

# Physically Based Grasping Control from Example

Nancy S. Pollard<sup>1†</sup> and Victor B. Zordan<sup>2‡</sup>

<sup>1</sup>School of Computer Science, Carnegie Mellon University

<sup>2</sup>Department of Computer Science, University of California Riverside

---

## Abstract

*Animated human characters in everyday scenarios must interact with the environment using their hands. Captured human motion can provide a database of realistic examples. However, examples involving contact are difficult to edit and retarget; realism can suffer when a grasp does not appear secure or when an apparent impact does not disturb the hand or the object. Physically based simulations can preserve plausibility through simulating interaction forces. However, such physical models must be driven by a controller, and creating effective controllers for new motion tasks remains a challenge. In this paper, we present a controller for physically based grasping that draws from motion capture data. Our controller explicitly includes passive and active components to uphold compliant yet controllable motion, and it adds compensation for movement of the arm and for gravity to make the behavior of passive and active components less dependent on the dynamics of arm motion. Given a set of motion capture grasp examples, our system solves for all but a small set of parameters for this controller automatically. We demonstrate results for tasks including grasping and two-hand interaction and show that a controller derived from a single motion capture example can be used to form grasps of different object geometries.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism: Animation

---

## 1. Introduction

Human dexterity is elegantly expressed through the use of our hands. However, dexterous behaviors such as grasping and manipulation are difficult to convey in animated human characters. While research on grasping has separately explored natural coordination patterns (e.g., [SFS02] [KCS03] [ES03]) and physically based control (e.g., [Ibe97] [BLTK93]), no system for grasping is yet available that exhibits the level of realism we see in motion capture data and also portrays physically plausible interactions between the hand and a grasped object.

This paper describes an approach which combines human motion data and physically based simulation with the goal of achieving compelling hand motion and generating convincing contact interactions. We have applied this algorithm to hand motions that involve sustained contact (Figure 1)



**Figure 1:** A handshake generated by our system. A grasp controller sequences the approach, grasp, release, and retreat. Joint limits and desired states for the controller are extracted automatically from motion data. Results reflect properties of the original motion and also display realistic physical interactions.

---

<sup>†</sup> nsp@cs.cmu.edu

<sup>‡</sup> vbz@cs.ucr.edu

and demonstrate examples of passive response, grasping, and two-hand interaction that were created using our technique. Our forward simulation acts based on a specialized controller which is derived from parameters almost exclusively extracted from motion capture data and which makes use of inverse dynamics as an *internal model* to compensate for torques produced in the hand due to motion of the arm and due to gravity.

Our emphasis in this paper is to allow a physically based hand to move in a fashion similar to a motion capture driven hand. In addition, when our physically based hand finds itself in a different environment or subject to unexpected disturbances, it should uphold the same high-quality movement. To meet these goals, we propose that an important aspect of human hand motion comes from its duality as both acting actively and passively - at all times. This dual nature gives the hand its compliance and other identifying qualities. Thus, an important aspect of this approach is the proper extraction and use of control parameters from motion capture examples, in particular to: capture the passive effects of the hand in a single neutral *setpoint* (or desired state); extract joint limits to keep the hand within viable bounds; define active setpoints that allow a simple state machine to control grasping; and, through these setpoints create a simple means for controlling the overall strength of a grasp. We show that a set of simple controls can be layered together to include each of these components in turn and that they allow us to generalize across different object geometries, even from a single motion example.

The contributions of this paper are to

- demonstrate results for grasping and interaction that combine realistic motion and physically plausible contact,
- present a technique for extracting passive and active parameters as well as joint limits from motion data,
- show that a simple control scheme with few parameters generates plausible responses to disturbances and generalizes to different object geometries,
- note that inverse dynamics compensation for arm motion and for gravity is important for generating pleasing motion with few setpoints.

## 2. Background

Research on grasping in computer graphics has focused in part on kinematic systems that select appropriate poses for the hand to grasp an object [AN99] [HBMT95] [RG91], and there has been a large amount of research in robotics to position contact points optimally on an object surface (see [Bic00] for an overview). Determination of hand poses for playing musical instruments has also been considered [KCM00] [ES03]. While these systems can create convincing hand postures or sequences of hand postures, they ignore the subtle physical interactions that occur as the hand makes contact with an object. Some recent research has focused on creating realistic physical models of the hand that are suitable for simulation (e.g., [AHS03]), but this work does not address the problem of controlling the hand to achieve specific task goals. Researchers in graphics and robotics have

developed controllers that allow the hand to dynamically conform to object shape (e.g., [Ibe97] [MT94] [BLTK93]). However, manual controller design for a high degree of freedom system such as the human hand remains a challenge. Our system extracts many of its parameters directly from motion data so that the grasping motion generated by the controller closely resembles human examples.

