Controlling the Apparent Inertia of Passive Human-Interactive Robots

Tom Worsnopp
Graduate Student
e-mail: greycloak@northwestern.edu

Michael Peshkin
e-mail: peshkin@northwestern.edu

Kevin Lynch
e-mail: kmlynch@northwestern.edu

J. Edward Colgate
e-mail: colgate@northwestern.edu

Mechanical Engineering Department,
Northwestern University,
Evanston, IL 60208

Passive robotic devices may exhibit a spatially varying apparent inertia perceptible to a human user. The apparent inertia is the projection of the inertia matrix onto the instantaneous direction of motion. The spatial variation is due to the configuration dependence of the inertia matrix and relevant to many passive mechanisms, including programmable constraint machines or "cobots," which use low-power steering actuators to choose the direction of motion. We develop two techniques for controlling the apparent inertia in cobots to emulate the desired inertial properties of a virtual object or mechanism. The first is a path-limiting method, which constraints the cobot to steer along certain paths where the apparent inertia and desired inertia are equivalent. The second uses a low-power actuator to control the apparent inertia by driving the device along its direction of motion. We illustrate these ideas for a two-link cobot we have built for experiments in human motor control and rehabilitation. For the actuated control method, we show that the power actuator can be relatively low power compared to the actuators of a traditional robot performing similar tasks. [DOI: 10.1115/1.2168165]

Keywords: inertia ellipse, apparent inertia, passive robot, cobot, effective mass

1 Introduction

When a human user interacts with a passive manipulandum, or more generally with a robot that is less than fully actuated, the natural dynamic properties of the mechanism will be apparent to the user. This is a problem if the purpose of the robot is to emulate a virtual object or mechanism; the illusion that the user is manipulating the virtual mechanism will be broken when the experienced inertia of the robot differs from the expected inertia of the virtual mechanism.

This paper studies the experienced dynamic properties of such mechanisms and describes passive and active approaches to controlling these dynamic properties. This work is motivated by our work on cobots. Cobots are passive, human-interactive robots that use mechanical rolling contacts to implement smooth constraint surfaces [1]. We have explored cobots as haptic interfaces to virtual environments [2], as assistive devices for material handling [3], as manipulanda for teleoperation [4], and as tools for the exploration of human motion control under motion constraint [5].

The operating principle of cobots is to use computer-controlled CVTs (continuously variable transmissions) to produce high-quality rolling constraints. In some cases, the CVT is no more than a steered rolling wheel [1]. In other cases, the CVT may be a complex mechanism [4,6]. Many designs have been explored for cobots of diverse workspace dimensionalities, sizes, and other requirements [4,7–9].

Although computer steering determines the path of a cobot endpoint (e.g., handle) through the cobot's workspace, the computer has no authority over the speed of the endpoint along that path. The speed of the endpoint is determined by the external forces (including those applied by the user and by the environment, e.g., gravity) and the inherent dynamics of the cobot itself. This means that cobots are passive devices, incapable of transmitting power to the user.

In some applications, the goal is to provide the user with the convincing feeling that he is manipulating a virtual mechanism, such as a rigid body, along a smooth, stiff constraint. Although the mechanical rolling contacts provide convincing constraint surfaces, the inertial properties of the actual robot mechanism will usually differ from those of the desired virtual mechanism. The key point is that the projection of the cobot's inertia matrix in the direction of motion varies as a function of the configuration (indicated in the inertia matrix, $M(\mathbf{q})$) and a unit tangent vector $\mathbf{T}$ (according to an appropriately chosen metric)

$$m_{\text{app}} = \mathbf{T}^T M(\mathbf{q})\mathbf{T}. \tag{1}$$

We call this projected inertia, m_{app}, the *apparent inertia along the path*, and the apparent inertia of the cobot is the only inertial property of the cobot that the user can sense at any instant.

The goal of this paper is to propose and analyze techniques for controlling the apparent inertia of a cobot. The unicycle two-link arm (UTLA) cobot (shown in Fig. 1 and discussed in Sec. 3) is used as a concrete example to demonstrate the concepts outlined in this paper.

We have explored two techniques for controlling the apparent inertia of a cobot:

1. *Path Limiting*. Limit the cobot to follow only certain paths in space. This technique reduces the number of degrees of freedom available to the cobot, but allows the cobot to passively control the apparent inertia felt by the user. If the cobot is redundant with respect to the task freedoms it must emulate, then the constraint on the cobot's motion may only affect internal degrees of freedom and not be apparent to the user. (Controlling effective inertia by manipulating internal degrees of freedom was also explored in [10] to minimize damage in a collision between a robot and its surroundings.) The degree to which the apparent inertia can be controlled is limited by the geometry and mass of the cobot.
2. *Low-Power Actuation*. Power the cobot along its direction of motion, but continue to use CVTs to implement constraint surfaces. The addition of an actuator provides the cobot with the ability to directly control the sensations felt by a user along the direction of motion. A benefit of this technique is that the cobot is not limited to certain paths in the workspace

Contributed by the Dynamic Systems, Measurement, and Control Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received February 7, 2005; final manuscript received November 14, 2005. Assoc. Editor: Sunil K. Agrawal.

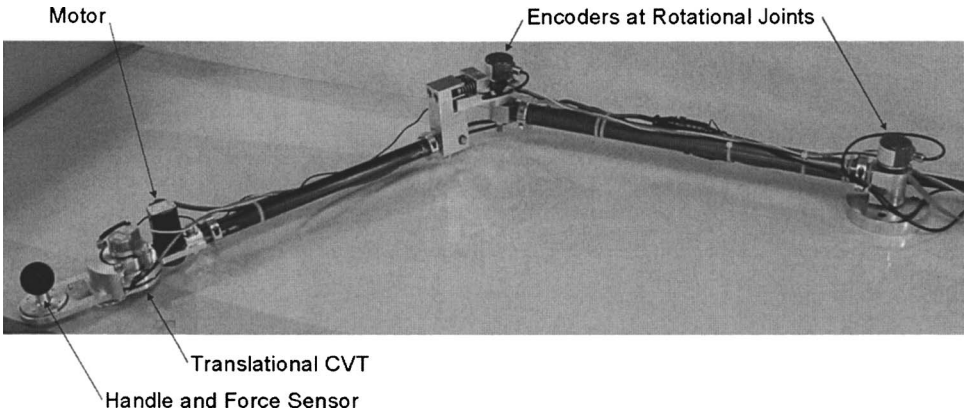

Fig. 1 The unicycle two-link arm (UTLA) cobot is a two-degree-of-freedom haptic device used to display high-quality, software-defined constraints over a workspace large enough for the full motion of a user's arm

and can explore most of the workspace (except near singularities). It is important to note that the additional actuator requires very little power to perform its function because the constraint surfaces are still passively enforced by frictional rolling constraints. The addition of this actuator does, however, mean that the cobot is no longer a purely passive device.

