Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Controlling the Apparent Inertia of Passive Human-Interactive Robots

Tom Worsnopp Michael Peshkin J. Edward Colgate Kevin Lynch
Laboratory for Intelligent Mechanical Systems: Mechanical Engineering Department
Northwestern University
Evanston, IL
greycloak@northwestern.edu peshkin@northwestern.edu colgate@northwestern.edu kmlynch@northwestern.edu

***Abstract*—We have been exploring the use of passive robotic mechanisms for the display of virtual surfaces. Cobots are one way of producing virtual surfaces using a passive mechanism. Unlike powered robots, the nonlinear dynamics of the passive mechanism (e.g., an arm) can be felt by the user as a spatially varying apparent inertia. This effect occurs in many passive designs, including but not limited to cobots. We explain the variable apparent inertia as the projection of the spatially-varying inertia matrix onto the direction of motion, and discuss several ways to control the apparent inertia.**

We explore apparent inertia in detail for the unicycle two link arm, a cobot we have developed for experiments in single-arm motor control studies and rehabilitation. Special paths ("iso-mass contours") are found for this mechanism along which the apparent inertia is constant.

Keywords-inertia ellipsoid; apparent inertia; passive robot; cobot; haptic interface

I. INTRODUCTION

Cobots are passive programmable devices which use mechanical rolling contacts to implement smooth constraint surfaces [2]. We have explored cobots as haptic interfaces to virtual environments [6], as assistive devices for material handling [10], as manipulanda for teleoperation [4], as tools for the exploration of human motion control under motion constraint [13], and are studying their use in rehabilitation.

The operating principle of cobots is to use computer-controllable CVTs (continuously variable transmissions) to produce high quality rolling constraints. In some cases, the CVT is no more than a steered rolling wheel [2]. In other cases the CVT may be a complex mechanism [8]. Many designs have been explored for cobots of diverse workspace dimensionalities, sizes, and other requirements [3], [4], [7], [11].

A key point is that, while computer steering determines the path of a cobot handle through a cobot's workspace, the computer has no authority over the speed of the handle along that path. The speed of the handle is entirely up to the external forces provided by the user (including gravity) and the inherent dynamics of the cobot itself. For example, if a user applies no forces, a cobot will continue to move indefinitely (in the absence of friction) with constant energy. As a cobot moves to other parts of its workspace, however, the handle speed can change due to the configuration-dependent dynamics of the cobot. Thus, even with no external force applied, the handle may speed up or slow down.

In some applications, the goal is to provide the user with the convincing feeling that he is manipulating a particular virtual object along a smooth constraint. While the mechanical rolling contacts provide convincing constraint surfaces, the user still feels the inertial properties of the actual cobot mechanism, which may or may not match the desired inertial properties of the virtual object. This effect may break the illusion of the virtual object. For some combinations of cobot architectures and virtual objects, this effect is not an issue. However, for the unicycle two-link arm (UTLA) cobot - illustrated in Figure 1 and described in Section III - the unmatched inertia characteristics of the actual 2R cobot and the virtual object may hinder the user's acceptance of the virtual object.

The key issue is that a cobot's inertia matrix, when projected to its current path, may be different than that of the virtual object's inertia matrix projected to the same path. We call the projected inertia 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. Even if the apparent inertias of both the cobot and the virtual object are instantaneously identical, their rates of change may be inconsistent. The goal of this paper is to identify the conditions under which a cobot can convincingly emulate the inertial properties of a virtual object. The UTLA cobot is used as a concrete example to illustrate the discussion of apparent inertia later in this paper.

There are three ways to emulate the inertial properties of a particular virtual object.

1. Limit the cobot to follow only certain paths in space. This form of emulation reduces the number of degrees of freedom available to the cobot, but allows the cobot to simulate the desired inertial properties by moving in the direction of a desired apparent inertia. Using this method, the inertial properties of the virtual object are limited by the geometry and mass of the cobot.
2. Use a cobot with redundant degrees-of-freedom, and use the redundancies to control the inertial properties. This is similar to the method outlined above, but it assumes that the cobot has additional degrees-of-freedom that are not necessary for the exploration of the entire workspace.

These redundant degrees-of-freedom can be used to select inertial properties [9], while the remaining degrees-of-freedom are used to explore the workspace. As with the first method of emulation, the inertial properties of the virtual object are limited by the physical geometry and mass of the cobot.

