, because it had already been built into the existing hardware. In the analysis, our assumption was that the digital sampling of the controller was fast enough so that we could approximate it by a continuous time PD controller.

Programmable variables in the HCTL's position control include the sampling time,  $T$ , as well as the parameters associated with the digital filter used to compensate for closed loop system stability:

$$D(z) = \frac{K(z - A/256)}{4(z + B/256)} \quad (3)$$

Since the HCTL-1000 was used in its position control mode, we did not directly send desired torque commands to the HCTL controller. When we needed to apply a particular torque, we calculated a virtual position error which, when multiplied by the DC gain of the position controller, yielded joint torques. The calculated joint torques equaled the desired torques as long as the trajectory was slow enough relative to the time constants associated with the HCTL controller that we could use the DC approximation.

The control system was tested with each of three different progressively more complicated control schemes, each building on the one(s) before: setpoint control without joint interpolation, setpoint control with joint interpolation, and hierarchical control (including setpoint control with joint interpolation at the lower level and PD plus feedforward at the high level).

## 4 Experimental Results

In the experimental results presented here, we used two different objects, each suited to meet the needs of the particular trajectory being tested. In the first set of trajectories, we used a metal bar that was loosely attached to the fingertips via pin joints. The parameters associated with this object are shown in the column labeled "beam" in Table 2. The second object used in the experiments was a cardboard box, the dimensions of which are also shown in Table 2, in the column labeled "box".

The HCTL-1000 parameters used in the generation of the figures shown were the chips' default parameters: Gain,  $K_d = 64$ ; Zero,  $A = 229$ ; Pole,  $B = 64$ ;

Sample Freq,  $\frac{1}{T} = 1.9$  KHz. The commanded trajectories for the objects' centers of mass were circular trajectories of radius 2.5 cm, centered at  $x = 1.3$  cm,  $y = 21.2$  cm relative to the midpoint between the two proximal motors, as shown in Figure 1, with frequencies of 1.0 Hz for the "beam" and 0.25 Hz for the "box". The orientation of the objects was to remain at zero throughout each movement. We chose the high level control gains by assuming the presence of an ideal low level controller which does a perfect job of controlling the fingers. As a result, we were able to ignore at the high level the finger dynamics and write a control law that considered only the object dynamics, which consisted of a simple second order system with a diagonal inertia matrix. To this simple, ideal system, we applied linear control theory and selected a rolloff above the trajectory frequencies and a damping ratio of 0.5, by choosing the high-level control gains to be  $K_v = 2$  and  $K_p = 4$ . These gains, which would have yielded exactly the desired second order response in the ideal case resulted in acceptable performance in the actual system. In each trial, we determined that collecting data for a period of 20 seconds (5 cycles each for the slow trajectories and 20 cycles each for the fast motions) was entirely sufficient for our purposes, yielding consistent results. In the slow movements we applied an internal force of  $3 \times 10^4$  dyne; due to hardware limitations, during rapid movements we used an object that we could loosely attach to the fingertips, enabling us to use smaller internal forces ( $3 \times 10^3$  dyne) without losing contact with the object.

Figures 4-6 depict the performance of the three different controllers, the setpoint controller (non-hierarchical), the setpoint controller with low-level joint velocity interpolation, and the hierarchical controller. The rather poor performance of the setpoint controller can be judged quickly by examining Figure 4(a), which shows the actual trajectory of the object's center of mass. A large overshoot, primarily in the horizontal,  $x$ , direction is evident.

The second controller that was used, the joint interpolation controller, was non-hierarchical as well. Although the trajectory shown in Figure 4(b), was found to be smoothed out more, we found the same overshooting of the goal trajectory that occurred in Figure 4(a). The problem that remained to be addressed was the object's mass, a parameter that was not taken into account at the lower level.

A significant improvement in the trajectory tracking performance was found when a high-level controller that corrected for the object's tracking errors was superimposed upon the existing low-level control

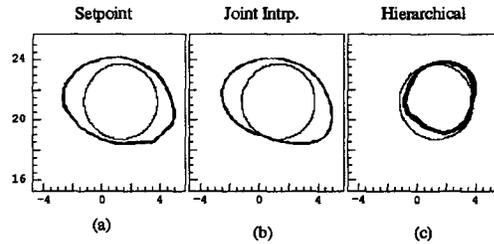


Figure 4: Actual and commanded object trajectories with setpoint controller (a), joint interpolation controller (b), and hierarchical controller (c) in Cartesian space with weighted beam; commanded circular trajectories are shown within each graph (fast trajectory).

structure. By calculating the torques required to bring the object back to the desired trajectory, the high-level controller was able to compensate for and minimize the trajectory errors in the object's (Cartesian) coordinates, as shown in Figure 4(c). The overshoot that so grossly disfigured the trajectories of the two non-hierarchical controllers disappeared, resulting in a much better overall tracking performance.

