Full-Arm Haptics in an Accessibility Task

Matthew Frey[*]
Department of Mechanical Engineering
University of Utah

David E. Johnson[†] John Hollerbach[‡]
School of Computing
University of Utah

Abstract

This paper develops haptic rendering for the Sarcos Dextrous Teleoperation System, a full-arm force-feedback device, and demonstrates a virtual prototyping application where a user must access a particular location in a mechanical system and apply forces. A user study suggests that users receiving full-arm force-feedback were able to more quickly access the desired portion of the mechanical system and could more easily generate required forces. Thus, for this task, the full-arm haptics were shown to have better virtual prototyping predictive capabilities and better ergonomic factors for users.

1 Introduction

Historically, designers have used an iterative process of constructing and testing physical models to determine accessibility and ergonomics for assemblers or users of complex devices or systems. When considering large systems, such physical models may be prohibitively expensive or time-consuming to build. There has been progress in the engineering community to replace such physical prototypes with *virtual prototypes* [3], simulations that allow many aspects of a design to be tested on a computer directly from a CAD model. Such simulations are only as useful as the design aspects they allow to be tested, and the fidelity with which such testing can be done.

The addition of touch cues, or haptic feedback, to virtual prototyping simulations is considered an important improvement for general usability and especially for tasks that rely on touch, such as assembling a mechanism. One serious limitation of current virtual prototyping work with haptic displays is that most existing devices apply forces only to the hand or fingers of the user. When used to test questions such as "can a technician get into that space to remove or install this part?", there are two main issues with current haptic displays. The most apparent problem is that of false positives, where the simulation reports that a part is reachable because only the hand collides with the environment, allowing the rest of the arm to make contact. Even if other feedback modalities, such as sound or visual cues, are used to show arm contact, those cues may be difficult to interpret. A less appreciated factor from the use of hand-based haptic displays is that of false negatives, where a hand-based display may give the sense of being unable to apply strong forces at a certain location. However, a full-arm display could simulate the full ergonomic scenario, where a user can brace a joint against part of the environment, thereby gaining additional leverage.

This paper develops a virtual prototyping environment for a full-arm haptic display, the Sarcos Dextrous Teleoperation System (DTS) force reflecting master [10] (see Figure 1). A haptic rendering system appropriate for full-arm force rendering is linked with a control scheme for full-arm haptic devices and used in a simple virtual prototyping application where users are asked to reach a location in a model and apply forces. A user study compares the benefit of full-arm haptics versus forces only applied to a hand for that application. The results demonstrate the utility of full-arm force-feedback in this virtual prototyping application.

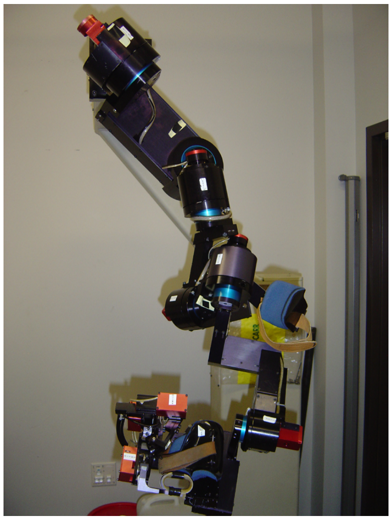

Figure 1: The Sarcos DTS Master Exoskeleton used for full-arm haptic rendering

2 Background

CAD packages have allowed designers to graphically explore designs. Although useful, visual cues alone can not sufficiently answer the questions asked by human factors engineers regarding the feasibility of human interaction with a given design or within a given environment. In support of these questions, virtual manikins were introduced [14] and have since become very useful in testing designs for physical and visual accessibility as well as ease of assembly and disassembly in a given environment [12]. However, though virtual manikins can answer questions about possible body motion and posture, they are unable to answer questions about comfort or the forces humans can exert and how those forces depend on posture [5]. To develop accurate force models for humans would require accurate biomechanical models of human muscles, muscle routing, as well as coordination and control. Research is being done in this area, but is not currently sufficient to develop necessary human force models for virtual prototyping.

A step beyond exploring virtual environments visually or with virtual manikins is to introduce a sense of touch, via haptics, to the human-computer interface. Coupled with visual and auditory cues, haptically rendered forces can greatly enhance a person's spatial awareness in a virtual environment, which is especially true in confined spaces or in environments where some objects occlude others [16]. There are several research examples where haptic devices have been used to manipulate models of components or even human limbs in a virtual environment to aid in the design process.

Over the past decade there has been significant effort to incor-

[*]e-mail: freym@cs.utah.edu
[†]e-mail:dejohnso@cs.utah.edu
[‡]e-mail:jmh@cs.utah.edu

Symposium on Haptic Interfaces for Virtual
Environments and Teleoperator Systems 2008
13-14 March, Reno, Nevada, USA
978-1-4244-2005-6/08/$25.00 ©2008 IEEE

porate haptic displays in virtual prototyping systems. Fisher and Vance [4] used a three degree of freedom (DOF) haptic device, the SensAble PHANToM, to allow humans to manipulate and assemble CAD models in a virtual environment. They concluded that haptic feedback makes manipulating complex parts through confined spaces much faster and more intuitive than the mouse and keyboard alone. It also gave users additional understanding about the geometry of the part, even when the part was visible. In this study, participants controlled the part which interacted with the virtual environment. There was no direct interaction between the participants hand or arm and the virtual world.

