

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/3450123>

# Vision-Assisted Control for Manipulation Using Virtual Fixtures

Article in IEEE Transactions on Robotics · January 2005

DOI: 10.1109/TRO.2004.829483 · Source: IEEE Xplore

CITATIONS

225

READS

338

5 authors, including:



**Panadda Marayong**

California State University, Long Beach

41 PUBLICATIONS 1,288 CITATIONS

[SEE PROFILE](#)



**Allison Okamura**

Stanford University

299 PUBLICATIONS 13,890 CITATIONS

[SEE PROFILE](#)



**Gregory D. Hager**

Johns Hopkins University

520 PUBLICATIONS 22,973 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



medical image processing [View project](#)



Dynamic Augmented Reality for Sensory Substitution in Robot-Assisted Surgical Systems [View project](#)

# Vision-Assisted Control for Manipulation Using Virtual Fixtures

Alessandro Bettini, Panadda Marayong, *Student Member, IEEE*, Samuel Lang, Allison M. Okamura, *Member, IEEE*, and Gregory D. Hager, *Senior Member, IEEE*

**Abstract**—We present the design and implementation of a vision-based system for cooperative manipulation at millimeter to micrometer scales. The system is based on an admittance control algorithm that implements a broad class of guidance modes called virtual fixtures. A virtual fixture, like a real fixture, limits the motion of a tool to a prescribed class or range of motions. We describe how both hard (unyielding) and soft (yielding) virtual fixtures can be implemented in this control framework. We then detail the construction of virtual fixtures for point positioning and curve following as well as extensions of these to tubes, cones, and sequences thereof. We also describe an implemented system using the JHU Steady Hand Robot. The system uses computer vision as a sensor for providing a reference trajectory, and the virtual fixture control algorithm then provides haptic feedback to implemented direct, shared manipulation. We provide extensive experimental results detailing both system performance and the effects of virtual fixtures on human speed and accuracy.

**Index Terms**—Human-machine systems, robot control, virtual fixtures, visual servoing.

## I. INTRODUCTION

MANY tasks currently require that humans perform high-precision, microscale manipulation. Examples include assembly of optoelectronic components, some types of microelectromechanical system (MEMS) assembly [2], [4], small manufactured parts such as miniature ball bearings, and a wide variety of surgical procedures. An example of the latter is the problem of retinal vein cannulation, the insertion of a small needle into blood vessels in the back of the eye for the purpose of injecting therapeutic drugs [25]. In this case, the scale of the blood vessel is on the order of 100  $\mu\text{m}$  in the outer diameter, with the lumen even smaller. By way of comparison, tremor and drift for skilled microsurgeons has been recorded to be on the order of 49  $\mu\text{m}$  rms [19]. In short, only the most skilled microsurgeons attempt to perform this task, and even then the outcome is not guaranteed.

In microscale manipulation, the challenges of physical scale make muscle tremor, motion accuracy, and position drift significant factors in task performance. Related problems include

severely reduced haptic sensation and limited depth perception [23]. However, preliminary evidence suggests that humans operating in collaboration with robotic mechanisms can surmount many of these barriers [3], [11]. In particular, recent preliminary tests using the Johns Hopkins “Steady Hand Robot” (SHR) [24] have suggested that human-robot cooperative systems in vitreoretinal microsurgery could have a significant impact on both the speed and efficacy of such procedures [12].

With the SHR, the tool being manipulated is simultaneously held by the operator and the robot. The robot is admittance-controlled, with the user commanding the robot through forces applied to the tool. This approach overcomes some of the limitations of: 1) teleoperation, where the user is physically disconnected from the environment; 2) autonomous robotic execution, which lacks high-level human judgement; and 3) freehand procedures, where the user may not have adequately precise control over motion and forces.

This paper specifically examines the implementation and efficacy of vision-based “virtual” fixtures [7], [8], [16], [18], [20]–[22] in the direct manipulation paradigm. Virtual fixtures, like real ones, are designed to restrict a mechanism to certain subspaces or volumes of motion. As such, virtual fixtures are ideal for use in a cooperative interface: judicious use of robotic sensing and control permits an operator to perform physical interactions with higher confidence and speed, yet retain direct control of the activity. Studies on telemanipulation systems have indicated that user performance on a given task can increase as much as 70% with the introduction of guidance using fixtures [20]. Previous telemanipulation systems (e.g., [1], [9], [10], [20]) have implemented these motion constraints within an impedance control regime. Cobots [17] implement predefined virtual fixtures using specialized hardware to steer cooperative robots along desired paths or away from undesirable regions. To our knowledge, these are the first reported results of a system based on admittance control that can apply virtual fixtures with varying strength and geometry based on sensory feedback.

The remainder of this paper details the development and testing of a compliant, vision-based virtual fixture algorithm that we have employed at both macroscopic and microscopic scales. We first describe control algorithms that can be used to create “hard” and “soft” virtual fixtures for an admittance control cooperative system. We demonstrate that the use of vision-based virtual fixtures significantly reduces positioning error and execution time in a macroscale task. We then extend these results to a broader class of multistep tasks and virtual fixture primitives. Finally, we provide quantitative experimental data and system analysis for manipulation at scales on the order of what we expect in vitreoretinal microsurgery.

Manuscript received January 14, 2003; revised September 29, 2003. This paper was recommended by Associate Editor C. Melchiorri and Editor I. Walker upon evaluation of the reviewers’ comments. This work was supported by the National Science Foundation under Grant EEC-9731478 and Grant IIS-0099770. This paper was presented in part at the IEEE/IEEE/RJS International Conference on Intelligent Robots and Systems, 2001.

The authors are with the Engineering Research Center for Computer Integrated Surgical Systems and Technology, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: bettini@cs.jhu.edu; pmarayong@jhu.edu; aokamura@jhu.edu; hager@cs.jhu.edu).

Digital Object Identifier 10.1109/TRO.2004.829483

## II. VIRTUAL FIXTURING SCHEME

In this paper, scalars are written normal face lowercase, vectors are written boldface lowercase, and matrices are written normal face uppercase. Matrix transpose is denoted by  $'$ . The robot is modeled as a Cartesian positioning device with tool tip position  $\mathbf{x}_a \in \mathbb{R}^3$  and control input  $\mathbf{v} = \dot{\mathbf{x}}_a \in \mathbb{R}^3$ , all expressed in the robot base frame. The robot is guided by applying forces  $\mathbf{f} \in \mathbb{R}^3$  on the manipulator handle, likewise expressed in robot base coordinates.

### A. Virtual Fixtures as a Control Law

In the Steady-Hand paradigm, the relationship between velocity and motion is derived by considering a “virtual contact” between the robot tool tip and the environment. In most cases, this contact is modeled by a linear viscous friction law

$$k\mathbf{v} = \mathbf{f} \quad (1)$$

or equivalently

$$\mathbf{v} = \frac{1}{k}\mathbf{f} \quad (2)$$

where  $\mathbf{f}$  is the user input (applied force),  $\mathbf{v}$  is the control output (robot velocity), and  $k > 0$  is the damping of the virtual contact. In what follows, it will be more convenient to talk in terms of an admittance  $c \equiv 1/k$ .

When using (2), the effect is that the manipulator is equally stiff in all directions: an isotropic admittance. A virtual fixture generalizes this model to include anisotropic admittances. In this paper, we specifically consider virtual fixtures on tool position. To this end, let us assume that we are given a  $3 \times n$  time-varying matrix  $\delta = \delta(t)$ ,  $0 < n < 3$ , with the property that  $\delta$  has column rank  $n$ . Intuitively,  $\delta$  represents the instantaneous “preferred” directions of motion for the tool tip. For example, if  $n$  is 1, the preferred direction is along a curve in space; if  $n$  is 2, the preferred directions span a plane.

From  $\delta$ , we define the projection operator

$$D_\delta \equiv \delta(\delta'\delta)^{-1}\delta'. \quad (3)$$

We note that the inverse above is guaranteed to exist due to the rank restrictions on  $\delta$ .

Consider now decomposing the input force vector  $\mathbf{f}$  into two components

$$\begin{aligned} \mathbf{f}_\delta &\equiv D_\delta \mathbf{f} \\ \mathbf{f}_\tau &\equiv \mathbf{f} - \mathbf{f}_\delta. \end{aligned} \quad (4)$$

From these definitions, it follows immediately that  $\mathbf{f}_\tau^T \delta = 0$ .

Combining (4) and (2), we can now write

$$\mathbf{v} = c(\mathbf{f}_\delta + \mathbf{f}_\tau). \quad (5)$$

Let us now introduce a new admittance  $c_\tau \in [0, 1]$  that attenuates the nonpreferred component of the force input  $\mathbf{f}_\tau$ . This results in

$$\begin{aligned} \mathbf{v} &= c(\mathbf{f}_\delta + c_\tau \mathbf{f}_\tau) \\ &= c(D_\delta + c_\tau(I - D_\delta))\mathbf{f} = G(c, c_\tau, \delta)\mathbf{f}. \end{aligned} \quad (6)$$

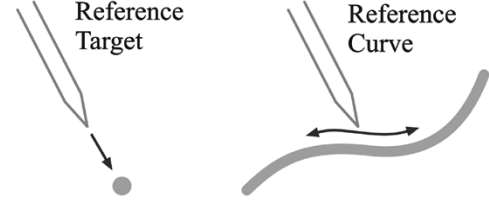


Fig. 1. Two types of reference direction fixtures. In the *Reference Target* case, the robot directs the tool toward a specific point. In the *Reference Curve* case, the robot directs the user motion to be along a path.

Thus, the final control law is in the general form of an admittance control with a time-varying gain matrix  $G$  depending on  $c$ ,  $c_\tau$ , and  $\delta$ . The value of  $c$  controls the overall admittance of the system. Choosing a value  $c_\tau < 1$  imposes the additional constraint that the robot is stiffer in the directions of motion orthogonal to  $\delta$ . We refer to the extreme case of  $c_\tau = 0$  as a *hard virtual fixture*, since it is not possible to move in any direction other than the preferred direction. All other cases will be referred to as *soft virtual fixtures*. We note that, in the case  $c_\tau = 1$ , we have an isotropic admittance as before.

### B. Virtual Fixtures for Targeting and Curve Following

In the remainder of this paper, we explore two types of guidance, shown in Fig. 1, for the case where the preferred direction is a curve in space ( $n = 1$ ). These are the following:

- **Reference target:** The desired path consists of a straight segment in three-dimensional (3-D) Cartesian space that connects the current position of the tool tip to a target point.
- **Reference curve:** The reference path is a fixed 3-D Cartesian space curve.

Some tasks require a combination of these guidance types. For example, to achieve retinal vein cannulation, a curve-following guidance mode is needed to direct the needle to track along a vessel in order to reach the vicinity of a desired target location. Once this target is in the visual field, a new targeting guidance mode is implemented to place the needle precisely over a vessel junction before insertion.

Both cases are treated using a *reference direction virtual fixture* that constrains tool tip motion to a given path. For the basic control law, it is assumed that the reference curve or target is known. (In practice, computer vision is used to determine this information as described in Section III.) In the reference target case, the path is the segment joining the tool position and the target position while, in the reference curve case, the path is the curve. In the *Reference Target* case, the motion ends when the target is reached.

