2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Haptic Interactions With Under-Actuated Robots Using Virtual Mechanisms

Greg R. Luecke, *Member, IEEE*, John A. Beckman

***Abstract*—Haptic interactions with computer generated simulations has become almost routine in Virtual Reality (VR) applications. While general interaction requires six degrees of freedom, some haptic devices are designed with fewer degrees of freedom, and so have problems representing general haptic contact. In this work, we present a new approach to the control of a general haptic device, and extend this approach to compensate for missing forces in under-actuated haptic devices. We present an experimental implementation using one of the most popular devices, the PHANTOM®, and show experimental results for a simple case of haptic interaction.**

I. Introduction

THE use of Virtual Reality (VR) has blossomed in recent years, finding applications in engineering, medicine, statistics, and many other areas where three dimensional images can aid in understanding of complex systems. In many applications, interaction with a virtual system can be enhanced with the sense of touch, and haptic feedback can be used to apply representative forces from the virtual environment to a human user. While there are many systems under development that provide haptic feedback, many fall into the category of research prototypes, developed to provide a specific type of force feedback suited to a particular problem. Often, the robotic systems at the core of the haptic device are expensive, one-of-a-kind manipulators that are not flexible enough to be converted for use on another problem [1].

Any haptic feedback device will have some drawbacks, mainly because the senses of kinesthetics and touch are so well developed in the human being that it is nearly impossible to reproduce all of the sensation that is normally felt in the most mundane physical interaction with the hands and fingers. In addition, many VR displays are affected when a mechanical device is introduced into the virtual environment, either occluding the view of the object of interest or adding significant weight that makes it difficult to use. Systems that add even a few simple force feedback options end up being extremely complex in terms of mechanical implementation, dynamics, and control [2].

Although other companies make commercially available haptic devices, such as the Force Dimension [3], by far the most mature haptic feedback device is the Phantom desktop haptic device [4]. The standardized hardware design and generalized interaction software has encouraged the use of the device in haptic research and brought the purchase price close to the consumer level. One drawback of the Phantom is that the six degree-of-freedom device has only three powered degrees of freedom, making it difficult to use for general haptic interactions. The three un-powered degrees of freedom make the Phantom an example of an under-actuated robot. While very useful for some applications, some tasks require constraint forces normally applied by these missing actuators.

In this work, we present a new approach to control of a general the general class of haptic devices. In the approach, we use a virtual robot manipulator to apply motion constraints that generate the haptic forces. We extend this approach to include an under-actuated robot to allow the use of this type of device as a general force feedback probe in a virtual environment. The approach provides a simple haptic control interface for any six DOF manipulator, and makes use of the motion kinematics to mask the effects of the missing actuators.

We will develop the equations for the general six degree of freedom case, and illustrate the concepts with a simpler 3 DOF example. Experimental results are presented to validate the control approach.

II. Virtual Mechanisms for Constraint Forces

Any haptic feedback device works by applying constraint forces to the hand of the user. For our haptic interaction, we choose to grasp the haptic device and "grow" a virtual probe from the end effector that is used for interaction with the virtual environment [5]. This approach offers two advantages over many conventional applications of haptic forces. First, by placing the haptic robot and hand away from the virtual image, there are none of the common occlusion problems as the haptic interaction occurs. Second, the virtual probe provides the user with a "direct" path to the point of interaction and serves as a visual cue between the contact and the forces.

An effective method for arranging the haptic device as a virtual probe is shown in Figure 1, where the end effector of a six degree of freedom robot is used as the grip end of a virtual probe. Interactions occur between the end of the virtual probe and objects in the virtual environment. We

Manuscript received September 14, 2007. This work was supported in part by the Virtual Reality Application Center and the Department of Mechanical Engineering, Iowa State University.

Dr. Greg R. Luecke is with the Iowa State University, Department of Mechanical Engineering, Ames, IA 50011(corresponding author phone: 515-294-5916; fax: 515-294-3261; e-mail: grluecke@iastate.edu).

J. A. Beckman is a Graduate Assistant at Iowa State University (e-mail: j.a.beckman@gmail.com).

also show how the probe can "bridge" the space between the haptic device and the virtual display.

In order for the haptic sensation to be effective, we need to program the robot to move appropriately for contact constraints between the virtual probe and the virtual environment.

