Haptic Interaction With Constrained Dynamic Systems
Eric L. Faulring, Kevin M. Lynch, J. Edward Colgate, Michael A. Peshkin,
Proc. International Conference on Robotics and Automation, Barcelona, 2005

Haptic Interaction With Constrained Dynamic Systems

Eric L. Faulring, Kevin M. Lynch, J. Edward Colgate and Michael A. Peshkin
Department of Mechanical Engineering
Northwestern University
Evanston, IL 60208
USA
e-faulring@northwestern.edu

***Abstract*—In this paper we are concerned with allowing the operator of a haptic display to interact with virtual systems having significant inertial dynamics and realistic constraints. We review the mathematical structure arising from the kinetic energy metric, required to create a virtual dynamics simulation consisting of rigid-body dynamics along with holonomic and/or nonholonomic motion constraints. We develop an admittance controller composed of feedforward and feedback terms, while preserving the integrity of the intended virtual dynamics simulation. This controller is implemented on the Cobotic Hand Controller, an admittance-type haptic display, and two examples are discussed.**

Index Terms – Haptics, Cobots, Rigid-Body Dynamics, Hybrid Control, Virtual Surfaces.

I. INTRODUCTION

A. Motivation and Goals

There are an increasing number of teleoperation based tasks or virtual environment interactions in which a high fidelity haptic master controller is desired. These include the control of a slave robot in medical surgeries, micro/nano-manipulation, undersea salvage, maintenance of nuclear plants and other hazardous environments, as well as interaction with computer-aided design models and other virtual environments. The execution of these tasks by an operator is affected by his/her sense of telepresence and the transparency of the master-slave relationship [1]. Physical cues provided by the master manipulandum or haptic display, be they feedback from the remote site, or assistive constraints on motion, improve operator performance and efficiency for many dexterous tasks. Consider a surgeon remotely controlling a slave robot's movements of a scalpel. In order to execute a precise incision, it is desirable for the motion of the scalpel to be constrained to a straight path at a certain depth, filtering out tremor and preventing damage to tissues beneath the incision or along either side. Such constraints can vastly simplify execution of a six-degree-of-freedom task in a teleoperation setting. Force feedback can also be used to inform the surgeon as to what impedance the scalpel is in contact with. The interactions or constraint forces imposed on the scalpel in the remote environment are played back to the operator via the haptic display.

Users of a teleoperator typically manipulate a master-manipulandum that may be quite different from the scalpel in the remote environment. The apparent inertia of the end-effector changes with configuration of the haptic display mechanism while that of the scalpel does not. In order to maintain the surgeon's sense of telepresence, we believe that the master-manipulandum's apparent inertia should be controlled to appear like that of the scalpel, retractor, or suturing tools that the surgeon is familiar with.

Rarely, in practice, are inertial dynamics of a haptic display masked or made to behave like those of the virtual tool. Perhaps because of the predominance of impedance displays, very little literature has addressed the haptic display of inertial systems. The vast majority of haptic devices (e.g., the ubiquitous Phantom [2]) are "impedance displays" – they sense motions applied by users and control force and torque in response. Most often, the user interacts with a virtual probe or stylus whose inertia is not simulated by the haptic display. That is, the inertia felt by the user is that of the haptic display itself, but since both this and the inertia of the virtual probe are small, the difference is not perceptible.

We have recently introduced a six-degree-of-freedom powered cobot and described its capabilities as a haptic interface [3]. This cobot is an example of an "admittance-type haptic display" – it senses the forces and torques applied by a user and controls motion in response. The Haptic Master [4] is another example of an admittance-type haptic display. While today's impedance and admittance displays may both be used to simulate a wide range of mechanical behaviors, they excel in different areas. Impedance displays are highly backdrivable and well-adapted to simulating low inertia, low damping environments. Admittance displays, on the other hand, are well-adapted to displaying rigid constraints, so-called "virtual surfaces" [5]. Unlike impedance displays, admittance displays must actively simulate inertia. While doing so increases controller complexity, it also allows for a higher level of fidelity, as illustrated in the scalpel example

The authors would like to acknowledge the support of the Department of Energy, grant number DE-FG07-01ER63288.

above. As another example, the user of an admittance display can throw it and reasonably expect proper behavior even once the device has left his hand.

In this paper, we take on the problem of simulating inertial systems, including holonomic and nonholonomic constraints.

B. Algorithms

Two sources of simulation techniques are the haptics literature and the computer graphics/animation literature. In the former, popular approaches to handling constraints are the god-object tracker and the virtual proxy algorithm [6, 7]. These algorithms track the location of a tool in the virtual world (e.g., the surgeon's scalpel), compute the points of interaction with that virtual world (a model of the patient), and ultimately compute contact forces due to inter-penetration depths. These forces are then used to compute a total reaction force to be displayed to the surgeon. Significant in the present context, these algorithms do not normally allow for the control of inertial dynamics tangent to surfaces or while in unconstrained situations. The motion of the god-object or virtual proxy along a constraint does not obey a dynamic model, but instead moves in response to the user's motion.