We take inspiration from controllers developed for dynamic simulations of full-body motion (e.g., [HWBO95], [LvF96], [FvdPT01]). Controller parameters have been learned in situations with a clear objective function such as distance traveled (e.g., [vF93], [Sim94], [GT95]). However, for grasping we expect that the objective function is less clear and that more guidance from motion data may be required to mimic the human characteristics of this behavior.

A number of researchers have explored systems that combine motion capture and simulation. In robotics, Kang and Ikeuchi [KI97] classify the type of a human grasp and then map that grasp to a robot hand in a procedural manner. In graphics, Borst and Indugula [BI05] use a forward simulation with proportional-derivative feedback control to track real-time motion capture data in order to display the user's hand interacting with objects in a virtual environment. For tasks other than grasping, Shapiro, Pighin, and Faloutsos [SPF03] show how hand designed controllers and motion capture playback can be combined by switching between simulation and playback modes when appropriate. Zordan and Hodgins [ZH02] propose a controller that tracks motion capture data and combine it with passive simulation for reactions in tasks such as boxing, and Yin, Cline, and Pai [YCP03] show that stiffness can be separated from quality of tracking by adding a feedforward term to the control equation. And Playter [Pla00] presents results for motion tracking combined with behavior based control for simulated human running. Our work differs in extracting a compact controller from motion data and also accommodating situations with sustained contact. Our goal is not to track the motion data, but to find a reduced representation that can replicate that data, with the belief that such a form will better support interpolation, extrapolation, plausible behavior in unexpected scenarios, and user control.

It may be desirable to create a controller which is motivated from actual human control, and researchers in computational neuroscience have considered a variety of models for human motor control. An ongoing controversy positions the equilibrium point hypothesis against the role of internal models (e.g., [HM03]). The equilibrium point hypothesis suggests that the human system coordinates movement by establishing a trajectory of equilibrium points. Differences between the current system state and equilibrium state result in forces driving the system, and smooth motion results from the system's natural dynamics as it heads toward equilibrium. In early work in this area, Bizzi and Polit [BP78] hypothesized that the target in a reaching task is encoded as a muscle activation "setpoint", and Flash [Fla87] found that relatively simple equilibrium point trajectories could explain observed reaching motions. In contrast, a representation of

control using internal models assumes that people have an internal model of their system dynamics and use this model to “compute” muscle activations required to control movement. This view is supported, for example, by experiments testing motor learning and control in environments with new dynamics (e.g., artificial force fields) [Kaw99]. Some researchers suggest that the two models are not incompatible (e.g., [FOL\*98]), and in our technique, we use a combination of the two approaches.

### 3. Control Overview

Our goal is a physically based simulation system for the hand that produces motion of quality comparable to motion capture data. The simulation infrastructure includes mechanisms for handling collision and contact and for maintaining the state of the physical system over time; our implementation of this system is reviewed in Section 7. The primary contribution of our paper, however, is the control algorithm that supplies joint torques to drive the hand during each timestep of the simulation.

To control a physically based hand for grasping, we explicitly include components for passive movement ( $\tau_P$ ) and active movement ( $\tau_A$ ) stemming from our belief that both must be present to create believable, compliant motion when the hand simulation is put under new conditions. We add dynamics compensation for arm motion ( $\tau_{ARM}$ ) and for gravity ( $\tau_G$ ), and assemble these components in the following equation:

$$\tau = \tau_P + \tau_A + \tau_{ARM} + \tau_G \quad (1)$$

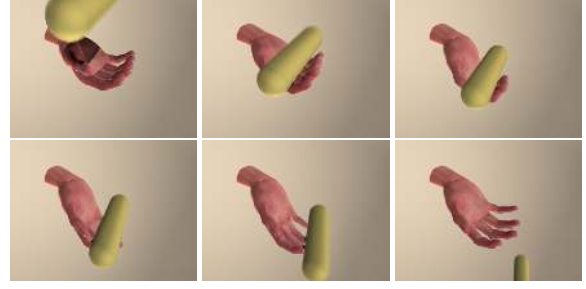
Parameters  $\tau_{ARM}$  and  $\tau_G$  are dynamics compensation terms, and their goal is to separate the intentional motion of the hand from secondary effects due to arm motion and gravity. The separation of arm motion from shaping of the hand for the grasp is supported by research on human grasping (e.g., see [MI94]), and the use of internal dynamic models is supported by research on human reaching [Kaw99].