1) *Defining the Reference Direction:* Given a manipulator Cartesian position  $\mathbf{x}_a$ , a reference direction  $\delta$  is defined as follows:

- 1) *Case 1: Reference Target:* The reference direction  $\delta_t$  is given by the difference between a fixed target position  $\mathbf{x}_t \in \mathbb{R}^3$  and the manipulator position as follows:

$$\delta_t(\mathbf{x}_a) = \mathbf{x}_t - \mathbf{x}_a. \quad (7)$$

- 2) *Case 2: Reference Curve:* Let us assume the reference curve is described by a parametric expression

$$\mathbf{p}(s) \equiv (x(s) \ y(s) \ z(s))^T, \quad s \in [0, 1]. \quad (8)$$

We define  $\hat{s}(\mathbf{x}_a)$  as that point on the curve at the minimum distance from the tool tip<sup>1</sup> as follows:

$$\|\mathbf{p}(\hat{s}(\mathbf{x}_a)) - \mathbf{x}_a\| = \min_{s \in [0, 1]} \|\mathbf{p}(s) - \mathbf{x}_a\|. \quad (9)$$

The reference direction  $\delta_p$  is then defined as the normalized tangent direction to the curve at that point

$$\mathbf{t}(\mathbf{x}_a) = \frac{d}{ds} \mathbf{p}(s) \Big|_{s=\hat{s}(\mathbf{x}_a)} \quad (10)$$

$$\delta_p(\mathbf{x}_a) = \frac{\mathbf{t}(\mathbf{x}_a)}{\|\mathbf{t}(\mathbf{x}_a)\|}. \quad (11)$$

In the reference target case, we can directly apply (6) with  $\delta = \delta_t$ . However, the case of a reference curve defining  $\delta = \delta_p$  would only work as expected provided the tool tip remains on the path. If the actual tool tip position were off the path, it would impose motion along a parallel curve rather than returning to the desired path. To address this problem, we define the Cartesian error  $\mathbf{e}$  as the difference between the evaluated position along the curve and the actual Cartesian position as follows:

$$\mathbf{e}(\mathbf{x}_a) = \mathbf{p}(\hat{s}(\mathbf{x}_a)) - \mathbf{x}_a. \quad (12)$$

A new direction  $\delta_c$  is defined as

$$\delta_c(\mathbf{x}_a) = \text{signum}(\mathbf{f} \cdot \delta_p(\mathbf{x}_a)) \delta_p(\mathbf{x}_a) + k_d \mathbf{e}(\mathbf{x}_a) \quad (13)$$

where the use of signum ensures the correct relationship between the tangent orientation and the motion direction, and  $k_d > 0$  is a scalar gain. By applying (6) with  $\delta = \delta_c$ , with nonzero error  $\mathbf{e}$ , the reference direction is chosen to tend to move the user back to the curve. As  $k_d$  is increased, tracking performance is improved and the stability margin is decreased.

Using the assumptions that  $\delta$  is full rank and the human input force is continuous, the admittance control law (6) is guaranteed to converge asymptotically. The complexity of the stability analysis increases substantially with the addition of the virtual fixture controller, multiple degrees of freedom, robot dynamics, and the robot's low-level proportional derivative (PD) controller. Although a complete stability analysis is beyond the scope of this paper, a summary of system behavior with select simplifications of the virtual fixture implementation in a planar case provides some useful insight.

Any preferred direction can be expressed in terms of an instantaneous preferred direction  $\delta_p$  lying along the  $x$  axis of a rotating frame. By applying an appropriate rotation matrix, any general two-dimensional (2-D) path can be achieved without affecting the result of this analysis. The preferred direction  $\delta_p$  and the Cartesian error  $\mathbf{e}$  can be expressed as

$$\begin{aligned} \delta_p &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \mathbf{e} &= \begin{bmatrix} 0 \\ y(t) \end{bmatrix} \end{aligned} \quad (14)$$

where  $y(t)$  is the position of the tool along the current  $y$  axis. Applying the projection operator (3) and the admittance control

law (6), where  $v$  is now the desired Cartesian velocity, the instantaneous desired velocity in the  $x$  and  $y$  directions ( $\dot{x}_d$  and  $\dot{y}_d$ ) can be expressed in terms of the virtual fixture control parameters ( $c$ ,  $c_\tau$ , and  $k_d$ ) and the input force in the  $x$  and  $y$  directions ( $f_x$  and  $f_y$ ) as follows:

$$\begin{aligned} \dot{x}_d &= c \left( \frac{(1 - c_\tau)}{1 + (k_d y)^2} + c_\tau \right) f_x + \frac{c k_d y (1 - c_\tau)}{1 + (k_d y)^2} f_y \\ \dot{y}_d &= \frac{c k_d y (1 - c_\tau)}{1 + (k_d y)^2} f_x + c \left( \frac{(k_d y)^2 (1 - c_\tau)}{1 + (k_d y)^2} + c_\tau \right) f_y. \end{aligned} \quad (15)$$

We assume an infinitely stiff, Cartesian robot that can be modeled with two uncoupled mass-damper equations, one for each degree of freedom. With the substitution of the desired velocity into the virtual fixture control law, the system is nonlinear. However, considering when the tool is accurately following the preferred direction ( $y \approx 0$ ), the transfer function relating the robot velocity to the human input force is

$$\frac{V_i(s)}{F_i(s)} = \frac{s + (k_p + k_v s)c}{m_i s^2 + (b_i + k_v)s + k_p} \quad (16)$$

where  $i = x, y$ ,  $k_v$ , and  $k_p$  are the derivative and proportional gains of the robot's low-level PD controller, respectively. In this case, the stability of the system depends only on robot dynamics and the selection of PD control gains. However, this simplified transfer function does not capture the effect of the closed-loop virtual fixture when  $y \neq 0$ . For the implementation described in this paper, selection of a proper set of controller parameters, including the value of  $k_d$  when  $y \neq 0$ , were determined experimentally.

2) *System Behavior Near the Target Point:* In the reference target case, the rank constraint on  $\delta$  is violated as the manipulator nears the target. A convenient way to address this problem is to define a spherical switching surface  $S(\rho)$  of radius  $\rho$  about the target point. External to  $S(\rho)$ , path following is active, while inside  $S(\rho)$  another control scheme is used.

One option is to fix the reference direction as the sphere is entered and then to use that reference direction as the basis for a virtual fixture with fixed direction. We have investigated the two extreme cases of this approach.

- 1) Use isotropic fixtures (set  $c_\tau = 1$ ) inside the sphere.
- 2) Use "hard" fixtures (set  $c_\tau = 0$ ) inside the sphere.

It is worth noting that, at the switching surface, there may be a velocity discontinuity (due to the change in admittance) that can be smoothed over several samples if desired.

A second option is to take autonomous control of the manipulator as the user enters the sphere. This implements a "snap-to" behavior. We have investigated two such approaches. The first is a simple linear control law

$$\mathbf{v} = k_c(\mathbf{x}_t - \mathbf{x}_a) \quad (17)$$

where  $\mathbf{x}_t$  and  $\mathbf{x}_a$  are, respectively, the target position and the tool current position, and  $k_c$  is a scalar gain. This proportional controller moves the tool along a straight line toward  $\mathbf{x}_t$  with a rate of convergence dependent on  $k_c$ .

Proportional velocity control typically creates velocity discontinuities at switching surface. If this is undesirable, it is possible to derive an acceleration version of this scheme. To realize such a controller, we record  $\delta$ , the desired direction of motion, the tool velocity  $\mathbf{v}_a$ , and position  $\mathbf{x}_a$  (read from the en-

<sup>1</sup>For the moment, we ignore the fact that this point is not always uniquely defined and return to this issue in Section III.

coders) when the tool tip enters the switching sphere. Define  $\tau = (I - D_\delta)\mathbf{v}_a$  and  $\mu = \delta \times \tau$ . Let  $R$  denote an orthonormal matrix computed from the vectors  $(\delta, \tau, \mu)$ . Using these definitions, the following second-order scheme is used to generate the Cartesian acceleration command  $\mathbf{a}$ :

$$\mathbf{a} = R^{-1}(K_v(\mathbf{v}_r - R\mathbf{v}_a) + K_p R(\mathbf{x}_t - \mathbf{x}_a)) \quad (18)$$

where  $K_v = \text{diag}(k_{v\delta}, k_{v\tau}, k_{v\mu})$ , and  $K_p = \text{diag}(k_{p\delta}, k_{p\tau}, k_{p\mu})$  are the diagonal matrices that contain the loop velocity and position gains, respectively, for each direction  $\delta, \tau$ , and  $\mu$ . The vector  $\mathbf{v}_r$  contains the reference velocity in the  $(\delta, \tau, \mu)$  frame. Its components along  $\tau$  and  $\mu$  are always zero, while the one along  $\delta$  can be used to specify a desired velocity profile along the line of approach. In order to limit transients away from  $\delta$ , it is important to use high position gains in the normal directions (high  $k_{p\tau}$  and  $k_{p\mu}$ ). To avoid position overshoot (obtaining critical damping or overdamping), appropriate position and velocity gains along  $\delta$  (i.e., low  $k_{p\delta}$  and high  $k_{v\delta}$ ) must be selected.

3) *Joining Curves and Points*: The structural invariance of the controller with respect to path following or positioning makes it straightforward to define a task that involves following a given path and positioning at a given target point. Such a reference path can be formalized as follows:

- path equation  $\mathbf{p}(s)$  (8);
- target point  $\mathbf{x}_t$ , lying along the path ( $\mathbf{x}_t = \mathbf{p}(s), s \in [0, 1]$ );
- spherical neighborhood  $S(\mu)$  of  $\mathbf{x}_t$  of radius  $\mu$ .

The sphere  $S(\mu)$  is a controller switching surface; outside  $S(\mu)$ , the task is a path-following problem, while inside  $S(\mu)$  it is a positioning problem. Assuming the same value of  $c_\tau$  for path and target configurations, the switch only involves a potential discontinuity in the reference direction. If this is undesirable, the change in reference direction can be smoothed over more than one sample time, for example, by taking a time-weighted combination of  $\delta_t(\mathbf{x}_a)$  and  $\delta_c(\mathbf{x}_a)$ . Similar modifications make it possible to compose paths of continuous curves with a finite set of nondifferentiable points. The reference direction virtual fixture (11) is still well defined, with the exception of nondifferentiable path connection points. Smoothing to create local  $C(1)$  continuity of the paths at the point of connection is the most natural method of avoiding this problem.

### C. Extension to Volumes

The previous path definitions assumed the goal was to confine motion to a lower dimensional subspace of the manipulator configuration space. While natural for precision operations, subspaces are overconstraining for situations where the motion constraints are inherently volumetric. For example, moving within a safe corridor of space or working within a delimited surgical field without leaving it. Here, we consider generalizing curve following and targeting to *stubes* (about a reference curve) and *cones* (funneled toward a reference point).