3. Power the cobot. The addition of an actuator provides the cobot with an energy source that can be used to directly influence the apparent inertia along the path. As with the second method, the cobot maintains its ability to explore the entire workspace. Unlike the first and second methods, however, the apparent inertia of the virtual object is limited by the power of the motor and the geometry of the cobot. Note that with the addition of a power source, the cobot is also no longer a passive device.

II. Modeling

Let $\vec{q}$ be a vector in $\Re^n$ of generalized coordinates, defining the cobot's configuration, $M(\vec{q})$ be the cobot's inertia matrix, $\vec{x}$ be a vector in $\Re^m$ of task space coordinates for the virtual object we wish to emulate, and $M_v(\vec{x})$ be the desired inertia matrix of the virtual object. The task space coordinates $\vec{x}$ can be expressed as a function of $\vec{q}$ by the kinematics

$$\vec{x} = f(\vec{q}), \tag{1}$$

from which the task space velocity is found to be

$$\dot{\vec{x}} = \frac{\partial f}{\partial \vec{q}} \frac{d\vec{q}}{ds} \dot{s} = J\vec{T}\dot{s}, \tag{2}$$

where $\vec{q}(s)$ is the cobot's path, parameterized by s, the Jacobian J is given by

$$J = \frac{\partial f}{\partial \vec{q}}, \tag{3}$$

and the path tangent $\vec{T}$ is given by

$$\vec{T} = \frac{d\vec{q}}{ds}. \tag{4}$$

For the cobot to emulate the apparent inertia of the virtual object, the cobot must follow a path $\vec{q}(s)$ such that the kinetic energy of the cobot is equivalent to the kinetic energy of the virtual object as it follows the path $f(\vec{q}(s))$. This condition can be written

$$\frac{1}{2}\dot{\vec{q}}^T M\dot{\vec{q}} = \frac{1}{2}\dot{\vec{x}}^T M_v \dot{\vec{x}}, \tag{5}$$

which can be reduced by canceling terms and substituting equations to

$$\vec{T}^T M\vec{T} = \left(J\vec{T}\right)^T M_v \left(J\vec{T}\right). \tag{6}$$

In general, the apparent inertia constraint of Equation 6 places one constraint on the n-dimensional velocity direction $\vec{T}$ as a function of the configuration $\vec{q}$. Of course the user of the cobot cares about the motion in the task space $\Re^m$, not the cobot's configuration space $\Re^n$. If n > m, the cobot is redundant with respect to the task space, and therefore the velocity constrain on $\dot{\vec{q}}$ may not be apparent in the task space. This reflects the second method for emulating the inertial properties of a virtual object, as described in Section I.

In the rest of this paper we will describe a two-degree-of-freedom system where n = m = 2, so the velocity constrain on $\dot{\vec{q}}$ appears as a velocity constraint on $\dot{\vec{x}}$. Specifically, we apply this formulation to the UTLA cobot in Section VI. First, however, we introduce the cobot, and present the application in which this formulation will be used.

III. The Unicycle Two-Link Arm (UTLA) Cobot

A. 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 a low inertia and little friction.

Based on these specifications, the UTLA was designed and built by Yambay [13].

B. Device Specifications

The UTLA, illustrated in Figure 1, is a one-wheeled, two-degree-of-freedom (DOF) 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 wheel that supports the cobot. There is a force sensor located beneath the handle.

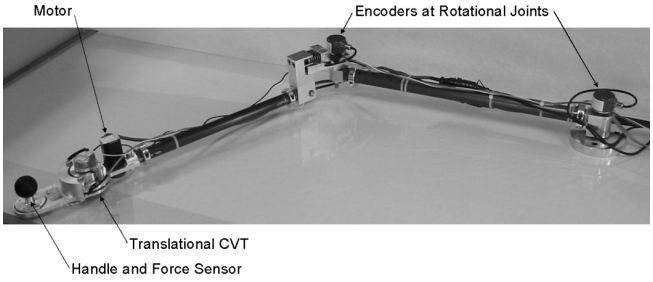

Figure 1 The unicycle two-link arm (UTLA) cobot is a two-degree-of-freedom haptic device used to display high-quality, software-defined constraints over a workspace large enough for the full motion of a user's arm.

