

Haptic Display of Constrained Dynamic Systems via Admittance Displays

Eric L. Faulring, *Member, IEEE*, Kevin M. Lynch, *Senior Member, IEEE*, J. Edward Colgate, *Member, IEEE*, and Michael A. Peshkin, *Member, IEEE*

Abstract— In the *Cobotic Hand Controller*, we have introduced an admittance display that can render very high impedances (up to its own structural stiffness). This is due to its use of infinitely variable transmissions. While admittance displays typically excel at rendering high impedances, the incorporation of infinitely variable transmissions in the *Cobotic Hand Controller* allows the stable display of a wide dynamic range, including low impedances. The existence of a display that excels at rendering high impedance constraints but has high-fidelity control of low impedances tangent to those constraints has led us to describe an admittance control architecture not often examined in the haptics community. In this paper we present an algorithm that enables rendering of rigid motion constraints while simultaneously preserving the physical integrity of the intended inertial dynamics tangent to those constraints, unlike conventional impedance control algorithms that focus on rendering reaction forces along contact normals with constraints. We examine this algorithm here, which is general to all admittance displays, and report on its implementation with the *Cobotic Hand Controller*. We offer examples of rigid bodies and linkages subject to holonomic and/or nonholonomic constraints.

I. INTRODUCTION

WE have recently introduced the *Cobotic Hand Controller* and described its capabilities as a high-fidelity six-degree-of-freedom admittance-controlled haptic interface [1].¹ Admittance-type haptic displays sense the forces and torques applied by a user and control motion in response. Admittance displays are reviewed in [4], [5], [6] and the *Haptic Master* [7] and *Steady Hand Robot* [8] are notable implementations of the admittance paradigm. Admittance displays usually contain a fixed-ratio transmission of significant reduction and are therefore non-backdrivable and consequently have limited dynamic range. Cobots however, use infinitely variable transmissions to relate motion of their joints and can therefore render a broad range of impedances by varying backdrivability. While cobots excel at rendering smooth rigid virtual surfaces [9], or bilateral constraints on motion, they are also able to render low impedances, actively simulating a virtual environment with little friction or mass to impede the operator. In this paper we present an algorithm enabling admittance displays with broad dynamic range to simultaneously render rigid bilateral constraints and accurately portray

inertial dynamics of virtual rigid bodies and linkages tangential to those constraints.

Although well-engineered admittance devices may have a higher dynamic range than their dual, impedance displays, they are rare due to cost and complexity. The required multi-degree-of-freedom force sensors, as well the extra engineering, machining and precision components involved in producing a stiff, tight tolerance mechanism, are costly. Successful commercial haptic displays are often impedance devices, which unlike admittance displays have low inertia and are highly backdrivable. Impedance displays include most notably the *Phantom* [10] and *Whole Arm Manipulator (WAM)* [11] along with those developed by [12], [13], [14], [15], [16], [17].

While today's impedance and admittance displays may both be used to simulate a wide range of mechanical behaviors, they excel in different areas due to the nature of their control and mechanical structures. Impedance displays are well-adapted to displaying low inertia, low damping environments, but have difficulty rendering energetically passive stiff constraints [18], [19]. Conversely, admittance displays such as [7], [8] are well-adapted to displaying rigid constraints but struggle to simulate unencumbered motion (cobots being an exception as they can vary their backdrivability). Unlike impedance displays, admittance displays must actively mask inertia and friction since the operator feels a large reflected inertia of the highly geared actuator and friction due to this gearing. The presence of these effects places limits on an admittance display's ability to render low impedances.

The requirement that admittance displays must simulate inertia (mask the display inertia) increases controller complexity, but also allows for a higher level of simulation fidelity. Rarely, in practice, are inertial dynamics of a haptic display masked or made to behave like those of the virtual tool. Users of an impedance-controlled haptic display typically manipulate an end-effector whose inertia may be quite different from that of the virtual inertia. The inertia of the end-effector changes with configuration of the display while that of the virtual inertia does not. It has been the practice of the haptics community (when using impedance displays) to control forces along the contact normals between a virtual tool and constraint, and allow (not control based on a dynamic model) motion orthogonal to these. Common methods include the *god-object tracker* [20], [21] and the *virtual-proxy* [22], [23] which compute forces of constraint due to interpenetration depths. Haptic rendering of these forces and of collisions is often executed via impulsive methods [24], [25], [26]. When the forces due to the inverse dynamics in the unconstrained

Manuscript received March 15, 2006. This work was supported by the DOE grant number DE-FG07-01ER63288.

The authors are with the Department of Mechanical Engineering at Northwestern University.

¹The word *cobot* is a contraction of collaborative and robot, meaning shared control between a human operator and a computer, and denotes a robot architecture introduced in [2], [3].

directions are neglected, 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 ignored by the designer. However, the Coriolis and inertial forces of the haptic display become increasingly apparent as velocities and accelerations increase. Conversely, algorithms designed for or amenable to the admittance paradigm ([27], [28], [29], [30], [31]) compute motions in the unconstrained directions, focusing solely on the forward-dynamics response (motion) to user applied forces, ignoring the constraint forces altogether as computed-torque control is often unnecessary for highly non-backdrivable admittance displays.

In this paper, we take on the problem of simulating inertial systems, including those subject to holonomic and nonholonomic constraints, similar to the methods of [31], [32], [33]. We provide a complete algorithm required for the dynamically correct rendering of a desired inertia matrix and constraint description via an admittance display, along with worked examples and their implementation on the Cobotic Hand Controller. In Section II, we define the manipulator task space and virtual environment frames of reference. The constrained Euler-Lagrange equations describing the inertial, constraint, and operator applied forces are reviewed. The Pfaffian description of holonomic and non-holonomic constraints is reviewed. A parametric description of constraints is provided to prevent drift off of holonomic constraints, and a feedback motion controller proposed. In Section III, three example constrained dynamic scenarios are worked through the framework presented in Section II, demonstrating the use of the framework to display holonomic and nonholonomic constraints as well as rigid-body and linkage inertias. Finally, in Section IV we provide data from the implementation of the examples on the Cobotic Hand Controller admittance display.

II. CONSTRAINED DYNAMIC SIMULATION

A. Haptic system framework

Figure 1 illustrates terminology we use to describe the operation of a haptic system. We term the physical device manipulated by the user the *manipulandum*, with task space coordinates x . For the six-degree-of-freedom Cobotic Hand Controller, coordinates x are the \mathbb{R}^6 coordinate representation of the $SE(3)$ workspace of the manipulandum. The *virtual tool* in the virtual environment has n generalized coordinates q . The virtual environment coordinates, q , are related to the coordinates of the *reference manipulandum*, x_r , by the kinematics

$$x_r = \varphi(q). \quad (1)$$

Due to control errors, the actual (i.e., measured) manipulandum coordinates, x , may not precisely coincide with the reference (i.e., desired) manipulandum coordinates, x_r .