1) *Tubes*: Suppose now that the task is described by a parametric curve equation  $\mathbf{p}(s)$  [defined as in (8)] representing a tube axis and a scalar value  $r$  representing a tube radius as depicted in Fig. 2. The tube boundary is a switching surface between unassisted motion (inside the tube) and virtual fixturing

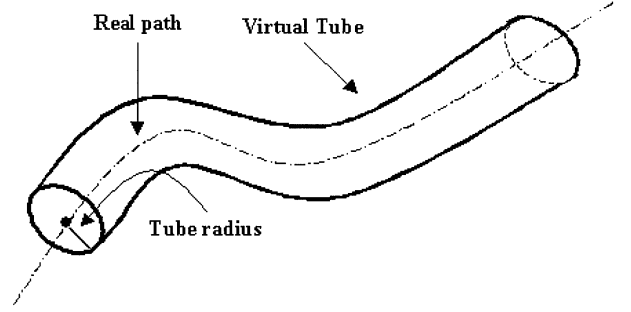


Fig. 2. Illustration of a virtual tube defined for a parametric curve.

behavior (outside the tube). Further, we define a switching transition region  $\epsilon$  with  $0 < \epsilon < r$  over which the gain discontinuity at the boundary is smoothed. We make use of  $\mathbf{e}$  as defined in (12) and note that motion is directed “outward” when  $\mathbf{e}(\mathbf{x}_a) \cdot \mathbf{f} < 0$  and “inward” otherwise. We then define a modified orthogonal admittance  $c_{tu}$  as follows:

$$c_{tu}(\mathbf{e}) = \begin{cases} c_\tau, & \|\mathbf{e}\| > r \\ c_\tau - \left[ \frac{r - \|\mathbf{e}\|}{\epsilon} \right]^n (c_\tau - 1), & (r - \epsilon < \|\mathbf{e}\| \leq r) \\ & \text{and } (\mathbf{e} \cdot \mathbf{f} < 0) \\ 1, & \text{otherwise} \end{cases} \quad (19)$$

where  $n \geq 1$  is a scalar value that shapes the switching region. In particular,  $n = 1$  generates a linear function while larger values tend to create a sharper transition from internal to external behavior. In all cases, there is a  $C_1$  discontinuity at the boundary of the region.

The last step is to define the reference direction fixture by defining the preferred direction  $\delta_{tu}$  by modifying  $\delta_p$  given in (11) to be

$$\delta_{tu}(\mathbf{x}_a) = \text{signum}(\mathbf{f} \cdot \delta_p(\mathbf{x}_a))\delta_p(\mathbf{x}_a) + k_d \mathbf{e}_t(\mathbf{x}_a) \quad (20)$$

where

$$\mathbf{e}_t(\mathbf{x}_a) = \begin{cases} \mathbf{0}, & \|\mathbf{e}(\mathbf{x}_a)\| \leq r \\ \frac{\mathbf{e}(\mathbf{x}_a)}{\|\mathbf{e}(\mathbf{x}_a)\|} (\|\mathbf{e}(\mathbf{x}_a)\| - r), & \|\mathbf{e}(\mathbf{x}_a)\| > r. \end{cases} \quad (21)$$

We then apply (6) with  $\delta = \delta_{tu}$  and  $c_\tau = c_{tu}(\mathbf{e})$ .

This scheme guarantees smooth transitions passing through the tube surface, as it connects the admittance values defined inside and outside the tube. We note that the discontinuity in  $\mathbf{e}_t$  will result in discontinuity of the *direction* of the apparent admittance of the mechanism, but not its magnitude. Implementing this scheme with  $c_\tau = 0$  corresponds to a situation where it is possible to freely move the tool inside the tube, but the inside surface of the tube has the property of a “hard” virtual fixture.

2) *Cones*: The idea of a cone is to create a “virtual chamfer” to guide a user to a high-precision target point as shown in Fig. 3. To this end, consider a target position  $\mathbf{x}_t$ , a fixed tool start position  $\mathbf{x}_s$ , and a base cone opening angle  $\alpha$ . The direction of the cone axis  $\mathbf{a}$  is

$$\mathbf{a} = \frac{(\mathbf{x}_t - \mathbf{x}_s)}{\|\mathbf{x}_t - \mathbf{x}_s\|}. \quad (22)$$

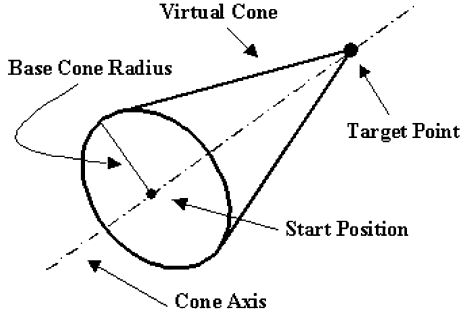


Fig. 3. Illustration of a virtual cone defined for a target point.

Given the current tool position in Cartesian space  $\mathbf{x}_a$ , we can decompose it into its orthogonal components relative to the cone axis as

$$\mathbf{n}_\perp(\mathbf{x}_a) = (I - \mathbf{a} \cdot \mathbf{a}^T)(\mathbf{x}_t - \mathbf{x}_a) \quad (23)$$

$$\mathbf{n}_\parallel(\mathbf{x}_a) = (\mathbf{a} \cdot \mathbf{a}^T)(\mathbf{x}_t - \mathbf{x}_a) \quad (24)$$

where  $I$  is the  $3 \times 3$  identity matrix. For convenience, we define the distance from the cone axis to the boundary as

$$\text{cone}(\mathbf{x}_a) = \tan(\alpha) \|\mathbf{n}_\parallel(\mathbf{x}_a)\|. \quad (25)$$

With these definitions, the tool tip is inside the cone when the following condition is satisfied:

$$\text{inside}(\mathbf{x}_a) \equiv \|\mathbf{n}_\perp(\mathbf{x}_a)\| \leq \text{cone}(\mathbf{x}_a). \quad (26)$$

As in the case of tubes, our goal is to define a smooth transition between the behavior of the robot inside and outside the cone. To this end, at each position  $\mathbf{x}_a$  inside the cone (with the exclusion of the points lying along the axis), it is possible to compute the direction  $\eta$ , normal to the direction  $\delta_t(\mathbf{x}_a)$ , co-planar to  $\mathbf{n}_\perp(\mathbf{x}_a)$ , and pointing inside the cone as follows:

$$\eta(\mathbf{x}_a) = \delta_t(\mathbf{x}_a) \times (\mathbf{n}_\perp(\mathbf{x}_a) \times \delta_t(\mathbf{x}_a)). \quad (27)$$

We consider the user force to be directed outward if

$$\mathbf{f} \cdot \eta(\mathbf{x}_a) \geq 0$$

and inward otherwise. We then introduce a switching region  $\epsilon$  where a point  $\mathbf{x}_a$  is in the region if

$$\text{trans}(\mathbf{x}_a) \equiv \text{cone}(\mathbf{x}_a) - \epsilon < \|\mathbf{n}_\perp(\mathbf{x}_a)\| \leq \text{cone}(\mathbf{x}_a).$$

We define the shaped gain in this region as

$$\tilde{c}(\mathbf{x}_a) = c_\tau - \left[ \frac{\text{cone}(\mathbf{x}_a) - \|\mathbf{n}_\perp(\mathbf{x}_a)\|}{\epsilon} \right]^n (c_\tau - 1)$$

where  $n \geq 1$  again shapes the switching region. Finally, we define the gain function as

$$c_{\text{co}}(\mathbf{x}_a) = \begin{cases} c_\tau, & \text{inside}(\mathbf{x}_a) \\ \tilde{c}(\mathbf{x}_a), & \text{trans}(\mathbf{x}_a) \text{ and } \mathbf{f} \cdot \eta(\mathbf{x}_a) \geq 0 \\ 1, & \text{otherwise.} \end{cases} \quad (28)$$

As with tubes, the first case, which corresponds to the tool being outside the cone, chooses the gain to create an anisotropic admittance, returning the user to the interior. The final case, when the tool is inside the cone, creates an isotropic admittance. The

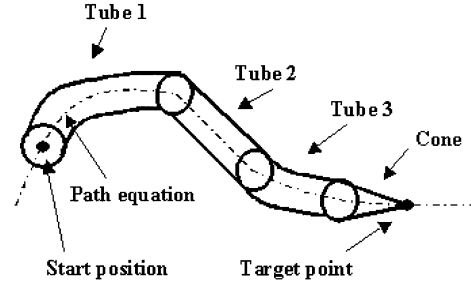


Fig. 4. Concatenation of virtual tubes and cones.

second case creates a smooth transition from inside to outside provided the motion of the tool is directed toward the outside. If, on the other hand, the tool motion is directed inward ( $\mathbf{f} \cdot \eta(\mathbf{x}_a) < 0$ ), then the gain reverts to the isotropic case.

The definition of  $\delta_t(\mathbf{x}_a)$  defined for the *Reference target* case (7) is modified to become

$$\delta_{\text{co}}(\mathbf{x}_a) = \text{signum}(\mathbf{f}^T \delta_t(\mathbf{x}_a)) \delta_t(\mathbf{x}_a) + k_d \mathbf{e}_{\text{co}}(\mathbf{x}_a) \quad (29)$$

with

$$\mathbf{e}_{\text{co}}(\mathbf{x}_a) = \begin{cases} \mathbf{0}, & \|\mathbf{n}_\perp(\mathbf{x}_a)\| \leq \text{cone}(\mathbf{x}_a) \\ \frac{\mathbf{n}_\perp(\|\mathbf{n}_\perp(\mathbf{x}_a)\| - \text{cone}(\mathbf{x}_a))}{\|\mathbf{n}_\perp(\mathbf{x}_a)\|}, & \text{otherwise.} \end{cases} \quad (30)$$

We then apply (6) with  $\delta = \delta_{\text{co}}$  and  $c_\tau = c_{\text{co}}(\mathbf{x}_a)$ .

This scheme again guarantees smooth transitions (with derivative discontinuity) while passing through the cone surface. As in the previous case, it follows that implementing this scheme with admittance gain equal to zero ( $c_\tau = 0$ ) corresponds to a situation where it is possible to freely move the tool inside the cone but it is not possible to exit from it. In this case, the system behaves as if a true “virtual chamfer” were funneling the user to the target point.

Finally, as we have defined it, there is a small region near the tip of the cone that is fully within the transition region. At the boundary of this region, we switch to a final positioning mode as described in the case of positioning at a point as discussed in Section II-B2.

**3) Joining Virtual Tubes and Cones:** Consider a case (Fig. 4) where the goal to follow is a continuous but possibly nondifferentiable curve to a fixed point. This can be modeled by a tube for each differentiable part of the path plus a final cone. For simplicity, we assume that all of the tubes have the same radius  $r$  and the same dimension  $\epsilon$  for the switching region. In this case, the proposed algorithm for tubes can still be applied, provided we extend the evaluation of the preferred direction to curves with the singular points, as described previously.

