Dynamic Guidance with Pseudoadmittance Virtual Fixtures

Zachary Pezzementi, Allison M. Okamura and Gregory D. Hager

***Abstract*— Human Machine Collaborative Systems (HMCS) have been developed to enhance sensation and suppress extraneous motions or forces during surgical tasks requiring precise motion. However, to date such systems have enforced constraints on the position or path of a tool, but have not considered the dynamics of motion. Also, the focus has been on the effect of guidance of motion during a task, rather than on the learning of motion skills through repetition. We present a pseudo-admittance framework for HMCS design to guide the user's velocity in such tasks. Two different fixture design approaches are analyzed, implemented and compared. Three tests are then conducted, showing the fixtures' promise for both guiding and learning motions with dynamics.**

I. INTRODUCTION

Many tasks requiring high-precision motion with specific dynamics are difficult for humans to perform and learn. For example, factors such as physiological tremor, poor visualization and imperceptibly small haptic cues make tasks in domains such as micro-assembly and microsurgery difficult or impossible to perform free-hand. Such tasks have an commensurately steep learning curve, since errors are difficult to detect and correct. We have developed Human Machine Collaborative Systems, to enhance sensation as well as suppress extraneous motions or forces, for training purposes or for guidance in actual procedures. Preliminary studies have shown that humans collaborating with such systems can overcome many of these difficulties [3], [8], improving speed and precision in procedures and lowering training time [6].

One form of cooperative control is provided by virtual fixtures [10]. Fixtures can vary widely from "virtual walls" to "virtual magnets" [11] to impedance planes or admittance contours [5], but they all provide some type of guidance of the position of the implement, whether it is to prevent the tool from entering a sensitive area or to guide it to a particular point or along a contour.

Previous work on virtual fixtures has concentrated on the position of the tool, largely disregarding higher-order motion characteristics. These higher-order characteristics can have a significant effect, though, on a wide variety of tasks. For example, one could not attempt to teach a user to juggle effectively without some sort of guidance of the velocity of the user's hands. Though less extreme, the same dynamics are important in many surgical motions, including the cutting of tissue, lancing, suturing and knot-tying. Here, we propose virtual fixtures to explicitly guide the velocity of the tool. To our knowledge, this is the first reported system to do so.

Furthermore, we propose a system by which an expert user may demonstrate a motion and automatically generate a virtual fixture to provide guidance for another user to carry out that same motion. For instance, an expert surgeon may demonstrate a motion to a student using the haptic device. The student would then practice the motion, getting haptic feedback, until (s)he is able to replicate the motion with good precision. Surgeons have been learning for decades to replicate surgical motions through repeated visual observation. We propose adding the possibility of repeated haptic observation.

Several previous telemanipulation systems have made use of impedance control [4], [5], [10] to implement haptic guidance. Conversely, virtual fixtures have been implemented on the Johns Hopkins "Steady Hand Robot" within an admittance control framework [9]. Impedance-type systems have clear advantages in transparency, and can usually be designed so that they can be overpowered by the user for safety reasons. On the other hand, admittance control provides greater precision and has a safety advantage in that it is naturally passive.

We have developed our guidance methods using a pseudo-admittance framework modeled on that reported in [2]. The primary differences from [2] are the use of a single device instead of a master-slave pair and novel guidance forces being applied to the proxy. This design, which implements an admittance-type controller on an impedance-type device using a proxy, provides many of the advantages of both systems.

In the remainder of this paper, we first establish a more formal definition of the dynamic guidance problem. We then describe a proxy-based implementation framework, and describe both impedance and admittance style guidance in this framework. Finally, we demonstrate the performance of both methods on three different dynamic guidance tasks.

II. PROBLEM STATEMENT

Our fundamental task is to guide a user along a desired trajectory with an associated velocity profile. More formally, we wish to specify, at any point in space and time, a direction, δ_d, in which to guide the user at a corresponding speed, v_d. In order to define this desired trajectory, let T denote a parametric curve in $\Re(3)$ which has a continuous,

This work was supported by the National Science Foundation (NSF).

The authors are all with the Engineering Research Center for Computer Integrated Surgical Systems and Technology (ERC-CISST), Johns Hopkins University, Baltimore, MD 21218, USA {zap,hager}@cs.jhu.edu, aokamura@jhu.edu

non-zero tangent vector:

$$T(s) = [x(s), y(s), z(s)]^T, s \in [0,1], \frac{dT}{ds} \neq 0 \tag{1}$$

Let $v(s)$ denote a scalar speed curve associated with T and let $\delta_p(s)$ denote the tangent to T at s:

$$\delta_p(s) = \left. \frac{\frac{dT}{ds}}{||\frac{dT}{ds}||} \right|_s \tag{2}$$