The $\dim(q)$ may be less than the $\dim(x)$ (e.g., $\dim(q)=2$ if we render a $2R$ mechanism with an independent set of coordinates, even though a haptic display with $\dim(x)=6$ may be used). The use of an independent set of coordinates, q , allows the inertia matrix of the linkage to be more easily described. Thus we may embed the holonomic constraints, that define a linkage from a multi-body system, into the

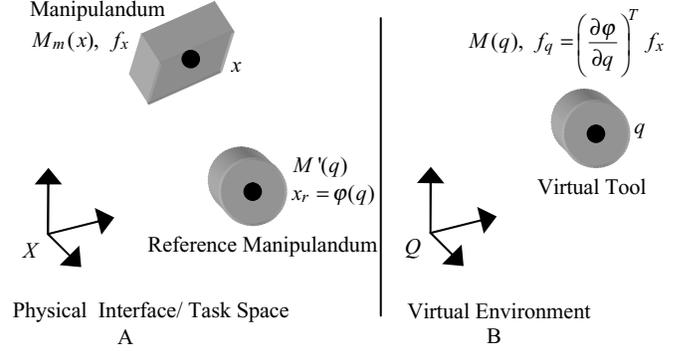


Fig. 1. A. Physical interface or task space coordinate frame. B. Virtual environment coordinate frame. The task space version of the simulated inertia $M(q)$, is $M'(q) = (\partial\varphi/\partial q)^{-T} M(q) (\partial\varphi/\partial q)^{-1}$, the operational space inertia [34]. The unmasked inertia matrix of the physical manipulandum is $M_m(x)$

kinematic mapping, $x_r = \varphi(q)$, between independent virtual and dependent manipulandum task space coordinates.

The reference manipulandum moves according to a physics simulation propagated in the virtual environment coordinates, and always exactly satisfies the virtual holonomic and non-holonomic constraints. We say that the holonomic constraints are “exactly satisfied” since the bilateral constraints to be rendered here are infinitely rigid algebraic constraints in the virtual environment, and are not described via springs or dampers.

The control scheme described in this paper can be summarized as follows: The user applies a generalized force, f_x , to the manipulandum, which is 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_q .

$$f_q = \left(\frac{\partial \varphi}{\partial q}\right)^T f_x \quad (2)$$

This mapping is single valued provided $\dim(q) \leq \dim(x)$ and that the row rank of $\frac{\partial \varphi}{\partial q} \geq \dim(q)$. We do not address the extension of this work to handle simulation of redundant virtual mechanisms.

Assuming that x is approximately x_r (one of the purposes of the feedback portion of the controller is to keep the manipulandum close to the reference manipulandum), the generalized force f_q is directly applied to the virtual tool at q .² The acceleration of the virtual tool, \ddot{q} , is then calculated via the constrained Euler-Lagrange equations and integrated forward. The acceleration is transformed by the kinematics φ to an acceleration of the manipulandum, \ddot{x} . This acceleration is a feedforward term applied to the reference manipulandum. In addition to this feedforward acceleration, a feedback acceleration is applied to compensate for small position and velocity errors between the manipulandum and the reference manipulandum. The result is a realistic display of the constrained dynamics of the virtual tool. A block diagram

²Thus the perception of inertia for an operator located at $\varphi^{-1}(x)$ will not significantly deviate from the desired configuration dependent inertial force computed at position $q = \varphi^{-1}(x_r)$.

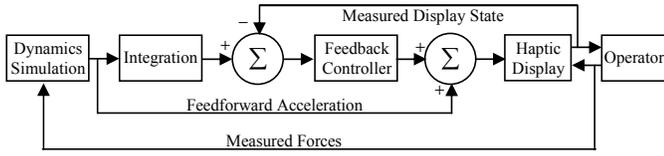


Fig. 2. The admittance-controlled haptic display renders motions in response to operator applied forces. Measured forces are the input to a dynamics simulation of a virtual environment. The output of the dynamics simulation yields a feedforward acceleration, and state for comparison with the measured state of the haptic display via a feedback controller.

of the admittance-type haptic control scheme described here is shown in Figure 2.

B. Euler-Lagrange formulation

The Euler-Lagrange dynamic equations

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = \tau + A(q)^T \lambda, \quad (3)$$

written in terms of n generalized coordinates q , and including m holonomic and/or nonholonomic constraints,

$$A(q)\dot{q} = 0, \quad (4)$$

describe the motion of the virtual tool. Note that the usual inner product of \mathbb{R}^3 cannot be used to relate applied force and desired motion when the coordinates, q , and inertial description, $M(q)$, represent topologies and inertial descriptions other than \mathbb{R}^3 and point masses [35]. The Euler-Lagrange equations must be used to relate force and motion via the desired inertial properties and constraints. The inertial forces are $M(q)\ddot{q}$ and $C(q, \dot{q})\dot{q}$. The constraint force magnitudes are represented by the $m \times 1$ vector λ . The forces not due to constraints or inertial properties (i.e., those due to an operator, virtual springs, dampers or gravity) are encompassed in τ .

C. Inertial properties

$M(q)$ is an $n \times n$ symmetric positive definite inertia matrix, preferably written in independent coordinates (prior to the inclusion of the constraints $A(q)$) for the simulated mechanical device (e.g., a rigid body or linkage). The inertia matrix,

$$M_{ij}(q) = \frac{\partial^2 T(q, \dot{q})}{\partial \dot{q}_i \partial \dot{q}_j} \quad i, j = 1 \dots n, \quad (5)$$

is obtained from the system kinetic energy, $T(q, \dot{q})$. The Coriolis forces of the virtual rigid body or linkage are

$$C(q, \dot{q})_{ij} = \sum_{k=1}^n \Gamma_{ij}^k(q) \dot{q}_k, \quad (6)$$

written in terms of the Christoffel symbols

$$\Gamma_{ij}^k(q) = \frac{1}{2} \left(\frac{\partial M_{ij}(q)}{\partial q_k} + \frac{\partial M_{ik}(q)}{\partial q_j} - \frac{\partial M_{kj}(q)}{\partial q_i} \right). \quad (7)$$

$i, j, k = 1 \dots n$

D. Nonholonomic and holonomic constraints

The virtual holonomic constraints,

$$H'(x) = 0, \quad (8)$$

and virtual nonholonomic constraints,

$$A'(x)\dot{x} = 0, \quad (9)$$

are natively defined in task space coordinates x . These constraints are written much more intuitively for rigid-body motion in task space coordinates, x , rather than in terms of a reduced set of virtual environment coordinates, q (e.g., a designer would use translational task space coordinates to constrain the end-point of a virtual linkage to move in a circle, and not the native joint angles of the linkage). The holonomic constraints can be differentiated and included in the $A'(x)$ matrix and we do so for the remainder of this work. The holonomic constraints reduce both the number of available motion freedoms and the dimension of the configuration space. The nonholonomic constraints also reduce the number of motion freedoms but do not reduce the dimension of the configuration space. The combined set of constraints are known as the Pfaffian constraints.

The dynamic equations will be written and solved in the coordinates q , so the Pfaffian constraints are projected to this subspace via

$$A(q) = A'(\varphi(q)) \left(\frac{\partial \varphi}{\partial q} \right). \quad (10)$$

Any rows of $A(q)$ that have all zeros after the projection can be removed.

Given n generalized coordinates, q , and m constraints, the tangent space at a given configuration has $n - m$ motion freedoms. $A(q)$ is an $m \times n$ matrix of Pfaffian constraints, either holonomic and/or nonholonomic, and λ is the vector of Lagrange multipliers representing the m constraint force magnitudes. The rows of $A(q)$ are the constraint force directions. If the rows of $A(q)$, $a_i(q)$, $i = 1 \dots m$, 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 $h_i(q) = c_i$.

E. External forces

External forces $\tau = f_q - \tau_d - \tau_s - g(q)$ are composed of user interaction forces, f_q , forces of virtual springs, τ_s , forces of virtual dampers, τ_d , and gravity forces or potential fields $g(q)$. Generalized user interaction forces are measured via a force torque sensor in an admittance-type haptic display. The springs and dampers may be defined in task space and the resulting forces mapped to the virtual environment via $(\partial \varphi / \partial q)^T$, or can be defined natively in virtual environment coordinates (e.g., damping applied to the end-point of a simulated linkage would likely be applied in task space, while damping at the joints of a linkage would be defined in the virtual environment coordinates of that linkage).