Parameters  $\tau_{ARM}$  and  $\tau_G$  are computed by solving for joint torques that would be required in the hand if velocities and accelerations in the palm and fingers were zero. More specifically, assume we write the dynamics equation for the system as follows

$$\tau_P + \tau_A + \tau_{ARM} + \tau_G + J^T f = I\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \quad (2)$$

where  $J^T f$  represents torques due to external forces,  $I$  the inertia of the system,  $V(\theta, \dot{\theta})$  velocity product terms, and  $G(\theta)$  the effects of gravity. In terms of this equation,  $\tau_G$  is set to exactly cancel  $G(\theta)$ , and  $\tau_{ARM}$  is set to cancel components of  $I\ddot{\theta} + V(\theta, \dot{\theta})$  that depend only on the motion of the arm (i.e., the arm acceleration term and any arm velocity product terms).

The dynamics compensation terms allow  $\tau_P$  and  $\tau_A$  to represent passive and active joint torques specifically relevant to grasping. The next sections describe how  $\tau_P$  and  $\tau_A$  are calculated from motion capture examples, and how we provide



**Figure 2:** We performed “drop” tests to empirically select gains for the passive controller. This figure shows response of the passive controller during one such drop test.

functionality for adjusting grasp forces within this control scheme.

### 4. Passive Hand Control ( $\tau_P$ )

Much of the human hand’s signature movement comes from its passive characteristics, both derived from its tendency toward a comfortable, neutral pose which it will return to when other excitations are not present and from the interplay of joint limits - as a limit is met a joint up the chain often provides additional passive “give” to extend the range of motion. We combine these two components to create a hand which acts passively in a fashion similar to a human hand. That is, without explicit internal actuation, we anticipate that our passive controller will yield a simulation which shows bias toward a natural equilibrium or neutral point,  $\theta_{NEUTRAL}$ , and will obey reasonable joint limits,  $\theta_{LIMIT}$ , for each degree of freedom. As such, the equation that we use for passive control is

$$\tau_P = I (k_S(\theta_{NEUTRAL} - \theta) + k_{JL}(\theta_{LIMIT} - \theta) - k_D\dot{\theta}) \quad (3)$$

Here  $\theta$  and  $\dot{\theta}$  are the current joint angle and joint angular velocity values for the system in axis-angle format. Parameter  $k_S$  is the stiffness used to drive the system toward the neutral pose,  $\theta_{NEUTRAL}$ , and parameter  $k_D$  is the damping constant. Stiffness term,  $k_{JL}$ , is used to drive the system back into the legal range of joint angles, and is only nonzero when a joint is outside its range. Joint limit  $\theta_{LIMIT}$  represents the currently active limits for any degrees of freedom that are outside their ranges. Parameter  $I$  is the inertia matrix of the bodies effected by the joint (moving from the wrist outward). Note that  $I$  depends on joint configuration and must be recomputed every timestep.

Of the parameters required for Equation 3, two are extracted from the motion data. Setpoint  $\theta_{NEUTRAL}$  is the mean pose in our dataset. Joint limits  $\theta_{LIMIT}$  are extremes observed in the motion capture library for each degree of freedom. These limits include extreme poses observed while grasping and during active exploration of range of motion. Our database includes motions where the actor is asked to move the hand and fingers to exercise each degree of free-

dom to its full extent without the assistance of external forces from the environment.

Three scalar parameters must then be set by the animator to fully define the passive controller:  $k_S$ ,  $k_{JL}$ , and  $k_D$ . These parameters determine the compliance of the hand, both in free motion and as it approaches a joint limit. We make these parameters the same at every joint so that the compliance throughout the hand is uniform. More specifically, consider Equation 2. From this equation we see that applying passive torque  $\tau_P$  results in a change in acceleration:  $\tau_P = I\Delta\ddot{\theta}$ . Comparing to Equation 3 we see that:

$$\Delta\ddot{\theta} = k_S(\theta_{NEUTRAL} - \theta) + k_{JL}(\theta_{LIMIT} - \theta) - k_D\dot{\theta} \quad (4)$$

In other words, given constant  $k_S$ ,  $k_{JL}$ , and  $k_D$ , the same values for  $(\theta_{NEUTRAL} - \theta)$ ,  $(\theta_{LIMIT} - \theta)$ , and  $\dot{\theta}$  will produce the same change in acceleration  $\ddot{\theta}$  at every joint.

We set parameters  $k_S$ ,  $k_{JL}$ , and  $k_D$  based on simple “drop” tests such as that shown in Figure 2. Specifically, we chose these parameters through trial and error by running similar drop tests a number of times until we obtained visually pleasing results consistent with our observations of human hand behavior. Given the right set of passive response experiments, we believe these parameters could be determined automatically, but they were not difficult for us to set. Once the values were tuned, we kept them fixed throughout our other experiments, with some small adjustments for the handshake (Table 1).