We should mention that the term "apparent inertia" has been used by others, often referencing techniques for varying the perceived inertia of a haptic interface or telemanipulator. For instance, Lee and Li [11,12] developed a control technique for a fully actuated conventional teleoperation robot to satisfy passivity and to give the robot desired inertial properties. We use apparent inertia in a more restrictive sense to mean the inertia perceived along a cobot's instantaneously available motion freedom.

In Sec. 2, we describe the general framework for controlling the apparent inertia of cobots and introduce the two control techniques. As an example application, in Sec. 3 we describe the UTLA cobot, a 2R cobot built for experiments in human arm motor control and rehabilitation. Users of the device expect it to feel similar to a point mass at the handle and are surprised by the varying apparent inertia. Thus, the goal of apparent inertia control for this device is to make it feel like a point mass. Section 4 describes the path-limiting method as applied to the UTLA and shows example paths that passively satisfy constant apparent inertia. Section 5 discusses the power requirements needed of a power actuator for extending the UTLA's capability to emulate desired apparent inertias. We conclude in Sec. 6.

We do not discuss implementations of the path-limiting controller and the low-power actuation controller in this paper. Path controllers have been described extensively in previous publications, and the UTLA does not currently possess a power actuator. Active control of apparent inertia is a challenging control problem, for cobots or conventional robots; see, for example, [12]. This paper also does not discuss simulation of the dynamics of the virtual mechanism being emulated; see, for example, [13].

2 Problem Formulation

Let $\mathbf{q} \in \mathbb{R}^n$ be a vector of generalized coordinates of the cobot mechanism, $\mathbf{x} \in \mathbb{R}^m$ be a vector of task space coordinates for the virtual object or mechanism we wish to emulate, and $\mathbf{x}=\mathbf{f}(\mathbf{q})$ be the kinematic mapping from cobot coordinates to task space coordinates. We assume $n \geqslant m$; the cobot must have at least as many degrees of freedom as the virtual mechanism to successfully emulate it. The inertia matrix of the cobot mechanism is $M(\mathbf{q})$ and the desired inertia of the virtual mechanism is $M_v(\mathbf{x})$.

Let $\mathbf{q}[s(t)]$ be the path taken by the cobot as the user manipulates it, where s is a path parameter. The velocity of the cobot can be expressed as $\dot{\mathbf{q}}=(d\mathbf{q}/ds)\dot{s}=T\dot{s}$, where T is the spatial path tangent. To provide the user with the convincing feeling that she is manipulating the virtual mechanism, the kinetic energy of the cobot must be equivalent to the expected kinetic energy of the virtual mechanism. (We neglect gravitational forces for the purposes of this paper.) This can be expressed as

$$\frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}} = \frac{1}{2}\dot{x}^T M_v(x)\dot{x}$$

$$T^T M(\mathbf{q})T = \left(\frac{dx}{ds}\right)^T M_v(x)\frac{dx}{ds} = \left(\frac{\partial f}{\partial q}\frac{dq}{ds}\right)^T M_v(x)\left(\frac{\partial f}{\partial q}\frac{dq}{ds}\right)$$

$$= T^T J^T M_v J T = T^T \tilde{M} T$$

where J is the Jacobian of the kinematic mapping and $\tilde{M} = J^T M_v J$ is the desired inertia of the emulated mechanism in cobot generalized coordinates.

For this relationship to be satisfied at all times, the derivatives of the left- and right-hand sides with respect to s must be equivalent at all times. Using primes to indicate differentiation with respect to s, this condition can be written (after rearranging)

$$2T^T(M-\tilde{M})\boldsymbol{\kappa} = T^T(\tilde{M}' - M')T$$

where $\boldsymbol{\kappa}=T'=d^2q/ds^2$ is the curvature of the cobot path. This equation can be written equivalently in terms of derivatives with respect to time

$$2\dot{\mathbf{q}}^T(M-\tilde{M})\ddot{\mathbf{q}} = \dot{\mathbf{q}}^T(\dot{\tilde{M}} - \dot{M})\dot{\mathbf{q}}$$

where

$$\ddot{\mathbf{q}} = M^{-1}(\mathbf{q})[\boldsymbol{\tau}_{\text{user}} + \boldsymbol{\tau}_{\text{cobot}} - C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}]$$

are the dynamics of the cobot, $\boldsymbol{\tau}_{\text{user}}$ are external forces provided by the user, and $\boldsymbol{\tau}_{\text{cobot}}$ are any active forces applied by the cobot.

We will explore two ways to satisfy these equivalent conditions. The first, the passive path-limiting method, steers the cobot's joints so that the curvature of the path $\boldsymbol{\kappa}$ satisfies the first condition. The second, the low-power actuation method, modifies the apparent inertia by the use of active forces τ_{cobot}, thereby modifying $\ddot{\mathbf{q}}$.

2.1 Path Limiting. In the path-limiting control mode, the cobot steering actuators choose any path curvature $\boldsymbol{\kappa} \in \mathbb{R}^n$ satisfying the condition $2T^T(M-\widetilde{M})\kappa = T^T(\widetilde{M}' - M')T$. All other terms in the equation are given by the current position and tangent direction of the cobot. This equation specifies one constraint on $\boldsymbol{\kappa}$, leaving an $(n-1)$-dimensional space of steering controls on the cobot consisting of any particular solution to the equation plus any element of the null space of $(T^T(M-\widetilde{M}))$. If the dimensions of the task space and the cobot's configuration space are equal $(m=n)$, this constraint manifests itself as a constraint on the possible paths that can be convincingly emulated. If the cobot is redundant $(n>m)$, however, the extra internal degrees of freedom of the cobot can be steered to maintain the desired apparent inertia along any path (up to inequality constraints and/or singularities due to the mass and geometry of the cobot).

In Sec. 4 the path-limiting method is applied to the UTLA to derive isomass contours: paths along which the apparent inertia is unchanging.

2.2 Low-Power Actuation. As the cobot is emulating a passive virtual mechanism, the user expects that work done on the virtual mechanism will result in a particular acceleration of the virtual mechanism. If the cobot is both passive and nonredundant, however, the acceleration felt by the user will not correspond to the expected acceleration—the apparent inertia will not equal the expected inertia. One solution to this problem is to provide the (otherwise passive) cobot with a single small power actuator. The control algorithm senses the forces $\boldsymbol{\tau}_{\text{user}}$ applied by the user and solves the single equation for the necessary torque to the power actuator $\boldsymbol{\tau}_{\text{cobot}}$. Additional feedback and feed-forward terms can be added to this nominal torque to track the desired inertia with greater precision in the presence of force sensor noise and control-loop bandwidth limitations. In Sec. 5, we simply look at the power requirements on the power actuator of the UTLA to emulate various point masses.