F. Solving for accelerations

Several steps are required for propagating the physics simulation of a virtual tool that is subject to constraints. The input to the simulation is the current state of the virtual tool, q and \dot{q} , as well as forces, f_q , applied by an operator. The output is the resulting acceleration, \ddot{q} , based on the simulated inertia, damping, springs, gravity and constraints. This acceleration is then integrated in a manner consistent with the configuration submanifold specified by holonomic constraints (Section II-G), and becomes the new state of the physics simulation.

In order to solve for the accelerations \ddot{q} , we first evaluate the Lagrange multipliers, λ , which are isolated by differentiating $A(q)\dot{q} = 0$ and inserting the result into the dynamic equations for \ddot{q} . This yields

$$\lambda = (AM^{-1}A^T)^{-1} \left(-\dot{A}\dot{q} + AM^{-1}(C(q, \dot{q})\dot{q} - \tau) \right). \quad (11)$$

The Lagrange multipliers, λ , represent the constraint force magnitudes due to the applied operator forces and the virtual environment dynamics. Subsequently we solve for the complete acceleration of the virtual tool,

$$\ddot{q} = M(q)^{-1} (\tau + A(q)^T \lambda - C(q, \dot{q})\dot{q}). \quad (12)$$

Plugging Equation 11 into Equation 12 provides the complete acceleration with the Lagrange multipliers removed,

$$\ddot{q} = -\tilde{A}\dot{A}\dot{q} + M^{-1}P_u(\tau - C\dot{q}), \quad (13)$$

where $\tilde{A} = M^{-1}A^T(AM^{-1}A^T)^{-1}$ and $P_u = I_{n \times n} - A^T(AM^{-1}A^T)^{-1}AM^{-1}$. The accelerations due to constraint forces are $-\tilde{A}\dot{A}\dot{q}$. The projection matrix P_u , as defined by Choset et al. [36], projects generalized forces to those that do work on the system (forces in the unconstrained directions).³ Thus the expressions $-\tilde{A}\dot{A}\dot{q}$ and $M^{-1}P_u(\tau - C\dot{q})$ yield the accelerations in the constrained and unconstrained directions, respectively.

G. Parametric formulation for integration

The reference acceleration, \ddot{q} , can now be integrated to yield \dot{q} and q . The integration method must keep the virtual tool on the constraint submanifold. We use a parametric approach in order to effectively integrate the equations of motion and to keep the reference position on the constraint submanifold. If b of the m constraints in $A(q)$ correspond to holonomic constraints, there exists an $(n - b)$ -dimensional submanifold Z of reachable configurations. The coordinates $z_j, j = 1 \dots (n - b)$ parameterize the configuration space of the constrained system, a submanifold of the ambient space X . A set of n functions $x = \psi(z)$ define the parametric description. As we define the Pfaffian constraints intuitively in task space as $A'(x)$, followed by a mapping to virtual environment space as $A(q)$, we also define the parametric description for the holonomic constraint surface in terms of task space coordinates x . These various spaces are illustrated in Figure 3. Generalized accelerations, \ddot{q} , are related to device

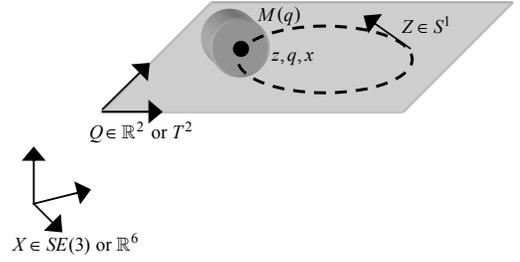


Fig. 3. The embedding of Q and Z in X illustrated through an example. A two-dimensional virtual environment frame Q is embedded in the $SE(3)$ or \mathbb{R}^6 task space, X , of our haptic display. Frame Q is that of a Cartesian (\mathbb{R}^2) or two-rotational (T^2) mechanism. The end-point is limited to travel on a circular path, the one-dimensional (S^1) submanifold Z , due to a holonomic constraint defined in task space. The virtual mechanism has native inertia $M(q)$ and is located at coordinates z, q and x in the various spaces.

accelerations \ddot{x}_r , via the kinematics

$$\ddot{x}_r = \frac{\partial \varphi}{\partial q} \ddot{q} + \dot{q}^T \frac{\partial^2 \varphi}{\partial q^2} \dot{q}. \quad (14)$$

These in turn are related to parametric accelerations, \ddot{z} , via the kinematics

$$\ddot{x}_r = \frac{\partial \psi}{\partial z} \ddot{z} + \dot{z}^T \frac{\partial^2 \psi}{\partial z^2} \dot{z}. \quad (15)$$

In order to integrate the parameters z from desired accelerations \ddot{q} , Equations 15 and 14 can be combined to yield

$$\ddot{z} = \left(\frac{\partial \psi}{\partial z} \right)^\dagger \left(\left(\frac{\partial \varphi}{\partial q} \ddot{q} + \dot{q}^T \frac{\partial^2 \varphi}{\partial q^2} \dot{q} \right) - \dot{z}^T \left(\frac{\partial^2 \psi}{\partial z^2} \dot{z} \right) \right). \quad (16)$$

Here $(\partial \psi / \partial z)^\dagger = ((\partial \psi / \partial z)^T (\partial \psi / \partial z))^{-1} (\partial \psi / \partial z)^T$ is the Moore-Penrose pseudo-inverse. The pseudo-inverse is merely effecting a change of coordinates, or a kinematic projection in this case. Numerical integration of \ddot{z} will yield a reference point on the configuration submanifold that is necessary in the next section 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. This projection to a parametric space and subsequent integration is essentially the *reference-cobot* method of Gillespie et al. [3].

H. Feedforward and feedback components

As shown in Figure 1, the actual manipulandum position, x , likely does not correspond exactly with the reference manipulandum position, x_r . Thus a controller is needed to make the actual manipulandum track the reference manipulandum, which is exactly tracking the dynamics simulation in the virtual environment.

The feedforward acceleration of the manipulandum, \ddot{x}_{ff} , is equal to the complete acceleration of the reference manipulandum, \ddot{x}_r . A proportional-integral-derivative (PID) feedback controller for \ddot{x}_{fb} will take the form

$$\ddot{x}_{fb} = K_p e + K_i \int e dt + K_d \dot{e}. \quad (17)$$

The tracking error, $e = (x_r - x)$, is the actual manipulandum state relative to the reference manipulandum state of the

³Khatib [34] and Liu and Li [31] are also excellent sources for more on these projection operations.

physics simulation. The reference manipulandum state, $x_r = \psi(z)$ and $\dot{x}_r = (\partial\psi/\partial z)\dot{z}$, can be utilized to generate feedback accelerations both in the constrained directions and in the free directions. K_p , K_i and K_d are feedback gain matrices.⁴ Summing the feedforward and feedback accelerations of the manipulandum yields the total acceleration command for the actual manipulandum,

$$\ddot{x} = \ddot{x}_{ff} + \ddot{x}_{fb}. \quad (18)$$

Device-specific kinematics convert task space commands \ddot{x} into joint level commands (and these into steering commands for cobots).

Algorithm 1 provides a summary of the virtual-environment simulation method. It includes the application of the Euler-Lagrange equations, the Pfaffian description of motion constraints, the parametric description of holonomic constraints and the integration and feedback control of motion. Since the haptic display is executing motion, it is imposing forces on itself through any end-effector distal to the load cell. Not shown, but required in Algorithm 1 and Figure 2 for proper dynamic display, is the fact that the weight and inertial effects of the mass of the end-effector should be counteracted, such that the forces between the object and environment or operator are properly conveyed.