Much more general algorithms for resolving dynamic collisions as well as the contact state between two rigid bodies have been developed by the computer graphics community. These methods compute constraint forces and impulses, and treat dynamics in all directions, normal as well as tangent [8, 9]. Generally speaking, however, these methods are designed for neither "hard" real-time (where computation time is always less than the actual integration time step) nor interface to haptic devices.

C. Haptic System Framework

Figure 1 illustrates the terminology we use in this paper to describe the operation of a haptic system. We call the physical device manipulated by the user the "manipulandum," with coordinates x and inertia matrix $M_m(x)$, the unmasked inertia of the end-effector. The "virtual tool" in the virtual environment has coordinates q_v, related to the coordinates of the manipulandum by the kinematics

$$q_v = \varphi(x). \tag{1}$$

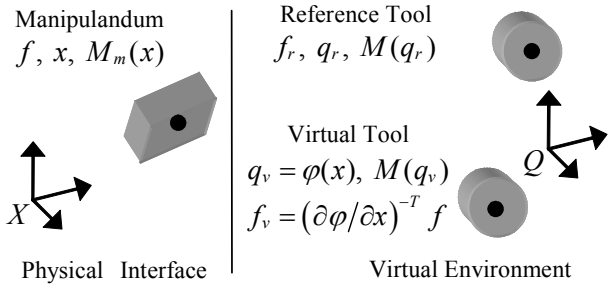

Figure 1. Physical interface and virtual environment.

The inertia matrix of the virtual tool, or our surgeon's scalpel, is $M(q_v)$. Because the coordinates of the virtual tool are directly determined by the manipulandum, due to control errors, the virtual tool may not exactly satisfy the holonomic and nonholonomic constraints describing the desired motion and configuration freedoms of the tool. Because of this, we define a "reference tool" with coordinates q_r and an inertia matrix $M(q_r)$. The reference tool moves according to a physics simulation and always exactly satisfies the virtual holonomic constraints

$$H(q_r) = 0 \tag{2}$$

and the virtual nonholonomic constraints

$$A(q_r)\dot{q}_r = 0. \tag{3}$$

Note that the holonomic constraints can be differentiated and included in the $A(q)$ matrix. We do this in the remainder of the paper. However, one must be careful to avoid creep resulting from integration errors of such a system. We employ a parametric method (discussed in detail later) in order to keep the holonomic constraints satisfied.

For our example of the surgeon, we could employ both holonomic and nonholonomic constraints, limiting the depth of the scalpel's incision and the direction of its motion respectively. The holonomic constraint reduces both the number of available motion freedoms and dimension of the configuration space, and the nonholonomic constraint reduces the number of motion freedoms but not the dimension of the configuration space.

The control scheme described in this paper can be summarized as follows. The user applies a generalized force f to the manipulandum, sensed by a six-degree-of-freedom load cell. This force is transformed by the kinematics φ to a force acting on the virtual tool f_v.

$$f_v = \left(\frac{\partial\varphi}{\partial x}\right)^{-T} f \tag{4}$$

Assuming that q_r is approximately q_v (one of the purposes of the controller is to keep the virtual tool close to the reference tool), the generalized force $f_r = f_v$ is applied to the reference tool, along with any other virtual external forces due to springs, dampers, and gravity. The acceleration of the constrained reference tool is then calculated, and the reference tool motion is integrated forward. This acceleration acts as a feedforward acceleration applied to the virtual tool. In addition to this feedforward acceleration, a (hopefully small) feedback acceleration is applied to the virtual tool to compensate small position and velocity errors between the reference tool and the virtual tool. The total acceleration is transformed by the kinematics φ to an acceleration of the manipulandum $\ddot{x}$. The result is a realistic display of the constrained dynamics of the virtual tool. Since the device is commanding motion, it is imposing forces on itself through

any mass distal to the load cell. Cancellation of the static and dynamic affects of this mass should be done.

D. Paper Summary

Section II describes the calculation of the acceleration of the reference tool, and Section III describes the controller that causes the virtual tool (and therefore the manipulandum) to track the reference tool. Examples of a disk rolling on a plane and a plate sliding tangent to a sphere are provided. Section IV presents experimental data from implementations of the examples with a haptic display.