3 UTLA and Virtual Paths

3.1 Device Motivation. The unicycle two-link arm (UTLA) cobot was developed as a research tool for arm-motion studies, such as those involving human-constraint interactions and the rehabilitation of stroke patients.

In order to be useful in both of these applications, the UTLA needed to satisfy a number of constraints. The cobot had to be able to operate in the horizontal plane (so that gravity would not play a role) and had to be able to render virtual paths over a two-dimensional workspace large enough for the full motion of a user's arm. It was also desirable that the cobot have low inertia and little friction.

Based on these specifications, the UTLA was designed and built by Yambay Valiente [5].

3.2 Device Specifications. The UTLA, illustrated in Fig. 1, is a two-degree-of-freedom cobot that consists of two links and two rotational joints connected to a fixed reference frame. Located at the end of the second link is a handle and a force sensor (located beneath the handle). The crucial steerable CVT used for cobot control is a simple wheel.

The UTLA is spring loaded at the elbow to apply a constant preload to the wheel. Friction between the table and the wheel prevents motion in directions perpendicular to the wheel's rolling direction. The wheel, which is steered via a traction drive by a motor, acts as a translational continuously variable transmission, controlling the direction of motion that the cobot endpoint can move in the *xy* plane. There are no other motors.

The UTLA has three encoders—one at each of the two rotational joints, and an additional one above the wheel to monitor its steering angle. By controlling the steering angle of the CVT (or its time derivative), the endpoint of the UTLA can be constrained to any desired path. This provides the UTLA with the ability to generate virtual paths through its workspace.

The UTLA controls the steering velocity of its CVT, $\dot{\phi}$, based on the properties of a software-defined path

$$\dot{\phi} = (\mathbf{v}_{\text{CVT}} \times \boldsymbol{\kappa}_{\text{CVT}}) \cdot \hat{\mathbf{k}} \tag{2}$$

where $\mathbf{v}_{\text{CVT}}$ is the translational velocity of the CVT (as measured by the joint encoders), $\boldsymbol{\kappa}_{\text{CVT}}$ is the software-defined curvature of the path at the CVT (including feedforward and feedback control terms), and $\hat{\mathbf{k}}$ is a unit vector normal to the *xy* plane in which the cobot operates. See [14] for more details about the implementation of a virtual path controller for the UTLA. Additional information about UTLA (including videos) may be found on the Web [15].

3.3 Varying Apparent Inertia. After the path controller had been implemented on the UTLA, a number of paths were tested and it became apparent that the nonlinear dynamics of the cobot were being felt by the user. For example, when the cobot endpoint is constrained to a circular path, as illustrated in Fig. 2 users report that the cobot seems to be "rising and falling" as it traverses the path.

The cobot is not actually rising and falling, however. The sensation corresponds to an increase and decrease in the speed of the cobot endpoint, which users interpreted as being due to variations in surface height, even though the surface is flat.

This problem is not specific to the UTLA. Any passive device with a configuration-dependent apparent inertia will experience these types of effects. For example, rehabilitation devices, such as the three-degree-of-freedom (3-DOF) mechanism in [16] and the passive balance mechanism presented in [17], will both experience variations in their apparent inertia as they move through their configuration spaces.

The change in speed felt by the user can be explained intuitively. Since the UTLA has no mechanism for storing energy, all energy associated with it is kinetic. Since it moves freely along its constrained path with very low friction, users apply very little force to the endpoint. The kinetic energy of the cobot is approximately constant as the user gently pushes it around the circular path. Note, however, that there are two places on the circle where link 1 must come to a stop and reverse its direction of motion (see Fig. 3). As link 1 stops, all of the kinetic energy must transfer to link 2. This results in an increase in angular velocity of link 2 and, therefore, an increase in speed of the endpoint. The user interprets this speeding up as if it was caused by a low point on the rolling surface.

To make the UTLA appear to the user as a point mass, we could simply place a heavy mass at the handle, dominating the configuration-dependent component of the inertia. This would require a very large mass, however, which is undesirable. For that reason, we describe the path-limiting and low-power actuation methods for the UTLA.

4 Path Limiting

4.1 Isomass Contours Defined. Based on the example in Sec. 3, it is apparent that there are paths for which the nonlinear dynamics of the cobot are felt by a user. Such sensations would be a distraction in the study of human-constraint interaction or stroke rehabilitation. There are, however, special paths along which the cobot feels like it has a constant mass. We refer to such paths as *isomass contours*.

4.2 Dynamics of the UTLA. To implement isomass contours on the real cobot, we use a full dynamic model of the cobot consisting of four mass parameters (two masses and two inertias of the two links). To simplify the equations in the derivation of the isomass contours, however, we adopt a two-parameter model that lumps the mass of each link at its distal endpoint. These lumped

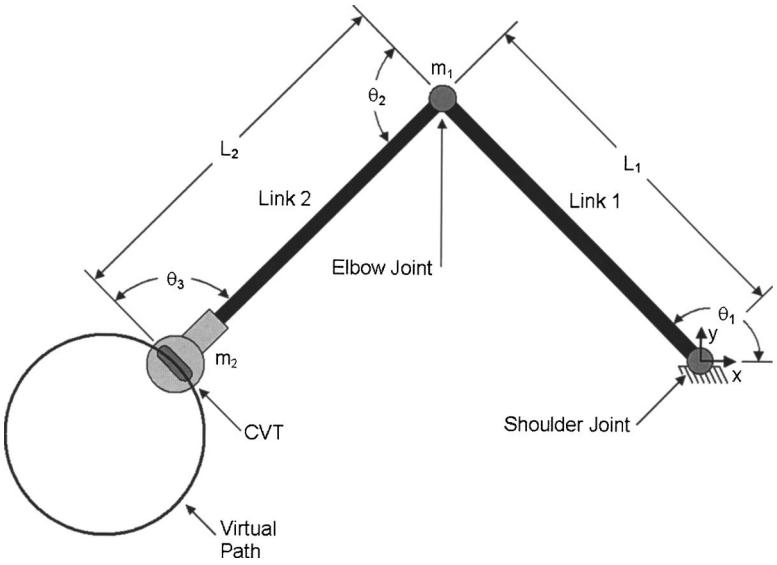

Fig. 2 The motion of the cobot is unintuitive to a user when following certain virtual paths. When the endpoint moves around a circular path, for example, it feels as if it is rising and falling. These effects are the result of the apparent inertia varying.

point masses have been chosen so that the inertia of the simplified model near the center of the workspace is very similar to the inertia of our more complete model, using mass distributions from a CAD model. Figure 2 illustrates this model of the UTLA, where $L_1=0.652$ m, $L_2=0.794$ m, $m_1=4.72$ kg, and $m_2=1.57$ kg. Qualitatively, the isomass contours for the two models are the same, but the simplified model allows us to demonstrate the main ideas behind the path-limiting method without unnecessarily cumbersome equations.