Using a haptic feedback device similar to the PHANToM called the Large Haptic Interface for Aeronautic Maintainability (LHIfAM), Borro et al [2] developed a haptic system, called Revima, to conduct accessibility, interference and maintainability analysis on aircraft. With this system, they were able to assemble an aircraft engine strictly in a virtual world. ITP, a supplier of low pressure turbines for Rolls-Royce who uses Revima, claims a 30 percent reduction in model inspection time. Engineers at ITP are able to gain a greater understanding of complex environments when compared to conventional visualization tools. However, even with a state-of-the-art haptic device like LHIfAM, information about the location and orientation of the user's arm is completely absent.

The Boeing Company has also been involved in developing ways for humans to more effectively interact with virtual environments using haptic feedback [16, 11]. The aim of their research has been to improve on traditional graphical techniques, such as the virtual manikin, to interact with virtual environments. Using haptic displays they were able to integrate a model of the human arm into a virtual environment to quickly and intuitively determine the ability of a human arm to reach desired positions. To move the arm, a user would effectively hold the wrist of the virtual arm. This type of system gives additional information about posture constraints imposed by human arm interaction with the environment. However, the system is again limited, because the user can only control and feel interaction forces and torques at one location on the virtual arm. Also, because the orientation of the user's arm does not necessarily match the orientation of the virtual arm, it is very difficult to draw accurate conclusions about force application or comfort.

Researchers at the University of Utah have also explored haptic feedback as a way to more effectively evaluate and interact with virtual models. To make interactions more realistic they have employed the Sarcos force reflecting master [10]. This haptic display allows users to interact more naturally feeling external forces from shoulder to fingertip as well as internal (grasping) contact forces. To date their research has focused on using only the hand to feel and manipulate virtual models.

2.1 Haptic Rendering Approaches

Fast collision detection between environment models and arm link models is critical in generating stiff and stable haptic forces. It is commonly accepted that the haptic control loop must run at around 1 kHz in order to maintain stability, yet it is difficult to detect model penetrations at these rates for all but the simplest environments. Recently, there have been significant advances in computing speeds, but there has also been significant research in the area of fast model-model haptic rendering.

One approach used by McNeely et al. [11] is to discretize the dynamic objects into surface points and associated normals while the static objects are represented by a spatial occupancy map called a voxmap. As the points on the dynamic object penetrate the voxmap of the environment, proportional restoring forces are created parallel to the normal on the dynamic objects surface. Another approach developed by Otaduy and Lin [13] uses a multiresolution hierarchy to reduce the number of polygons that are tested. This simplification approach increases performance without noticeably reducing haptic perception. Johnson and Willemsen [7, 8, 9] introduced a method where local minimum distances between polygonal models are generated through descent of spatialized normal cone hierarchies (SNCH). These local minima are then fed to a very quick local update algorithm that updates the position of the local minima at haptic rates. By connecting the surface points at a local minimum with virtual springs, forces are generated to keep models apart.

2.2 Research Goals and Approach

Up to this point users have been limited by haptic displays that only reflect forces to their hand or by haptic rendering algorithms that could not render the forces between the arm and environment. In contrast, a person using the Sarcos arm can very naturally move and complete tasks in a virtual environment, while feeling nearly the same forces in the same positions that a person would feel when interacting with a physical model. The goal of this paper is to develop methods for full-arm haptic rendering and to test, using the Sarcos Dextrous Arm Master, whether full-arm haptic interaction with virtual prototyping environments provides a benefit over current haptic displays.

3 Experimental Setup

To investigate if there is a benefit to using full-arm haptics, a study was conducted where participants were exposed to several environments where they were asked to complete a simple task by reaching a prescribed location and applying a prescribed force. For each environment the participant experienced two scenarios. In the first, only forces resulting from collisions between the hand model and the virtual environment were rendered to the participant. Users were asked to avoid collisions between their arm and the environment while completing a task. Collisions involving the upper arm and forearm were indicated by visual change in the arm model's color. In the second scenario, full-arm haptics was used to render collisions between the environment and the entire arm model while the user was completing the task.

3.1 Hardware and Computational Environment

The hardware setup consisted of five main components: two PC workstations, the Sarcos arm, signal conditioning and amplification boards, and a 19 inch CRT display as shown in Figure 2. The first computer generates the graphics and performs a global search for local distance minima between the arm and environment. This computer is networked to the second computer, which performs local updates to the minima at the haptic control rate of 1 kHz and converts forces generated by the minima into joint torques. This local update computer also controls the Sarcos arm's joint torques while receiving position and joint torque feedback from the manipulator's sensors through the signal conditioning boards.

3.1.1 Sarcos DTS Master

Full-arm haptic rendering is made possible by the DTS Master shown in Figure 1. The DTS was originally developed for remote dextrous manipulation using a linked robotic arm [6]. The master arm is a ten degree of freedom robotic mechanism. The first seven degrees mimic those found in the human arm from the shoulder to wrist. The remaining three degrees are used by the fingers and thumb on the master. Note that while a 6DOF haptic device can be used to constrain the position and orientation of the hand, such devices leave the elbow free to move. Thus, these more common devices cannot correctly simulate the interaction of the arm with an environment.