We begin the development of the haptic control by examining any real probe as it comes in contact with a flat surface in the environment. As the human grips the handle, forces are applied that oppose the contact forces between the probe and the environment. Figure 2 shows the general arrangement of a probe, where we use two finger-tip forces, F_{F1} and F_{F2}, to apply an arbitrary normal, axial, and moment load to the probe. In the general case, these loads may not be distributed between two fingers as shown, but in all cases, the loads are applied to balance any interaction load, R_x and R_y. The interactive forces in the virtual environment may be computed using whatever combination of statics and dynamics modeling is desired. The challenge in haptics is to translate the effects of these force to the robot end-effector and thus into the hand of the user.

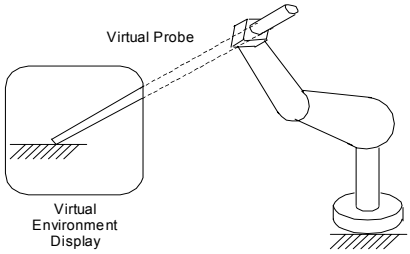

Fig. 1. Virtual probe attached to a general manipulator.

For our control approach, we can consider the probe to be a specific kind of manipulator in the virtual environment, a 5 DOF manipulator with two prismatic joints and three revolute joints, "PPRRR" in classic robotics terms, as shown in Figure 3. Our haptic control approach is based on the use of this "virtual manipulator" together with the actual haptic robot.

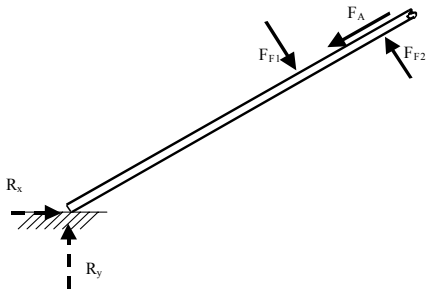

Fig. 2. Hand and environment forces applied to a probe.

For the purposes of illustrating the control approach in this work, we will simplify the example to that shown in Figure 4, the planer equivalent case, where we also fix the manipulator to the flat surface with a revolute joint, as though there is enough friction to keep the tip of the virtual manipulator from moving. These simplifications remove the unilateral nature of the force interactions, and while these effects are not trivial, they can be incorporated on top of the control scheme developed here.

III. Haptic Constraint Using the Virtual Manipulator

The human being applies external loads to the probe in the form of a generalized force vector composed of three Cartesian forces and three moments about those Cartesian axes. In our simplified example, the applied force vector, F_H, has x- and y-components and a moment about the z-axis.

$$F_H = \begin{Bmatrix} f_x \\ f_y \\ n_z \end{Bmatrix} \tag{1}$$

In this case the virtual manipulator (VM) is constrained to a single degree of freedom, rotating along the circular arc shown in Figure 4. We emphasize that the derivation of the control equations for this simple case also hold for the general case in Figure 3, where the virtual probe is constrained in only one direction [6].

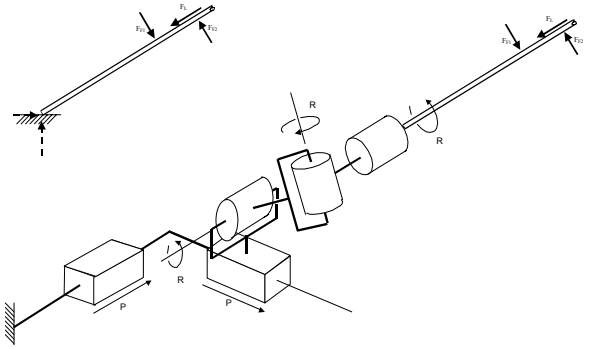

Fig. 3. A five degree of freedom virtual mechanism is used to model the probe. Note that the three revolute joints will intersect at a point.