## 5. Active Control ( $\tau_A$ )

To activate the hand for a grasping behavior, we include active torque,  $\tau_A$ , controlled via the simple finite state machine (FSM) shown in Figure 3. This state machine is mostly independent of time and instead relies on the distance from an object to trigger different actions leading up to and following a grasp. The states of the FSM were selected based on observations of the motion of several grasp examples and we believe will generalize to many other grasp activities.

The equation for the active torque is

$$\tau_A = I(k_S(\theta_{DES} - \theta) + k_D(\dot{\theta}_{DES} - \dot{\theta})) \quad (5)$$

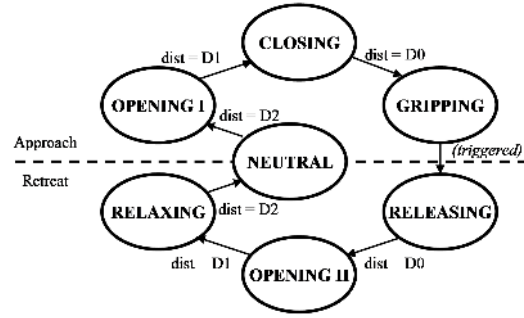
where the stiffness and damping parameters are identical to those used in the passive controller and  $I$  is the inertia matrix for the outboard bodies as in Equation 3. Parameters  $\theta_{DES}$  and  $\dot{\theta}_{DES}$  are desired values for joint angles and joint angular velocities toward which the hand will be driven. They depend on the current state of the FSM and on distance from hand to object. In particular, for each state,  $\theta_{DES}$  and  $\dot{\theta}_{DES}$  will be a blend between setpoints.

$$\theta_{DES} = BP_S \quad (6)$$

$$\dot{\theta}_{DES} = BP_D \quad (7)$$

where  $B$  is the blend function and  $P_S$  and  $P_D$  are the position and velocity elements of the setpoints.

We have six setpoints  $(\theta_i, \dot{\theta}_i)$ , which are expressed for al-



<b>NEUTRAL:</b>	<b>OPENING I:</b>	<b>CLOSING:</b>	<b>GRIPPING:</b>
$B[0] = 1$	$B[0] = 1 - B[1]$	$B[1] = 1 - B[2]$	$B[2] = 1$
	$B[1] = \frac{D2 - dist}{D2 - D1}$	$B[2] = \frac{D1 - dist}{D1 - D0}$	
<b>NEUTRAL:</b>	<b>RELAXING:</b>	<b>OPENING II:</b>	<b>RELEASING:</b>
$B[5] = 1$	$B[5] = 1 - B[4]$	$B[4] = 1 - B[3]$	$B[3] = 1$
	$B[4] = \frac{D2 - dist}{D2 - D1}$	$B[3] = \frac{D1 - dist}{D1 - D0}$	

Figure 3: Finite state machine and active setpoints.

gebraic manipulation as

$$P_S = [\theta_0 \ \theta_1 \ \dots \ \theta_5]^T \quad (8)$$

$$P_D = [\dot{\theta}_0 \ \dot{\theta}_1 \ \dots \ \dot{\theta}_5]^T \quad (9)$$

Blend function  $B$  is dependent on state and distance from hand to object. The blend functions are shown in Figure 3. For example, in the NEUTRAL state,

$$B = [1 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (10)$$

In the CLOSING state,

$$B = [0 \ (1-a) \ a \ 0 \ 0 \ 0] \quad (11)$$

with

$$a = \frac{D1 - dist}{D1 - D0} \quad (12)$$

In other words, during the CLOSING state we linearly blend from setpoint  $(\theta_1, \dot{\theta}_1)$  to setpoint  $(\theta_2, \dot{\theta}_2)$  as the distance decreases from  $D1$  to  $D0$ . When the distance reaches  $D0$ , the state transitions to GRIPPING.

### 5.1. Extracting Active Setpoints from Motion Capture

To create an active controller that results in natural looking motion, we use the simple control model proposed here and fit the parameters of that model using data from our motion library. Assume we have a single grasping trajectory we are trying to match. This grasping trajectory gives us the data required to solve for setpoints  $P_S$  and  $P_D$ . Combining Equations 1 and 5 leads to the following expression.

$$k_S(\theta_{DES} - \theta) + k_D(\dot{\theta}_{DES} - \dot{\theta}) = I^{-1}(\tau - \tau_{ARM} - \tau_G - \tau_P) \quad (13)$$

which can be rewritten as

$$k_S B P_S + k_D B P_D = k_S \theta + k_D \dot{\theta} + I^{-1} (\tau - \tau_{ARM} - \tau_G - \tau_P) \quad (14)$$

where all parameters except  $P_S$  and  $P_D$  are available or can be computed from the motion capture example. Each frame of the example grasp gives us one such equation, and we combine these equations into a single large linear system that can be solved for  $P_S$  and  $P_D$  using least squares.