In addition to high resolution potentiometers measuring each joint position, there are also load cells positioned at each joint so joint torques can be accurately measured. With a workspace very close to that of the human arm, high resolution torque and position

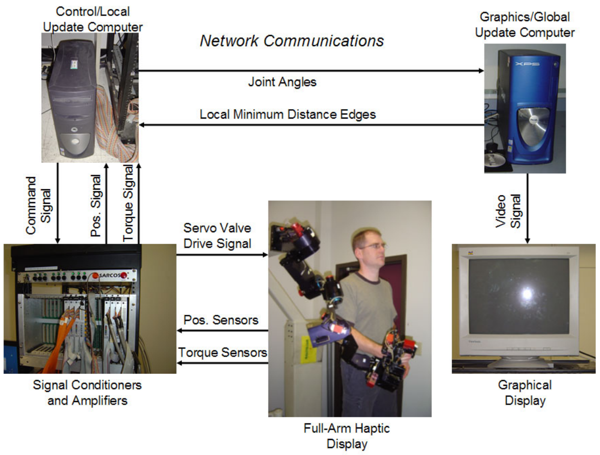

Figure 2: A depiction of the full-arm haptic rendering hardware setup.

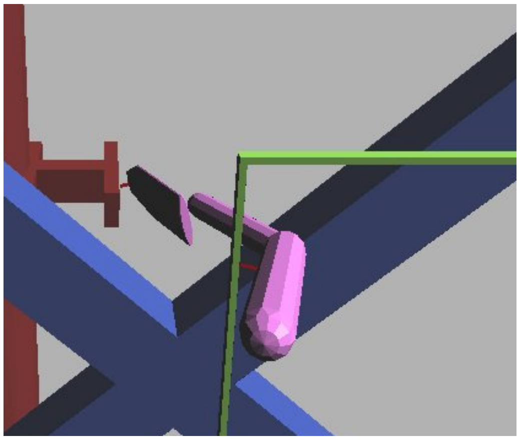

Figure 3: Local minima in distance create spring-like repulsive forces, shown here with red lines, between the arm linkages and the environment

sensors, and the potentially high stiffness provided by powerful hydraulic actuators, the Sarcos DTS Master remains a novel device for full-arm haptic rendering.

3.2 Haptic Rendering For Full-Arm Haptics

While haptic rendering of model-model contact is still considered technically challenging, scaling the rendering algorithms to track multiple contacts between a full-arm model and a reasonably complex environment is even more so. The approach used is to split the computation into time-critical and not critical components, and to distribute the computation over multiple platforms.

Conceptually, the collision or contact forces between arm and environment are characterized by spring-like repulsive forces generated when the distance between the arm and environmental objects falls below a specified small threshold (Figure 3). Coupled with gravitational forces, the contact forces are translated into joint torques through the kinematics of the arm model. These joint torques are then cascaded through all proximal joints back to the shoulder to provide haptic feedback across the entire arm. The motion between the arm model and the environment is assumed to be quasi-static or slow enough that dynamic forces can be ignored.

3.2.1 Haptic Rendering Algorithm Modifications

The normal cone hierarchy approach from [9] has been adapted for full-arm haptic rendering. Two aspects of that approach were useful for this project. First, it provides forces to resist penetration, rather than requiring penetration to generate forces. Second, the algorithm uses a global phase and local update phase and the local phase was relatively easy to port to the real-time system controlling the arm. Some alternative approaches are more monolithic in form. In the normal cone approach, each link of the arm model is tested in turn against all the models in the environment. The collection of local minima between each link and the environment can be represented by spring forces that generate forces which are later converted into torques at the arm joints.

This approach is well suited for the Sarcos arm hardware setup. The slower normal cone algorithm is run on the graphics workstation where speed is not as critical to stability. The faster local descent algorithm is run on the workstation which controls the manipulator forces at haptic rates. While the local algorithm is awaiting new minima from the global algorithm it continues to update the old minima using new positions from the arm. This allows the two workstation to run asynchronously with little effect on stability. However, when the arm approaches a new potential contact it is critical to ensure that the global algorithm produces new minima before penetration occurs. To minimize the chance of this possibility, the global algorithm was modified to pass minima longer than the force generation threshold. The cutoff length of minima must be balanced against global update rates and the number of minima passed over the network. If the global algorithm length threshold is too large, performance will suffer as more minima must be checked by the global algorithm, passed over the network, and updated by the local algorithm.

One drawback to the normal cone hierarchy algorithm occurs when the colliding surfaces are flat or concave and have densely packed triangles. In this scenario a large number of local minima can be generated within a small area, increasing computational load but providing little in terms of haptic stability or fidelity. To minimize this effect, as the algorithm generates local minima between two adjacent models, the newly created minima are checked for their proximity to the previously generated and accepted minima. Minima with endpoints falling withing a specified radius of another's endpoints are discarded before use by the local descent and force generation algorithm. Once a new minimum is accepted, it becomes a new baseline against which future solutions are checked.

3.2.2 Networking and Data Flow

To reduce network communications, identical copies of the arm and environmental models were stored on both the global and local-update computers. Joint angles were received at a rate of 100 Hz from the local-update computer to update the position and orientation of the global models. The graphics and minimum distance solutions were then generated on the 3.0 GHz Intel P4 based computer running Linux. These minima were passed back to the local-update computer, an Intel 2.8 GHz P4 computer running VxWorks 5.5, as they were generated.

With real-time joint angles available, the local-update computer makes adjustments to the minima using the local descent algorithm. Next, the minima are converted to spring forces which produce torques from the joint torque controller.

Finally, command voltages generated by torque controller were output through an D/A converter, through an amplifier, and finally to the Sarcos servovalves. High resolution potentiometers measured joint angles while strain gauge load sensors measured the torque at each joint. These signals were passed back through the signal conditioning boards to an A/D converter to be used by the local update computer for control and model updates.