Algorithm 1 Virtual environment algorithm

- Require:** Timestep T
Require: Initial parametric coordinates z_o
Require: Kinematics from virtual environment to task $x_r = \varphi(q)$
Require: Desired virtual inertial properties $M(q)$
Require: Pfaffian constraint description $A(q) = A'(\varphi(q)) \frac{\partial\varphi(q)}{\partial q}$
Require: Parametric description $x_r = \psi(z)$
Require: Damping and stiffness descriptions τ_d, τ_s
- 1: **Initialize** $z \rightarrow z_o, q \rightarrow \varphi^{-1}(\psi(z_o)), \dot{z} \rightarrow 0$ and $\dot{q} \rightarrow 0$
 - 2: **repeat**
 - 3: **Measure** x and \dot{x}
 - 4: **Measure** f_x , **Evaluate** $f_q = \frac{\partial\varphi(q)}{\partial q}^T f_x$
 - 5: **Evaluate** τ_s, τ_d and $g(q)$
 - 6: **Evaluate** $\tau = f_q - \tau_d - \tau_s - g(q)$
 - 7: **Evaluate** $M(q), C(q, \dot{q}), A(q), A(q, \dot{q}), \tilde{A}(q)$ and $P_u(q)$
 - 8: **Evaluate** $\ddot{q} = -\tilde{A}\dot{q} + M^{-1}P_u(\tau - C\dot{q})$
 - 9: **Evaluate** $\ddot{x}_{ff} = \ddot{x}_r = \frac{\partial\varphi}{\partial q}\ddot{q} + \dot{q}^T \frac{\partial^2\varphi}{\partial q^2}\dot{q}$
 - 10: **Evaluate** $\ddot{z} = (\partial\psi/\partial z)^T (\ddot{x}_r - \dot{z}^T (\partial^2\psi/\partial z^2)\dot{z})$
 - 11: **Integrate** $z_{new} = z + \dot{z}T + \frac{1}{2}\ddot{z}T^2$ and $\dot{z}_{new} = \dot{z} + \ddot{z}T$
 - 12: **Evaluate** $x_r = \psi(z_{new}), \dot{x}_r = \frac{\partial\psi}{\partial z}\dot{z}_{new}$
 - 13: **Evaluate** $e = x_r - x$ and $\dot{e} = \dot{x}_r - \dot{x}$
 - 14: **Evaluate** $q = \varphi^{-1}(x_r), \dot{q} = (\partial\varphi/\partial q)^T \dot{x}_r$
 - 15: **Evaluate** $\ddot{x}_{fb} = K_p e + K_i \int e dt + K_d \dot{e}$
 - 16: **Evaluate** $\ddot{x} = \ddot{x}_{ff} + \ddot{x}_{fb}$
 - 17: **Output** Send motion commands to display
 - 18: **Delay** for haptic timestep T
 - 19: **until** haptic simulation ends
-

⁴In Faulring [37] we performed the feedback in joint space on a joint by joint basis rather than in task space coordinates. Reference motions in task space were mapped to desired joint motions and these were compared with actual joint motion. The PID gains consisted of a set of 3 scalars for each joint. This made the tuning of gains easier for the specific application, but we present the algorithm here in terms of task space coordinates for generality.

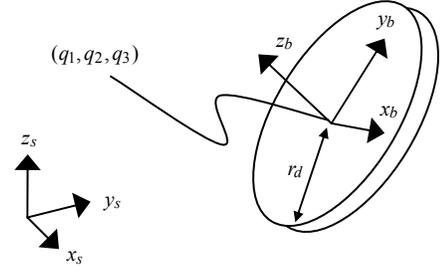


Fig. 4. Definition of translational coordinates. Subscripts s and b indicate the spatial and body frame axes. Since the virtual tool is a rigid body, frame Q and task space frame X are equivalent for this example.

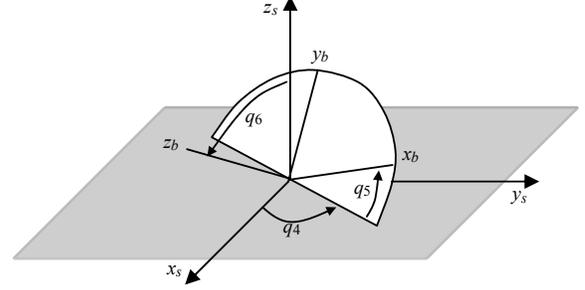


Fig. 5. Definition of rotational coordinates.

III. EXAMPLE CONSTRAINT SCENARIOS

A. Upright rolling disk

In order to demonstrate a physics simulation consisting of both holonomic and nonholonomic constraints, we consider a disk with inertia $M(q)$ as a rigid body in $SE(3)$, confined to rolling upright on a plane without slipping. Figures 4 and 5 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. The notation s indicates the generalized coordinate inertial (spatial) frame, and b the body frame of the disk. Although dynamics are to be implemented for a rigid body in this example, we have chosen an \mathbb{R}^6 coordinate representation and not the special Euclidean group $SE(3)$. Our choice of coordinates leads to integration problems near Euler angle singularities. However, we have avoided much more complex integration issues for the implicit $SO(3)$ representation of orientation [38], [39], [40]. Euler angle coordinate representation singularities are not an issue here since they can be placed outside of the workspace of our device. We focus on our algorithm, not the shortcomings of a particular choice of coordinates. The 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 and followed by a rotation of q_5 about the new z_b allows transformations from the body to the inertial frame. The mapping $x = \varphi(q)$ will be identity for this example if we also describe the task space of our display by three translational coordinates (x_1, x_2, x_3) and three Euler angles (x_4, x_5, x_6) of the same parameterization.

The kinetic energy of a rigid body in terms of coordinates q is defined

$$T(q, \dot{q}) = \frac{m}{2}(\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2) + \frac{1}{2}\omega(q, \dot{q})^T J\omega(q, \dot{q}). \quad (19)$$

Here m is the desired mass of the virtual tool and J the constant body fixed inertia. The body fixed angular velocities, $\omega(q, \dot{q})$, are extracted from the skew symmetric matrix $\hat{\omega}(q, \dot{q}) = R(q)^T \dot{R}(q, \dot{q})$ (see Appendix I). Evaluating Equation 5 for our choice of Euler angles yields

$$M(q) = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & M(q)_{44} & M(q)_{45} & M(q)_{46} \\ 0 & 0 & 0 & M(q)_{54} & M(q)_{55} & 0 \\ 0 & 0 & 0 & M(q)_{64} & 0 & M(q)_{66} \end{bmatrix}. \quad (20)$$

$$M(q)_{44} = (J_{xx}s^2q_5 + J_{yy}c^2q_5)s^2q_6 + J_{zz}c^2q_6$$

$$M(q)_{45} = M(q)_{54} = cq_6J_{zz}$$

$$M(q)_{46} = M(q)_{64} = cq_5(J_{xx} - J_{yy})sq_5sq_6$$

$$M(q)_{55} = J_{zz}$$

$$M(q)_{66} = c^2q_5J_{xx} + s^2q_5J_{yy}$$

The rotational inertia principle mass moments of the body frame inertia J are J_{xx} , J_{yy} and J_{zz} . The abbreviations sq and cq are short for $\sin(q)$ and $\cos(q)$. Equations 6 and 7 allow computation of the Coriolis matrix but we leave out this complex expression for brevity.