Note that to make Equation 13 linear in the unknowns, we are suggesting that we can take the difference of two rotations by subtracting their vector form representations. In our system, we model all joints as ball joints and operate on rotations expressed in axis-angle format. While this is a very poor approximation in general, we found that it worked well for extracting setpoints from the motion data. We believe the reason this works in our case is that the joint motions required for grasping are mostly rotations about a single axis or an axis that changes slowly throughout the movement.

## 6. Grip Strength

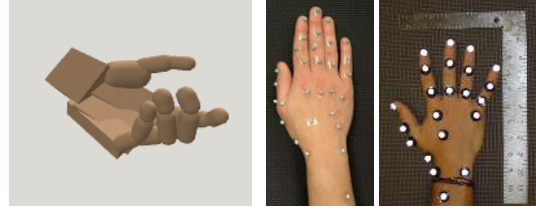
In situations of sustained contact, there is one difficulty with Equation 13—the actual joint torques  $\tau$  cannot be determined without some knowledge of contact forces, and these contact forces are generally not available from motion capture data. Without a representation of actual torques during contact, our setpoints may lead to a hand which is too “weak” to support the object in the grasp.

Because actual contact forces are not known, rather than attempting to account for them explicitly, we take advantage of the fact that careful force control is not needed to create the visual appearance of grasps and interactions. Instead, we rely on our synthetic grasps to create their own force balance as contacts are made and as the object is pulled into the hand. Our solution, then, is based on the intuition that attempting to close the hand further will result in greater force applied to the object. To implement this idea, we assume that the setpoint for a firm grasp can be determined by extrapolating the motion between an open pose and a closed pose. The extent of the extrapolation is adjusted by setting a single parameter  $\alpha$  to get the desired appearance for the grasp. Control of this parameter  $\alpha$  can then be left to the animator.

Note, from Figure 3, without the grip force parameter, only Setpoint 2 is active, i.e.,  $B[2] = 1$  during the GRIPPING state. When the grip force parameter  $\alpha$  is incorporated, then instead of simply using setpoint  $(\theta_2, \dot{\theta}_2)$  during the GRIPPING state, we add to this setpoint the value  $\alpha(\theta_2 - \theta_1, \dot{\theta}_2 - \dot{\theta}_1)$ . The resulting setpoint is equivalent to closing the hand farther along the joint space line represented by the closing portion of the motion. Blend functions for the GRIPPING state then become:

$$B[2] = 1 + \alpha \quad (15)$$

$$B[1] = -\alpha \quad (16)$$



**Figure 4:** (Left) Rigid-body articulation for the physical simulation. These primitive models were used in self- and object collision detection and reactions. (Center/Right) Marker sets used for hand motion capture for one- and two-hand examples, respectively.

## 7. Implementation and Results

Our physical model of the hand, seen in Figure 4, has 19 ball joints, for 57 total degrees of freedom. We use ODE, the Open Dynamics Engine ([www.ode.org](http://www.ode.org)), to simulate motion of the hand and use the simplified model of the hand geometry shown in Figure 4 for the physics simulation. Link lengths, masses, and inertias for the hand model come from measurements of our human actor and assumptions about average density.

The ODE simulation engine takes care of detecting collisions, creating and breaking contact, and adding contact forces to the hand during the simulation. Contact is handled by adding a special type of “contact constraint,” and the user has control over parameters such as coefficient of friction. The ODE simulation is a rigid-body approximation of the hand, which is potentially problematic given that the human hand is quite compliant. However, two things allow us to get away with this approximation: the stiffness at the joints is low, providing the hand with a great deal of “give” in response to external forces, and ODE’s contact constraints can be made soft, effectively adding compliance to the system at the points of contact. The simulation’s computation time varies depending on the number of contacts present at a given simulation time; we recorded computation speeds from 5 to 20 fps on a 3.2 Ghz Athlon processor with the simulation timestep for the examples set at 0.2 ms.

Our hand motion capture library contains cylinder grasps, a range-of-motion test where the actor attempts to exercise all degrees of freedom, and two-hand interactions including handshakes. Hand motions were captured using a Vicon optical motion capture system. The Vicon system gives us 3D positions of the markers over time and, to obtain joint data, we adapted the technique from Zordan and van der Horst [ZH03]. For our examples we used two marker sets with 22 and 30 markers shown in Figure 4.