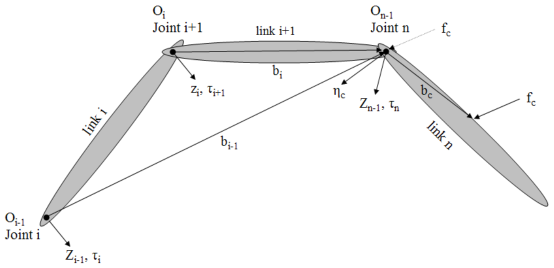

Figure 4: Collision force and resulting joint torques

3.2.3 Joint Torque Calculations

In calculating the joint torques, an assumption is made that the joint axes on the Sarcos Master are nearly collinear with the joints of the human user's arm. In addition, we assume that a change in the robotic arm's joint angle represents an equal change in the user's joint. This will not always be exact, but for this study, the assumption is mostly important for ergonomic reasons rather than for the fidelity of the display. We are mostly concerned that the user's arm is not twisted unrealistically while trying to apply forces and as long as the user is reasonably coupled this will be true. However, the user's arm dimensions clearly can effect the display of forces and the impact of different user dimensions while using a fixed arm remains a topic of further study.

Before any contact forces can be applied to the arm model and before joint torques can be accurately calculated, the spatial relationship between the links of the arm must be defined. This is done using Denavit-Hartenberg (DH) parameters [15]. The DH parameters used are the physical DH parameters of the Sarcos Master. However, these could be varied along with the graphical model of the human arm to explore virtual environments with different arm sizes.

The next step is to determine the joint torques resulting from an external force applied anywhere on the arm model (Figure 4). Because a contact force can be applied anywhere on the model of the arm, a single force Jacobian relating the force applied at the end effector and the numerous joint torques will not suffice. Instead, we create a force Jacobian for each link in collision assuming the end effector to be located at the end of link $i-1$ if the contact occurs on link i. The contact force will be translated to the new end effector becoming a force and torque. The joint torque of the contact link, joint i, can be calculated by crossing the new end effector torque with joint axis of the contact link. Next, all proximal joint torques are calculated using a tailored force Jacobian.

For each link the Force Jacobian is tailored such that joint torques are only generated for the joints leading up to, but not including the contact link's joint. If the collision occurs on link n, the torques for joints 1 through $n-1$ are defined as

$$\begin{bmatrix}\tau_1\\ \vdots \\ \tau_{n-1}\end{bmatrix} = J_f \begin{bmatrix} f_c \\ \eta_c\end{bmatrix} = \begin{bmatrix} (z_0\times b_0)^T & z_0^T \\ \vdots & \vdots \\ (z_{n-2}\times b_{n-2})^T & z_{n-2}^T\end{bmatrix}\begin{bmatrix} f_c\\ \eta_c\end{bmatrix} \tag{1}$$

where

- z_i is the axis of rotation at joint $i+1$
- τ_i is the joint torque at joint i about axis z_{i-1}
- b_i is the vector from joint $n-i$ to joint n
- f_c is the force at the point of contact on link n
- η_c is the torque created by f_c at joint n

Finally, the joint torque at joint n is simply

$$\tau_n = z_{n-1}\cdot\eta_c \tag{2}$$

Note that all vectors in the above equation 1 and 2 have been transformed back to the base coordinate system. This torque calculation is completed for every contact minimum, and the results are summed to generate the contact joint torques for each of the arms joints.

3.3 Joint Torque Control

The control scheme used for the DTS Master in this research was an admittance type controller. In developing a control scheme we took advantage of the joint position and torque sensors located at each joint. The position sensors simply measured the relative angles between two links. These joint angles were used in the torque controller and to position the virtual arm. The torque sensors on each joint measured the relative torque between the two links. These torques were used in a feedback loop to ensure forces applied to the users arm were minimized while the virtual arm was in the free state and accurately displayed while in the contact state.

A typical admittance controller attempts to minimize the torque error, or difference between the externally applied and virtual contact forces, by changing the manipulators position or velocity. However, the goal of the haptic display is to simply apply forces back to the user. To accomplish this, a very simplistic control scheme might be used where the hydraulic actuators, which generate torque, are controlled by a proportional gain applied to the torque error for each joint. A derivative or integral component might also be added to decrease the response time and increase steady state accuracy. Unfortunately, controlling torque directly with this type of controller proves to be difficult due to the noise associated with torque sensors. The real challenge is in achieving a high enough gain so that the joint torques can be effectively driven to near zero, not including the gravity compensating torques, when the virtual arm is in the free state. Without sufficiently high gains, the manipulator would feel excessively sluggish and burdensome to the user. Unfortunately, increasing the proportional gain of the controller reduces the stability margin for second order or higher systems. The sensor noise can be filtered out, but this introduces more destabilizing lag. Achieving stability while maintaining an acceptable controller gain can be very problematic.

3.3.1 Admittance Control Approach

Bilodeau and Papdopoulos [1] developed a slightly more complex controller termed model-based impedance control, using the SACOS DTS, for hydraulic manipulators interacting with their environment. In this admittance type controller, forces or torques are measured, and, through an impedance filter, positions are commanded. Essentially, torques or forces from interaction with the environment along with a desired trajectory are fed into a linear model of the system. The impedance model outputs a new position to which the manipulator is then commanded by a position feedback loop.