The configuration of the cobot can be specified by the (x,y) coordinates, with the UTLA cobot always in a right-handed configuration. The equations of motion of the UTLA—expressed in Cartesian coordinates $\mathbf{q}=(x,y)$ at the endpoint for convenience—can be written

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \tag{3}$$

where $\boldsymbol{\tau}$ is the force vector applied at the endpoint, $M(\mathbf{q})$ is the inertia matrix in terms of the endpoint coordinates, $\ddot{\mathbf{q}}$ is the acceleration of the endpoint, and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ is a vector containing centrifugal and Coriolis terms.

4.3 Apparent Inertia. In Sec. 1, we defined the apparent inertia as the projection of the inertia matrix onto the direction of motion

$$m_{\text{app}} = \mathbf{T}^{\mathbf{T}} M(\mathbf{q})\mathbf{T} \tag{4}$$

where the inertia matrix of the UTLA, expressed in the joint angles θ_1 and θ_2 as defined in Fig. 2, is

$$M(\theta) = \begin{bmatrix} \dfrac{m_1 + m_2 - m_2\cos(2\theta_2) + m_1\cos[2(\theta_1+\theta_2)]}{2\sin^2(\theta_2)} & \dfrac{m_1\sin[2(\theta_1+\theta_2)]}{2\sin^2(\theta_2)} \\ \dfrac{m_1\sin[2(\theta_1+\theta_2)]}{2\sin^2(\theta_2)} & \dfrac{m_1 + m_2 - m_2\cos(2\theta_2) - m_1\cos[2(\theta_1+\theta_2)]}{2\sin^2(\theta_2)} \end{bmatrix} \tag{5}$$

The unit vector in the direction of motion by the (Euclidean) kinetic energy metric associated with our desired virtual object (a point mass) is

$$\mathbf{T} = \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix} \tag{6}$$

and ϕ is the angle between the direction of motion of the wheel and the x-axis. Thus, the apparent inertia of the UTLA can be computed as

$$m_{\text{app}} = \frac{m_1 + m_2 - m_2\cos(2\theta_2) + m_1\cos[2(\theta_1+\theta_2-\phi)]}{\sin^2(\theta_2)} \tag{7}$$

Inspection of (7) reveals that it is possible to determine the direction of motion ϕ of the minimum and the maximum apparent inertia, for a given configuration (θ_1 and θ_2 are dictated by the configuration $\mathbf{q}$). The minimum occurs when the fourth term in (7), $m_1\cos[2(\theta_1+\theta_2-\phi)]$, is smallest, and maximum when this term is greatest.

Therefore, the direction of motion for the minimum apparent inertia corresponds to $\phi=\theta_1+\theta_2-\pi/2$, which is always perpendicular to link 2 and corresponds only to rotations about the elbow joint. The direction of motion for the maximum apparent inertia corresponds to $\phi=\theta_1+\theta_2$, which is always parallel to link 2 and, therefore, perpendicular to the direction of motion for the minimum apparent inertia.

Figure 4 illustrates the apparent inertia of the UTLA as a func-

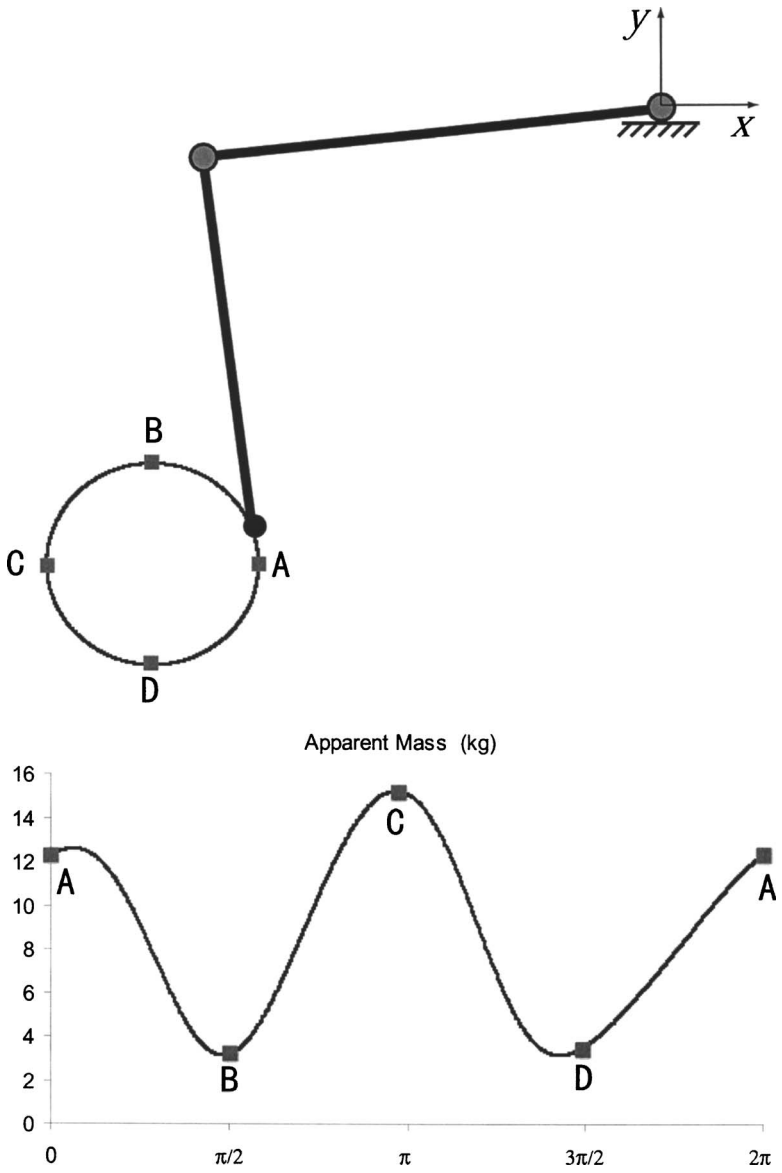

Fig. 3 The apparent inertia of the cobot varies between 3 kg and 15 kg as a circle is traversed. The radius of the circle shown is 20 cm.

tion of θ_2 and ϕ. Note that the shoulder joint angle θ_1 is fixed at zero. The plot of apparent inertia for any other shoulder angle θ_1 is identical with the ϕ axis replaced by $\phi - \theta_1$. Also note that the plot does not extend to the workspace boundaries, where the apparent inertia of the UTLA goes to infinity.