We define a switching sphere of radius  $\mu$  about the final target point. At this point, the controller switches to a cone model, where we pick  $\alpha$  so that  $\mu \sin(\alpha) = r$ . We take the tool start position  $\mathbf{x}_s$  as the path point corresponding to the last tool position before entering the sphere. When passing the sphere, this scheme does not present any discontinuity problem provided the user is inside the tube (as the controller is providing an isotropic

admittance). If this is not the case, then the orientation change at the switch should be smoothed over multiple samples.

#### D. Velocity Scaling

We further shape the robot velocities in two ways. First, we implement a two-mode scaling of velocities to allow both slow, fine positioning and large, coarse motion. To do so, we compute a gain of the form

$$k_f(s_1, s_2, \kappa) = \begin{cases} s_1, & \|\mathbf{v}\| \leq \kappa \\ s_1(1 + s_2(\|\mathbf{v}\| - \kappa)), & \|\mathbf{v}\| > \kappa \end{cases} \quad (31)$$

where  $\|\mathbf{v}\|$  is the norm of the commanded Cartesian velocity. With this gain definition, pushing lightly on the handle moves it slowly (up to a velocity threshold,  $\kappa$ ) whereas pushing hard on the handle quickly increases velocity. We also impose a maximum velocity limit  $v_{\max}$  on the system.

The overall system loop is thus

$$\mathbf{v} = k_f(s_1, s_2, \kappa)[c(D_\delta + c_\tau(I - D_\delta))\mathbf{f}]. \quad (32)$$

### III. IMPLEMENTATION AND EXPERIMENTAL RESULTS

We have implemented the algorithms described above and performed experiments demonstrating the system's performance using macroscopic and microscopic scales. In this section, we present three classes of experiments. First, we performed experiments with the robot operating autonomously, setting the force based on the observed path direction, to provide baseline data. Second, we acquired performance data from multiple users to quantify the effect of admittance on path following. Finally, we performed a broader set of tests to evaluate the correctness and performance of the complete class of virtual fixtures defined above.

#### A. Experimental System

The algorithms described above were implemented on the JHU SHR [12]. The robot was equipped with a vision sensor and a force-sensing handle on the end effector. We chose to execute two-dimensional tasks (defining a task plane), so only the first two translational joints of the robot base were used. We performed experiments using a CCD camera with a lens of 8 mm focal length at the macro scale and a grin lens endoscope (Insight IE 3000) at the micro scale. The vision sensor always viewed the task plane, allowing reading of the motion references and real-time display of task execution (Fig. 5). The center of the image was graphically marked, and users were instructed to perform path-following and positioning tasks relative to this mark. The force sensor on the handle was used to record user inputs to the admittance control law. The force sensor resolution is 12.5 mN and force values are expressed as multiples of this base unit.

For the *path-following* case, the overall control gain (31) was  $k_f(0.0008, 0.003, 2)$  at macro scale and  $k_f(0.0003, 0.003, 2)$  at micro scale. At the macro scale, the velocity was saturated at 10 mm/s while, at micro scale, at 2 mm/s [hence, users experienced the velocity ramp up in (31)]. In addition, for the *positioning* case, the maximum velocity was reduced linearly to 0.1 mm/s within a circle of radius 10 mm from the target. At a distance

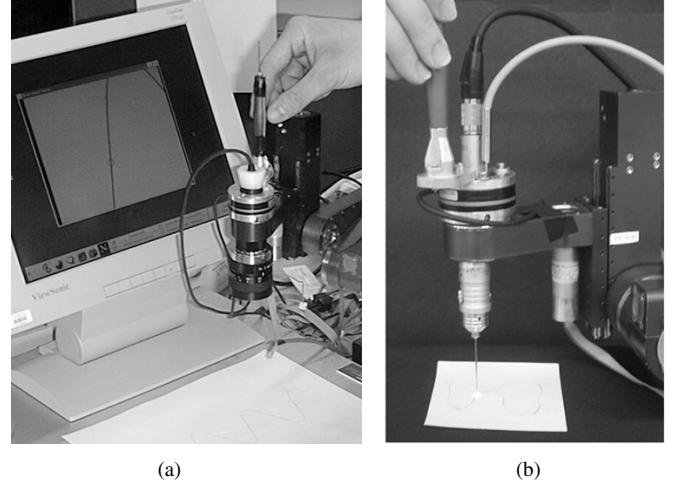


Fig. 5. Experimental setup of the SHR using virtual fixtures to assist in path-following and positioning tasks. (a) Macroscale. (b) Microscale.

of 1 mm from the target, the system switched to one of the four fine positioning schemes.

For the *positioning* case, the task was specified in absolute robot coordinates, and control was likewise based on absolute position error. A visual cue (a black mark on a sheet of paper) was created at the appropriate location to guide the user. For the *path-following* case, at macro scale, the path is furnished to both the system and the user by printing a sine curve (35 mm amplitude, 70 mm wavelength, and 0.54 mm width) on the task plane (in black on white paper). A similar approach was taken to demonstrate the system performance at the micro scale. However, at this scale, it was not possible to print a sufficiently smooth curve, so we instead embedded a human hair (about 80  $\mu\text{m}$  in diameter) in glue on a yellow sheet of paper.

Visual tracking (XVision system [5]) was used to sense (in real time) the local position and tangent direction of the path. Subpixel interpolation was used to increase the precision of these measurements. The vision and control subsystems executed on two different PCs and data exchanged over a local network. The control system operated at 100 Hz, using the most recent available data from the vision system and handle force sensor. The vision system recorded the direction and error estimates at each sample time: the macro scale used  $640 \times 480$  pixel images at 30 Hz, while the micro scale used  $320 \times 240$  pixel images at 60 Hz.

#### B. Baseline Results With Autonomous Robot

In order to establish baseline performance for the control system without a human in the loop, we executed path following (Section II-B) at the macroscale along the sine path with  $c_\tau = 0$ , generating Cartesian velocities along the tangent (as calculated by visual tracking) to the curve. The camera was positioned 200 mm from the paper, yielding a working area of 614.1 mm<sup>2</sup> with a pixel footprint of 0.066 mm on the working surface.

This task was executed twice: once with a constant velocity norm of  $v = 1$  mm/s and once with a constant velocity norm of  $v = 10$  mm/s. In this first case, the tool moves very slowly and the resultant error can be considered to be a lower bound on system tracking error. The second velocity is at saturation and

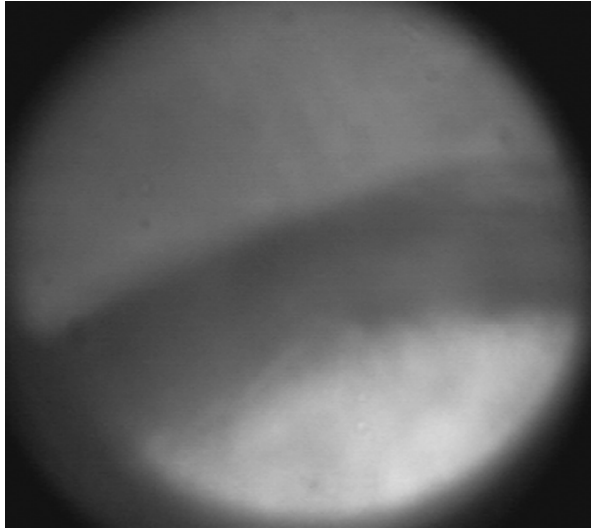


Fig. 6. Endoscope image of the 80- $\mu\text{m}$ -diameter human hair used as the path in microscale experiments.

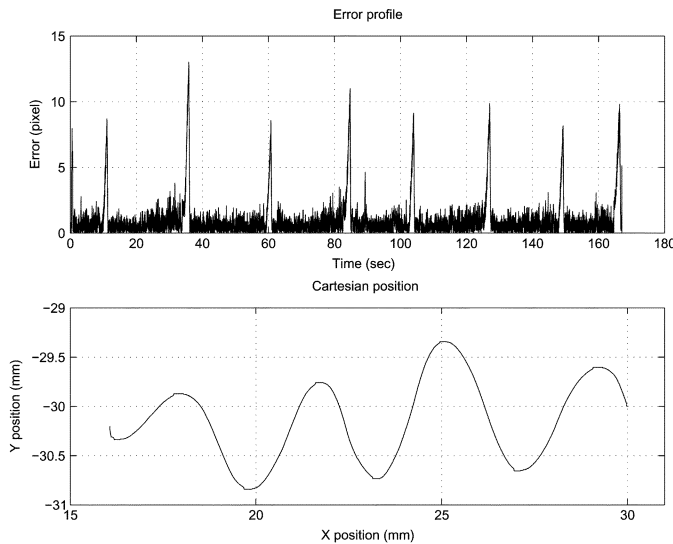


Fig. 7. Error profile and Cartesian positions for an autonomous robot when following an 80- $\mu\text{m}$  hair at the microscale. Velocity  $v = 0.1$  mm/s.

can be considered an upper bound. The average error for  $v = 1$  mm/s is 0.28 pixel, while for  $v = 10$  mm/s it is 0.53 pixel. Also, in the case  $v = 1$  mm/s, the encoder positions were taken as a measurement of the underlying path.

We performed this experiment at the microscale as well. In this case, the velocity norm was  $v = 0.1$  mm/s and the endoscope was about 150  $\mu\text{m}$  above the working surface, yielding a pixel footprint of about 1  $\mu\text{m}$  (Fig. 6). Fig. 7 reports the error profile and the Cartesian positions read from the encoders during the motion. The average error is 0.96 pixel, but it can be seen that much of the tracking error occurs in regions where the robot motors switch direction. The low scale of operation (peak amplitude is about 20  $\mu\text{m}$ ) suffers from the effects of the low-level PID servo-controllers that control the motors. Simply put, the abstraction of the robot as a kinematic device devoid of dynamics breaks down. If we disregard these peaks, then the error reduces to 0.6 pixel. As for the macroscale

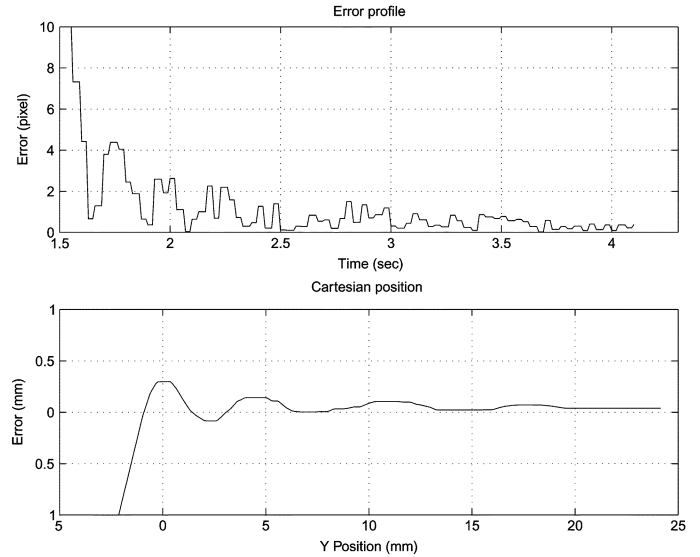


Fig. 8. Error profile and Cartesian positions for an independent robot when approaching a linear path at the macroscale. Velocity  $v = 10$  mm/s. The graphics are enlarged to highlight the details.

case, considering the low visual error, we take the measured Cartesian path a good estimate of the true one.