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

$$q_3 = r_d \rightarrow \dot{q}_3 = 0 \quad (21)$$

and

$$q_6 = \frac{\pi}{2} \rightarrow \dot{q}_6 = 0. \quad (22)$$

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

$$r_d \dot{q}_5 \cos(q_4) + \dot{q}_1 = 0 \quad (23)$$

and

$$r_d \dot{q}_5 \sin(q_4) + \dot{q}_2 = 0. \quad (24)$$

The constraint matrix is given by

$$A(q) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & r_d \cos(q_4) & 0 \\ 0 & 1 & 0 & 0 & r_d \sin(q_4) & 0 \end{bmatrix}. \quad (25)$$

Since the virtual environment inertia for this example is that of a rigid body, our mapping from virtual environment space to task space, $x_r = \varphi(q)$, is the identity map. Thus the constraint descriptions $A'(x)$ and $A(q)$ are identical, and we have written them directly in terms of q . The reachable configuration submanifold is now four-dimensional and can be parameterized

$$x_r = \psi(z) = \begin{bmatrix} z_1 \\ z_2 \\ r_d \\ z_3 \\ z_4 \\ \frac{\pi}{2} \end{bmatrix}. \quad (26)$$

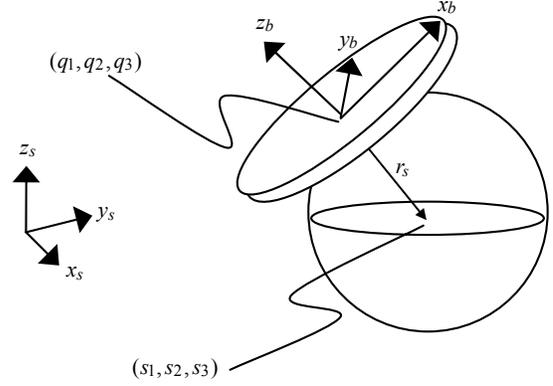


Fig. 6. This virtual environment consists of a disk required to stay tangent to a sphere, but allowed to spin about the contact normal. Subscripts s and b indicate the spatial and body frame axes.

If the designer would like to incorporate forces due to damping or stiffness, Appendices I and II can be utilized to generate τ_d and τ_s , translational and rotational damping and stiffness natively defined in $SE(3)$. These forces will have an impact on motion only if they have components tangential to the constraints.

B. Disk sliding on sphere

As a second example of the algorithm presented here, consider a disk as shown in Figure 6, whose center must stay in contact with a sphere of radius r_s located at (s_1, s_2, s_3) . The reference disk can move in two degrees of freedom about the sphere that it is constrained to, and can 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 4 and 5. In order to establish the constraint equations for this scenario, consider the rotation matrix $R(q_4, q_5, q_6)$ multiplied by the vector $(0, 0, r_s)^T$ in the body fixed frame,

$$\begin{bmatrix} \dots & \dots & \sin(q_5) \sin(q_6) \\ \dots & \dots & \cos(q_5) \sin(q_6) \\ \dots & \dots & \cos(q_6) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ r_s \end{bmatrix} = \begin{bmatrix} q_1 - s_1 \\ q_2 - s_2 \\ q_3 - s_3 \end{bmatrix}. \quad (27)$$

This yields the current inertial frame 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 entries of the rotation matrix irrelevant to this example are not shown. We can then pull out three holonomic constraint equations (Equations 28 through 30).

$$q_1 - r_s \sin(q_5) \sin(q_6) - s_1 = 0 \quad (28)$$

$$q_2 - r_s \cos(q_5) \sin(q_6) - s_2 = 0 \quad (29)$$

$$q_3 - r_s \cos(q_6) - s_3 = 0 \quad (30)$$

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

$$A(q) = \begin{bmatrix} 1 & 0 & 0 & 0 & -r_s \cos(q_5) \sin(q_6) & -r_s \sin(q_5) \cos(q_6) \\ 0 & 1 & 0 & 0 & r_s \sin(q_5) \sin(q_6) & -r_s \cos(q_5) \cos(q_6) \\ 0 & 0 & 1 & 0 & 0 & r_s \sin(q_6) \end{bmatrix}. \quad (31)$$

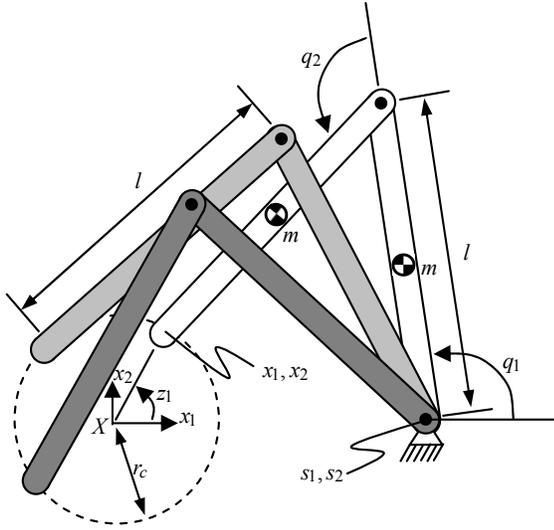


Fig. 7. This virtual environment consists of a 2R manipulator, with base affixed in the X frame at (s_1, s_2) , whose endpoint is constrained to move along a circle of radius r_c in task space.

Again, since the transformation φ between Q and X is identity for this example, we have written $A(q)$ directly rather than $A'(x)$. The reachable configuration submanifold is now three-dimensional and can be parameterized

$$x_r = \psi(z) = \begin{bmatrix} s_1 + r_s \sin(z_2) \sin(z_3) \\ s_2 + r_s \cos(z_2) \sin(z_3) \\ s_3 + r_s \cos(z_3) \\ z_1 \\ z_2 \\ z_3 \end{bmatrix}. \quad (32)$$

Again, if the designer would like to incorporate forces due to damping or stiffness, Appendices I and II can be utilized to generate wrenches τ_d and τ_s on a rigid body.

C. 2R manipulator constrained to a circle

As a third example of the algorithm presented here, consider a 2R manipulator, with two links of length, l , each with uniformly distributed mass, m , as shown in Figure 7. The kinetic energy of the manipulator,

$$T(q) = \frac{ml^2}{6} ((5 + 3 \cos(q_2)) \dot{q}_1^2 + (2 + 3 \cos(q_2)) \dot{q}_1 \dot{q}_2 + \dot{q}_2^2), \quad (33)$$

is written as the sum of the translational and rotational energy in both links. The virtual environment space inertia matrix,

$$M(q) = \begin{bmatrix} \frac{ml^2}{3} (5 + 3 \cos(q_2)) & \frac{ml^2}{6} (2 + 3 \cos(q_2)) \\ \frac{ml^2}{6} (2 + 3 \cos(q_2)) & \frac{ml^2}{3} \end{bmatrix}, \quad (34)$$

is computed using Equation 5. The virtual environment space Coriolis matrix,

$$C(q, \dot{q}) = \begin{bmatrix} -\frac{ml^2}{2} \sin(q_2) \dot{q}_2 & -\frac{ml^2}{2} \sin(q_2) (\dot{q}_1 + \dot{q}_2) \\ \frac{ml^2}{2} \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}, \quad (35)$$

is similarly computed from the kinetic energy via Equations 6 and 7. For this example, the Jacobian $(\partial\varphi/\partial q)$ between

virtual environment coordinates and task space coordinates, is no longer identity. Rather, the functions φ are given by

$$x = \varphi(q) = \begin{bmatrix} s_1 + l \cos(q_1) + l \cos(q_1 + q_2) \\ s_2 + l \sin(q_1) + l \sin(q_1 + q_2) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (36)$$

The projection $(\partial\varphi/\partial q)^T$ will now be non-square, and will map the six-dimensional force and torques of the user in task space to the two torques on the virtual manipulator. As stated earlier, this mapping is single valued provided $\dim(q) \leq \dim(x)$ and that the row rank of $\frac{\partial\varphi}{\partial q} \geq \dim(q)$. Conversely, once the dynamics have been performed in the reduced set of coordinates, Equation 14 will project the manipulator motion up from virtual environment coordinates to task space coordinates.