4.4 Inertia Ellipses. Figure 5 illustrates the apparent inertia in terms of another, perhaps more familiar, representation: an inertia ellipse [18–20]. In this representation, the size of the ellipse in any direction is inversely proportional to the square root of the apparent inertia in that direction. Thus, the major axis of the inertia ellipse corresponds to the direction of minimum apparent inertia and the minor axis to the direction of maximum apparent inertia. The inertia ellipse at any given configuration represents the velocity vectors that result in the same kinetic energy.

Using this representation, it is possible to graphically determine the directions of motion that have a particular apparent inertia by intersecting the inertia ellipse of the cobot with the inertia ellipse of the virtual mechanism. In our case, the virtual mechanism is a point mass, which has a circular inertia ellipse. Figure 6 illustrates the endpoint of the cobot and its associated inertia ellipse. A dashed circle is used to illustrate the inertia ellipse for a point mass, and arrows are used to indicate the directions along which the cobot endpoint feels identical to the point mass.

4.5 Isomass Contours Analyzed. The concepts above set the stage for how to generate an isomass contour. Given a configuration, we can determine the directions of motion in which the apparent inertia of the cobot matches that of the point mass. This can be accomplished by solving for ϕ in (7). The computed configuration and directions of motion can then be used as a starting point from which we can numerically generate the isomass contours.

In order to generate smooth isomass contours, we must also determine the rate of change of the direction of motion along which the apparent inertia is invariant. If we know the rate of change of the direction of motion, we can numerically integrate it to compute each adjacent point on the isomass contour from a starting configuration.

It is possible to solve for the rate of change of the direction of motion, $d\phi/ds$, along which the apparent inertia is invariant by

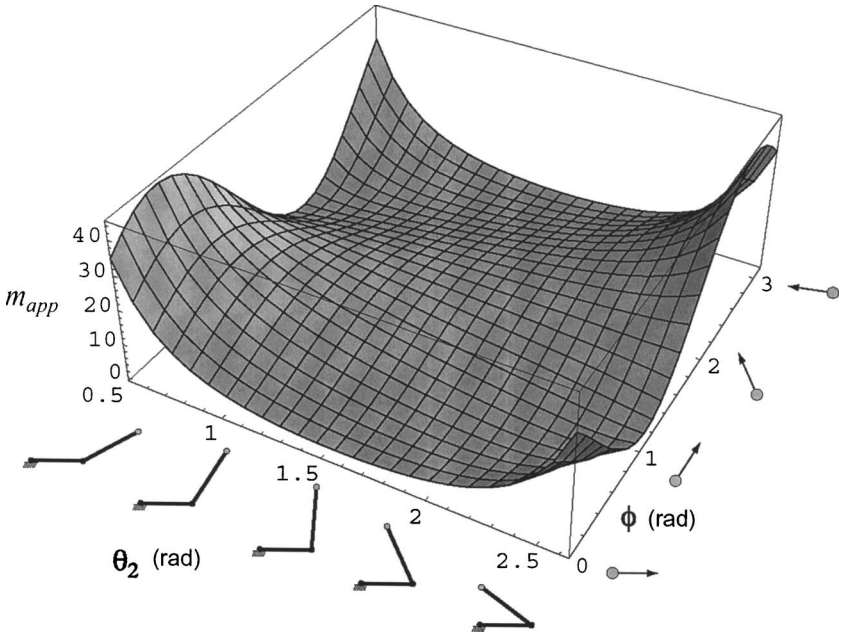

Fig. 4 A three-dimensional plot of the apparent inertia (in kilograms) of the simplified UTLA model is shown as a function of the second joint angle (θ_2) and the angle of motion (ϕ) of the endpoint. The shoulder joint angle (θ_1) has no affect on the apparent inertia (it only shifts the values) and, thus, is excluded from consideration.

setting the derivative of the apparent inertia, dm_{app}/ds, with respect to path parameter s, to zero. In the case of the UTLA, the rate of change of direction of motion, $d\phi/ds$, is

$$\frac{d\phi}{ds} = \frac{d\theta_1}{ds} + \frac{d\theta_2}{ds}\left[\frac{\cos(\theta_1 - \phi)}{\sin(\theta_2)\sin(\theta_1 + \theta_2 - \phi)}\right] \tag{8}$$

We can numerically integrate this differential equation to find isomass contours starting from a given initial configuration and with a given apparent inertia.

Figure 7 illustrates a pair of isomass contours numerically computed using this technique. In this example, the apparent inertia along the iso-mass contours is $m_{app}=3.30$ kg. The four branches that extend away from the starting configuration correspond to the directions in which the apparent inertia feels like a 3.30 kg point mass (i.e., the directions in which the inertia ellipse intersects the point mass circle).

We see that three ends of the two isomass contours terminate at the workspace boundaries (indicated by the large and small dashed circles). The fourth branch, however, does not. It instead spirals toward a limit cycle circle centered at the shoulder joint. It is worth noting that there exist sections of the contours that are approximately straight lines through the workspace. Such contours have potential for utility in human-constraint interaction studies, where movement along a straight line is of interest [21].

Also illustrated in Fig. 7 is the inertia ellipse for the starting configuration ($x=-1.027$ m, $y=0.0$ m). For the sake of comparison, an additional inertia ellipse is illustrated for a different starting configuration. The inertia ellipses for both configurations are drawn with their diameters tangent to the path indicated by the two double-headed arrows. Despite the differences in the shapes

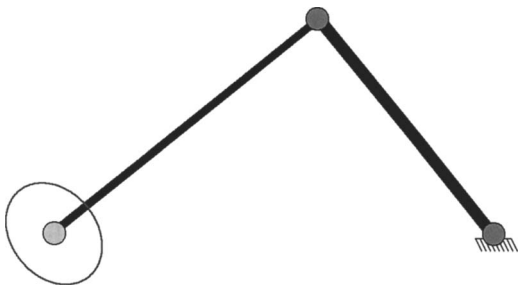

Fig. 5 The long axis of the inertia ellipse corresponds to a direction of low apparent inertia, and the short axis corresponds to a direction of high apparent inertia

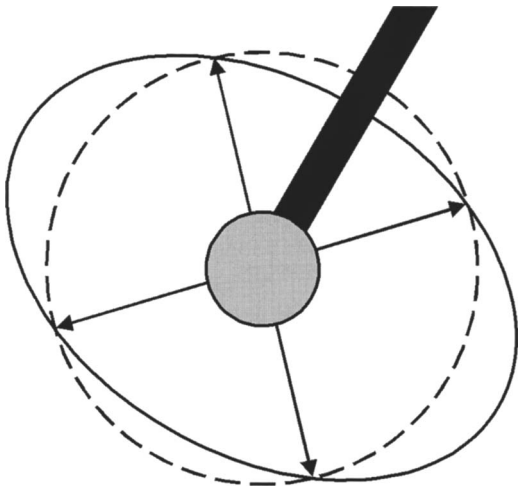

Fig. 6 The intersection of the inertia ellipse of the cobot with the inertia ellipse of a point mass (i.e., a circle) can be used to determine the directions of motion in which the cobot will feel like the point mass. These directions are indicated by the arrows. Note that the cobot cannot emulate masses that are too large (small inertia circles) or too small (large inertia circles).