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 actuated via a traction drive by a motor, acts as a translational continuously variable transmission (CVT) [8].

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 UTLA can constrain and allow certain motions. 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:

$$\dot{\phi} = \vec{v}_{CVT} \times \vec{\kappa}_{CVT} \cdot \hat{k} - \dot{\theta}_{12}, \tag{7}$$

where $\vec{v}_{CVT}$ is the translational velocity of the CVT, $\vec{\kappa}_{CVT}$ is the software-defined curvature of the path at the CVT (including feedforward and feedback control terms), $\hat{k}$ is a unit vector normal to the x-y plane in which the cobot operates, and $\dot{\theta}_{12}$ is the sum of the angular velocity of Link 2. See [12] for more details about the implementation of a virtual path controller. Additional information, as well as a video of the UTLA, may be viewed at http://lims.mech.northwestern.edu/projects/utla/.

IV. The UTLA and Virtual Paths

Once the virtual path controller had been implemented for 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 users are constrained to a circular path - as illustrated in Figure 2 - they feel like the cobot is rising and falling as it traverses the path.

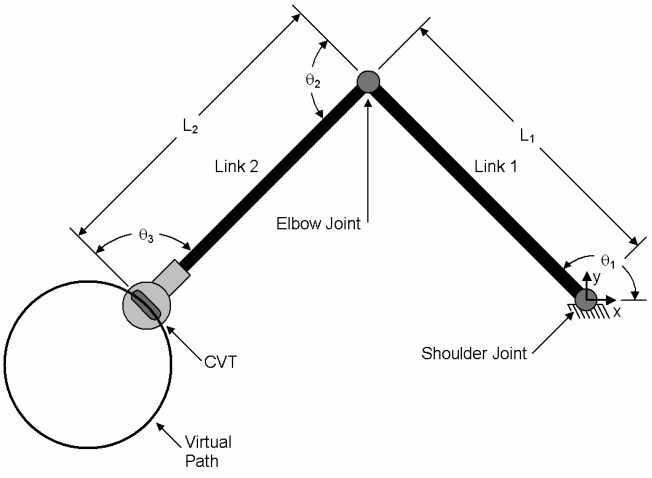

Figure 2 The motion of the cobot is unintuitive to a user when following certain virtual paths. When the handle moves around a circular path, for example, it feels like it is rising and falling.

The cobot is not actually rising and falling, however. The rising and falling sensation corresponds to an increase and decrease in the speed of the cobot handle. The changes in speed are unintuitive for most users since the motion doesn't coincide with the motion of a point mass.

The cause for the change in speed can be explained by considering the kinetic energy of the cobot. Since the UTLA has no mechanism for storing energy, all energy associated with it is kinetic. Furthermore, the small amount of energy lost to friction at the two rotational joints is compensated by the small forces applied at the handle by the user. This means that 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. As Link 1 stops, all of the kinetic energy must shift into Link 2. This results in an increase in angular velocity of Link 2, and therefore an increase in speed of the handle.

V. Iso-Mass Contours

Based on the example in Section IV, it is apparent that there are paths for which the nonlinear dynamics of the cobot are felt by a user. Such paths don't appear very useful in the study of human-constraint interaction or stroke rehabilitation. Ideally, a user should not feel the inertia of the cobot varying along the paths in such studies, as such changes could distract them or taint their interactions. This line of reasoning led us to search for paths along which the cobot feels like it has a constant mass (i.e., along which the cobot feels like a point mass). We refer to such paths as iso-mass contours. In order to generate such contours for the UTLA cobot, we apply the formulation derived in Section II.

VI. Dynamics of the UTLA

Since the majority of the weight is distributed at the ends of the two links of the cobot, the UTLA can be modeled as two point masses connected by two massless links. Figure 3 illustrates this model of the UTLA, where L_1 = 0.652m, L_2 = 0.794m, m_1 = 4.72kg, and m_2 = 1.57kg.