To demonstrate the dynamic behavior of the system when the tool is positioned off-path, the tool was positioned at  $x = 35$  mm,  $y = -10$  mm and a linear path (approximately the  $x$  axis) was used as the reference path. The robot was commanded to move at a constant velocity with a norm of  $v = 10$  mm/s (the maximum allowed) and the nonpreferred admittance was set to  $c_T = 0$ . Fig. 8 reports the error profile and the Cartesian positions of tool trajectory. It is possible to see how the tool, although it is running at the maximum velocity allowed and the error from the path is large, quickly reduced the error to the normal path-following range (less than 0.4 pixel).

### C. Path Following: Human Performance Results in Macroscale

A study was performed to quantitatively gauge the effect of admittance level on human-machine performance (measured by time and accuracy) in the macroscale setup. In the experiment, eight subjects were instructed to follow the reference path. The camera was positioned 125 mm from the paper, yielding a pixel footprint of 0.044 mm on the working surface. They were asked to move the robot as quickly as was comfortable, while not sacrificing accuracy. During the experiment, four different admittance ratios:  $c_T = 0$  (hard virtual fixturing),  $c_T = 0.3$  (medium virtual fixturing),  $c_T = 0.6$  (soft virtual fixturing), and  $c_T = 1$  (no virtual fixturing/free motion) values were applied at random. Prior to beginning each trial, the subjects were told which scenario they were about to perform. The subjects performed three trials for each admittance value. Time was recorded in seconds and error in pixels. Error in pixels is the sum of error collected from the starting point to the end point averaged over the total sampling time. Each subject was allowed to have at least one practice on each test scenario prior to the experiment.

The experimental results are shown in Table I, with the average of error and time of the eight subjects performing the path



TABLE I

AVERAGE SUM OF ERROR AND EXECUTION TIME IN THE PATH-FOLLOWING TASK FOR EIGHT SUBJECTS UNDER VARIOUS ADMITTANCE CONDITIONS AT THE MACRO-SCALE

$c_\tau$	Average error (pixels)	Average Execution time (sec)
0	1.133	18.710
0.3	1.438	19.647
0.6	2.241	20.425
1	4.238	24.745

TABLE II

T-TEST RESULTS OF PATH FOLLOWING ON AVERAGES OF EXECUTION TIME AND ERROR FOR ALL SUBJECTS. P-VALUE IS BASED ON 14 DOF, AND THE ITALICIZED ADMITTANCE LEVELS PROVIDED LOWER AVERAGE TIME AND ERROR VALUES

$c_\tau$	Time t-value	Time p-value	Error t-value	Error p-value
1 vs. <i>0.6</i>	1.978	0.035	3.113	0.003
1 vs. <i>0.3</i>	2.357	0.02	4.501	0
1 vs. <i>0</i>	2.865	0.007	5.057	0
<i>0.6</i> vs. <i>0.3</i>	0.742	0.24	3.225	0.004
<i>0.6</i> vs. <i>0</i>	1.851	0.04	4.862	0
<i>0.3</i> vs. <i>0</i>	1.069	0.15	1.839	0.04

following task. Table II shows the t-test results for the comparison between each admittance ratio used in the experiment.

A large performance improvement is evident between  $c_\tau = 1$  (free motion) and  $c_\tau = 0$  (hard virtual fixturing) for execution time ( $t_{14} = 2.865, p = 0.007$ ) and error ( $t_{14} = 5.057, p = 0$ ), as shown in Table II. All subjects moved faster with higher accuracy along the path with at least a small degree of assistance from the virtual fixture, as indicated by a sharp drop in error between free motion ( $c_\tau = 1$ ) and  $c_\tau = 0.6$ . Comparison between the p-values for execution time and error indicates that guidance provided by the virtual fixture guarantees improvement in error reduction. However, the high p-values for execution time comparisons between  $c_\tau = 0.6, c_\tau = 0.3$ , and  $c_\tau = 0$  indicate that a decrease in admittance does not necessarily improve execution time. However, none of the subjects performed worse in both time and error simultaneously when admittance was decreased. Fig. 9 shows a typical error profile of a subject performing the path-following task. The profiles demonstrate reduction in error and execution time as guidance provided by the virtual fixture increases. Hard virtual fixturing ( $c_\tau = 0$ ) offers the best performance improvement.

A varying level of admittance allows a user to deviate from the preferred path if necessary (e.g., due to vision errors and unmodeled tasks). In order to examine the effect of different  $c_\tau$  values on a user's ability to generate motions *not* dictated by the virtual fixture, an "avoidance region," a 10-mm-radius circle, was added near the center of the path. The users were told to follow the path as closely as possible and leave it only to avoid the area inside the circle (this is only possible when  $c_\tau \neq 0$ ). To make the task as consistent as possible, the users were asked to avoid the circular region by traveling around it on the right half of its perimeter. In addition, they were told to leave and return to the curve as close to the intersection between the curve and the circle as possible. The results are shown in Table III with the

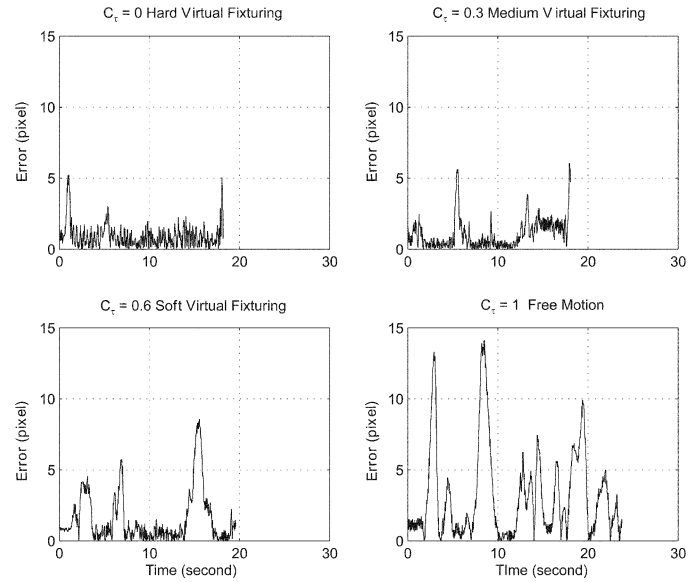


Fig. 9. Error profiles of a subject for the *path-following* case at the macroscale with different orthogonal admittance values.

TABLE III

AVERAGE OBSTACLE AVOIDANCE TIME IN THE PATH-FOLLOWING TASK FOR EIGHT SUBJECTS UNDER VARIOUS ADMITTANCE CONDITIONS AT THE MACRO-SCALE

$c_\tau$	Average Total Time (sec)	Average Obstacle Avoidance Time(sec)	Average Path Following Time(sec)
0.3	33.615	14.681	18.934
0.6	31.213	11.871	19.343
1	33.055	9.317	23.738

TABLE IV

T-TEST RESULTS OF PATH FOLLOWING ON AVERAGES OF OBSTACLE AVOIDANCE TIME FOR ALL SUBJECTS. P-VALUE IS BASED ON 14 DOF, AND THE ITALICIZED ADMITTANCE LEVELS PROVIDED LOWER AVERAGE TIME VALUES

$c_\tau$	Avoidance Time	p-value
1 vs. <i>0.6</i>	1.171	0.13
1 vs. <i>0.3</i>	2.233	0.02
<i>0.6</i> vs. <i>0.3</i>	0.991	0.15

t-test results of the obstacle avoidance time compared between each value of  $c_\tau$  shown in Table IV.

With higher levels of guidance, execution time increased significantly due to the effort involved in shifting from the path to avoid the circle. Increasing  $c_\tau$  (using less virtual fixture guidance) produced a larger average error and a longer time for path following, but the lower damping dropped the object avoidance time. The case of  $c_\tau = 1$  (free motion) offers the best time reduction in obstacle avoidance, though not significantly different from the case when  $c_\tau = 0.6$ , which improves error and time significantly over the free motion (Table II). Therefore, having virtual fixture guidance, even at a minimal level, improves the performance of macroscale tasks. The error profiles of the path following with obstacle avoidance task are shown in Fig. 10 with the off-scale portion indicating when the user leaves the path to avoid the circle.

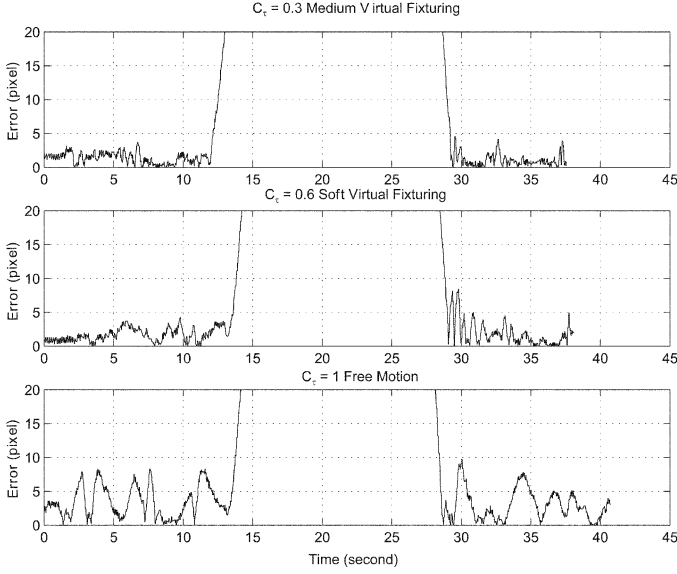


Fig. 10. Error profiles of a subject for the *obstacle avoidance* case at the macroscale with different orthogonal admittance values.

TABLE V  
ERROR AND EXECUTION TIME FOR AN EXPERT USER UNDER VARIOUS ADMITTANCE CONDITIONS AT THE MICRO-SCALE

$c_T$	Average error (pixel)	Average Execution time (sec)
0	3.20	26.86
0.3	4.97	34.85
0.6	7.64	45.66
1	8.15	55.75

#### D. Path Following: Results in the Microscale

At the microscopic scale, no obstacle was presented. The same four admittance conditions were used:  $c_T = 0, 0.3, 0.6, 1$ . A single expert user performed the path-following task on a piece of curly human hair as quickly and accurately as possible. Table V summarizes the results, and Fig. 11 shows the error and velocity profiles.

From Figs. 7 and 11, it can be seen that much of the tracking error again occurs in regions where the robot motors switch direction. An additional problem is that the robot no longer appears stiff at this scale—even a light touch on the force handle introduces significant perturbations of the endpoint position. Still, apart from the peaks due to switching, the average error is 1.44 pixels, which corresponds to less than  $2 \mu\text{m}$ . This demonstrates that the virtual fixturing scheme is effective in following a path, even when external disturbances introduce unexpected errors. In comparison to the macroscale case, the execution time at the microscale varies far more significantly with orthogonal admittance. This seems to be due to the fact that the user is only provided with a narrow view of the hair from the endoscope (Fig. 6). When the user moves quickly, he or she does not have a wide enough view to plan future velocities/position trajectories, so it is difficult to stay on the curve. Adding an additional control term related to the curvature of the line would improve this.