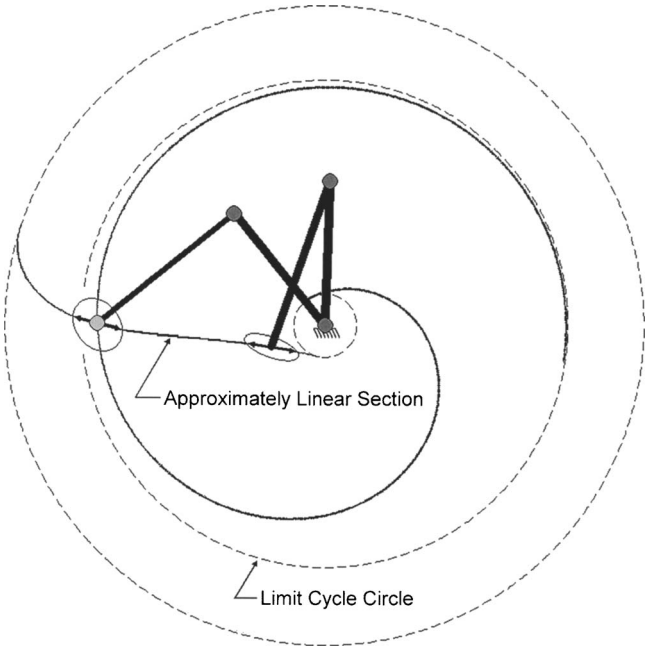

Fig. 7 The UTLA cobot is shown with two different inertia ellipses for two different configurations. The diameter of both ellipses tangent to the isomass contour are the same length, as indicated by the two double-headed arrows. A set of isomass contours is illustrated with the UTLA positioned in a starting configuration for an apparent inertia of m_{app}=3.30 kg. Note that three ends of the two contours terminate at the boundaries of the workspace, while the fourth one spirals toward a limit cycle circle centered at the shoulder.

of the inertia ellipses, the lengths of the diameters along the path are the same, which is indicative of the fact that the path has a constant apparent inertia.

4.6 Summary of Path-Limiting Technique. We have presented a technique for generating isomass contours for the UTLA cobot. Although this example is specific to the UTLA cobot, the general technique presented in Sec. 2 can be used to control the apparent inertia of any cobot throughout its workspace. If the cobot is redundant, the steering constraint may not be observable to the user; it may only affect internal degrees of freedom, allowing the exploration of arbitrary paths. An alternative technique, which requires an additional power motor, discussed next, allows a nonredundant cobot to emulate arbitrary inertias over large portions of its workspace.

5 Low-Power Actuation

If we attach a driving motor to the UTLA's wheel, allowing the cobot to drive itself forward and backward, we can directly control the apparent inertia of the device along arbitrary paths. In this section, we calculate the power that such a motor would require for typical interaction tasks. The result is a relatively low-power motor.

In order for a comparable fully powered manipulandum to control its dynamics and implement constraints over as large a workspace as the UTLA, it must be equipped with very large actuators. For example, the manipulandum described in [22] uses two 3 KW motors (PMI Motion Technologies, Model No. JR24M4CH) to implement constraints and manipulate the dynamics of a patient's hand and arm. In contrast, the UTLA, with the addition of a power motor, could perform comparable tasks with motors more than an order of magnitude smaller in power: a 70 W motor (an Allen-Bradley Y-1002-1 motor and 2098-DSD-005 drive) is currently used to steer the CVT, and the calculations below illustrate that an 80 W motor could be used to control the dynamics of the UTLA over most of its workspace.

If we think in terms of kinetic energy, the additional motor serves to add energy to the system when the apparent inertia of the UTLA is greater than that of the point mass and subtract kinetic energy when the apparent inertia is less than that of the point mass. The addition of energy would cause the UTLA to accelerate, whereas the removal of energy would cause the UTLA to decelerate.

It should be noted that the goal of this section is to demonstrate the feasibility of using two low-power actuators (one to control the nonholonomic constraint and one to control the dynamics) rather than two large actuators. In this section, we do not discuss the implementation of a low-power actuation controller as we have not implemented such a controller on the UTLA (the UTLA does not currently posses a power actuator). However, path controllers have been described extensively in previous publications.

5.1 Maximum Motor Power. In order to determine the maximum power the motor requires to operate over the entire workspace, we need to determine how much torque the motor must apply and how fast the motor must rotate. When considered in the workspace, the power can be computed as follows:

$$P_{\text{motor}} = \boldsymbol{\tau}_{\text{cobot}} \cdot \mathbf{v}_{\text{wheel}} \tag{9}$$

where P_{motor} is the power of the motor, $\boldsymbol{\tau}_{\text{cobot}}$ is the task space force applied by the motor through the wheel in the direction of motion, and $\mathbf{v}_{\text{wheel}}$ is the velocity vector along which the wheel is translating in the direction of motion.

With the addition of the motor to the UTLA, there are two force sources: the user and the motor. The equation of motion for the UTLA with a motor is

$$\boldsymbol{\tau}_{\text{cobot}} + \boldsymbol{\tau}_{\text{user}} = \{[M(\theta)\ddot{\mathbf{q}} + \mathbf{C}(\theta, \dot{\theta})\dot{\theta}] \cdot \mathbf{T}\}\mathbf{T} \tag{10}$$

where $\boldsymbol{\tau}_{\text{user}}$ is the user applied force in the direction of motion, $M(\theta)$ is the inertia matrix, $\ddot{\mathbf{q}}$ is the acceleration vector of the endpoint, $\mathbf{T}$ is a vector indicating the direction of motion, and $\mathbf{C}(\theta, \dot{\theta})\dot{\theta}$ is a vector containing centrifugal and Coriolis terms in the $\mathbf{q}$ coordinates. $\mathbf{C}(\theta, \dot{\theta})\dot{\theta}$ for the UTLA is

$$\mathbf{C}(\theta, \dot{\theta})\dot{\theta} = \begin{bmatrix} \dfrac{m_1 \cos(\theta_1 + \theta_2)[L_1 \cos(\theta_2)\dot{\theta}_1^2 + L_2(\dot{\theta}_1 + \dot{\theta}_2)^2]}{\sin^2(\theta_2)} \\ \dfrac{m_1 \sin(\theta_1 + \theta_2)[L_1 \cos(\theta_2)\dot{\theta}_1^2 + L_2(\dot{\theta}_1 + \dot{\theta}_2)^2]}{\sin^2(\theta_2)} \end{bmatrix} \tag{11}$$