In Figure 7, the end-point of the 2R manipulator is constrained to a circle of radius r_c . This holonomic constraint expressed in task space coordinates is

$$x_1^2 + x_2^2 = r_c^2. \quad (37)$$

This may be written as a Pfaffian constraint,

$$A'(x) = [2x_1 \ 2x_2 \ 0 \ 0 \ 0 \ 0], \quad (38)$$

and mapped to virtual environment coordinates,

$$A(q)^T = \begin{bmatrix} 2l((c(q_1 + q_2) + cq_1) s_2 - (s(q_1 + q_2) + sq_1) s_1) \\ 2l(c(q_1 + q_2) s_2 - s(q_1 + q_2) s_1 - lsq_2) \end{bmatrix}, \quad (39)$$

via $A(q) = A'(\varphi(q))(\partial\varphi/\partial q)$. The reachable configuration submanifold is one-dimensional and is parameterized

$$x_r = \psi(z) = \begin{bmatrix} r_c \cos(z_1) \\ r_c \sin(z_1) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (40)$$

Again, if the designer would like to incorporate forces due to damping or stiffness applied to a rigid-body end-effector of the simulated linkage, Appendices I and II could be utilized to generate wrenches τ_d and τ_s , and these then projected via $(\partial\varphi/\partial q)^T$ to the virtual environment coordinates. Alternately, the designer may add either damping or stiffness directly in the virtual environment coordinates (e.g., friction or compliance at the joints of the simulated linkage).

IV. EXPERIMENTATION WITH THE COBOTIC HAND CONTROLLER

The three examples from Sections III-A, III-B and III-C were implemented on the Cobic Hand Controller. The implementations used slightly different notation than described above, since a different Euler angle parameterization was used [37].

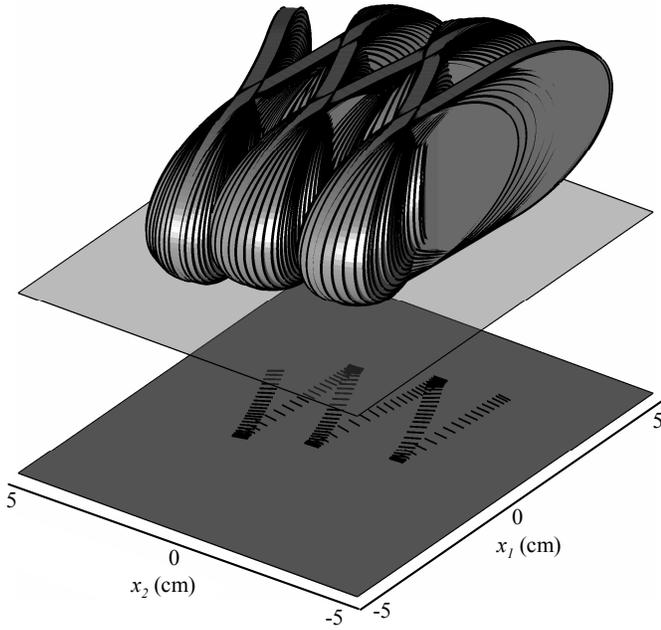


Fig. 8. A disk shown at 0.1 second intervals in rolling contact with a planar surface. The shadow shows the line contact of the disk with the plane, or “tire tracks,” as the disk makes “parallel parking” motions.

A. Upright rolling disk, continued

The six-centimeter-diameter disk in Figure 8 is in rolling contact with the plane. It is allowed to spin and roll, but is unable to slide sideways. It is able to move sideways by “parallel parking” motions, or by simply turning and driving in the desired direction. The simulated disk has mass 0.25 kg, all principal-axis inertias are 0.0025 kgm², translational damping is 1.0 Nsm⁻¹, and rotational damping is 0.1 Nmsrad⁻¹.⁵ In Figure 9, 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 are on the order of the position resolution of the Cobotic Hand Controller. Thus these infinitely rigid constraints are rendered correctly. This performance is in significant contrast to impedance displays that can only render constraints of a few kN/m. Also reported is the percent error in the rolling constraint, $0.01|v - r_d\omega|/(|v| + \epsilon)$, where $\epsilon = 0.001$ m/s, $r_d\omega = -(0.03)\dot{x}_5$ and $v = \dot{x}_1 \cos(x_4) + \dot{x}_2 \sin(x_4)$ is in m/s.

B. Disk sliding on a sphere, continued

Figure 10 portrays a two-centimeter-diameter disk constrained to remain tangent to a ten-centimeter-diameter sphere, but permitted to rotate about its contact normal with the sphere. The disk has the same inertial properties as in the upright rolling disk example, although we have changed its dimensions. In Figure 11, several metrics indicating the feedback motion controller’s performance are reported. Again, the infinitely rigid constraints are accurately displayed by the

⁵To put these simulated values in perspective, in [37] we show that with the transmissions all set to a 1:1 ratio, or a 45 degree steering angle of the wheels, the cobot has an apparent inertia of 11 kg (this extends from 6 kg to infinity if the steering angle is varied from 90 degrees to 0 degrees), and backdrive friction forces on the order of 7.2 N.

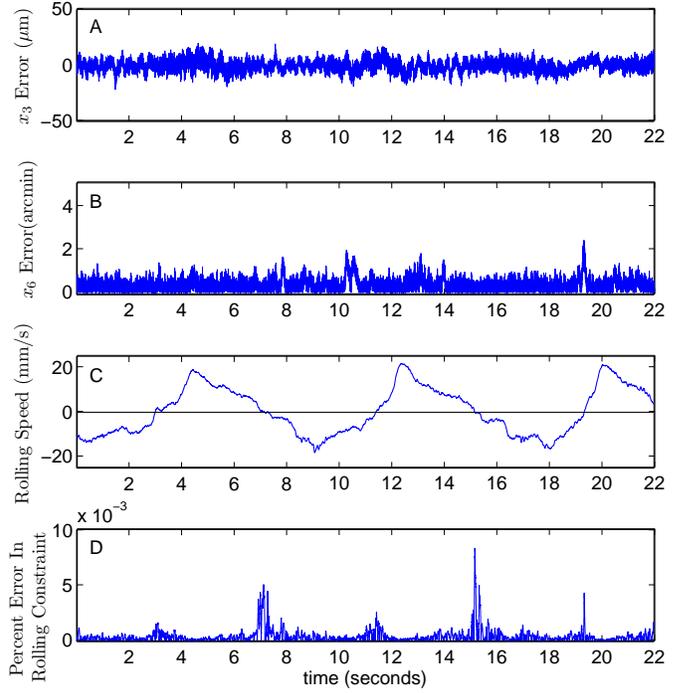


Fig. 9. Experimental data recorded during the implementation of the upright rolling disk example with the Cobotic Hand Controller. All data are computed from the measured state of the display (x and \dot{x}). A) Error in the height of disk center. B) Error in the disk’s upright orientation. The noise present in A and B is partially due to analog sensor noise and are nearly equivalent to what would be obtained when the device is not moving. C) Rolling velocity $v = \dot{x}_1 \cos(x_4) + \dot{x}_2 \sin(x_4)$ of the disk. D) The percent of error in the rolling constraint is $0.01|v - r_d\omega|/(|v| + \epsilon)$ where $r_d\omega = -(0.03)\dot{x}_5$, v is in m/s and $\epsilon = 0.001$ m/s. ϵ prevents this metric from reporting infinite error at near zero speed.

algorithm in conjunction with the Cobotic Hand Controller as is shown by the small radial errors (Subplot A of Figure 11) during the application of large radial forces (Subplot C of Figure 11).