Then, the desired velocity, $\vec{\boldsymbol{v}}_d$, at s is

$$\vec{\boldsymbol{v}}_d(s) = v(s)\delta_p(s) \tag{3}$$

This specifies the desired direction and velocity as long as the user remains on the curve. But if the tool position deviates from the trajectory, we wish to guide it back toward the curve. To accomplish this, let $\vec{\boldsymbol{x}}_m \in \Re(3)$ be a point in a neighborhood about T and define $c(T, \vec{\boldsymbol{x}}_m)$ to be a mapping from a parameter value s to a to point on the curve T of minimum distance to $\vec{\boldsymbol{x}}_m$. A correction direction, δ_c, is defined in terms of this difference as:

$$\delta_c(\vec{\boldsymbol{x}}_m) = T\left(c(T, \vec{\boldsymbol{x}}_m)\right) - \vec{\boldsymbol{x}}_m \tag{4}$$

The reference direction, δ_d, at $\vec{\boldsymbol{x}}_m$ is a linear combination of the tangential direction and correction direction, parameterized by k_e:

$$\delta_f(\vec{\boldsymbol{x}}_m) = \delta_p(c(T, \vec{\boldsymbol{x}}_m)) + k_e\delta_c(\vec{\boldsymbol{x}}_m) \tag{5}$$

$$\delta_d(\vec{\boldsymbol{x}}_m) = \frac{\delta_f(\vec{\boldsymbol{x}}_m)}{||\delta_f(\vec{\boldsymbol{x}}_m)||} \tag{6}$$

Finally, the desired speed, v_d, at a point $\vec{\boldsymbol{x}}_m$ in a neighborhood about the curve is given by $v_d(\vec{\boldsymbol{x}}_m) = v\left(c(T, \vec{\boldsymbol{x}}_m)\right)$.

For notational simplicity, in what follows, the position argument to δ_d and v_d is dropped with the understanding that it is taken to be the current position of the tool tip.

III. MODEL

A virtual environment is constructed to guide the user along the trajectory, within a pseudoadmittance framework. The physical device (master) does not interact directly with the virtual environment. The virtual environment only acts directly on a virtual version of the tool (the proxy), which is connected to the master through a virtual spring (the virtual coupling).

The motion model of the master and proxy is based on that described in [2], but modified to involve only one device. The proxy can then simulate an admittance controller, while the virtual coupling transduces displacements between the master and proxy into forces on the proxy, which are fed back to the master.

The motion of the proxy is described in terms of the proxy position $\vec{\boldsymbol{x}}_p$ and a global force scaling constant, K_s

$$\dot{\vec{\boldsymbol{x}}}_p = K_s \vec{\boldsymbol{f}}_r \tag{7}$$

The virtual interaction force between the proxy and the master, $\vec{\boldsymbol{f}}_r$, is given by Hooke's Law with K_r as the spring constant and x_m representing the location of the master. This force acts as a simplified version of the virtual coupling presented in [7]. There is also a damping factor, given by b, between the proxy and the environment.

$$\vec{\boldsymbol{f}}_r = K_r\left(\vec{\boldsymbol{x}}_m - \vec{\boldsymbol{x}}_p\right) - b\dot{\vec{\boldsymbol{x}}}_p \tag{8}$$

In the standard pseudoadmittance framework, we define the force of the controller on the master as

$$\vec{\boldsymbol{f}}_{\text{cm}} = K_r\left(\vec{\boldsymbol{x}}_p - \vec{\boldsymbol{x}}_m\right) \tag{9}$$

which is in this case the force of the virtual coupling. The exogenous human input force representing all external input forces is denoted $\vec{\boldsymbol{f}}_h$, with associated inertias defined by $\boldsymbol{M}(\theta)$, where θ is the set of joint angles. The motion of the master is then given by

$$\boldsymbol{M}(\theta)\ddot{\vec{\boldsymbol{x}}}_m = \vec{\boldsymbol{f}}_{\text{cm}} + \vec{\boldsymbol{f}}_h \tag{10}$$

The master is only connected to the proxy through the virtual coupling. In reality, there is a gravitational input force on the master. This force is compensated for separately, and the force pair is neglected in this analysis. Through algebra, we arrive at the proxy and master position equations

$$\vec{\boldsymbol{x}}_m = \vec{\boldsymbol{x}}_p - \frac{1}{K_r}\left(\boldsymbol{M}(\theta)\ddot{\vec{\boldsymbol{x}}}_m - \vec{\boldsymbol{f}}_h\right) \tag{11}$$

$$\vec{\boldsymbol{x}}_p = \vec{\boldsymbol{x}}_m - \left(\frac{1}{K_s} + b\right)\frac{1}{K_r}\dot{\vec{\boldsymbol{x}}}_p \tag{12}$$

Virtual fixtures are then applied to the proxy, and their effects are transferred to the master through the spring interaction. The workspace consists of a vector field of desired tool velocities at any given point in space-time.