In the figures showing data in "box-plot" form (Figures 5-6), the top and bottom of the boxes correspond to the twenty-fifth and the seventy-fifth percentiles of the given variables, while the horizontal lines through the boxes correspond to the median values of the variables. The vertical lines have ends that extend beyond the quartiles by a distance equal to one and one-half times the inter-quartile range.

In Figure 5 we show the calculated box position error, for the setpoint controller (a), the setpoint controller with interpolation (b), and the hierarchical controller (c) for the same trajectories that were depicted in Figure 4. For each controller, the error is given by  $\sqrt{(X_{act} - X_d)^2 + (Y_{act} - Y_d)^2}$ . Again, the marked improvement of the performance of the hierarchical controller over the performance of both single-level controllers is evident.

The improvement in the object's position tracking that we found when using the hierarchical controller was, unfortunately, not mirrored in the object's orientation, as can be seen in Figure 6. One possible cause is a poor approximation of the moment of inertia of the object. Due to the configuration of the system, a very slight shift in the fingertip position can cause a relatively large orientation error in the object. Thus, any error in the calculation of the object's moment of inertia can cause an error in the orientation portion of the feedback control, which, in turn, can cause a significant orientation error in the object.

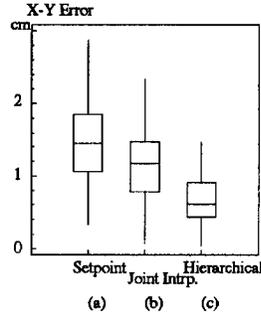


Figure 5: Deviation (in cm) from commanded trajectories of actual object trajectories with setpoint controller (a), joint-interpolation controller (b), and hierarchical controller (c) when using weighted beam (fast trajectory).

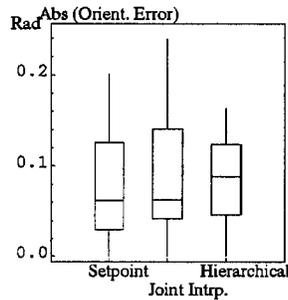


Figure 6: Deviation (in radians) from commanded object orientation of actual object orientation with setpoint controller (a), joint interpolation controller (b), and hierarchical controller (c) from commanded trajectories when using weighted beam (fast trajectory).

The reasons for using the beam in the controller comparisons were to enable us to track high-speed trajectories with heavy objects without saturating the motor output. However, we wish to point out that the system is, in fact, capable of standard “grasping” manipulation. We tested the performance when the three controllers were used to manipulate the lighter object (“box”) in tracking the slow (0.25 Hz) trajectory. As expected, at slow speeds and when using small object masses, the differences between the single-level controllers, which did not take into account the object’s parameters, and the hierarchical controller became insignificant. The object’s trajectory, box plot of position error, and orientation error as a function of time are shown for the three controllers in Figure 7.

The commanded internal force exerted on the object by the hand was kept at a constant value,  $3 \times 10^4$  dyne, in all of the slow trajectory experiments shown here. This value was chosen, since it seemed an

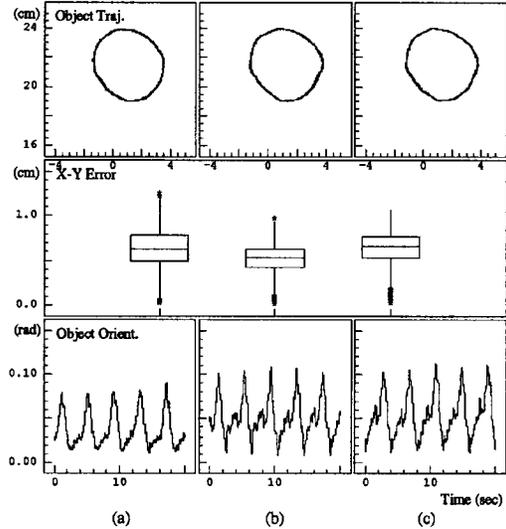


Figure 7: Object trajectory, box plot of object position error, and object orientation error as a function of time for setpoint controller (a), setpoint controller with joint interpolation (b), and hierarchical controller (c) (slow trajectory).

adequate compromise between holding the object with a minimum of slipping and forcing the grasp to become unstable. This potential instability due to a large “internal force” was a real problem that needed to be addressed: the grasp and Jacobian matrices, used in the internal force calculation, were updated at the relatively slow speed of the high-level control loop. As a result, the calculated “internal” force contained a small, at times significant, portion that lay outside of the grasp matrix’s null space. An even more significant contributing factor, perhaps, was the amplifier gain “drift” over time, which made precise calibration of the system an extremely difficult task, and one that would need to be repeated throughout the life of the robot components. We stress this difficulty in calibration, because it goes beyond the need to precisely measure the kinematic parameters associated with the robotic hand and the object. In general, we found that high commanded internal forces interfered significantly with proper tracking behavior and with stability.