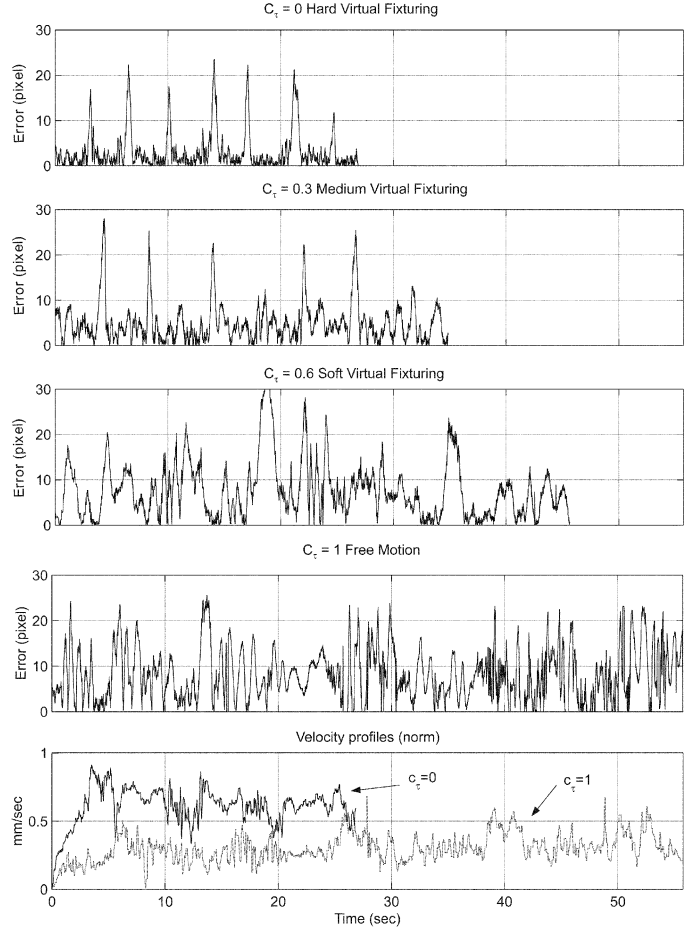


Fig. 11. Error and velocity profiles for the *path-following* case at the microscale for various orthogonal admittance values.

TABLE VI  
EXECUTION TIMES FOR FOUR DIFFERENT FINE POSITIONING SCHEMES AT THE MACRO-SCALE

Scheme	free motion	hard virtual fixture	Vel	Acc
T (sec)	22.54	0.72	1.04	3.03

#### E. Positioning Tests

Recall that path following and positioning use the same methods, but positioning suffers from singularity problems in the neighborhood of the target point. In Section II-B2, we described four schemes to use within this neighborhood. The first two vary the  $c_T$  value in  $S$ , creating free motion ( $c_T = 0$ ) and a hard virtual fixture ( $c_T = 1$ ). The last two are velocity and acceleration controllers that autonomously move to the target position, regardless of force applied by the user.

To compare these schemes, we executed the task of moving the robot from  $(0, 0)$  to  $(-35 \text{ mm}, 0)$ , with  $c_T = 1$  (for all cases),  $S$  of radius 1 mm, and a positioning error tolerance of  $5 \mu\text{m}$ . Velocity reduction is disabled inside  $S$  when using the two autonomous schemes. Table VI summarizes the times involved to reach the target after the controller switch. Figs. 12 and 13 show the Cartesian positions and the velocity profiles for the four approaches.

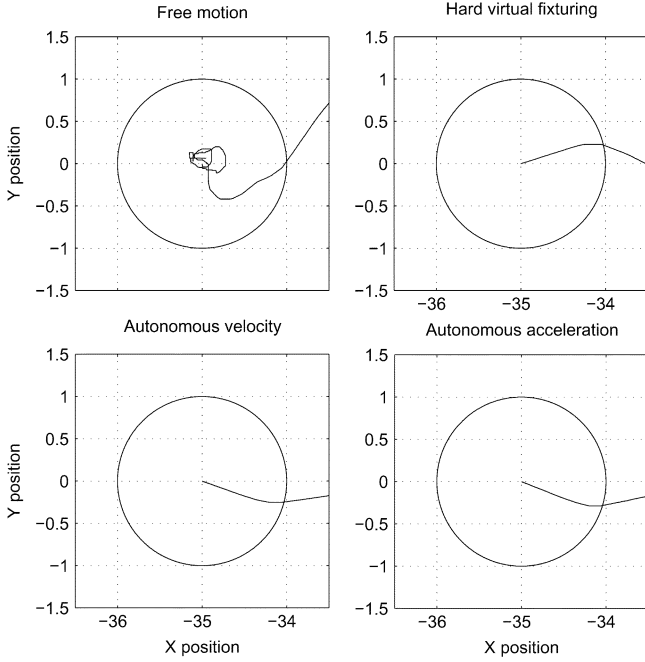


Fig. 12. Detail of target acquisition using the four proposed fine positioning schemes at the macroscale. The circle indicates the switching surface.

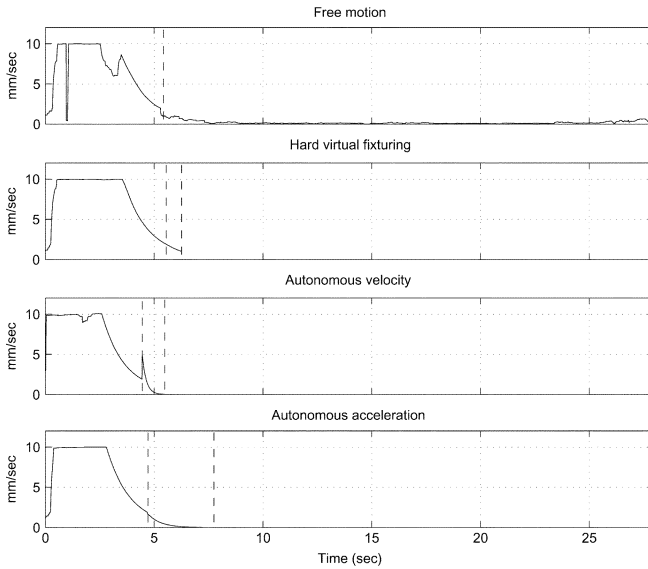


Fig. 13. Velocity profiles for the four proposed fine positioning schemes. The left and right dashed vertical lines mark the switching and motion termination events, respectively.

With the free motion scheme, the user takes much longer to acquire the target point. This is mainly due to the fact that the control loop is closed using encoder measurements from the robot. As a result of minor kinematic inaccuracies due to backlash and flex, it is difficult to determine the exact (encoder-based) robot position relative to the target on the screen. In contrast, the hard virtual fixturing scheme is fast and effective, as it can exploit the maximum possible velocity and is using the internal encoders as a reference. Furthermore, the velocity profile for this scheme is not discontinuous at the instant of switching.

The autonomous schemes are also effective, although they are somewhat slower than the hard virtual fixturing scheme be-

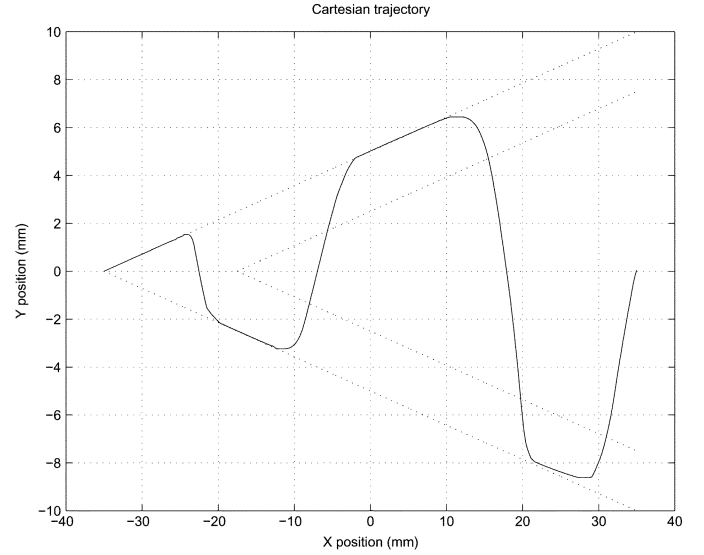


Fig. 14. Cartesian trajectory of the tool while positioning when a cone is defined. The larger cone delimits the free motion zone. The smaller cone delimits the unsmoothed zone.

cause the velocity is forced to zero as the target is approached. In the velocity controller, there is a discontinuity in the velocity profile, but it is too small to be felt by the user. The acceleration scheme avoids any discontinuity in the velocity profile, however it is slower than both hard virtual fixturing and autonomous velocity control. It is notable that the assistive, rather than the autonomous, schemes have the best performance for execution time.

#### F. Volumetric Primitives

Here we demonstrate the behavior of the volumetric primitives.

1) *Cones*: The first demonstration shows how the system behaves in the targeting case when a cone is defined as described in Section II-C2. The virtual fixture is hard (outside admittance  $c_\tau = 0$ ). The initial and target positions were respectively  $(x = 35 \text{ mm}, y = 0)$  and  $(x = -35 \text{ mm}, y = 0)$ . The cone base radius was  $\xi = 10 \text{ mm}$  (cone opening of  $8.13^\circ$ ) and the size of the switching transition region was  $\epsilon = 0.5 \text{ mm}$ . We used a linear function to smooth the damping discontinuity [ $n = 1$  in (28)]. The velocity reduction and the terminal switch to a fine positioning mode were implemented as described for the case of a target point (Section II-B2).

Figs. 14 and 15 report, respectively, the tool Cartesian trajectory and the force and position profiles during the motion. Fig. 14 shows the cone constraint acting on the tool motion. Inside the cone it is possible to move in every direction while, when approaching the perimeter, it becomes difficult to move in directions that point outside. In particular, when the tool reaches the boundary, it is not possible to go through the cone. As the cone perimeter is approached, the path is smoother than during motion toward the interior. This is due to dependence of admittance on motion direction in (28).

Fig. 15 reports the position and force profiles during the motion as function of the time. The  $y$  direction is normal to the boundary, so we see the  $y$  position is bounded although the user

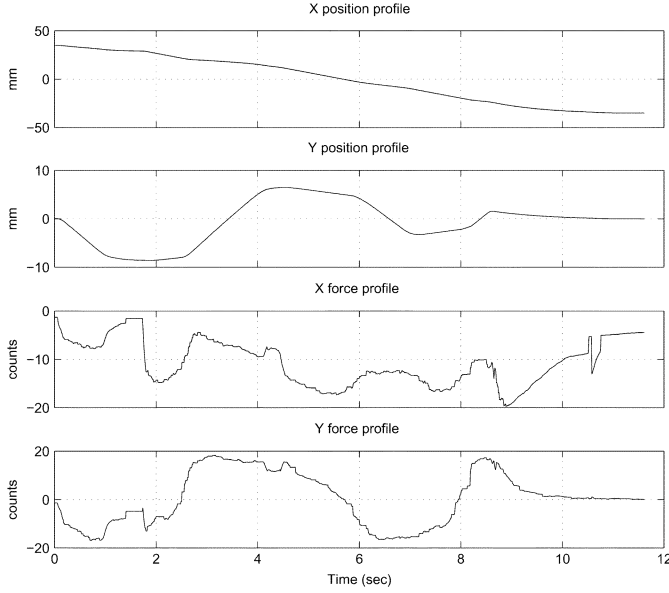


Fig. 15. Position and force profiles while positioning and a cone is defined.

is pushing in that direction. Also, the  $y$  component of the position appears as a delayed version of the  $y$  component of the force when the user tries to approach the perimeter, while the position seems to follow the force profile when leaving from it. This behavior is again due to the direction-dependent gain switch that occurs near the cone perimeter.