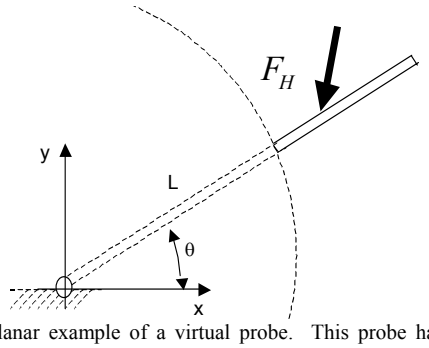

Fig. 4. Planar example of a virtual probe. This probe has only a single degree of freedom

As with any robot, we can relate the Cartesian position, x_v, with the joint angles, θ_v, using the forward kinematics of the VM, Φ_v. For the example in Figure 4, the forward kinematics are simple.

$$x_V = \begin{Bmatrix} x \\ y \\ \theta_z \end{Bmatrix} = \Phi_V(\theta_V) = \begin{Bmatrix} L\cos(\theta) \\ L\sin(\theta) \\ \theta \end{Bmatrix} \tag{2}$$

The virtual joint torque at the pivot, τ_v, is easily found using the Jacobian relationship for the virtual manipulator, J_v, and if the objective of the control was to prevent any

motion, this joint torque could be applied to the VM and no motion would occur.

$$\tau_V = J_V^T F_H \tag{3}$$

We can also ask what applied end effector forces can be balanced by a particular VM joint torque, but we find that equation (3) cannot be inverted directly because the Jacobian is not square. We will use a Moore-Penrose pseudo-inverse to find the end effector force that has the smallest L_2 vector norm, F^*_H.

$$F_H^* = J_V \left(J_V^T J_V\right)^{-1} \tau_V \tag{4}$$

Note that this is not exactly the same force as the original applied force, but is the component of the applied force in the range space of the VM Jacobian. All of the other components of the force applied by the human would be reacted by the structure of the VM. In the most general case, we may want to weight the pseudo-inverse to allow us to adjust which axes of the VM are most heavily used. We choose diagonal matrices, W_A, in the weighted pseudo-inverse.

$$F_H^* = W_A J_V \left(J_V^T W_A J_V\right)^{-1} \tau_V \tag{5}$$

We can use a simple control law to prevent all motion of the haptic manipulator. In the Cartesian space, this is equivalent to fixing the end effector in place with springs. In Figure 5, these springs are shown schematically for the three Cartesian directions, and for a general 6 DOF manipulator, there would also be torsional springs attached in the rotational directions.

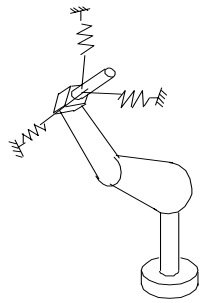

Fig. 5. Simple position-error feedback acts as a spring to prevent motion of the haptic device.

Using these springs provides some compliance to the haptic robot, and we see that in the static case, the forces in the springs will be equal to the forces applied by the human. The simple way to introduce these springs is to use error feedback. We can treat the desired location of the robot as the origin, and then develop a vector of restoring forces according to the error feedback control law:

$$F_H = \left(K_p e + K_v \dot{e}\right) \tag{7}$$

Here, the position error is $e = (x_d - x_R)$, the velocity error is $\dot{e} = (0 - x_d) = -x_d$, and F_H is the applied force from the human operator. The proportional and velocity gains are diagonal matrices, K_p and K_v.

In many cases, the robot coordinate transforms and Jacobian relationships are available in canned routines within the robot controller, and these can be used to generate the forces directly in the Cartesian workspace. If these transforms are not available and the restoring forces are required in the robot joint space, the robot Jacobian, J_R, can be used to convert the Cartesian control forces, F_H, to robot joint torques, τ_R, but note that this is not the VM Jacobian.

$$\tau_R = J_R^T F_H = J_R^T \left(K_p e + K_v \dot{e}\right) \tag{8}$$

It is worth noting that in the case of the Phantom, only the three linear directions are powered, and the rotational directions cannot apply or react torque. Our approach to control of the haptic manipulator will combine the motion kinematics of the virtual manipulator and the control of the haptic robot to generate appropriate constraint forces.

In order to allow motion in the unconstrained directions of the virtual manipulator, it is necessary to apply *all* of the robot control forces that are resisting general motion *except* the set of equivalent robot forces that are resisting motion of the virtual manipulator. Simple subtraction will allow this:

$$\begin{aligned} f_{motion} &= F_H - F_H^* \\ &= F_H - J_V \left(J_V^T J_V\right)^{-1} J_V^T F_H \\ &= \left(I - J_V \left(J_V^T J_V\right)^{-1} J_V^T\right) F_H \end{aligned} \tag{9}$$

This is the expression for the robot control forces, f_{motion}, required to resist motion in all of the directions constrained by the virtual manipulator, while at the same time allowing motion in the unconstrained directions. The right-hand-side of equation (9) is a null space filter matrix for the virtual mechanism. This matrix removes the component of robot resistive forces only in the free-motion direction of the virtual mechanism.