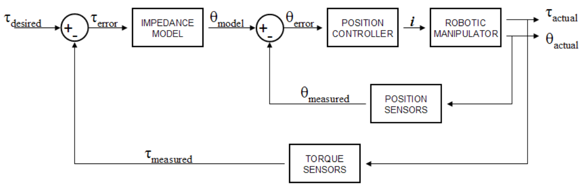

Figure 5: Admittance control diagram

The first component of the controller is the impedance model. To reduce the computational load, we assumed the manipulator was always in a quasistatic state. For a single revolute actuator, the system behavior for a linear, second order model is given by [1]

$$J_m(\ddot{\theta}_m) + B_m(\dot{\theta}_m) + K_m(\theta_m - \theta_d) = \tau \quad (3)$$

where J_m, B_m, and K_m are the desired impedance model parameters, θ_m and θ_d are the impedance model and desired joint angles, τ is the torque applied to the model. The torque is equal to

$$\tau = (\tau_u + \tau_{ga}) - (\tau_c + \tau_{ge}) \quad (4)$$

where τ_u is the torque applied by the user, τ_c is the torque due to contact with the virtual environment, τ_{gm} is the measured torque due to gravity, and τ_{ge} is the expected torque due to gravity. The joint angle, θ_m, was solved through a double integration of the torque. This double integration can cause the model to drift away from the manipulators true joint angle, especially at the joint stops or during high accelerations due to contact. To prevent destabilizing divergence of the model angle and the true joint angle, θ_d was set to the manipulators joint angle. Tuning K_m was an important factor in ensuring low impedance in the free state and rigid and stable contact in the collision state.

Equation 5 represents the Laplace Transform of the position error, or difference between the impedance model and manipulator joint angles, which is the input to the position controller. Inspection of this equation reveals that the position error equals τ/K_m in the steady state condition as the second and third terms cancel. It also reveals that change in the position of the manipulator effectively reduces τ by the difference between the second and third term. This difference stabilizes the arm by reducing controller gain during movement where large torque spikes can occur. In the absence of a good dynamic model of the manipulator, this control scheme worked well.

$$\theta_{error} = \frac{\tau}{J_m s^2 + B_m s + K_m} + \frac{K_m \theta_d}{J_m s^2 + B_m s + K_m} - \theta_d \quad (5)$$

There are several options for the position controller to include PID, adaptive, and sliding mode schemes [1]. For this haptic application, a simple sliding mode scheme was used for two reasons. Most importantly, the sliding mode controller will saturate, limiting the command current to the powerful hydraulic actuators. If network communications fail or the computer controlling the arm locks up, the rate of movement of the arm will be limited. Second, the robust sliding mode controller design will attempt to force decay of θ_{error} to mimic a simple first order system, minimizing oscillations. The controller used operates almost exclusively in the continuous region, which effectively makes it a simple PD position controller, but tuning becomes more intuitive.

To make a given position error decay like a linear, first order model, the sliding surface

$$s = \dot{\theta}_{error} + \lambda \theta_{error} = 0 \quad (6)$$

is used. Because a human will be directly acted upon by the actuator, chatter is unacceptable. To prevent this, the control law

$$u = -B_o sat(s/\varepsilon) \quad (7)$$

is used where the saturation function replaces the traditional signum function. In addition, s is replaced by s/ε where epsilon can be varied to change the slope of the saturation function. The use of the saturation function introduces steady state position error and the system will no longer have a time constant of $1/\lambda$ like the ideal first order system. However, the actual time constant of the systems position response is much less important than the actuators torque response because the arm is positioned by the user. Also, the position error is the difference between the impedance model and the manipulator, and is only present when the user is pushing on a link or when the virtual arm contacts the virtual environment. In effect, the position error is used to develop torque at the joints as opposed to controlling position.

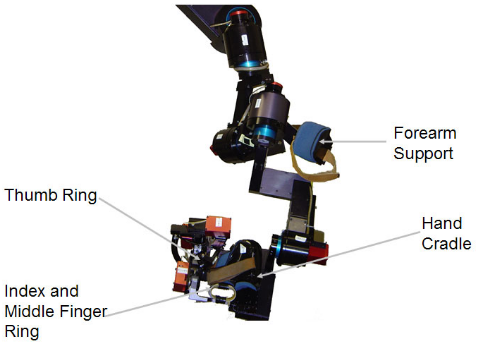

Figure 6: Sarcos Master coupling to the human arm

While this controller is composed of an idealized impedance model of the manipulator and a robust position controller, tuning of the controller parameters leads to a controller that has little in common with the physical manipulator or virtual environment. The impedance models moment of inertia is much lower than the physical link, and the stiffness of the position controller is much too low to ensure accurate tracking between the impedance model and manipulator. Instead, the controller can be viewed as a torque filter with position feedback, as seen in Equation (8), where the control input, u, is

$$u = \frac{B_o}{\varepsilon}\left((s+\lambda)\left(\frac{\tau}{J_m s^2 + B_m s + K_m} + \frac{K_m \theta_d}{J_m s^2 + B_m s + K_m} - \theta_d\right)\right) \quad (8)$$

The most significant difference between the simple torque controller and the admittance controller appears during joint movement. In the case of the simple torque controller there can be a large spike in the command signal as the joint begins to move; this spike is significantly reduced in the case of the admittance controller. This spike can destabilize the manipulator and therefore requires a lower proportional gain to ensure stability. The benefit of the admittance controller developed here is that the position feedback is used to reduce the gain of the controller during joint movement. This allows higher controller gains which ultimately reduces the burden on the user and allows stiffer virtual environments to be modeled.