II. Simulation of a Dynamic Haptic Environment

A. Outline

There are several steps in propagating the physics simulation of a reference tool subject to constraints. The input to the simulation is the current state of the reference tool, q_r and $\dot{q}_r$, as well as forces f_r applied by an operator. The output is the resulting acceleration, $\ddot{q}_r$, based on the simulated inertia, damping, springs, gravity and constraints. This acceleration is then integrated in such a manner consistent with the configuration submanifold specified by holonomic constraints, and becomes the new state of the physics simulation.

B. Euler-Lagrange Formulation

The Euler-Lagrange dynamic equations, including holonomic and/or nonholonomic constraints, are expressed

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau + A(q)^T \lambda \tag{5}$$

$$A(q)\dot{q} = 0\,. \tag{6}$$

These are expressed in terms of n generalized coordinates q. $M(q)$ is an $n \times n$ symmetric positive definite inertia matrix. The Coriolis forces are represented by the vector $C(q,\dot{q})\dot{q}$. $A(q)$ is an $m \times n$ matrix of Pfaffian constraints, either holonomic or non-holonomic, and λ is the vector of Lagrange multipliers representing the m constraint force magnitudes. The rows of $A(q)$ are the constraint force directions. External forces $\tau = f_v - \tau_d - \tau_s - g(q)$ are composed of user interaction forces f_v, forces of virtual springs τ_s and forces of virtual dampers τ_d. Generalized user interaction forces f_v are measured via a force torque sensor in an admittance-type haptic display. Virtual gravity and potential fields are represented by $g(q)$.

Equations 5 and 6 apply for any simulated mechanical device. For the particularly interesting case of a rigid body reference tool, generalized coordinates can be given by three translational coordinates (q_1, q_2, q_3) and three Euler angles (q_4, q_5, q_6). We have chosen an Euler angle set such that any singularities are out of the workspace of our specific manipulandum. Euler angles allow us to work in generalized coordinates rather than with the special orthogonal group $SO(3)$.

$M(q)$ for the case of a rigid body can be expressed

$$M(q) = \begin{bmatrix} m\mathrm{I}_{3x3} & 0 \\ 0 & R(q)JR(q)^T \end{bmatrix}. \tag{7}$$

Here m is the desired mass of the virtual tool, I_{3x3} the identity matrix, and J the constant body fixed inertia adjusted by the rotation matrix $R(q)$ to the generalized coordinate inertial frame.

C. Constraints

Given n generalized coordinates and m constraints, the tangent space at a given configuration has $n-m$ motion freedoms. If the rows of $A(q)$, $a_i(q)$, $i = 1...m$, can be represented by $\partial h_i / \partial q = a_i(q)$ for some real-valued functions $h_i(q)$, the constraint is said to be holonomic or integrable and can be written as $h_i(q) = c_i$.

If b of the m constraints in $A(q)$ correspond to holonomic constraints, there exists an $n-(m-b)$-dimensional integrable submanifold Z of reachable configurations. The coordinates $z_j, j = 1...(n-(m-b))$ parameterize the configuration space of the constrained system, a submanifold of the ambient space Q. n functions $q = \psi(z)$ define the parametric description. While nonholonomic constraints reduce the apparent motion freedoms, they do not decrease the dimensionality of the reachable configuration submanifold.

D. Parametric Formulation For Integration

In order to propagate the simulation forward, accelerations of the reference tool, $\ddot{q}_r$, must be computed and then integrated to yield new reference tool velocities and positions. In order to solve for the acceleration, we first evaluate the Lagrange multipliers λ. In order to isolate the Lagrange multipliers, we first differentiate $A(q)\dot{q} = 0$ and plug it into the dynamic equations for $\ddot{q}$, yielding the following expression for the Lagrange multipliers [10, 11]:

$$\lambda = (AM^{-1}A^T)^{-1}(-\dot{A}\dot{q} + AM^{-1}(C(q,\dot{q})\dot{q} - \tau)) \tag{8}$$

Now Equation 5 can be solved for the complete acceleration $\ddot{q}_r$ of the reference tool.

$$\ddot{q}_r = M(q)^{-1}(\tau + A(q)^T\lambda - C(q,\dot{q})\dot{q}) \tag{9}$$

Now we can integrate the reference acceleration $\ddot{q}_r$ to yield $\dot{q}_r$ and q_r. The integration method must keep the reference tool on the constraint submanifold. We suggest a parametric approach in order to effectively integrate the equations of motion and keep the reference position on the constraint submanifold. Generalized accelerations $\ddot{q}_r$ are related to parametric accelerations $\ddot{z}$, via the following relationship:

$$\ddot{q}_r = \frac{\partial \psi}{\partial z}\ddot{z} + \dot{z}^T \frac{\partial^2 \psi}{\partial z^2}\dot{z} \tag{10}$$

In order to integrate the parameters z from desired accelerations $\ddot{q}_r$, Equation 10 can be rearranged to yield

$$\ddot{z} = \left(\frac{\partial \psi}{\partial z}\right)^{\dagger}\left(\ddot{q}_r - \dot{z}^T\left(\frac{\partial^2 \psi}{\partial z^2}\right)\dot{z}\right). \quad (11)$$

Here $(\partial\psi/\partial z)^{\dagger} = ((\partial\psi/\partial z)^T(\partial\psi/\partial z))^{-1}(\partial\psi/\partial z)^T$ is the pseudo-inverse. The pseudo-inverse is merely performing a change of coordinates, or a kinematic projection here. Numerical integration of $\ddot{z}$ will yield a reference point on the configuration submanifold which is necessary later in the paper for the computation of feedback terms.