Since a user interacting with the UTLA expects to feel a point mass, $\boldsymbol{\tau}_{\text{user}}$ can be expressed as

$$\boldsymbol{\tau}_{\text{user}} = (m_{pt}\ddot{\mathbf{q}}_{pt} \cdot \mathbf{T})\mathbf{T} \tag{12}$$

where m_{pt} is the point mass the user expects to interact with and $\ddot{\mathbf{q}}_{pt}$ is the acceleration the user expects to undergo when moving the point mass. Substituting (12) into (10) provides us with an equation for the force that the motor must apply,

$$\boldsymbol{\tau}_{\text{cobot}} = \{[M(\theta)\ddot{\mathbf{q}}_{pt} + \mathbf{C}(\theta, \dot{\theta})\dot{\theta} - m_{pt}\ddot{\mathbf{q}}_{pt}] \cdot \mathbf{T}\}\mathbf{T} \tag{13}$$

Using Eq. (13), it is possible to numerically compute the power for every direction of motion, at every configuration. Since the configurations of the endpoint at the same radial distance will have the same maximum power, we need only compute the maximum power for all radial distances of the endpoint.

In order to compute the power at a given configuration and given direction, we must assign a few parameters. We must know how much force a user is likely to apply and how fast a user is likely to move their hand. Experimenting with the UTLA indicated that users are unlikely to move their hands faster than

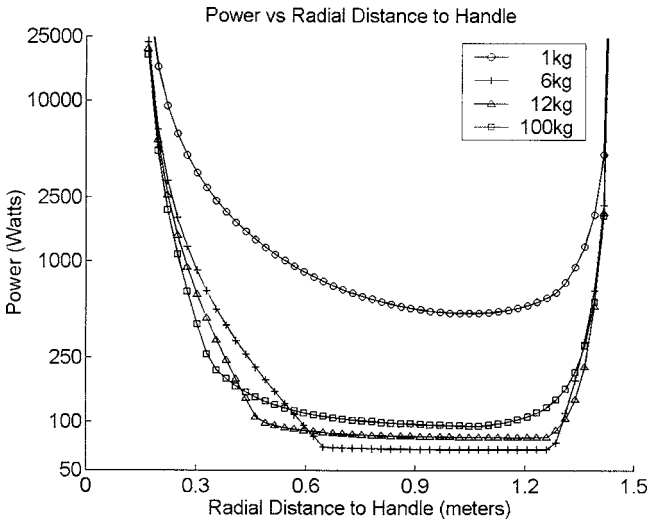

Fig. 8 Numerically computed maximum power for a wide range of point mass values for every configuration

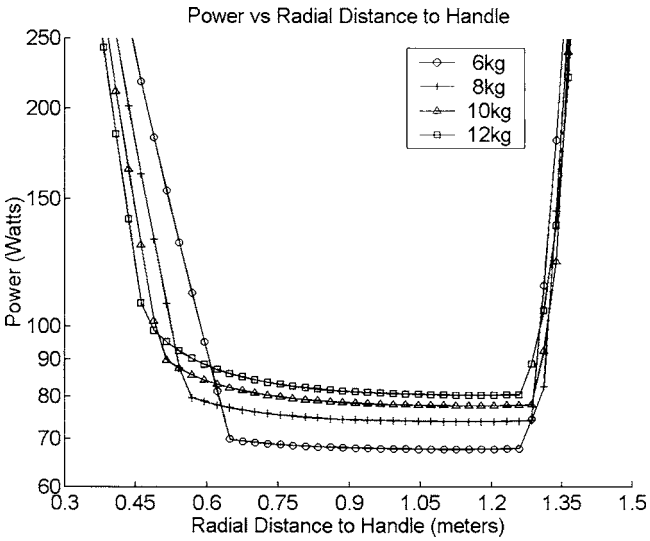

Fig. 9 Numerically computed maximum power for a smaller range of point mass values for every configuration

1.5 m/s or apply forces greater than 60 N. These values are for healthy subjects and provide a fairly liberal calculation for a motor power.

The only remaining parameter to assign is the size of the point mass with which we want the user to interact. This parameter has a significant effect on the maximum motor power.

Figure 8 shows the power requirements for different point masses. Emulating a small point mass, 1 kg for example, requires a very powerful motor. Even limiting the endpoint to a radial distance between 0.7 m and 1.2 m, an 800 W motor would be required. The power is necessary to overcome the actual inertia of the UTLA and make it accelerate as quickly as a 1 kg point mass would.

At the other extreme, emulating a mass of 100 kg would allow the UTLA to operate over most of its workspace using a motor smaller than 250 W. The downside to emulating such a large mass is that the cobot would be unresponsive to user input. The mass we would most likely emulate is between these two extremes: a point mass with low power requirements, but that is still responsive to user input.

Figure 9 shows power requirements over a smaller range of point masses. This range of point masses provides us with some reasonable options for both motor power and workspace coverage. If we emulate a point mass of 8 kg, for example, we can explore the workspace from a radius of 0.53 m to 1.27 m using an 80 W motor. This very large workspace (0.7 m^2) is illustrated in Fig. 10. The physical boundaries (edges of the table) are illustrated on the top, bottom, and right sides. Additionally, the radial boundaries are shown as solid circular arcs. The singularities are illustrated as dotted arcs.