4 An Accessibility and Force Feasibility Study

While hand-only haptic rendering has proved to be beneficial [2, 16, 4] there is a seemingly obvious benefits to haptically rendering forces across the entire human arm. With full-arm haptic rendering designers would be able to quickly evaluate their virtual environments for accessibility and ergonomics as influenced by the arm's orientation as well as obstacles in the environment. In tight space this information can prove to be invaluable. However, because of the additional complexity and cost of full-arm haptic rendering, these disadvantages must be offset by an increase in speed and accuracy during the evaluation of virtual environments.

In order to evaluate this hypothesis, a study was designed to quantify the benefits of allowing full-arm interaction with a virtual environment vs. hand-only haptic rendering by addressing the following questions:

- Are virtual model evaluation times reduced?

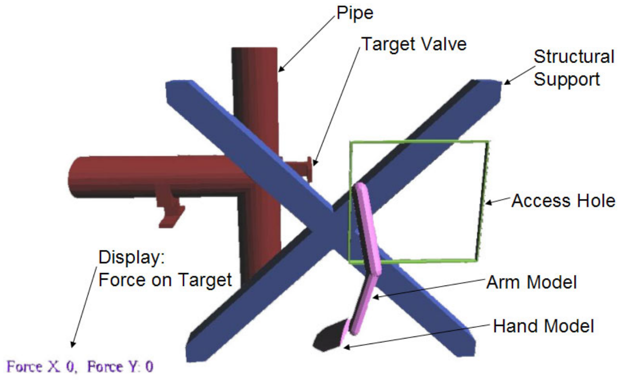

Figure 7: Experiment environment

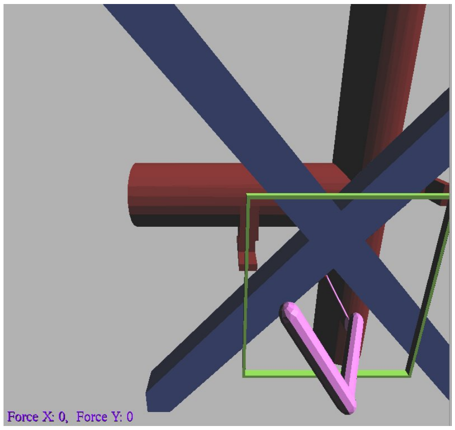

Figure 8: One of the test environments.

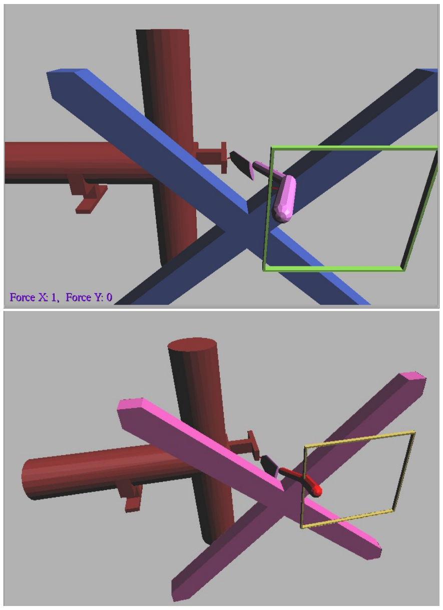

Figure 9: The figure on top shows the normal (no penetration) color scheme. The figure below shows the upper and forearm (red) penetrating the access hole (yellow) and structural member pink).

- Are model evaluations more accurate in determining if a task is possible?
- How does full-arm haptic rendering affect a person's ability to apply a force or torque to the target?

4.1 Experimental Setup

The tests used the Sarcos Master exoskeleton for all haptic rendering. This manipulator couples to the user's hand and arm with a ring for the thumb and index finger, a cradle for the hand, and a pad and strap near the elbow joint for the forearm, as shown in Figure 6. Because there was no direct connection between the manipulator and the user's upper arm, it was assumed that the location of each participant's right shoulder remained fixed throughout the trials. Participants also stood on a platform which was adjusted to ensure that their elbow and the elbow joint of the manipulator were at nearly equal heights. The subject viewed simulation graphics on a CRT screen at roughly head height.

In the experiments, participants were asked to evaluate potential valve placements for the plumbing in a commercial or industrial building setting shown in Figure 7. Their task was to navigate their arm through an opening in a wall, around some structural support behind the wall, and apply a force to a gate style valve as if to close it. We timed each trial to determine how long the participant needed to evaluate the environment and determine if the task was possible. To determine each participant's accuracy in evaluating the virtual environments we measured the participants success in reaching the target where the tasks ranged from impossible to easy. Finally, to understand the benefits of full-arm haptic rendering during force application, we recorded the maximum force applied by each participant to the target.

The participants experienced the five environments twice in random order, for a total of ten trials, once with full-arm haptic rendering (arm forces on) and once with hand-only haptic rendering (arm forces off). The environments represented a range of difficulty in reaching the valve, including one environment for which was impossible (Environment 2). Some example environments are shown in Figures 8, 7, and 9, representing environments 1, 4, and 5, respectively.

Participants would start each trial with their hand at their side. When the clock started they would physically move their hand and arm around the obstacles while watching their movements on the CRT monitor. Once they could reach the target with their hand, they would apply a sustained force of at least one unit as indicated on the monitor. At this point the clock was stopped, and the participants were asked to apply maximum force to the target holding that force for at least one second.