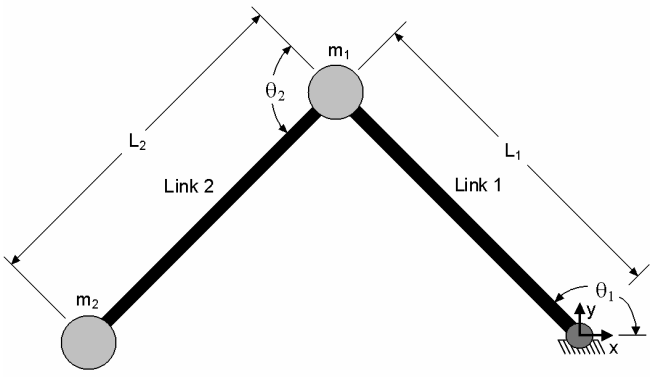

Figure 3 The UTLA cobot is modeled as two point masses at the end of two massless links, since most of the mass is located at the end of the two links.

The configuration of the cobot is specified by the (x, y) coordinates, with the UTLA cobot always in a right-handed orientation.

The equations of motion of the UTLA, expressed in $\vec{x} = (x, y)$ Cartesian coordinates at the handle for convenience, can be written

$$\vec{F} = M(\vec{x})\ddot{\vec{x}} + C(\vec{x}, \dot{\vec{x}})\dot{\vec{x}} + \vec{G}(\vec{x}), \tag{8}$$

where $\vec{F}$ is the force vector applied at the handle, $M(\vec{x})$ is the inertia matrix, $\ddot{\vec{x}}$ is a vector of the acceleration of the handle, $C(\vec{x}, \dot{\vec{x}})$ is a matrix containing centrifugal and Coriolis terms, $\dot{\vec{x}}$ is a vector of the velocity of the handle and $\vec{G}(\vec{x})$ is a vector containing the terms due to gravity. $\vec{G}(\vec{x})$ is, of course, zero since the UTLA is operating in a horizontal plane.

Applying the formulation derived in Section II to this example, we find that by selecting $M(\vec{x})$ in task space, the Jacobian J is the identity matrix

$$J = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{9}$$

and the inertia matrix for the desired inertia is

$$M_v(\vec{x}) = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}, \tag{10}$$

where m is the desired apparent inertia of the virtual object we wish to emulate (i.e., a point mass). Substituting these quantities into Equation 6, and solving for the apparent inertia, we find that

$$m = \vec{T}^T M(\vec{x})\vec{T}, \tag{11}$$

where the inertia matrix, $M(\vec{x})$, is

$$M(\vec{x}) = \begin{bmatrix} \frac{m_1 + m_2 - m_2\cos(2\theta_2) + m_1\cos(2(\theta_1+\theta_2))}{2\sin^2(\theta_2)} & \frac{m_1\sin(2(\theta_1+\theta_2))}{2\sin^2(\theta_2)} \\ \frac{m_1\sin(2(\theta_1+\theta_2))}{2\sin^2(\theta_2)} & \frac{m_1 + m_2 - m_2\cos(2\theta_2) - m_1\cos(2(\theta_1+\theta_2))}{2\sin^2(\theta_2)} \end{bmatrix}, \tag{12}$$

the unit tangent of the path being followed, $\vec{T}$, is

$$\vec{T} = \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix}, \tag{13}$$

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

$$m = \frac{m_1 + m_2 - m_2\cos(2\theta_2) + m_1\cos[2(\theta_1+\theta_2-\phi)]}{\sin^2(\theta_2)}. \tag{14}$$

VII. Results

A. Apparent Inertia

Inspection of Equation 14 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). The minimum occurs when the fourth term in Equation 14, $m_1\cos[2(\theta_1+\theta_2-\phi)]$, 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 to rotations about the elbow joint only. This is illustrated in Figure 4.

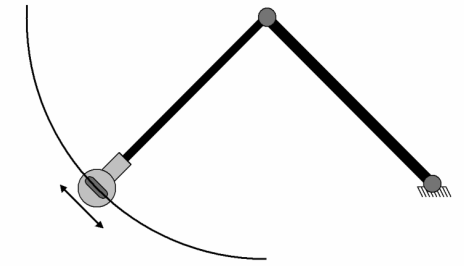

Figure 4 The direction of motion for minimum apparent inertia at a given configuration is always perpendicular to Link 2.

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. This is illustrated in Figure 5.