While the set of equivalent motion forces in equation (9) is valid for any applied end effector force and is independent of the control force generation scheme, the error feedback control approach automatically generates control forces that resist the applied human forces. The haptic feedback forces may be slightly affected by the un-modeled dynamic forces in the error feedback approach.

In order to implement motion, we need to set the desired position of the robot to match the current position of the virtual manipulator. We can continue to assume that the desired velocity for the haptic robot is zero, although this is not required.

$$\begin{aligned} e &= X_d - X_R = X_V - X_R \\ \dot{e} &= \dot{X}_d - \dot{X}_R = -{}^V\dot{X}_R \end{aligned} \tag{10}$$

In the most basic form, this error feedback control can be accomplished with a set of fixed gains on each actuator axis of the haptic robot, so that the gain matrices, K_p and K_v are diagonal. In practice, it is often necessary to tune these feedback gains for various postures of the robot to compensate for changes in dynamic forces as the robot changes posture. We will assume that the feedback gains are diagonal and fixed, although position-dependent values of these gains are possible. This approach generates haptic robot joint torques that apply haptic forces on the user for the constrained motion.

$$\begin{aligned} \tau_R &= J_R^T F_H \\ &= J_R^T \left(I - J_V \left(J_V^T J_V\right)^{-1} J_V^T\right) \left(K_p e + K_v \dot{e}\right) \end{aligned} \tag{11}$$

While this control law assures that the haptic manipulator

will move only along the trajectory of the virtual manipulator, there are some nuances associated with this motion that bear some discussion. Along with the current position of the haptic robot, the most important aspect of the control law is the location of the virtual manipulator.

IV. Position of the Virtual Manipulator

If we examine the position of the haptic robot relative to the virtual manipulator in Figure 6, it is not clear where on the allowable trajectory that we should place the VM. Should the VM be at the position of the heavy dashed line, perpendicular to the VM trajectory? This will work, but it will result in an error vector that is large in terms of the orientation of the handle. Is it more accurate to position the VM so that the handle is in the same orientation? This will work, too, but will result in a larger position error.

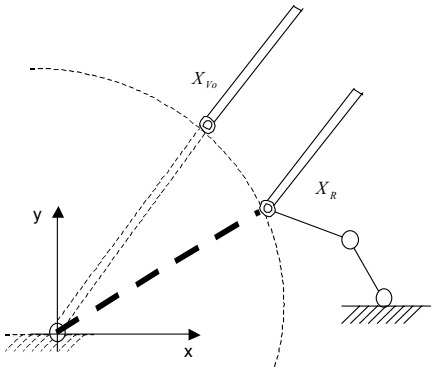

Fig. 6. The haptic robot and the virtual manipulator trajectory do not specify the VM location. The VM may be placed anywhere on the allowable trajectory.

The fact of the matter is that the position of the VM is arbitrary, it is, after all, virtual. Anywhere the VM is placed along the allowable trajectory, the control law in Equation (14) will move the haptic robot to that position. We will point out, however, that placing the VM outside the two points mentioned earlier will result in both the position and the orientation errors being larger than they need to be.

We will examine the error in order to find some criteria for placing the VM relative to the haptic robot.

$$e = X_V - X_R \tag{12}$$

Using both the robot and VM kinematics, we can expand the expression for the error using a Taylor series about some arbitrary initial position of the VM.

$$X_V - X_R = \Phi_V(\theta_V) - \Phi_R(\theta_R) \tag{13}$$

$$X_V = \Phi_{Vo} + \left.\frac{\partial \Phi_V}{\partial \theta_{V1}}\right|_{\theta_{Vo}} \Delta\theta_{V1} + \left.\frac{\partial \Phi_V}{\partial \theta_{V1}}\right|_{\theta_{Vo}} \Delta\theta_{V2} + \cdots + H.O.T. \tag{14}$$

$$X_R = \Phi_{Ro} + \left.\frac{\partial \Phi_R}{\partial \theta_{R1}}\right|_{\theta_{Ro}} \Delta\theta_{R1} + \left.\frac{\partial \Phi_R}{\partial \theta_{R1}}\right|_{\theta_{Ro}} \Delta\theta_{R2} + \cdots + H.O.T. \tag{15}$$