Before beginning the trials, participants were given time to explore the virtual environment with hand-only as well as full-arm haptic rendering. During that time they also practiced moving their arm from the starting position, with their arm at their side, to the target valve. Once the user felt comfortable with the system, usually after about 10-15 minutes of practice, the trials began. In the hand-only haptic rendering trials, when an arm link penetrated the environment the user would not feel a resisting force, but both the arm link and the environment would change color to visually indicate penetration shown in Figure 9. Also, the participants head movements were not tracked so the user's frame of reference was static. Typically a technician would be able to move to a position where they could see the valve through the opening in the wall before blindly reaching in to complete the task. Since the participants frame of reference was fixed, the wall was not displayed, and the opening was framed with thin rectangular tubes. Because of the somewhat awkward movement required to get through the hole in the wall, if participants wasted more than 5 seconds during this stage they were asked to return their arm to their side and restart the trial.

Table 1: Normalized Task Completion Times

Env. #	Arm Force Off Time	Arm Force On Time	% Reduction
1	1.11	0.63	43
2	1.48	1.27	14
3	1.06	.81	24
4	.85	.54	37
5	1.54	.72	53

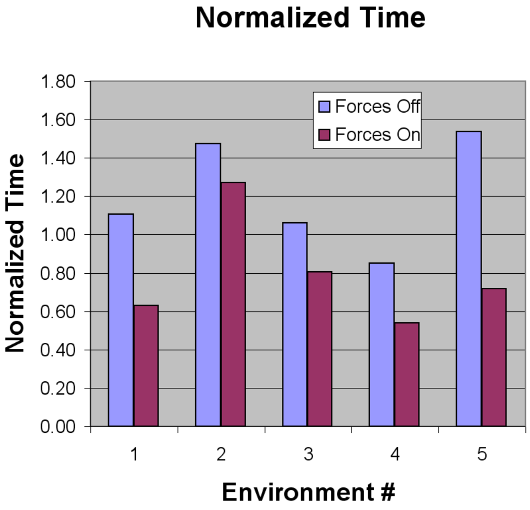

Figure 10: Normalized task completion times comparing arm forces on and arm forces off.

Task accuracy was simply a measure of whether or not the participant could successfully complete all possible tasks. If the participant stated they could not get to the target and apply the minimum force, the trial ended. Of the 5 environments, only environment 2 was impossible to complete, while only environment 3 was clearly possible without interacting with the environment.

In addition to the objective measurements of time, accuracy, and force, participants were also asked the following questions. Was it easier or more difficult to reach the target valve with the arm forces on, and why? Was it easier or more difficult to apply force to the target valve with the arm forces on, and why?

4.1.1 Participant Characteristics

Ten individuals participated in the experiment. Nine of the participants were either aircraft engineers or technicians having spent significant time working in the confined spaces of various aircraft. However, only 3 of the participants had any experience with haptic applications and only 1 was familiar with the setup used in these experiments.

4.2 Results

4.2.1 Evaluation Time

Evaluation times are shown in Figure 10 and Table 1. To account for slow careful movements of some participants versus the aggressive movements of others, the data is normalized by the cumulative time for each participant.

Table 2: Normalized Force Application with Incorrect Assessments Removed

Env. #	Arm Force Off Magnitude	Arm Force On Magnitude	% Force Increase
1	.72	.9	25
2	impossible task	impossible task	N/A
3	1.56	2.07	33
4	1.24	1.52	23
5	.48	.70	47

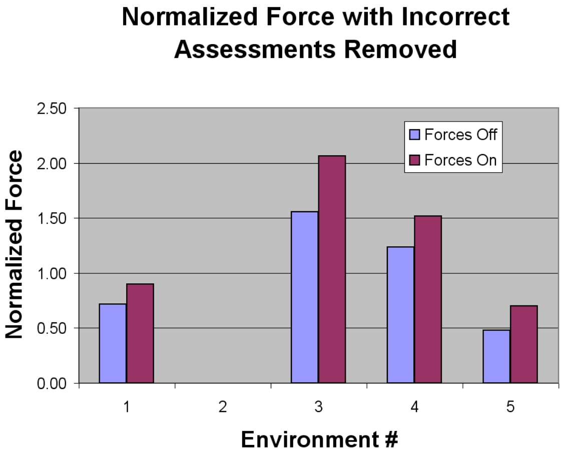

Figure 11: Normalized force application with incorrect assessments removed comparing arm forces on and arm forces off.

4.2.2 Assessment Accuracy

Of the 5 environments, only environment 2 made reaching the target impossible. However, environments 1, 4, and 5 were designed so that visually it was difficult to determine if the target was reachable. Without full-arm haptic rendering, participants were wrong 20% of the time in environment 1 and 50% of the time in environment 5.

4.2.3 Force Application

Maximum force application is shown in Figure 11 and Table 2. To account for the differences in strength of the participants, the data is again normalized by the cumulative force applied for all trials for each participant. In addition, because some participants did not complete the task for environments 1 and 5 with the arm forces off, the corresponding task with arm forces on was excluded as not to skew the data.