The figures in this section assume zero mass for the power

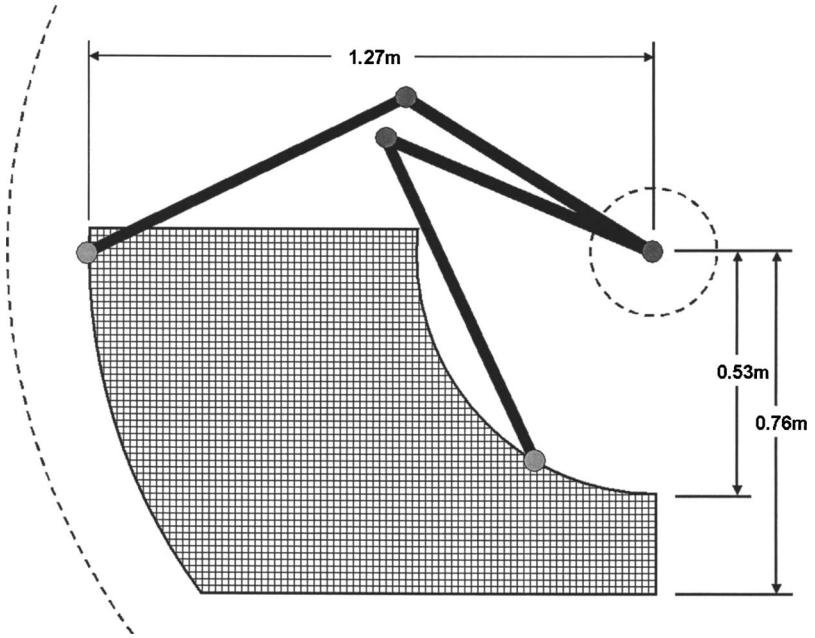

Fig. 10 Using an 80 W motor, the UTLA can emulate an 8 kg mass over a very large workspace (0.7 m^2). The boundaries of the workspace are physically limited by the table edges on the top, bottom, and right side.

motor. Nonzero mass would change the plots somewhat, making the most power-efficient mass to emulate a somewhat larger value.

5.2 Summary of Low-Power Actuation Technique. The low inertia, stiffness, and friction of the UTLA in the direction of motion means it is possible to use a small motor to modify the dynamics of the UTLA. Additionally, the CVT used to passively implement stiff constraint surfaces requires a very small motor to steer the wheel. This means that the UTLA with two low-power motors (one for steering, one for modifying the apparent inertia) should be able to perform the same functions as a comparable manipulandum which requires two massive motors. The result could be a safer and cheaper manipulandum for human-robot interaction research.

6 Conclusion

We have presented two techniques for controlling the apparent inertia of passive or "nearly passive" robotic mechanisms. These techniques are particularly applicable to cobots, and they make possible the development of very low-power, safe, and inexpensive manipulanda for human-robot interaction research.

Acknowledgment

This work was supported under NSF Grant No. IIS0082957.

References

[1] Colgate, J. E., Wannasuphoprasit, W., and Peshkin, M., 1996, "Cobots: Robots for Collaboration With Human Operators," *International Mechanical Engineering Congress and Exposition*, ASME, New York, pp. 433–440.

[2] Moore, C. A., Peshkin, M. A., and Colgate, J. E., 2003, "Cobot Implementation of Virtual Paths and 3D Virtual Surfaces," IEEE Trans. Rob. Autom., **19**(2), pp. 347–351.

[3] Wannasuphoprasit, W., Akella, P., Peshkin, M., and Colgate, J. E., 1998, "Cobots: A Novel Material Handling Technology," *Proceedings of IMECE*, ASME, New York, pp. 171–178.

[4] Faulring, E., Colgate, J. E., and Peshkin, M. A., 2004, "A High Performance 6-DOF Haptic Cobot," *IEEE International Conference on Robotics and Automation*, IEEE, New York, Vol. 2, pp. 1980–1985.

[5] Yambay Valiente, M. Y., 2001, "Design of a Unicycle Cobot," M.S. thesis, Northwestern University, Evanston, IL.

[6] Moore, C. A., 1997, "Continuously Variable Transmission for Serial Link Cobot Architectures," M.S. thesis, Northwestern University.

[7] Colgate, J. E., Peshkin, M. A., and Wannasuphoprasit, W., 1996, "Nonholonomic Haptic Display," *IEEE International Conference on Robotics and Automation, Minneapolis*, IEEE, New York, Vol. 1, pp. 539–544.

[8] Moore, C. A., Peshkin, M. A., and Colgate, J. E., 1999, "Design of a 3R Cobot Using Continuously Variable Transmissions," *IEEE International Conference on Robotics and Automation*, 4, pp. 3249–3254.

[9] Wannasuphoprasit, W., Gillespie, R. B., Colgate, J. E., and Peshkin, M. A., 1997, "Cobot Control," *IEEE International Conference on Robotics and Automation*, IEEE, New York, pp. 3571–3577.

[10] Walker, I. D., 1994, "Impact Configurations and Measures for Kinematically Redundant and Multiple Armed Robot Systems," IEEE Trans. Rob. Autom., **10**(5), pp. 670–683.

[11] Lee, D., and Li, P. Y., 2003, "Passive Bilateral Feedforward Control of Linear Dynamically Similar Teleoperated Manipulators," IEEE Trans. Rob. Autom., **19**(3), pp. 443–456.

[12] Lee, D., and Li, P. Y., 2005, "Passive Bilateral Control and Tool Dynamics Rendering for Nonlinear Mechanical Teleoperators," IEEE Transactions on Robotics, **21**(5), pp. 936–951.

[13] Faulring, E., Lynch, K. M., Colgate, J. E., and Peshkin, M. A., 2005, "Haptic Interaction With Constrained Dynamic Systems," *IEEE International Conference on Robotics and Automation*, IEEE, New York, pp. 2469–2475.

[14] Worsnopp, T., 2003, "Design of a Unicycle Cobot Controller," M.S. thesis, Northwestern University, Evanston, IL.

[15] http://lims.mech.northwestern.edu/projects/utla/, October, 2005.

[16] Matsuoka, Y., and Miller, L. C., 1999, "Domestic Rehabilitation and Learning of Task-Specific Movements," *International Conf on Rehabilitation Robotics*, pp. 177–182.

[17] Rahman, T., Ramanathan, R., Seliktar, R., and Harwin, W., 1995, "A Simple Technique to Passively Gravity-Balance Articulated Mechanisms," J. Mech. Des., **117**(4), pp. 655–658.

[18] Asada, H., 1983, "A Geometrical Representation of Manipulator Dynamics and Its Application to Arm Design," ASME J. Dyn. Syst., Meas., Control, **105**, pp. 751–756.

[19] Khatib, O., and Burdick, J., 1987, "Dynamic Optimization in Manipulator Design: The Operational Space Formulation," *The International Journal of Robotics and Automation*, **2**(2), pp. 90–98.

[20] Yoshikawa, T., 1985, "Manipulability of Robotic Mechanisms," Int. J. Robot. Res., **4**(2), pp. 3–9.

[21] Pan, P., Peshkin, M. A., Colgate, J. E., and Lynch, K. M., 2004, "Static Single-Arm Force Generation With Kinematic Constraints," J. Neurophysiol., **93**, pp. 2752–2765.

[22] Scheidt, R. A., Reinkensmeyer, D. J., Conditt, M. A., Rymer, W. Z., and Mussa-Ivaldi, F. A., 2000, "Persistence of Motor Adaptation During Constrained, Multi-Joint, Arm Movements," Bull. Am. Phys. Soc., **84**(2), pp. 853–862.