Note that Φ_o is the assumed initial position of the VM, and that the definition of the partial derivatives on the right is simply the Jacobian. If we neglect the higher order terms (H.O.T) we can use this relationship to find changes in the VM position that will eliminate the error between the two. We can choose not to move the haptic robot: after all, that takes actual energy, while moving the VM is done in software.

$$e \cong \Phi_{Vo} + J_V \Delta\theta_V - \Phi_{Ro} - J_R \underbrace{\Delta\theta_R}_{\substack{=0 \\ \text{Do not move haptic robot}}} \tag{16}$$

Our objective is to drive the error to zero, and using the current positions of the VM and haptic robot we can compute the change in position of the VM to eliminate error.

$$X_R - X_V = e = -J_V \Delta\theta_V \tag{17}$$

Unfortunately, the VM Jacobian is non-square by definition, and the physical meaning in this case is that there may not be *any* position of the VM that will actually match the position of the haptic robot in Figure 6. However, we will again use the Moore-Penrose pseudo-inverse to find a VM position that minimizes the L_2 vector norm of the error vector. In the most general case, we can use weight matrices to influence the final position of the VM, for instance, we may want to weight the linear error more heavily than the angular error. Again, we will choose diagonal elements in this error weight matrix, W_e.

$$\Delta\theta_V^* = \left(J_{Vo}^T W_e J_{Vo}\right)^{-1} J_{Vo}^T W_e \left(X_R - X_{Vo}\right) \tag{18}$$

Note that the VM Jacobian is a function of the current value of the VM joint parameters.

$$J_{Vo} = f\left(\theta_{Vo}\right) \tag{19}$$

The new value of the VM joint parameters is found using the change from Equation (22).

$$\theta_{Vnew} = \theta_{Vo} + \Delta\theta_V^* \tag{20}$$

Because of the approximation in equation (16) and due to the nonlinear nature of the robotic system, the VM position found in equation (20) may not be exact, especially if the initial positions are very far apart. It may be necessary to iterate to a final solution for the "best" position of the VM.

We use the virtual manipulator forward kinematics to find the Cartesian coordinates for the position of the virtual manipulator.

$$x_V^* = \Phi_V\left(\theta_V^*\right) \tag{21}$$

Once this new position of the virtual manipulator is known, we can find the error between the two robots.

$$e = X_V^* - X_R \tag{22}$$

Note that the error between the two robots is now completely in the weighted range space of the virtual manipulator, because of the use of Equation (22) in defining the position of the VM.

$$\left(J_{Vo}^T W_e J_{Vo}\right)^{-1} J_{Vo}^T W_e \, e = e \tag{23}$$

Using this approach to control for the haptic robot, the joint-level torque commands are computed based on the error, and the control approach automatically allows the haptic robot to move along the hyper-surface defined by the virtual manipulator free space, while automatically applying constraint forces in all other directions.

While the error can be in any combination of the VM range and null space, the control forces on the haptic robot resist the applied forces that would normally be reacted by the structure of the virtual manipulator. Applied forces that would be reacted by the virtual manipulator joint actuators are not opposed by the haptic robot and allow motion along the allowable trajectory of the virtual manipulator.

V. Control for Under-actuated Robots

There are several haptic feedback devices that are designed to apply only the three Cartesian forces in the x-, y-, and z-directions. We will use the Phantom as the example haptic robot as we develop the control strategy for under-actuated robots.

The virtual manipulator control approach is shown with an under-actuated robot in Figure 8. In our planar example, the robot has 3 DOF to allow general planar motion, but has an un-powered joint on the distal link, indicated by the double circle. The virtual manipulator control strategy will allow the robot to constrain the user to the circle, but will not be able to constrain the handle to the correct angle. This is shown experimentally in Figure 9, using the control law from equation (9). Because the third element of the robot force vector is zero—no actuator on the distal joint—we could also use the weighted pseudo-inverse of equation (5) where the weights are used to specifically prevent any of the control forces from being assigned to the un-actuated joints.

$$W_A = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{24}$$

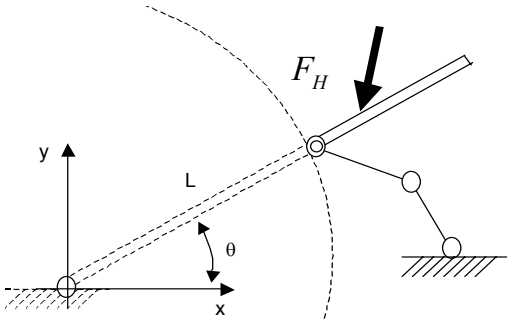

Fig. 8. An under-actuated robot used with a virtual manipulator.