4.2.4 Statistical Confidence Measures

In order to test the statistical significance of the measurements, a matched pair t-test was performed to test the probability of a zero mean difference between results with arm forces on and arm forces off. The one-tail p-values are reported in Table 3. The results show a strong confidence that the improved performance in many of the environments is statistically significant. In environment two, however, there is a not measurable statistical significance. This environment is the one in which it was impossible to reach the goal, so it is not unreasonable that it took a long time to determine that no matter the state of the arm forces. Environment three also shows weaker significance. In this environment, the arm had to move through a fairly complicated series of configurations to reach the goal. When

Table 3: P-values from Matched Pair T-tests Comparing Arm Forces On and Off

	Env. 1	Env. 2	Env. 3	Env. 4	Env. 5
Time Test	0.004	0.26	0.12	0.005	0.00002
Forces Test	0.13	N/A	0.05	0.05	0.046

arm forces were off, people tended to collide with the environment, move through the difficult part, then try to get out of colliding status. So, while using the full-arm haptics people were forced to maintain a valid sequence of motions, while the hand-only haptics made illegal motions possible, invalidating their ability to test accessibility. The arm force performance benefits are not as strongly statistically significant as the time performance, although still measurable.

4.2.5 Subjective Assessment of the Benefits of Full-Arm Haptic Rendering

At the end of each experiment, participants were asked if it was easier or more difficult to reach the target when they could feel the arm contacting the environment. In every case, participants felt it was easier to navigate through the environment. Below are some of the responses:

- You could feel your way through the environment.
- It was much more difficult to move your hand to the target while watching for color change.
- Feeling resistance is more natural than watching for color change.
- The forces guide the path your arm takes so there is less to focus on.

They were also asked if they felt it was easier or more difficult to apply maximum force to the target. All but two participants felt they could apply more force to the target when they could feel the arm contacting the environment. Of the two, one felt there was no difference while the other felt it was more difficult. Below are some of the responses from those who found force application easier:

- You could use the environment to gain leverage while pushing on the target.
- You could concentrate on applying force to the target instead of worrying about the arm changing color.

5 Discussion and Conclusions

To demonstrate full-arm haptic rendering we used a distributed computation with normal cone hierarchy and local descent methods for computing minimum distances, tailored force jacobians to compute individual joint torques, and position-based impedance for joint torque control. While there is room for improvement, this demonstration system proved useful in showing the potential benefits of full-arm haptic rendering.

Using the Sarcos DTS Master robotic arm we were able to demonstrate the benefits of full-arm haptic rendering, namely, improved speed in determining accessibility and improved ergonomics, resulting in easier application of forces. While the additional cost is not justifiable in all situations, there are clear benefits in complex virtual environments where obstacles interfere with a persons ability to reach and apply a force or torque to a target. Further studies are warranted to explore the contribution of full-arm haptics to task completion.

Acknowledgements

This work was supported in part by NSF grants DMI-9978603 and CCF-0541402.

References

[1] G. Bilodeau and E. Papadopoulos. A model-based impedance control scheme for high-performance hydraulic joints. In *Proc. 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 1308–1313, October 1998.

[2] D. Borro, J. Savall, A. Amundarain, A. Garcia-Alonso, and L. Matey. A large haptic device for aircraft engine maintainability. *Computer Graphics and Applications, IEEE*, 24(6):70–74, Nov.-Dec. 2004.

[3] F. Dai, W. Felger, T. Fruhauf, M. Gobel, D. Reiners, and G. Zachmann. Virtual prototyping examples for automotive industries, 1996.

[4] A. Fischer and J. Vance. Phantom haptic device implemented in a projection screen virtual environment. In *ACM International Conference Proceedings Series*, volume 39, pages 225–229, 2003.

[5] J. Hollerbach. Some current issues in haptics research. In *Proc. IEEE Intl. Conf. Robotics and Automation*, pages 757–762, April 24-28 2000.

[6] S. Jacobsen, F. M. Smith, E. K. Iversen, and D. K. Backman. High performance, high dexterity, force reflecting teleoperator. In *Proc.38th Conference on Remote Systems Technology*, volume 2, pages 180–185, 1990.

[7] D. Johnson and E. Cohen. Spatialized normal cone hierarchies. In *Proc. 2001 ACM Symposium on Interactive 3D Graphics*, pages 129–134, March 19-21 2001.

[8] D. Johnson and P. Willemsen. Six degree-of-freedom haptic rendering of complex polygonal models. In *Proc. 11th Symposium on Haptic Interfaces for Virtual Envrironment and Teleoperator Systems, HAPTICS 2003*, page 229.

[9] D. Johnson and P. Willemsen. Accelerated haptic rendering of polygonal models throgh local descent. In *Proc. 12th International Symposium on HAPTICS*, pages 2566–2573, March 27-28 2004.

[10] H. Maekawa and J. Hollerbach. Haptic display for object grasping and manipulating virtual environments. In *Proc. IEEE Intl. Conf. Robotics and Automation, (Leuven, Belgium)*, pages 16–21, May 16-21 1998.

[11] W. McNeely, K. Puterbaugh, and J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *ACM SIGGRAPH 99 Conf., Los Angeles, CA*, pages 401–408, August 1999.

[12] M. Morrissey. Human-centric design. *Mechanical Engineering*, 120(7):60, July 1998.

[13] M. A. Otaduy and M. C. Lin. Sensation preserving simplification for haptic rendering. In *Proc. from ACM SIGGRAPH*, pages 543–553, 2003.

[14] C. B. Phillips and N. I. Badler. Jack: a toolkit for manipulating articulated figures. In *UIST '88: Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software*, pages 221–229, New York, NY, USA, 1988. ACM Press.

[15] L. Sciavicco and B. Siciliano. *Modeling and Control of Robot Manipulators*. McGraw-Hill, 1996.

[16] J. Troy. Haptic control of a simplified human model with multibody dynamics. In *Phantom Users Group Conf., Aspen, CO*, pages 43–46, October 2000.