The method of integration should be chosen carefully to avoid numerical problems over time, but our key concern in this paper is the instantaneous constrained dynamics. Also, although we are doing dynamics for a rigid body, we have chosen coordinates and not a Lie group representation utilizing the special orthogonal group for rotations. While our choice of coordinates has integration problems near singularities, we have avoided much more complex integration issues for the implicit $SO(3)$ representation of orientation [12]. We have no issues with the singularities of an Euler angle coordinate representation since they can be placed out of the workspace of our specific six-degree-of-freedom device.

E. Example 1

In order to demonstrate a physics simulation consisting of both holonomic and nonholonomic constraints, we consider a disk with inertia $M(q)$ as our rigid body in $SE(3)$, confined to roll upright on a plane without slipping. Figures 2 and 3 define the translational coordinates (q_1, q_2, q_3) and rotational coordinates (q_4, q_5, q_6) we have chosen to represent the disk. s indicates the generalized coordinate inertial frame, and b the body frame of the disk.

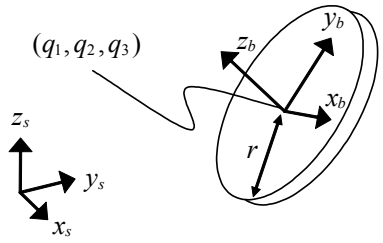

Figure 2. Definition of translational coordinates.

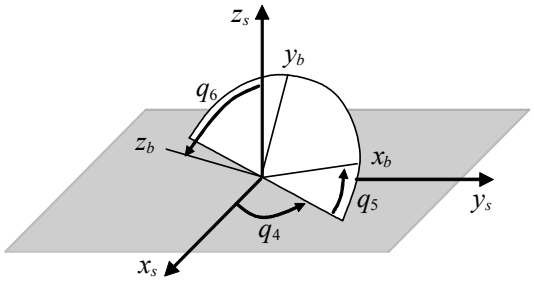

Figure 3. Definition of rotational coordinates.

Restricting a disk of radius r to roll on the plane $q_3 = 0$ and stand upright yields the holonomic constraints

$$q_3 = r \;\rightarrow\; \dot{q}_3 = 0 \quad (12)$$

$$q_6 = \frac{\pi}{2} \;\rightarrow\; \dot{q}_6 = 0 \quad (13)$$

Assuming the holonomic constraints are satisfied, imposing the no-slip rolling constraint yields the following two nonholonomic constraints.

$$r\dot{q}_5\cos(q_4) + \dot{q}_1 = 0 \quad (14)$$

$$r\dot{q}_5\sin(q_4) + \dot{q}_2 = 0 \quad (15)$$

Thus the constraint matrix is

$$A(q) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & r\cos(q_4) & 0 \\ 0 & 1 & 0 & 0 & r\sin(q_4) & 0 \end{pmatrix}. \quad (16)$$

The reachable configuration submanifold is now four-dimensional and can be parametrized

$$q = \psi(z) = \begin{pmatrix} z_1 \\ z_2 \\ r \\ z_3 \\ z_4 \\ \pi/2 \end{pmatrix}. \quad (17)$$

The inertia matrix $M(q)$ is given in Equation 7. Given a generalized force f_r and current state q_r and $\dot{q}_r$, we compute the acceleration of the reference tool $\ddot{q}_r$. Since we have the current z and $\dot{z}$, we can use Equation 11 to project the acceleration of the reference tool $\ddot{q}_r$ to the parametric acceleration $\ddot{z}$. Integrating $\ddot{z}$ yields new values of $\dot{z}$ and z from which we then can compute new values $q_r = \psi(z)$ and $\dot{q}_r = (\partial\psi/\partial z)\dot{z}$. These terms will be used in the development of feedback controllers in the next section.

III. CONTROLLER DESIGN

A. Feedforward and Feedback Components

As shown in Figure 1, the virtual tool analog of the manipulandum position is $q_v = \varphi(x)$. The virtual environment tool position q_v likely does not correspond exactly with the reference tool position q_r. Thus a controller is needed to make the virtual tool track the reference tool, nulling out errors both orthogonal to and along the constraint submanifold.