The position of the virtual manipulator is important to the control of this system. The double dashed line in Figure 6 shows the system when the VM is placed at the position defined by the un-actuated robot handle, not at the position on the constraint circle that is closest to the handle. We can find this position of the VM using equation (18), by assigning the weight matrix with zeros along the powered joints while leaving a weight on the un-powered joint.

$$W_e = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \tag{25}$$

This forces all of the error between the haptic robot and the VM to be defined by the un-powered joint. This error will be used to generate a constraint force with the haptic robot.

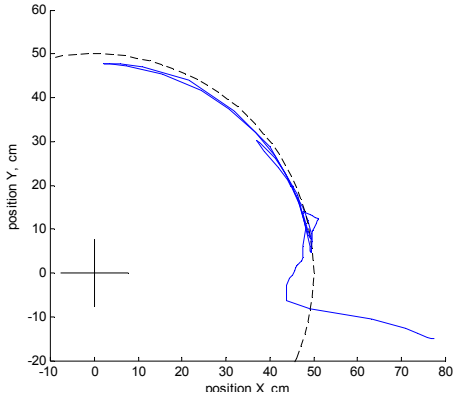

Fig. 9. Trajectory of the haptic robot using the virtual manipulator constraint. Note that while the robot moves along the circle, the handle is free to rotate and is not constrained to the angle of the VM.

The weighted pseudo-inverse of equation (18) assigns all the error between the VM and the haptic robot to the position of the un-powered joint. Because this error lies completely in the range space of the VM, we can add a force to the haptic robot using the same simple error feedback control law. This force, F_e, will move the haptic robot toward the VM position.

$$F_e = K_p e + K_v \dot{e} \tag{26}$$

This force is completely in the range space of the VM, and we can use equation (3) to find the equivalent joint force, $\boldsymbol{\tau}_v$. The weighted equation (5) computes the equivalent robot force, and using the weights in equation (24), there will be no components of this force assigned to the un-powered joint!

Overall, the control for under-actuated haptic manipulators can be expressed as two parts, one part generated from the error and passed through the null space filter of the VM, and the second part generated from the error in the range space of the VM and passed through the range space filter of the VM.

$$f_{motion} = \underbrace{\left(I - W_A J_V \left(J_V^T W_A J_V\right)^{-1} J_V^T\right)\left(K_p e + K_v \dot{e}\right)}_{\text{Null space filter}} + \underbrace{W_A J_V \left(J_V^T W_A J_V\right)^{-1} J_V^T \left(K_p e^* + K_v \dot{e}^*\right)}_{\text{Range space filter}} \tag{26}$$

By weighting these filters properly, we assign all the motive forces to the powered joints and use the error in the un-powered joint to generate constraint forces along the trajectory.

VI. Experimental Results

We programmed the control law of equation (26) using a PHANTOM® Omni™ while constraining the motion in the third, z-direction, to be in a plane. By grasping the un-powered handle, we can move the VM along the circle. The initial position of the haptic robot was far from the constraint trajectory, so large forces were generated that

moved the hand to the circle. This is shown in Figure 10, where the haptic robot handle is the thick line and the force is shown as an arrow. The force is not quite directing the handle directly to the constraint circle, but slightly toward the VM position corresponding to the handle position.

Once the haptic robot reaches the constraint, the forces become small, unless the user tries to pull the robot away from the circle. As the user rotates the handle, the control generates a force to move the robot to the "new" position of the VM. The user can choose to be passive, and the haptic robot moves to the new position, as shown if Figure 11. Here, the constraint forces are small and serve to keep the handle on the VM trajectory.

The user can also choose to resist the motion. A real probe would bend in this case and the user would feel the forces shown in Figure 2. The experimental forces are shown in Figure 12, although the only true measured force is the one applied by the robot. The other two finger forces are calculated to statically balance the handle. The directions of the forces are correct for a bending probe, as shown (try it yourself with a pencil!). The probe has been added to show visually how the robot applies forces based on the un-powered joint positions, and to show that these forces are physically consistent with the forces that result from an contact with an actual probe.