C. 2R manipulator constrained to a circle, continued

Figure 12 depicts experimental results of the implementation of the virtual 2R manipulator constrained to a 5 cm radius circle. The links each have length 15 cm, mass 2.0 kg, and damping at the joints of 0.01 Nmsrad⁻¹. The base of the manipulator is 21 cm from the center of the circular constraint. The operator manipulates the virtual linkage for a few seconds and then releases it. The operator was attempting to maintain a constant velocity tangential to the constraint. This is difficult as it requires that the kinetic energy in the virtual manipulator rise and fall, and thus the user must alternately push and hold back the manipulator to maintain the constant tangential velocity. Once released, the kinetic energy stored in the linkage decreases monotonically due to damping at the joints, but the kinetic energy in each individual joint rises and falls. While Figure 9 parts A and B and Figure 11 parts A and B demonstrate small errors in the constrained directions, and Figure 9 part D demonstrates effective maintenance of a nonholonomic constraint, Figure 12 part C demonstrates small errors in the free motion direction, the direction in which

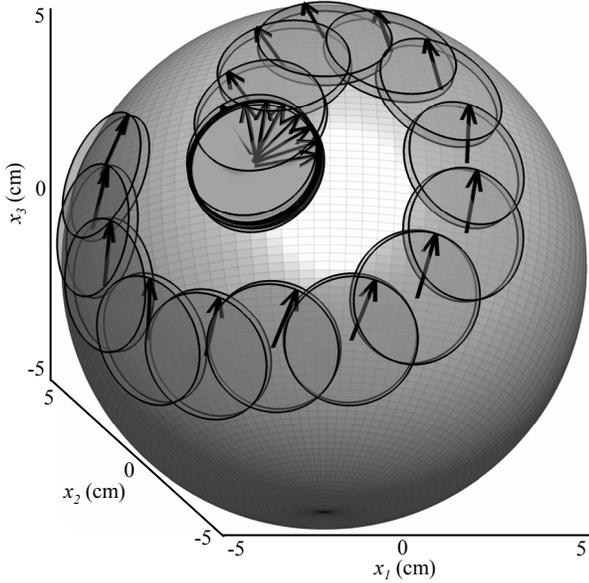


Fig. 10. A simulated disk, shown at one second intervals, is constrained to track a virtual sphere while remaining tangent to it, but is permitted to rotate about the contact normal. The arrows indicate the orientation of the disk on the sphere.

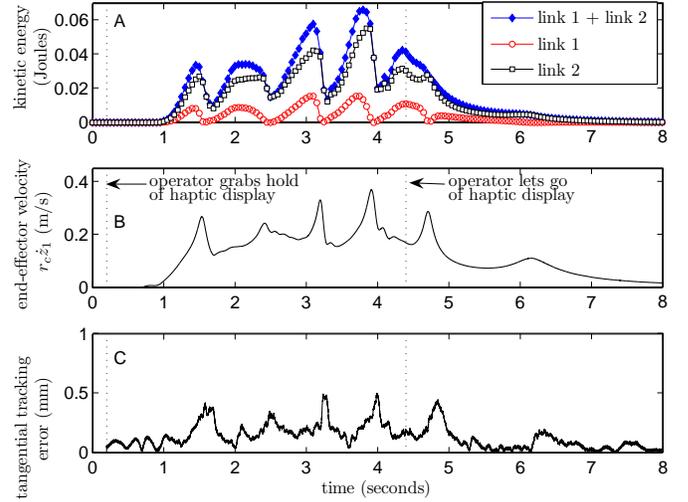


Fig. 12. The operator grabs hold of the end-effector and manipulates the virtual linkage. A) The total kinetic energy in the manipulator and the kinetic energy in the individual links. B) The speed of the end-point tangential to the circular task space constraint. C) The error between the desired and actual position of the end-effector along the circular path. Once the operator releases the linkage, the kinetic energy stored in the linkage decays monotonically due to virtual damping at the manipulator joints. Note that the kinetic energy of the base link (link 1) goes to zero each time it reverses direction. The kinetic energy of the distal link (link 2) never approaches zero as it has translational kinetic energy even when it has no rotational velocity. Note the increase in endpoint speed at times 4.7 s and 6.2 s even while the total manipulator kinetic energy is decaying.

desired dynamic behaviors are imposed.

V. CONCLUSIONS

We have outlined a virtual environment simulation and integration algorithm for the haptic display of rigidly constrained dynamic systems. Assuming the current active constraints are known, we form the constrained Euler-Lagrange equations. These allow us to solve for the dynamically correct motion in response to the measured operator applied forces with respect to the simulated inertia and active constraints. We derived a combined feedforward and feedback controller for the motion of the actual manipulandum relative to a reference manipulandum. In addition to point masses, the framework presented here can also be used to simulate the configuration-dependent inertia of rigid bodies and of linkages. Example constraint scenarios have been provided, including the first display of nonholonomic constraints with a cobot, and data derived from their implementation on the Cobic Hand Controller are reported. The user feels both inertial and viscous forces that vary over a wide dynamic range. The proper coupling of translational and rotational dynamics in response to a generalized force, which is composed of both forces and torques, is conveyed, even in the presence of constraints on motion. While this algorithm is general to all admittance displays, it was motivated by the Cobic Hand Controller haptic display. Data presented here demonstrate the ability of the Cobic Hand Controller to simultaneously render extremely stiff constraints and dynamically correct motion tangent to those constraints.

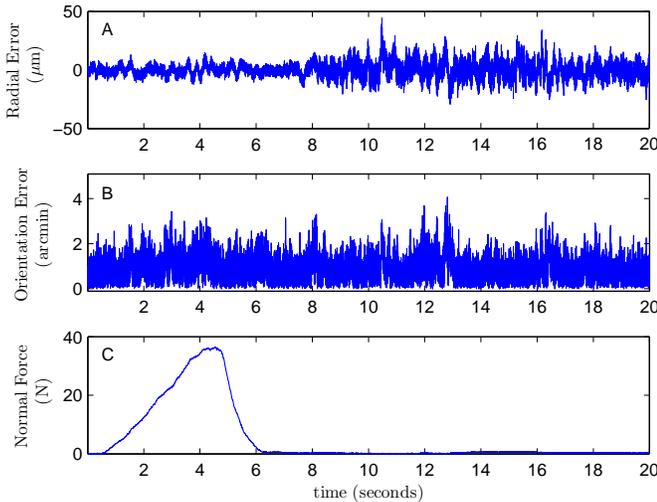


Fig. 11. Experimental data recorded from an implementation of the disk sliding on sphere example with the Cobic Hand Controller. All data are computed from the measured state of the display (x and \dot{x}). A) Translational deviation of the disk from the surface of the sphere. B) Error in the disk orientation as computed from the angle between the surface normal of the sphere and the surface normal of the disk at the point of contact. C) Force applied along the contact normal between the disk and sphere. Note the small radial error even during the application of a large radial force. This demonstrates the ability of the display to render very rigid constraints.