The feedforward acceleration of the virtual tool $\ddot{q}_{vff}$ is equal to the complete acceleration of the reference tool $\ddot{q}_r$.

A PID feedback acceleration controller for $\ddot{q}_{vfb}$ will take the following form:

$$\ddot{q}_{vfb} = K_p e + K_i \int e\,dt + K_d \dot{e} \quad (18)$$

$e = (q_r - q_v)$ is the tracking error of the virtual tool relative to the reference tool state of the physics simulation. K_p, K_i and K_d are feedback gain matrices.

Summing the feedforward and feedback accelerations of the virtual tool yields the total acceleration command for the virtual tool.

$$\ddot{q}_v = \ddot{q}_{vff} + \ddot{q}_{vfb} \tag{19}$$

Utilizing the kinematics φ, this acceleration is mapped back to the physical interface yielding an acceleration command $\ddot{x} = (\partial\varphi/\partial x)^{-1}(\ddot{q} - \dot{x}^T(\partial^2\varphi/\partial x^2)\dot{x})$. Subsequently, device specific kinematics convert $\ddot{x}$ into joint level commands. For cobots, these joint level commands will be desired velocities of mechanical continuously variable transmission elements [13-15]. A block diagram of an admittance-type haptic control system is shown in Figure 4. Implicit in this diagram is the need to cancel the static and dynamic effects of a haptic display mass distal to the force sensor.

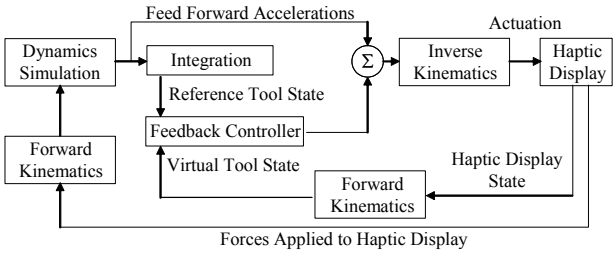

Figure 4. Block diagram of the Dynamics Simulation, Integration, Feedback Controller and Haptic Display.

B. Example 1 Continued

Returning to the example of the disk, we now see that the reference disk state $q_r = \psi(z)$ and $\dot{q}_r = (\partial\psi/\partial z)\dot{z}$ is utilized to generate feedback accelerations both in the constrained directions and the free directions. The feedback terms attempt to keep the velocity vector of the virtual disk to be pure rolling in the plane and obeying the no-slip rolling condition. Feedback terms will null out any deviations of the virtual disk from upright, and deviations of the virtual disk from the plane. Feedback will also assist the feedforward terms in keeping the virtual disk spinning and rolling to keep up with the reference disk.

C. Example 2

As an example of the mathematical tools presented here, consider a disk as shown in Figure 5, whose center must stay in contact with a sphere of radius r located at (s_1, s_2, s_3). Thus the reference disk can move in two degrees of freedom about the sphere that it is constrained to, and rotate about the contact normal axis.

The reference disk, nominally residing in six-dimensional $SE(3)$, is given a set of generalized coordinates q in the same fashion as in Figures 2 and 3. In order to establish the constraint equations for this scenario, consider a rotation matrix $R(q_4, q_5, q_6)$, composed of a rotation of q_4 about z_b, followed by a rotation of q_6 about the new x_b, followed by a rotation of q_5 about the new z_b. This matrix multiplied by the vector $(0, 0, -r)^T$ in the body fixed frame will yield the current generalized coordinates (q_1, q_2, q_3) of the virtual tool less the location of the sphere (s_1, s_2, s_3). The cells of the rotation matrix irrelevant to this example are not shown.

$$\begin{bmatrix} \cdots & \cdots & \sin(q_5)\sin(q_6) \\ \cdots & \cdots & \cos(q_5)\sin(q_6) \\ \cdots & \cdots & \cos(q_6) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} = \begin{bmatrix} q_1 - s_1 \\ q_2 - s_2 \\ q_3 - s_3 \end{bmatrix} \tag{20}$$

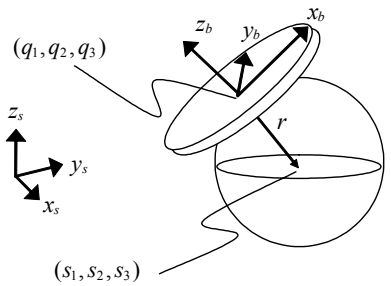

Figure 5. This virtual environment consists of a reference disk required to stay tangent to a sphere.