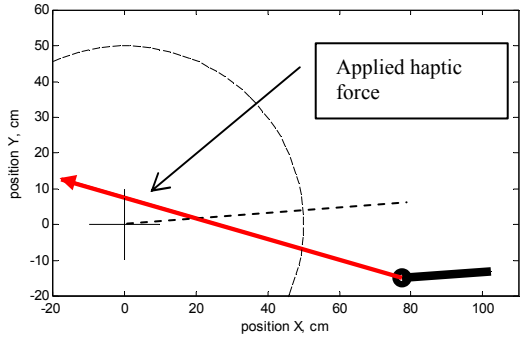

Fig. 10. When there is a large error between the haptic robot and the VM, a large force is generated to move the haptic robot to the constraint.

While the forces that the user feels in this case are representative of the actual forces, they are not identical. Because the two finger forces are reacting the virtual probe moment and shear in a different way, the loads are also different. However, the feeling is representative and provides a good feel to the user.

VII. Conclusion

Haptic feedback is useful in many VR applications, and the cost of these devices has fallen in recent years to a point that is close to allowing the general consumer to add this tool to their standard computer configuration. We show in this work how the most popular haptic device can be programmed to feel like a virtual probe. The approach is general, and can be applied to any haptic device, under-actuated or not.

The control approach uses simple error feedback for generating the motion forces, so the motion is stable without any need for a force sensor on the haptic device.

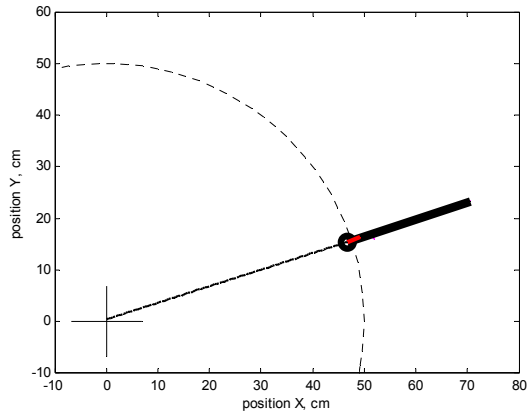

Fig. 11. When there is little error between the haptic robot and the VM, the constraint forces are small. In this case the constraint is perpendicular to the VM trajectory, constraining the user to the path.

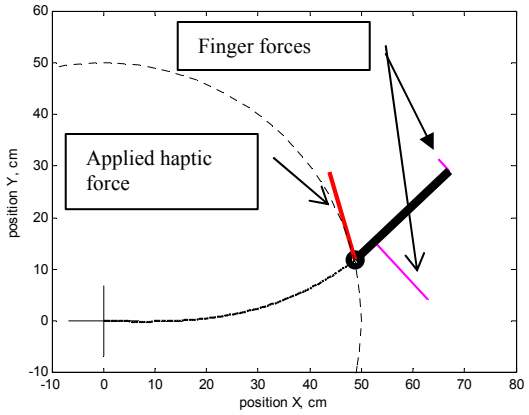

Fig. 12. Reaction forces during virtual haptic interaction. The finger forces are not measured, but show how the user would feel the virtual bending moment when the probe is in contact.

References

[1] Tholey, G, and Desai, J., "Design and development of a general purpose 7 DOF haptic device", Proceedings - IEEE Virtual Reality, v 2006, IEEE Virtual Reality, Haptics Symposium and Symposium on 3D User Interface, 2006, 2006, p 84

[2] Frisoli, A., Simoncini, F., and Bergamasco, M., "Kinematic design of a two contact points haptic interface for the thumb and index fingers of the hand", *Journal of Mechanical Design, Transactions of the ASME*, v 129, n 5, May, 2007, p 520-529

[3] Force Dimension, http://www.forcedimension.com/fd/avs/home/

[4] SensAble Technologies, http://www.sensable.com/

[5] Luecke, G.R., Edwards, J.C., and Miller B.E. 1997, "Virtual Cooperating Manipulator Control For Haptic Interaction with NURBS Surfaces," Proceedings of the International Conference on Robotics and Automation, pg 112-117, Albuquerque, NM, April, 1997.

[6] Zafer, N, and Luecke,G.R., "A geometric constraint-based approach to force and motion coupling between real and virtual mechanisms", *Advanced Robotics*, Vol. 21, No. 12, pp. 1393–1410 (2007)