All of our experiments used the FSM and blend functions shown in Figure 3. To keep the hand from contacting the object too quickly, setpoint  $B[1]$  was set to 1 during the closing state of the grasps. The control parameters used appear in Table 1. We set distance parameters based on visual inspection of the motions, but optimization could be used to adjust

Parameter	$k_S$	$k_D$	$k_{JL}$	D1	D2
Units	$1/s^2$	$1/s$	$1/s^2$	m	m
Grasps	200	50	5000	0.05	0.3
Handshake	300	60	5000	0.09	0.2

**Table 1:** Parameter values used in our experiments.  $D0$  is defined as zero for all cases.



**Figure 5:** Frames from a handshake sequence generated by our system. The poses for the two hands are different. Each hand is running its own controller, with setpoints blended based on distance between the hands.

these parameters automatically for the best fit to the motion data.

The videos illustrate the effects of components of our system. Joint limits are important for creating plausible response to disturbances, because the operating stiffness of our hand is quite low. The supplementary video shows a comparison of the drop test with and without joint limits. Inverse dynamics compensation is important for reducing finger lag due to arm motion, especially during a situation such as the handshake, where the setpoint is constant, but the arm motion is highly dynamic. The supplementary video shows the handshake with and without the use of parameter  $\tau_{ARM}$ . Without inverse dynamics compensation, waving of the fingers is visible as the arm accelerates and decelerates. When inverse dynamics compensation is enabled, much of the extra finger motion is eliminated. Grasp force parameter  $\alpha$  is surprisingly effective at conveying different levels of strength for a grasp. The main video shows a comparison of the same grasp executed at varying levels of  $\alpha$  (varying from 0.0 to 0.5).

Two-hand interaction and other grasping results appear in Figures 1, 5, and 6. We tested the flexibility of our controller by using setpoints derived from a single motion cap-

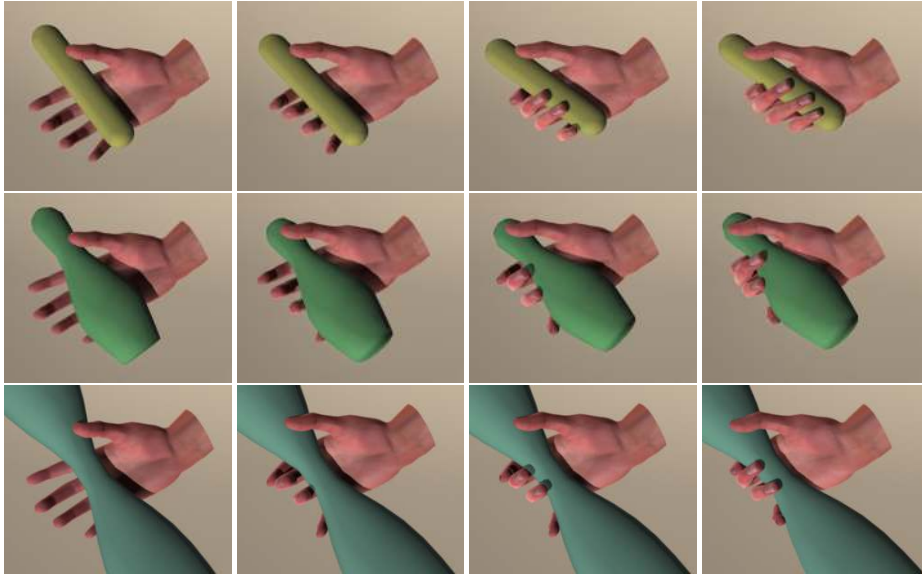
ture example to grasp objects of different geometries. Figure 6 shows some examples, and additional grasps are shown in the main video. Figures 1 and 5 show a handshake generated by our system. In this experiment we raised the stiffness and damping to allow the hands to respond more reasonably to forces from the other hand (Table 1). The size of one of the hands differs significantly from that of the actor from which the motion data was collected. However, our system is able to obtain a secure grasp for the handshake despite this difference from the original motion capture scenario.

## 8. Discussion

The primary contribution of this paper is a technique for creating controllers for grasping and two-hand interaction that produce hand motion with the richness of motion capture data. We proposed that an important aspect of human hand motion comes from how it combines passive and active control, and we showed how a layered set of controllers could be created to obtain good response to external disturbances, create motion similar to motion capture data, demonstrate plausible compliance when acquiring the grasp, and provide some control over grasp forces.