We can then pull out three holonomic constraint equations:

$$q_1 - r\sin(q_5)\sin(q_6) - s_1 = 0 \tag{21}$$

$$q_2 - r\cos(q_5)\sin(q_6) - s_2 = 0 \tag{22}$$

$$q_3 + r\cos(q_6) - s_3 = 0 \tag{23}$$

Note the lack of dependence on q_4, the allowed spin of the disk about its z_b axis which is the contact normal. These three holonomic constraint equations can be differentiated to yield

$$A(q) = \begin{bmatrix} 1 & 0 & 0 & 0 & -r\,\mathrm{c}q_5\,\mathrm{s}q_6 & -r\,\mathrm{s}q_5\,\mathrm{c}q_6 \\ 0 & 1 & 0 & 0 & r\,\mathrm{s}q_5\,\mathrm{s}q_6 & -r\,\mathrm{c}q_5\,\mathrm{c}q_6 \\ 0 & 0 & 1 & 0 & 0 & r\,\mathrm{s}q_6 \end{bmatrix}. \tag{24}$$

The reachable configuration submanifold is now three-dimensional and can be parametrized

$$q = \psi(z) = \begin{pmatrix} s_1 + r\sin(z_2)\sin(z_3) \\ s_2 + r\cos(z_2)\sin(z_3) \\ s_3 + r\cos(z_3) \\ z_1 \\ z_2 \\ z_3 \end{pmatrix}. \tag{25}$$

Like Example 1, these functions can be differentiated to yield the necessary relations for transformation between generalized coordinate accelerations and parameter accelerations for integration along the parametrized submanifold.

IV. IMPLEMENTATION

A. The Cobotic Hand Controller

The two examples presented in this paper, among many others, have been implemented with the Cobotic Hand Controller (Figure 6), a six-degree-of-freedom, fully-actuated admittance-type haptic display [3]. Figures 7-10

depict actual recorded data from the implementations of Examples 1 and 2. In them we show that the holonomic and nonholonomic constraints are well satisfied.

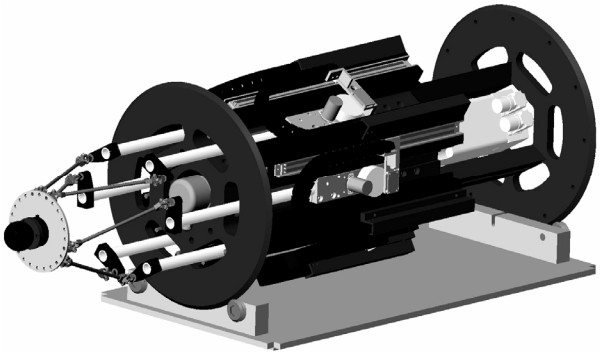

Figure 6. The Cobotic Hand Controller, a six-degree-of-freedom haptic display. An operator interacts with the spherical manipulandum at left. The device can display an extremely large range of impedances due to its use of continuously variable transmissions in its parallel architecture.

B. Example 1 Continued

The six-centimeter-diameter disk in Figure 7 is in rolling contact with the plane. It is allowed to spin and roll, but is unable to slide sideways. It is able to make sideways motion by parallel parking, or it could simply turn and drive in that direction. The simulated disk has mass 0.25 kg, all principal-axis inertias are 0.0025 kgm^2, translational damping is 1.0 Ns/m, and rotational damping is 0.1 Nms/rad.

In Figure 8, several metrics of the implementation are reported. The error in the height of the disk and orientation of the disk are shown to be negligible and on the order of the 25 μm position resolution of the Cobotic Hand Controller. Also reported is the percent error in the rolling constraint, $0.01|V - r\omega|/(|V|+\varepsilon)$, where $r\omega = -0.03\dot{q}_5$, $V = \dot{q}_1 \cos q_4 + \dot{q}_2 \sin q_4$ and $\varepsilon = 1.0$ mm/s.

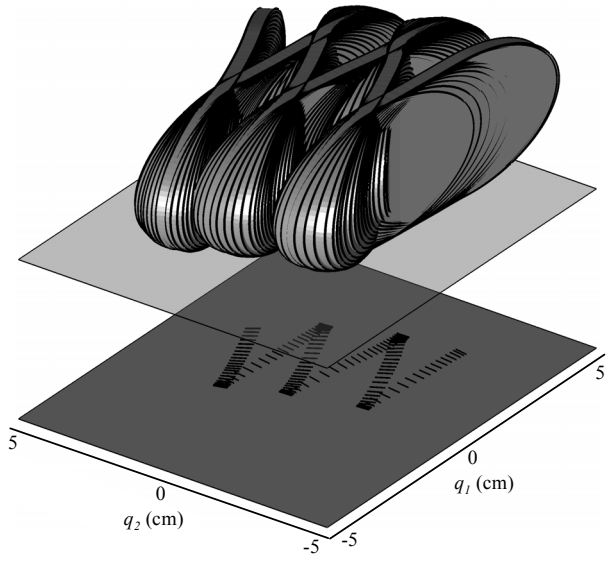

Figure 7. A disc shown at 0.1 second intervals in rolling contact with a planar surface. The shadow shows the line contact of the disc with the plane, or "tire tracks" as the disk makes "parallel parking" motions.