#### 4.1 Discussion of Results

The results discussed here represent an experimental confirmation of the predicted stability of the hierarchical control structure under the conditions given. In particular, the assumptions regarding the separability of the choice of control parameters as well as the

time scales of adjacent control structures have been validated experimentally for the planar, two-fingered system used here. Furthermore, the validity of the assumption that the frequencies inherent in the trajectories were low enough to merit the steady-state approximation of the HCTL-1000 gain was confirmed.

We surmise that the poor tracking performance of the single level controllers was a result of neglecting the object's dynamics in the overall control scheme. The speed at which the single-level controllers needed to be operated placed strict limits on the amount of computation that could be completed within the control loop, thereby effectively excluding the possibility of moving the computation of the dynamics of the entire system into the lower level. Clearly, a scheme making use of the best of both controller complexity and controller speed was required.

There are many areas in the design, analysis, and testing of hierarchical control algorithms pertaining to grasping that need to be investigated more fully. A more complete discussion of these areas may be found in [3].

## 5 Conclusion

Based upon the experiments performed on Styx thus far, the most effective control scheme in fast movement of heavy objects is the hierarchical control scheme. Hierarchical control schemes have the advantage of being able to run simple, lower levels at high speeds, thus rapidly correcting for tracking errors in fast movements, while running at lower speeds more complicated higher levels that improve the overall performance by incorporating system dynamics far removed from the low level actuators. Although the experimental results presented here are based solely upon work done with a simple, planar system, the advantages of using a hierarchical control scheme can easily be applied to more complicated systems; in the implementation of the control structure, we made no use of simplifying assumptions based upon the simplicity of the system.

Our comparison of hierarchical and single-level control schemes has at once provided an indication of the advantages of using hierarchical control algorithms in grasping control and introduced a variety of open research questions relating to the theory and the application of hierarchical control in multi-fingered grasping situations.

## Acknowledgements

This research was supported in part by NSF under grant DMC-84-51129 and under grant ECS-87-19298 and by the U.S. Department of Health and Human

Services under grant PHS GM07379-14. In addition, the authors would like to thank Dr. S. Lehman, of the Bioengineering Graduate Group at the University of California at Berkeley, for many helpful discussions and comments relating to the work presented here.

## References

- [1] S. Arimoto, F. Miyazaki, and S. Kawamura. Cooperative motion control of multiple robot arms or fingers. In *IEEE Intl. Conf. on Robotics and Automation*, pages 1407-1412, 1987.
- [2] J. Demmel, G. Lafferrier, J Schwartz, and M. Sharir. Theoretical and experimental studies using a multifinger planar manipulator. *IEEE Intl. Conf. on Robotics and Automation*, pages 390-395, 1988.
- [3] K. Hollerbach. Hierarchical control in grasping and manipulation with a two-fingered planar hand. Master's thesis, University of California at Berkeley, 1991. (also available as ERL MEMO UCB/ERL M91/113).
- [4] S. Jacobsen, J. Wood, K. Bigger, and E. Iverson. The Utah/MIT hand: Work in progress. *Intl. J. of Robotics and Control*, 4(3):221-250, 1986.
- [5] D. Koditschek. Natural motion for robot arms. In *IEEE Control and Decision Conference*, pages 733-735, 1984.
- [6] P.V. Kokotovic, H. K. Khalil, and J. O'Reilly. *Singular Perturbation Methods in Control: Analysis and Design*. Academic Press, New York, 1986.
- [7] Z. Li, P. Hsu, and S. Sastry. On kinematics and control of multifingered hands. In *IEEE Intl. Conf. on Robotics and Automation*, pages 384-389, 1988.
- [8] M. T. Mason and Jr. J. K. Salisbury. *Robot Hands and the Mechanics of Manipulation*. The MIT Press, Cambridge, Massachusetts, 1985.
- [9] R. M. Murray, D. C. Deno, K. S. J. Pister, and S. S. Sastry. Control primitives for robot systems. *IEEE Trans. on Systems, Man and Cybernetics*, 1992. (to appear).
- [10] R. M. Murray and S. S. Sastry. Control experiments in planar manipulation and grasping. *IEEE Intl. Conf. on Robotics and Automation*, pages 624-629, 1989.
- [11] R. M. Murray and S. S. Sastry. Grasping and manipulation using multifingered robot hands. In R. W. Brockett, editor, *Robotics: Proc. of Symp. in Applied Mathematics, Volume 41*, pages 91-128. American Mathematical Society, 1990.
- [12] Y. Nakamura, K. Nagai, and T. Yoshikawa. Mechanics of coordinative manipulation of multiple robot mechanisms. In *IEEE Intl. Conf. on Robotics and Automation*, pages 991-998, 1987.
- [13] Van-Duc Nguyen. Constructing force-closure grasps. *Intl. J. of Robotics and Control*, 7(3):3-16, 1988.
- [14] S. T. Venkataraman and T. E. Djaferis. Multivariable feedback control of the JPL/Stanford hand. In *IEEE Intl. Conf. on Robotics and Automation*, pages 77-82, 1987.