Setpoint control with torque controllers at the joints made it easy for us to separate passive response from active control of the hand, but it was useful for other reasons as well. First, this control scheme behaves in a consistent way across the contact boundary. We do not have to change parameters or switch control modes when transitioning from free space motion to motion with contact. The control algorithm is exactly the same before, during, and after contact and behaves well in situations with sustained contact. Second, setpoint controllers have the advantage of simplicity. We have very few setpoints (just six), and they have a meaningful interpretation (neutral, open, close, grasp, release, relax, and return to neutral). Having a small number of setpoints is useful because the setpoints can result in meaningful hand poses, which may be (re-)used in simple ways. As an illustration, Figure 7 shows an example where the user interactively switches between open and close setpoints to move the object within the grasp. We believe setpoint control will make it easier to obtain high quality results for re-parameterization to grasp new objects and to adapt timing and distance to new scenarios. There are alternative controller forms that could be selected – tracking control as in Zordan and Hodgins [ZH02] or Yin, Cline, and Pai [YCP03], or explicit force control for the duration of the grasp. However, similar simplicity and consistency across the contact boundary would be more difficult to achieve using a tracking or mixed position/force control approach.

To create pleasing motion with very few setpoints, we found that inverse dynamics compensation for arm motion was important. In addition to eliminating secondary motion of the fingers, inverse dynamics compensation has the advantage of separating the effects of arm and hand motion, which allows us to run our controllers with arm motion from sources other than the original motion capture data. There are other ways we could address the problem of arm motion



**Figure 6:** Grasps obtained from a single motion capture example.



**Figure 7:** Simple interactive object manipulation can be performed by giving the animator direct control over setpoint sequencing. Here, the animator is choosing the timing between open and close setpoints derived from the motion data.

influencing hand dynamics, such as tracking a dense representation of the motion data with high stiffness. However, stiffness has been set to obtain desired passive response of the hand, and we do not want to change that.

We also found that it was important to provide some intuitive control over apparent grip strength. To do this, we chose to modify the grip setpoint while leaving the compliance of the hand fixed, which kept our controller simple and predictable. There is some evidence that this assumption is not biologically correct and that stiffness in the hand may increase with increasing force (e.g., [HH97]), although the situation may be more complex during grasping [MF98]. Understanding how/whether stiffness must be controlled to create humanlike responses to disturbance forces from a grasped object is one topic of future work.

One advantage of our approach is to allow motion capture data to be generalized to new skeletons (Figure 5) and new object geometries (Figure 6) while maintaining the physically plausible appearance of contact events and finger positions. Our algorithm is limited, however, to generating grasps that are similar in character to the motion capture examples from which they were derived. Our controller has no knowledge of the intended task and is not able to orient the hand or adjust the fingers to obtain a “better” grasp.

In addition, our algorithm for modulating between softer and firmer grasps does not explicitly balance contact forces. It works because the grasps we explored were whole-hand or enveloping grasps, where simply closing the hand more tightly is sufficient to pull the object into the grasp and generate greater force on the object. For these types of grasps, forces are balanced automatically as the hand settles into its equilibrium position.

For future work, we are interested in extending the controller to better capture the natural coupling between degrees of freedom in the hand, incorporating compliance of the skin and tissue, and exploring automatic parameterization of controllers based on measures such as object size. We would also like to develop a similar controller for arm motion that is derived from motion data and serves the hand appropriately to align it with the object to be grasped.

**Acknowledgments** We would like to thank: Moshe Mahler for help with the video and rendering; Jiaxin Fu, Ying Li, Jose Medina, and Ting Ting Wu for help in processing the motion capture data; and Adriano Macchietto for software support. We would also like to acknowledge NSF grants CCF-0343161, IIS-0326322, ECS-0325383, CNS-0224419, and CNS-0423546.