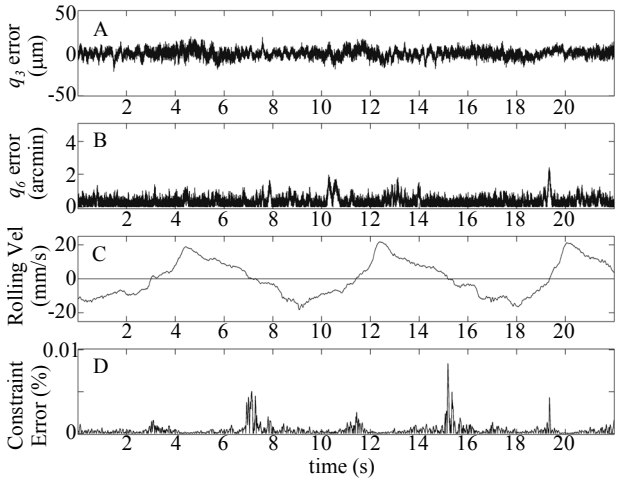

Figure 8. This is experimental data recorded from an implementation of Example 1 with the Cobotic Hand Controller. All data is computed from "virtual" coordinates, or physically meaured parameters of the display. Note that the position resolution of the Cobotic Hand Controller is 25 μm. A) Error in the height of disk center. B) Error in the disk's upright orientation. C) Rolling velocity $V = \dot{q}_1 \cos q_4 + \dot{q}_2 \sin q_4$ of the disk. D) The percent of error in the rolling constraint is $0.01|V - r\omega|/(|V|+\varepsilon)$ for $r\omega = -0.03\dot{q}_5$ and $\varepsilon = 1.0$ mm/s.

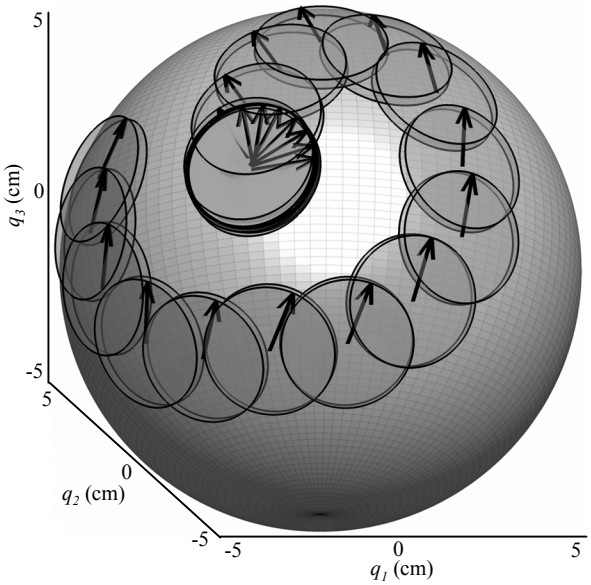

Figure 9. A simulated disc, shown at one second intervals, is constrained to track a virtual sphere while staying tangent to it, but is allowed to rotate about the contact normal. The arrows indicate the orientation of the disk on the sphere.

C. Example 2 Continued

Figure 9 portrays a two-centimeter-diameter disk constrained to stay tangent to a ten-centimeter-diameter sphere, but allowed to rotate about its contact normal with the sphere. The disk has the same inertial properties as in Example 1, even though we have changed its dimensions. In Figure 10, several metrics of the implementation are reported. Also demonstrated is the ability of the Cobotic Hand Controller to impart rigid constraints, as it suffers no significant position errors even when subjected to a 35 N load normal to the constraint surface.

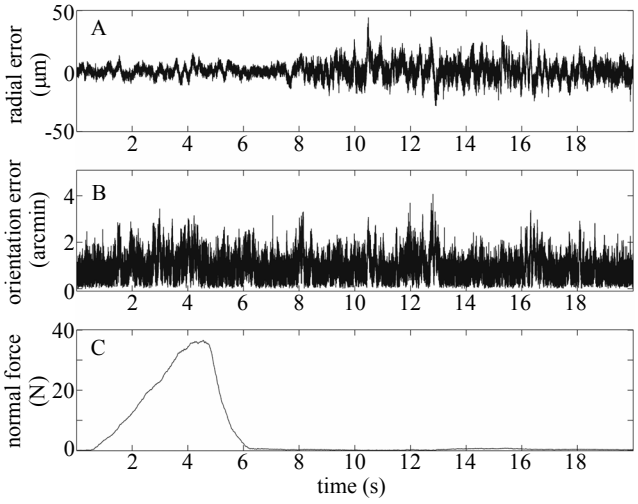

Figure 10. This is experimental data recorded from an implementation of Example 2 with the Cobotic Hand Controller. All data is computed from "virtual" coordinates, or physically meaured parameters of the display. Again, the position resolution of the Hand Controller is 25 μm. A) Translational deviation of the disk from the surface of the sphere . B) Error in the disk orientation as computed from the angle between the surface normal of the sphere and surface normal of the disk at their contact point. C) Normal force applied by the operator along the contact normal. Note that the cobot is able to effectively ignore this large force, incurring no significant position error.