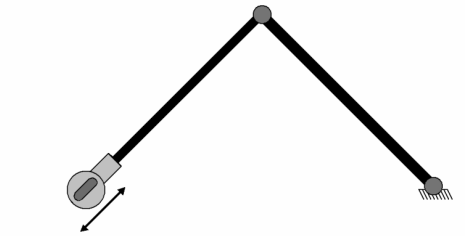

Figure 5 The direction of motion for maximum apparent inertia at a given configuration is always parallel to Link 2.

The concepts outlined above reveal a number of relevant properties about apparent inertia for a given configuration. It is not possible to generate an apparent inertia less than the minimum or greater than the maximum. There are two directions in which the cobot can move that produce an apparent inertia between the minimum and maximum.

B. Inertia Ellipses

Figure 6 illustrates the apparent inertia in terms of another, perhaps more familiar, representation: an inertia ellipse [1], [5], [14]. In this representation, the radius of the ellipse is inversely proportional to the square root of the apparent inertia. 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 radii of the ellipse represent the velocity magnitude required for the same kinetic energy in all directions from the given configuration.

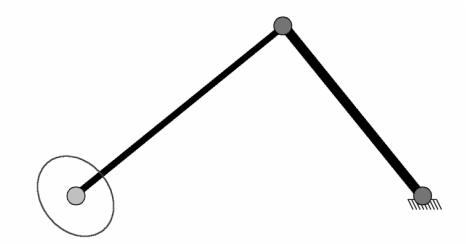

Figure 6 The idea of apparent inertia can also be understood in terms of an inertia ellipse. The major axis corresponds to the direction of minimum apparent inertia, and the minor axis to the direction of maximum apparent inertia.

Using this representation, it is possible to determine the directions of motion for a desired apparent inertia by intersecting the inertia ellipse of the cobot with the inertia ellipse of the desired mass. In our case, the desired mass is a point mass, which is represented by an inertia ellipse with a constant radius (i.e., a circle). Figure 7 illustrates the handle of the cobot and its associated inertia ellipse. A dashed circle is used to illustrate the inertia ellipse for the point mass, and arrows are used to indicate the cobot path directions along which the cobot handle feels identical to the point mass.

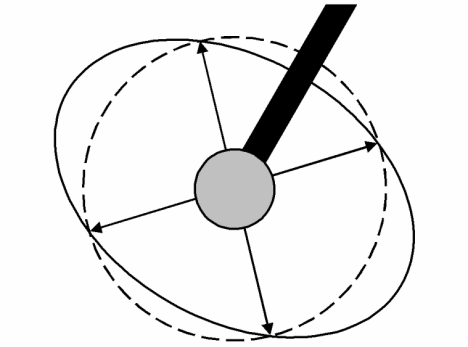

Figure 7 The intersection of the inertia ellipse of the cobot with the inertia ellipse of a point mass (i.e., a circle) can be used to determine the directions of motion in which the cobot will feel like the point mass. These directions are indicated by the arrows.

C. Iso-Mass Contours

The concepts above set the stage for how to generate an iso-mass contour from a given starting configuration. By setting the rate of change of the apparent inertia to zero, the steering velocity of the CVT, $\dot{\phi}$, can be solved for:

$$\dot{\phi} = \dot{\theta}_1 + \dot{\theta}_2 \frac{\cos(\theta_1 - \phi)}{\sin(\theta_2)\sin(\theta_1 + \theta_2 - \phi)}. \qquad \textbf{(15)}$$

We can numerically integrate this differential equation to find iso-mass contours starting from a given initial configuration and with a given apparent inertia (i.e., with a given direction of motion).

Figure 8 illustrates one set of iso-mass contours generated using this technique. Also shown is the inertia ellipse for the starting configuration (x=−1.027m, y=0.0m), and the boundaries of the workspace (depicted as dashed lines). In this example, the apparent inertia is m=3.30kg.