APPENDIX I

REPRESENTATION OF DAMPING IN $SE(3)$

A generalized damping wrench, τ_d , may be computed in the rigid body frame by

$$\tau_d = \begin{bmatrix} b_t I_{3 \times 3} & 0 \\ 0 & b_r I_{3 \times 3} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (41)$$

where b_t and b_r are the translational and rotational damping coefficients respectively. The body frame angular velocities can be pulled out of the skew symmetric matrix of body frame velocities,

$$\hat{\omega} = R^T \dot{R} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (42)$$

The vector of the body frame translational velocities is given by

$$v = R^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}. \quad (43)$$

APPENDIX II

REPRESENTATION OF STIFFNESS IN $SE(3)$

The generalized spring wrench on a rigid body, τ_s , may be computed in the body frame by

$$\tau_s = - \begin{bmatrix} k_t I_{3 \times 3} & 0 \\ 0 & k_r \theta I_{3 \times 3} \end{bmatrix} \begin{bmatrix} R^T (p_o - [q_1, q_2, q_3]^T) \\ K \end{bmatrix}. \quad (44)$$

k_t and k_r are the translational and rotational spring constants, respectively. The body frame translational spring origin, $R(q)^T (p_o - [q_1, q_2, q_3]^T)$, is computed from the world frame spring origin, p_o . K and θ are the axis-angle representation of the current orientation, where the axis and angle are computed from the net rotation from body frame to the spring origin, $\Delta R = R^T R_o$, where R_o is the spring origin [21].

$$\theta = \arccos \left(\frac{\text{trace}(\Delta R) - 1}{2} \right) \quad (45)$$

$$K = \frac{1}{2 \sin(\theta)} \begin{bmatrix} \Delta R_{32} - \Delta R_{23} \\ \Delta R_{13} - \Delta R_{31} \\ \Delta R_{21} - \Delta R_{12} \end{bmatrix} \quad (46)$$

REFERENCES

- [1] E. Faulring, J. Colgate, and M. Peshkin, "A high performance 6-DOF haptic cobot," in *IEEE International Conference on Robotics and Automation*, New Orleans, LA, 2004, pp. 1980–1985.
- [2] M. Peshkin, J. Colgate, W. Wannasuphprasit, C. Moore, R. Gillespie, and P. Akella, "Cobot architecture," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.
- [3] R. Gillespie, J. Colgate, and M. Peshkin, "A general framework for cobot control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 391–401, 2001.
- [4] V. Hayward and O. Astley, "Performance measures for haptic interfaces," in *Robotics Research: The 7th International Symposium*, 1996, pp. 195–207.
- [5] C. Carignan and K. Cleary, "Closed-loop force control for haptic simulation of virtual environments," *Haptics-e*, vol. 1, no. 2, 2000.
- [6] T. Yoshikawa, "Force control of robot manipulators," in *IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 220–226.
- [7] R. Van der Linde, P. Lammertse, E. Frederiksen, and B. Ruiters, "The HapticMaster, a new high-performance haptic interface," in *EuroHaptics*, Edinburgh, U.K., 2002.
- [8] J. Abbott, G. Hager, and A. Okamura, "Steady-hand teleoperation with virtual fixtures," in *12th IEEE International Workshop on Robot and Human Interactive Communication*, 2003, pp. 145–151.
- [9] L. Rosenberg, "Virtual fixtures: Perceptual overlays enhance operator performance in telepresence tasks," Ph.D. Dissertation, Stanford University, 1994.
- [10] T. Massie and J. Salisbury, "The PHANTOM haptic interface: A device for probing virtual objects," in *ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, vol. DSC 55, Chicago, IL, 1994, pp. 295–302.
- [11] K. Salisbury, B. Eberman, M. Levin, and W. Townsend, "The design and control of an experimental whole-arm manipulator," in *The Fifth International Symposium on Robotics Research*, 1990, pp. 233–241.
- [12] F. Dimension, "Delta Haptic Device," February 2004, http://www.forcedimension.com/fd/avs/home/products/6-dof_delta/.
- [13] R. Adams, M. Moreyra, and B. Hannaford, "Excalibur, a three-axis force display," in *ASME Winter Annual Meeting Haptics Symposium*, Nashville, TN, 1999.
- [14] Quanser, "3 DOF Planar Pantograph," August 2005, <http://www.quanser.com/english/html/products/>.
- [15] B. Adelstein and M. Rosen, "Design and implementation of a force reflecting manipulum for manual control research," in *Advances in Robotics*, ASME DSC, vol. 42, Anaheim, CA, 1992, pp. 1–12.
- [16] C. Lee, D. Lawrence, and L. Pao, "A high-bandwidth force-controlled haptic interface," in *Symposium on Haptic Interfaces for Teleoperation and Virtual Reality*, ASME IMECE, 2000.
- [17] P. Millman and J. Colgate, "Design of a four degree-of-freedom force reflecting manipulum with a specified force-torque workspace," in *IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 1488–1493.
- [18] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 426–434, 1989.
- [19] J. Brown and J. Colgate, "Passive implementation of multibody simulations for haptic display," in *ASME IMECE DSC*, vol. 61, 1997, pp. 85–92.
- [20] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995, pp. 146–151.
- [21] W. McNeely, K. Puterbaugh, and J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *ACM SIGGRAPH*, 1999, pp. 401–408.
- [22] D. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997, pp. 345–352.
- [23] D. Ruspini and O. Khatib, "Haptic display for human interaction with virtual dynamic environments," *Journal of Robotic Systems*, vol. 18, no. 12, pp. 769–783, 2001.
- [24] D. Baraff, "Analytical methods for dynamic simulation of non-penetrating rigid bodies," *Computer Graphics*, vol. 23, no. 3, pp. 223–232, 1989.
- [25] B. Mirtich and J. Canny, "Impulse-based simulation of rigid bodies," in *Symposium on Interactive 3D Graphics*, 1995, pp. 181–188.
- [26] B. Chang and J. Colgate, "Real-time impulse-based simulation of rigid body systems for haptic display," in *ASME DSC*, Houston, TX, 1997, pp. 1–8.
- [27] P. Lee, D. Ruspini, and O. Khatib, "Dynamic simulation of interactive robotic environment," in *IEEE International Conference on Robotics and Automation*, San Diego, CA, 1994, pp. 1147–1152.
- [28] P. Mitiguy and T. Kane, "Motion variables leading to efficient equations of motion," *International Journal of Robotics Research*, vol. 15, no. 5, pp. 522–532, 1996.
- [29] R. Gillespie and J. Colgate, "A survey of multibody dynamics for virtual environments," in *ASME DSC*, vol. 61, Dallas, TX, 1997, pp. 45–54.
- [30] R. Gillespie, "Kane's equations for haptic display of multibody systems," *Haptics-e*, vol. 3, no. 2, 2003.
- [31] G. Liu and Z. Li, "A unified geometric approach to modelling and control of constrained mechanical systems," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 574–587, 2002.
- [32] X. Yun and N. Sarkar, "Unified formulation of robotic systems with holonomic and nonholonomic constraints," vol. 14, no. 4, pp. 640–650, 1998.

- [33] W. Blajer, "A geometrical interpretation and uniform matrix formulation of multibody system dynamics," *Journal of Applied Mathematics and Mechanics*, vol. 81, no. 4, pp. 247–259, 2001.
- [34] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [35] J. Duffy, "The fallacy of the modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces," *Journal of Robotic Systems*, vol. 7, no. 2, pp. 139–144, 1990.
- [36] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementations (Intelligent robotics and autonomous agents)*. Cambridge, MA: The MIT Press, 2005.
- [37] E. Faulring, "The cobotic hand controller: Design, control and analysis of a novel haptic display," Ph.D. Dissertation, Northwestern University, 2005.
- [38] W. Blajer, "A geometric unification of constrained system dynamics," *Multibody System Dynamics*, vol. 1, no. 1, pp. 3–21, 1997.
- [39] P. Dehombreux, O. Verlinden, and C. Conti, "An implicit multistage integration method including projection for the numerical simulation of constrained multibody systems," *Multibody System Dynamics*, vol. 1, no. 4, pp. 405–424, 1997.
- [40] R. Muharlamov, "On the equations of kinematics and dynamics of constrained mechanical systems," *Multibody System Dynamics*, vol. 6, no. 1, pp. 17–28, 2001.

Draft