These simulations and others have striking realism when displayed on the Cobotic Hand Controller. The smooth rigid feeling of the constraints that the Cobotic Hand Controller displays can hardly be expressed in plots of data. The user feels both inertial and viscous forces that can be varied over a wide dynamic range. The proper coupling of translational and rotational dynamics in response to a generalized force, composed of both forces and torques is conveyed.

V. CONCLUSIONS

Users of haptic displays should be able to interact with systems having significant inertial dynamics and realistic rigid constraints. In this paper we have shown how to accurately display constrained dynamic systems. The methods provided can be extended to NURBS-type representations of haptic environments. In addition to rigid bodies, the framework presented here can also be used to simulate the configuration-dependent inertia of linkages.

A physics simulation and integration method has been outlined and a feedforward term computed for the reference tool. Subsequently a combined feedforward and feedback controller for the tracking error of a virtual tool relative to a reference tool was derived for an admittance-controlled system. Example constraint scenarios were provided, and data reported from their implementation on the Cobotic Hand Controller.

Ongoing research with the Cobotic Hand Controller will quantify its bandwidth and stable impedance range over the space of virtual stiffness, virtual damping and virtual inertia parameter values. Also underway is research to characterize the impact on impedance range of both the rotational-to-linear-motion continuously variable transmission elements utilized in the Cobotic Hand Controller's joints, and the exploitation of the redundancy in the actuation architecture of the Cobotic Hand Controller [3, 16].

REFERENCES

[1] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA: MIT Press, 1992.

[2] T. H. Massie and J. K. Salisbury, "The PHANTOM haptic interface: a device for probing virtual objects," in *Proc. International Mechanical Engineering Exposition and Congress*, Chicago, 1994, 295-302.

[3] E. L. Faulring, J. E. Colgate, and M. A. Peshkin, "A high performance 6-DOF haptic cobot," in *Proc. IEEE International Conference on Robotics and Automation*, New Orleans, LA, 2004, 1980-1985.

[4] H. Yano, (2004, February) Technical Performance of Haptic Master. [Online]. Available: http://intron.kz.tsukuba.ac.jp/vrlab_web/hapticmaster/detail/perform.html

[5] L. B. Rosenberg, "Virtual fixtures: perceptual overlays enhance operator performance in telepresence tasks," Ph.D. dissertation, Stanford University, Stanford, 1994.

[6] C. B. Zilles and J. K. Salisbury, "A Constraint-based God-object Method For Haptic Display," in *Proc. International Conference on Intelligent Robots and Systems*, 1995, 146-151.

[7] D. Ruspini, K. Kolarov, and O. Khatib, "The Haptic Display of Complex Graphical Environments," *Computer Graphics*, vol. 31, pp. 345-352, 1997.

[8] D. Ruspini and O. Khatib, "A framework for multi-contact multi-body dynamic simulation and haptic display," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000,

[9] D. Baraff, "Analytical methods for dynamic simulation of non-penetrating rigid bodies," *Computer Graphics*, vol. 23, pp. 223-232, 1989.

[10] G. Liu and Z. Li, "A unified geometric approach to modeling and control constrained mechanical systems," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 574-587, 2002.

[11] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, Inc., 1994.

[12] P. E. Crouch and R. Grossman, "Numerical integration of ordinary differential equations on manifolds," *Journal of Nonlinear Science*, vol. 3, pp. 1-33, 1993.

[13] R. B. Gillespie, J. E. Colgate, and M. A. Peshkin, "A general framework for cobot control," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 391-401, 2001.

[14] M. A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. A. Moore, R. B. Gillespie, and P. Akella, "Cobot architecture," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 377-390, 2001.

[15] W. Wannasuphoprasit, R. B. Gillespie, J. E. Colgate, and M. A. Peshkin, "Cobot Control," in *Proc. International Conference on Robotics and Automation*, Albuquerque, NM, 1997, 3571-3576.

[16] C. A. Moore, Jr., M. A. Peshkin, and J. E. Colgate, "Cobot implementation of virtual paths and 3D virtual surfaces," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 347-351, 2003.