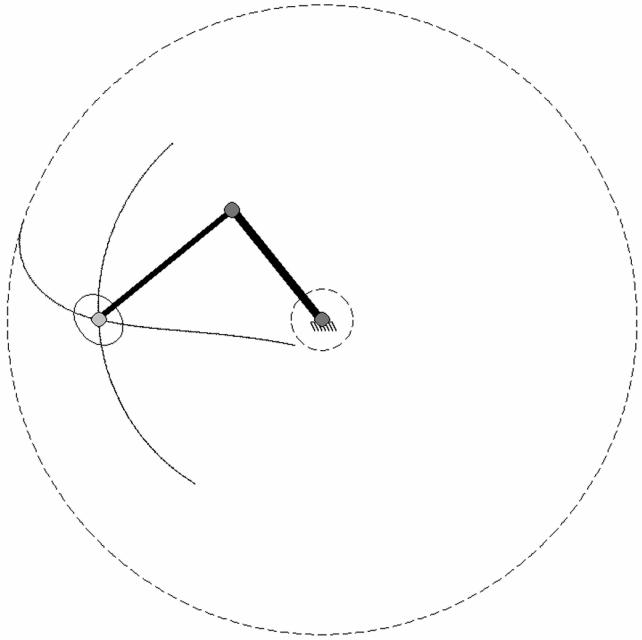

Figure 8 The starting configuration is illustrated along with the inertia ellipse and the set of iso-mass contours for the apparent inertia, m=3.30kg. The dashed lines indicate the edges of the workspace.

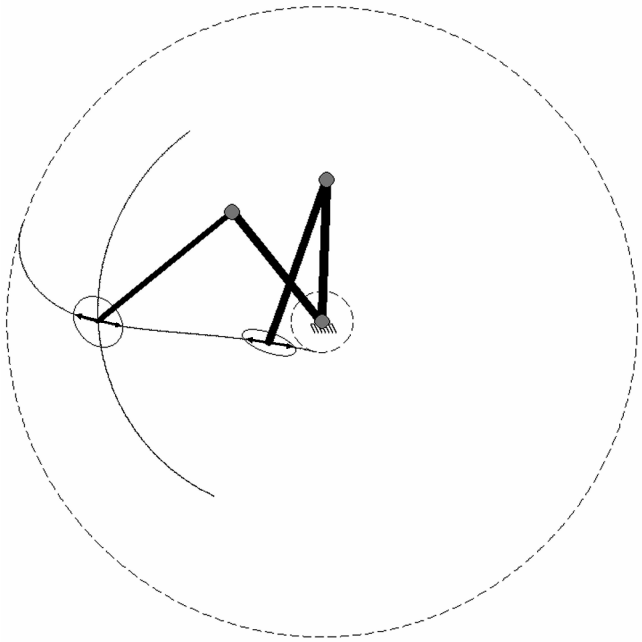

Figure 9 The UTLA cobot has two different inertia ellipses for two different configurations. The diameter of both ellipses tangent to the iso-mass contour are the same length, as indicated by the two double-headed arrows.

In order to provide a point for comparison, Figure 9 illustrates the UTLA in two different configurations. 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 of the inertia ellipses, the lengths of the diameters are the same, which is indicative of the fact that the path does, in fact, have a constant apparent inertia.

The effect of propagating the contours presented in Figure 8 and Figure 9 further through the workspace is illustrated in Figure 10. We see that three ends of the two iso-mass contours terminate at the workspace boundaries. The fourth direction of motion, however, does not. It instead spirals towards a limit cycle circle centered at the shoulder joint. It is worth noting that while most iso-mass contours are circular (spiraling away from the starting configuration), there are also contours that are approximately linear in the task space. Such contours have greater potential for utility in human-constrain interaction studies, where movement along a straight line is of interest.

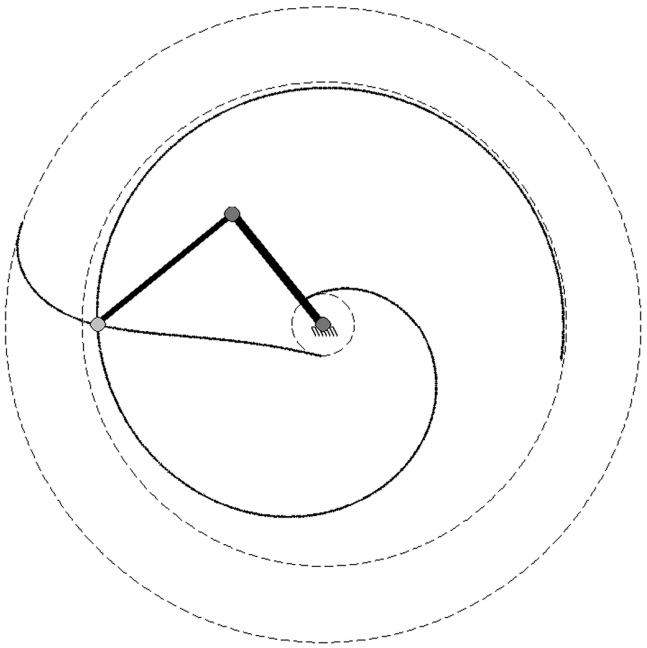

Figure 10 Iso-mass contours: three ends of the two contours terminate at the boundaries of the workspace, while the fourth one spirals towards a limit cycle circle centered at the shoulder, and indicated by the dashed circle.