## References

- [AHS03] ALBRECHT I., HABER J., SEIDEL H.-P.: Construction and animation of anatomically based human hand models. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2003). 2
- [AN99] AYDIN Y., NAKAJIMA M.: Database guided computer animation of human grasping using forward and inverse kinematics. *Computers and Graphics* 23 (1999), 145–154. 2
- [BI05] BORST C. W., INDUGULA A. P.: Realistic virtual grasping. In *IEEE Virtual Reality* (2005). 2
- [Bic00] BICCHI A.: Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on Robotics and Automation* 16, 6 (2000), 652–662. 2
- [BLTK93] BEKEY G. A., LIU H., TOMOVIC R., KARPLUS W. J.: Knowledge-based control of grasping in robot hands using heuristics from human motor skills. *IEEE Transactions on Robotics and Automation* 9, 6 (1993), 709–722. 1, 2
- [BP78] BIZZI E., POLIT A.: Processes controlling arm movements in monkeys. *Science* 201 (1978), 1235–1237. 2
- [ES03] ELKOURA G., SINGH K.: Handrix: Animating the human hand. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2003). 1, 2
- [Fla87] FLASH T.: The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics* 57, 4-5 (1987), 257–274. 2
- [FOL\*98] FELDMAN A. G., OSTRY D. J., LEVIN M. F., GRIBBLE P. L., MITNITSKI A. B.: Recent tests of the equilibrium-point hypothesis (lambda model). *Motor Control* 2 (1998), 189–205. 3
- [FvdPT01] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable controllers for physics-based character animation. In *SIGGRAPH 01 Proceedings* (2001). 2
- [GK96] GOMI H., KAWATO M.: Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science* 272 (1996), 117–120.
- [GT95] GRZESZCZUK R., TERZOPOULOS D.: Automated learning of muscle-actuated locomotion through control abstraction. In *SIGGRAPH 95 Proceedings* (1995), pp. 63–70. 2
- [HBMT95] HUANG Z., BOULIC R., MAGNENAT THALMANN N., THALMANN D.: A multi-sensor approach for grasping and 3D interaction. In *Proc. Computer Graphics International* (1995). 2
- [HH97] HAJIAN A. Z., HOWE R. D.: Identification of the mechanical impedance at the human finger tip. *Journal of Biomechanical Engineering* 119 (1997), 109–114. 7
- [HM03] HINDER M. R., MILNER T. E.: The case of an internal dynamics model versus equilibrium point control in human movement. *Journal of Physiology* 549 (2003), 953–963. 2
- [HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *SIGGRAPH 95 Proceedings* (1995), pp. 71–78. 2
- [Ibe97] IBERALL T.: Human prehension and dextrous robot hands. *International Journal of Robotics Research* 16, 3 (June 1997). 1, 2
- [Kaw99] KAWATO M.: Internal models for motor control and trajectory planning. *Current Opinions in Neurobiology* 9 (1999), 718–727. 3
- [KCM00] KIM J., CORDIER F., MAGNENAT THALMANN N.: Neural network-based violinist's hand animation. In *Proc. Computer Graphics International* (2000). 2
- [KCS03] KAMPER D. G., CRUZ E. G., SIEGEL M. P.: Stereotypical fingertip trajectories during grasp. *Journal of Neurophysiology* 90 (2003), 3702–3710. 1
- [KI97] KANG S. B., IKEUCHI K.: Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps. *IEEE Transactions on Robotics and Automation* 13, 1 (1997), 81–95. 2
- [LvF96] LASZLO J., VAN DE PANNE M., FIUME E.: Limit cycle control and its application to the animation of balancing and walking. In *SIGGRAPH 96 Proceedings* (1996), pp. 155–162. 2
- [MF98] MILNER T. E., FRANKLIN D. W.: Characterization of multijoint finger stiffness: Dependence on finger posture and force direction. *IEEE Transactions on Biomedical Engineering* 45, 11 (1998), 1363–1375. 7
- [MI94] MACKENZIE C. L., IBERALL T.: *The Grasping Hand*. North-Holland, 1994. 3
- [MT94] MAS SANSO R., THALMANN D.: A hand control and automatic grasping system for synthetic actors. In *Eurographics '94* (1994). 2
- [Pla00] PLAYTER R.: Physics-based simulation of running using motion capture. In *Course notes for SIGGRAPH 2000* (2000). 2
- [RG91] RIJPKEMA H., GIRARD M.: Computer animation of knowledge-based human grasping. In *Computer Graphics (SIGGRAPH 91 Proceedings)* (1991), vol. 25, pp. 339–348. 2
- [SFS02] SANTELLO M., FLANDERS M., SOECHTING J. F.: Patterns of hand motion during grasping and the influence of sensory guidance. *Journal of Neuroscience* 22 (2002), 1426–1435. 1
- [Sim94] SIMS K.: Evolving virtual creatures. In *SIGGRAPH 94 Proceedings* (1994), pp. 15–22. 2
- [SPF03] SHAPIRO A., PIGHIN F., FALOUTSOS P.: Hybrid control for interactive character animation. In *Pacific Graphics* (2003), pp. 455–461. 2
- [vF93] VAN DE PANNE M., FIUME E.: Sensor-actuator networks. In *SIGGRAPH 93 Proceedings* (1993), pp. 335–342. 2
- [YCP03] YIN K., CLINE M. B., PAI D. K.: Motion perturbation based on simple neuromotor control models. In *Pacific Graphics* (2003). 2, 6
- [ZH02] ZORDAN V. B., HODGINS J. K.: Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2002). 2, 6
- [ZH03] ZORDAN V. B., HORST N. V. D.: Mapping optical motion capture data to skeletal motion using a physical model. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2003), pp. 140–145. 5