2) *Tubes*: This demonstration shows how the system behaves in the *Reference curve* case when a tube is modeled as described in Section II-C1. The tube was defined by drawing a sine curve in black on white paper (35 mm amplitude, 70 mm wavelength, and 3.4 mm width). The visual tracker, following both edges of the curve, was able to determine locally the perimeter positions and tangents and then estimate the tube width. A hard virtual fixture was used ( $c_T = 0$ ) and a linear law was used to smooth the admittance discontinuity ( $n = 1$ ) in (19). The switching region was about one fourth of the tube radius ( $r = 25.8$  pixels and  $\epsilon = r/4 = 6.45$  pixels). The starting position was (35 mm, 0) while the ending position was (−35 mm, 0).

Fig. 16 reports the vision error (with respect to the center line of the path) and the estimated position of the tube boundaries. The boundary estimation is somewhat noisy, especially in areas of high curvature, but the system is effective in keeping the tool inside the tube. The tool error appears to chatter at the boundary, but this is an artifact of the scale of the diagram (one pixel is 0.066 mm). Finally, Fig. 17 shows the position and force profiles, as a function of time, for this experiment.

### G. Extended Tasks

In this section, we briefly describe the results of executing tasks composed of multiple segments.

1) *Curve and Target*: We instructed the user to follow half of the sinusoidal path and position the robot at the Cartesian position (0,0) (approximately the center of the path). The system was set to switch from path following to the positioning as soon as the robot was within 10 mm of the target. For both segments, an orthogonal admittance ratio of  $c_T = 0.6$  was used.

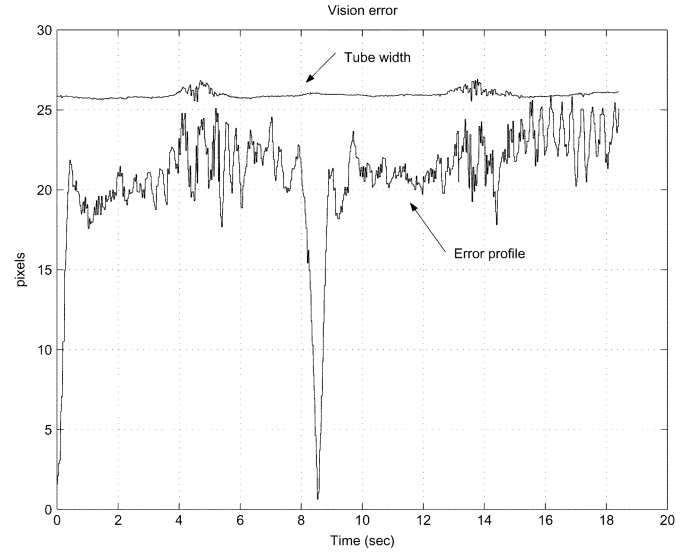


Fig. 16. Vision error profile while following a path and a tube is defined. The error is compared with the tube's estimated dimension.

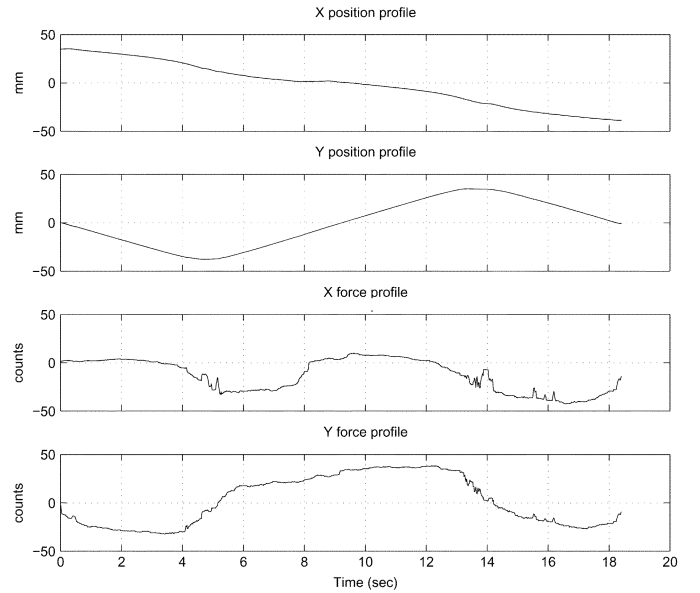


Fig. 17. Position and force profiles during the motion while following a path and a tube is defined.

In Figs. 18 and 19, the Cartesian positions and the velocity profile are shown. It is important to note that, in actuality, no velocity discontinuities are present during the controller switch. The jump in the error diagram is due only to a change in error definition: the path-following error (vision error, in pixels) is replaced by positioning error (robot encoder error, in mm).

2) *Tube and Cone*: The final demonstration shows the system behavior when both a tube and a cone are combined. We used the same thick sine path of the previous experiment, starting in the same position and imposing a stop condition in (0, 0) (approximately the center of the path). For both tube and cone, the orthogonal admittance ratio was  $c_T = 0$ , and a linear law was used to smooth the admittance discontinuity [ $n = 1$  in (19) and (28)]. The system was set to switch from tube to cone when the tool was within 10 mm of the target.

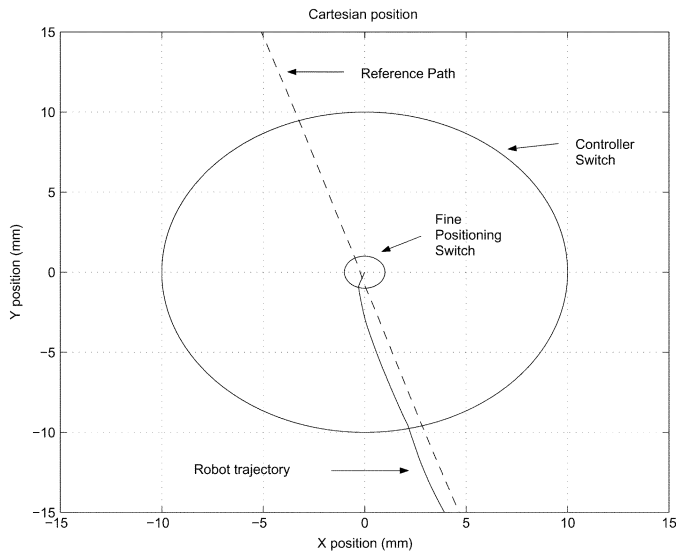


Fig. 18. Details of the Cartesian positions in the switch between controllers for the combined task of path following and positioning in the target.

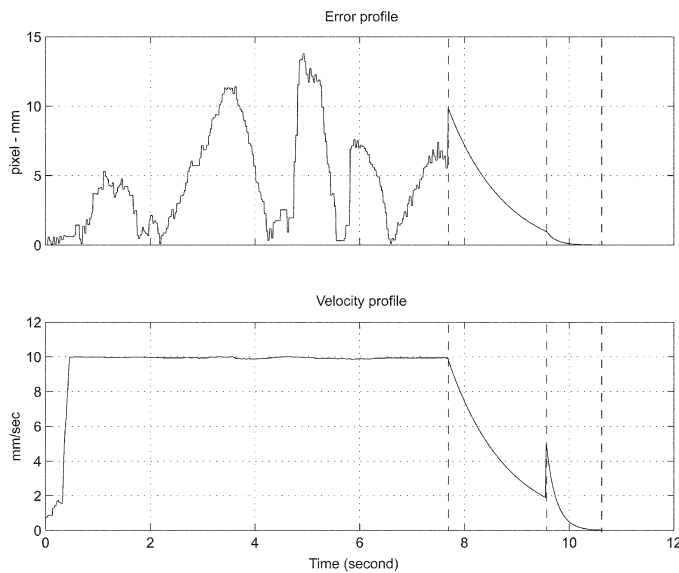


Fig. 19. Error and velocity profile for the combined task of path following and positioning in the target. The vertical lines mark, from left to right, the control switch time, the fine positioning switch time, and the motion termination time. Errors before and after the control switch time are expressed in pixels and millimeters, respectively.

The switching region was dimensioned as one fourth of the tube radius (estimated by the tracker) and after the switch, the tube parameters ( $r$  and  $\epsilon = r/4$ ) converted from pixels to millimeters, were used to correctly dimension the cone.

The behavior in the tube part is similar to the previous experiment, as shown in Fig. 20. The jump in the error profile at the switch is again due to a different error definition. Finally, Fig. 21 details the position and force profiles during the motion.

#### IV. CONCLUSION

In the context of Human-Machine Collaborative Systems (HMCS), we have proposed a control law for implementing virtual fixtures using anisotropic admittance. The basic virtual

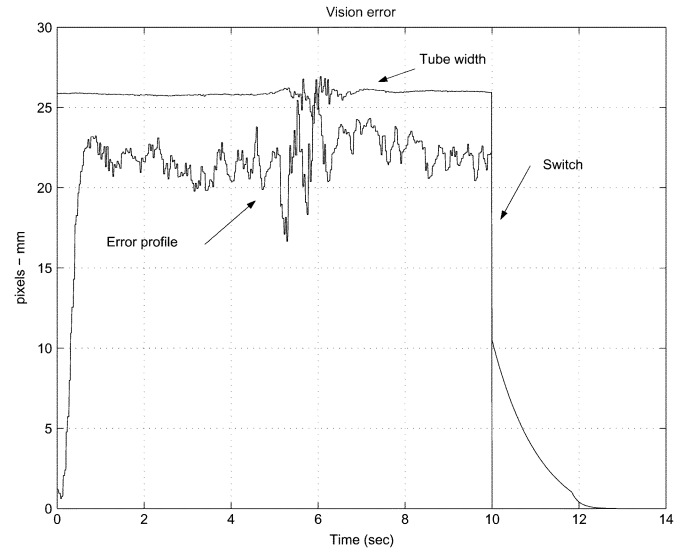


Fig. 20. Vision error profile while following a path and positioning and, a tube and a cone are defined. The error is compared with the tube's estimated dimension.