VIII. Conclusions and Future Work

We have presented a technique for generating iso-mass contours for the UTLA cobot. While our research has focused primarily on emulating a specific inertial property - that of a point mass - for a specific device, we believe that many interesting inertial properties can be emulated for any degree-of-freedom passive robot. Thus, the formulation of a general strategy for emulating any inertial property in higher dimensional spaces is one of our objectives for future research in this domain.

Iso-mass contours are potentially very useful, and could be used in human arm-motion studies. However, they are also limiting since they constrain the cobot to a subset of paths in the workspace. Ideally, we would like to be able to emulate a point mass and still be able to explore the entire workspace. In order to do this, we must be able to alter the apparent inertia of the system. There are at least two techniques for gaining control of the apparent inertia: add a degree of freedom to the system or add an energy source to the system. Exploration of these two techniques will be the subject of future work.

Acknowledgments

We gratefully acknowledge NSF grant IIS-0082957.

References

[1] H. Asada, "Dynamic Analysis and Design of Robot Manipulators Using Inertia Ellipsoids," Proceedings of the IEEE International Conference on Robotics, Atlanta, March 1984.

[2] J.E. Colgate, W. Wannasuphoprasit, and M. Peshkin, "Cobots: Robots for Collaboration with Human Operators," International Mechanical Engineering Congress and Exposition, Atlanta. pp. 433-440, ASME, 1996.

[3] J.E. Colgate, M.A. Peshkin, W. Wannasuphoprasit, "Nonholonomic Haptic Display," IEEE International Conference on Robotics and Automation, Minneapolis, MN, Vol. 1, pp. 539-544, 1996.

[4] E. Faulring, J.E. Colgate, M.A. Peshkin, "A High Performance 6-DOF Haptic Cobot," submitted to IEEE International Conference on Robotics and Atomation, New Orleans, LA, 2004.

[5] O. Khatib, J. Burdick, "Dynamic Optimization in Manipulator Design: The Operational Space Formulation," The International Journal of Robotics and Automation, vol.2, no.2, 1987, pp. 90-98.

[6] C.A. Moore, M.A. Peshkin, J.E. Colgate, "Cobot Implementation of Virtual Paths and 3D Virtual Surfaces," IEEE Transactions on Robotics and Automation, 19(2): 347-351, 2003.

[7] C.A. Moore, M.A. Peshkin, J.E. Colgate, "Design of a 3R Cobot Using Continuously Variable Transmissions," IEEE International Conference on Robotics and Automation, Detroit, MI, 1999.

[8] C.A. Moore, "Continuously Variable Transmission for Serial Link Cobot Architectures," Masters Thesis, Northwestern University, 1997.

[9] I.D. Walker, "Impact Configurations and Measures for Kinematically Redundant and Multiple Armed Robot Systems," IEEE Transactions on Robotics and Automation, Vol. 10, No. 5, 1994.

[10] W. Wannasuphoprasit, P. Akella, M. Peshkin, J.E. Colgate, "Cobots: A Novel Material Handling Technology," Proceedings of IMECE, 1998.

[11] W. Wannasuphoprasit, R.B. Gillespie, J.E. Colgate, M.A. Peshkin, "Cobot Control," IEEE International Conference on Robotics and Automation, Albuquerque, NM, pp. 3571-3577, 1997.

[12] T. Worsnopp, "Design of a Unicycle Cobot Controller," M.S. Thesis, Northwestern University, 2003.

[13] M.Y. Yambay Valiente, "Design of a Unicycle Cobot," M.S. Thesis, Northwestern University, 2001.

[14] T. Yoshikawa, "Manipulability of Robotic Mechanisms," The International Journal of Robotics Research, Vol. 4, No. 2, MIT Press, 1985.