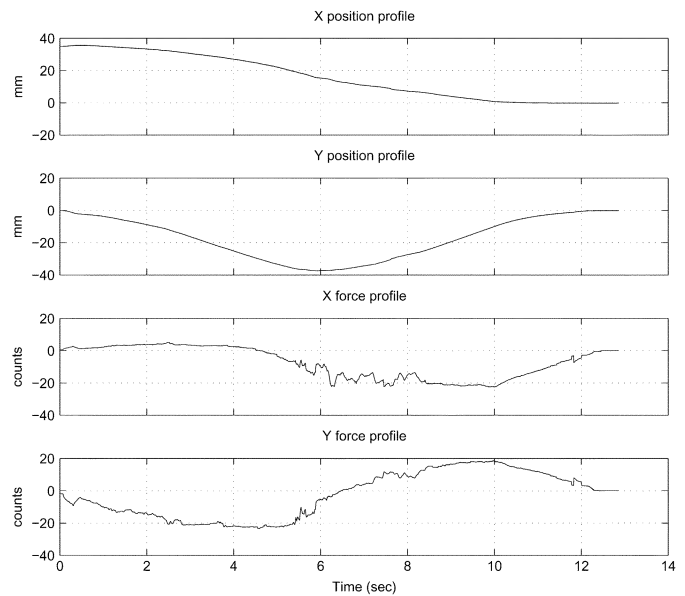


Fig. 21. Position and force profiles while following a path and positioning, and a tube and a cone are defined.

fixture provides guidance along a path, and specialized virtual fixture geometries such as tubes and cones can be used to confine motion to a volume. The JHU SHR, equipped with a camera (at the macroscale) and an endoscope (at the microscale) was used to validate the proposed controller and fixtures while executing planar tasks.

Autonomous robot and human-machine cooperation experiments, performed for path following and positioning tasks, validated the basic implementation of the controller. The fine positioning method of switching from soft to hard virtual fixturing showed the best performance for approaching a target. Other experiments, performed with virtual cones and tubes, demonstrated how virtual fixturing techniques allow users to move easily inside subspaces and smoothly between connected volumes.

As expected, the system demonstrates a tradeoff between user performance (measured by average error and execution time) and user control. We observed that assisted execution always outperforms unassisted execution for hard, medium, and soft virtual fixturing cases. In particular, hard virtual fixtures (with admittance ratio  $c_\tau = 0$ ) provide the best performance, although they seriously limit the user's ability to control the direction of motion. This could be problematic in cases where the virtual fixture geometry, determined by computer vision, is inaccurate. However, we also demonstrated that even soft virtual fixturing (with admittance ratio  $c_\tau = 0.6$ ) improves user performance with statistical significance, while maintaining a great deal of user control. At the macroscale, a hard virtual fixture resulted in improvements of 78% in error and 32% in execution time, with respect to unassisted execution. At the microscale, the improvement is 60% in error and 52% in execution time. Furthermore, not considering motor transients of the SHR, the improvement in error rises to 82%. Users reacted similarly to the different admittance levels at the macro and micro scales.

For application to vitreoretinal surgery, the necessary performance of the system at the microscale must be characterized. Additionally, improving performance beyond the results obtained in this work presents significant challenges in robot design and control. For example, the grin lens endoscope provided a pixel footprint of  $1\ \mu\text{m}$ , which may be too large for smooth control in the execution of some microscale tasks. Thus, a different imaging technology may be required. The resolution of the position sensors and motor torque output, as well as robot dynamics, also contribute to controller error. Perhaps the most demanding design challenge is the inherent flexibility of the robot. A detailed model of the system may be used to cancel out this effect, control laws can compensate for unactuated image motion, or the robot could simply be designed in a stiffer (e.g., parallel) configuration.

We are currently continuing to further develop and evaluate the proposed framework, both in a general context and for the specific application of vitreoretinal eye surgery. In recent work, we have developed the generalization of admittance-control based virtual fixtures to more general configuration spaces [6], [14]. With this, we plan to perform a comprehensive user study to quantify average performance enhancement and the learning effects involved in our approach. The larger problem of developing HMCS systems requires not only control methods for providing assistance, but also sophisticated environment and "human intent" sensing. By developing detailed task models and comparing user inputs to those models, we hope to provide real-time, context-based assistance [13], [15].

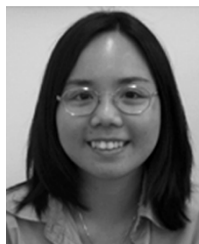
## REFERENCES

- [1] S. E. Everett and R. V. Dubey, "Vision-based end-effector alignment assistance for teleoperation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1999, pp. 543–549.
- [2] J. T. Feddema and T. R. Christenson, "Parallel assembly of high aspect ratio microstructures," *Proc. SPIE*, vol. 3834, pp. 153–164, 1999.
- [3] J. T. Feddema and R. W. Simon, "Visual servoing and cad-driven microassembly," in *IEEE Robot Automat. Mag.*, vol. 54, Dec. 1998, pp. 17–24.
- [4] J. T. Feddema, P. Xavier, and R. Brown, "Micro-assembly planning with van der Waals force," in *Proc. IEEE Int. Symp. Assembly and Task Planning*, 1999, pp. 32–38.
- [5] G. D. Hager and K. Toyama, "The "XVision" system: A general purpose substrate for real-time vision applications," *Comp. Vision, Image Understanding*, vol. 69, no. 1, pp. 23–27, Jan. 1998.
- [6] G. D. Hager, Human-machine cooperative manipulation with vision-based motion constraints, presented at *Proc. Workshop Visual Servoing with IROS*. [Online] <http://www.cs.jhu.edu/CIRL/new/publications.html>
- [7] G. Hirzinger, G. Grunwald, B. Brunner, and H. Heindl, "A sensor-based telerobotic system for the space robot experiment ROTEX," in *Proc. 2nd Int. Symp. Experimental Robotics*, Toulouse, France, 1991.
- [8] J. M. Hollerbach and H. Maekawa, "Haptic display for object grasping and manipulating in virtual environment," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1998, pp. 2566–2573.
- [9] L. D. Joly, "Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1994, pp. 357–362.
- [10] K. Kosuge, K. Takeo, and T. Fukuda, "Unified approach for teleoperation of virtual and real environment—Manipulation based on reference dynamics," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 938–943.
- [11] R. Kumar, T. M. Goradia, A. Barnes, P. Jensen, L. L. Whitcomb, D. Stoianovici, L. M. Auer, and R. H. Taylor, "Performance of robotic augmentation in common dextrous surgical motions," in *Proc. Medical Image Computing and Computer-Assisted Interventions (MICCAI-99)*, vol. 1679, 1999, pp. 1108–1115.
- [12] R. Kumar, G. D. Hager, P. Jensen, and R. H. Taylor, "An augmentation system for fine manipulation," in *Proc. Medical Image Computing and Computer Assisted Intervention*, 2000, pp. 956–965.
- [13] M. Li and A. M. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in *Proc. 11th Int. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, pp. 125–131.
- [14] P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures," in *Proc. ICRA*, 2003, pp. 1954–1959.
- [15] J. Nolin, P. Stemniski, and A. M. Okamura, "Activation cues and force scaling methods for virtual fixtures," in *Proc. 11th Int. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, pp. 404–409.
- [16] R. Paul and C. Sayers, "An operator interface for teleprogramming employing synthetic fixtures," *Presence*, vol. 3, no. 4, pp. 309–320, 1994.
- [17] M. A. Peshkin, J. E. Colgate, W. Wannasuphprasit, C. A. Moore, B. Gillespie, and P. Akella, "Cobot architecture," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 377–390, Aug. 2001.
- [18] F. Lai and R. D. Howe, "Virtual Fixtures for Robotics Endoscopic Surgery," Harvard Univ., Cambridge, MA, Biorobotic Laboratory Rep., 2000.
- [19] C. N. Riviere and P. K. Khosla, "Characteristics of hand motion of eye surgeons," in *Proc. IEEE 19th Annu. Int. Conf.*, vol. 4, Oct. 30–Nov. 2 1997, pp. 1690–1693. 199.
- [20] L. Rosenberg, "Virtual Fixtures," Ph.D. Thesis, Dept. of Mechanical Engineering, Stanford Univ., Stanford, CA, 1994.
- [21] C. A. Sayers, *Remote Control Robotics*. London, U.K.: Springer-Verlag, 1998.
- [22] Z. Stanisic, S. Payandeh, and E. Jackson, "Virtual fixtures as an aid for teleoperation," in *Proc. 9th Can. Aeronautic and Space Institute Conf.*, 1996, pp. 18–23.
- [23] S. Tamai, "History of microsurgery—From the beginning until the end of the 1970s," *Microsurgery*, vol. 14, no. 1, p. 6, 1993.
- [24] R. Taylor, P. Jensen, and L. Whitcomb *et al.*, "Steady Hand robotic system for microsurgical augmentation," *Int. J. Robot. Res.*, vol. 18, no. 12, pp. 1201–1210, 1999.
- [25] J. N. Weiss and L. A. Bynoe, "Treatment of central retinal vein occlusion by injection of tissue plasminogen activator into a retinal vein," *Ophthalmology*, vol. 108, pp. 2249–2257, 2001.



**Alessandro Bettini** received the "Laurea" degree in computer engineering and the Ph.D. degree in control systems engineering from the Università degli Studi di Roma "La Sapienza," Rome, Italy, in 1997 and 2002, respectively.

From October 2000 to July 2001, he was a visiting student with the ERC-CISST of the Johns Hopkins University, Baltimore, MD. His research interests lie in the area of control systems engineering.



**Panadda Marayong** (S'02) received the B.S. degree from the Florida Institute of Technology, Melbourne, and the M.S. degree in mechanical engineering from Johns Hopkins University, Baltimore, MD, in 2003, where she is currently working toward the Ph.D. degree in mechanical engineering.

Her research interests include cooperative manipulation, virtual fixture-assisted manipulation, and human-machine interaction.

Ms. Marayong is a member of Tau Beta Pi and Pi Tau Sigma.



**Samuel Lang** received the M.S. degree in computer science from the Johns Hopkins University, Baltimore, MD, in 2002.

While at Johns Hopkins, he developed software components for microscale visual tracking and virtual fixture manipulation. His research interests include software systems and computer vision.



**Allison M. Okamura** (S'98–M'04) received the B.S. degree from the University of California, Berkeley, in 1994 and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1996 and 2000, respectively, all in mechanical engineering.

She is currently an Assistant Professor of Mechanical Engineering and a faculty member of the Engineering Research Center for Computer-Integrated Surgical Systems and Technology, The Johns Hopkins University, Baltimore, MD. Her research interests include haptic display in virtual

environments and teleoperation systems, medical robotics, and exploration with robotic fingers.



**Gregory D. Hager** (S'86–M'90–SM'02) received the B.A. degree (*summa cum laude*) in computer science and mathematics from Luther College, Decorah, IA, in 1983, and the M.S. and Ph.D. degrees in computer science from the University of Pennsylvania, Philadelphia, in 1985 and 1988, respectively.

From 1988 to 1990, he was a Fulbright Junior Research Fellow with the University of Karlsruhe, Karlsruhe, Germany, and the Fraunhofer Institute IITB, Karlsruhe. From 1991 until 1999, he was with the

Computer Science Department, Yale University, New Haven, CT. In 1999, he joined the Computer Science Department, The Johns Hopkins University, Baltimore, MD, where he is now a full Professor and a faculty member with the Center for Computer Integrated Surgical Systems and Technology. His current research interests include visual tracking, vision-based control, medical robotics, and human-computer interaction.