IV. FIXTURES

A. Active Fixture

The simplest way to provide guidance along the trajectory omits the pseudoadmittance framework and applies a force in the preferred direction, δ_d, directly to the master to effect a velocity of $\vec{\boldsymbol{v}}_d = v(s)\delta_d$. Specifying a proportional control coefficient, k_{pf}, the dynamics are the same as in Equation (10), but we redefine the force of the controller on the master:

$$\vec{\boldsymbol{f}}_{cm} = k_{\text{pf}}\left(\vec{\boldsymbol{v}}_d - \dot{\vec{\boldsymbol{x}}}_m\right) \tag{13}$$

Converting this fixture into the pseudoadmittance framework affords greater functionality and gives us what we shall refer to as the active fixture. As shown in [2], if a second controllable device is available to hold the actual tool, it can be set up as a slave connected to the proxy through a second virtual spring. Such a setup allows force and velocity scaling as well as the elimination of physiological tremor, in addition to whatever guidance the fixture provides.

As with the previous fixture, at any given point the active fixture applies a force, $\vec{\boldsymbol{f}}_{\text{dir}}$, to the proxy to effect a velocity of $\vec{\boldsymbol{v}}_d = v(s)\delta_d$. This force and the spring force govern the proxy's motion. The pseudoadmittance virtual coupling is in place, so $\vec{\boldsymbol{f}}_{\text{cm}}$ is defined as in Equation (9). Then

$$\vec{\boldsymbol{f}}_{\text{dir}} = k_{\text{pf}}\left(\vec{\boldsymbol{v}}_d - \left(\dot{\vec{\boldsymbol{x}}}_p^T\delta_d\right)\delta_d\right) \tag{14}$$

$$\dot{\vec{\boldsymbol{x}}}_p = k_S\left(K_r\left(\vec{\boldsymbol{x}}_m - \vec{\boldsymbol{x}}_p\right) + \vec{\boldsymbol{f}}_{\text{dir}}\right) \tag{15}$$

The speed guidance portion of the fixture can be omitted by instead choosing $\vec{\boldsymbol{f}}_{\text{dir}} = k_{\text{pf}}\delta_d$, giving the active fixture without speed guidance, which will be useful for evaluating the effectiveness of speed guidance.

The fixtures suggested thus far apply a force, and therefore a motion output, even when there is no human input. The tool may therefore move without contact with the user. This property is often considered undesirable, particularly in safety-critical situations such as surgery. Thus, we now consider the development of a passive fixture.

B. Passive Fixture

In our implementation of the passive fixture, two types of guidance fixtures are applied, one guiding direction of motion and the other speed. The two are composed to form a single fixture which guides velocity.

1) Direction guidance: Our direction guidance fixture is based on the method described in [1]. Force is applied to the proxy through the virtual coupling force, $\vec{\boldsymbol{f}}_{\text{cm}}$, as defined in Equation (9). This force is then considered in terms of the components aligned with the reference direction ($\vec{\boldsymbol{f}}_\delta$) and perpendicular to it ($\vec{\boldsymbol{f}}_\tau$):

$$\vec{\boldsymbol{f}}_\delta = \left(\vec{\boldsymbol{f}}_{\text{cm}}^T\delta_d\right)\delta_d \tag{16}$$

$$\vec{\boldsymbol{f}}_\tau = \vec{\boldsymbol{f}}_{\text{cm}} - \left(\vec{\boldsymbol{f}}_{\text{cm}}^T\delta_d\right)\delta_d \tag{17}$$

An admittance control law is then applied in terms of each component individually, with coefficients k_δ and k_τ:

$$\vec{\boldsymbol{v}}_\delta = k_\delta\vec{\boldsymbol{f}}_\delta \tag{18}$$

$$\vec{\boldsymbol{v}}_\tau = k_\delta k_\tau\vec{\boldsymbol{f}}_\tau \tag{19}$$

where $\vec{\boldsymbol{v}}_\delta$ and $\vec{\boldsymbol{v}}_\tau$ are the components of the resulting proxy velocity in the directions δ_d and τ respectively. By defining $D = \delta_d\delta_d^T$, and k_S as the contribution of the speed controller (described next), we can express the control law as

$$\dot{\vec{\boldsymbol{x}}}_p = k_S k_\delta\left(D + k_\tau\left(I - D\right)\right)\vec{\boldsymbol{f}}_{\text{cm}} \tag{20}$$

As currently presented, motions in the direction $-\delta_d$ receive the same gain as those in the direction δ_d. One further modification is made to attenuate motion opposite the preferred direction, making the guidance unidirectional. When $\delta_d^T\vec{\boldsymbol{f}}$ is non-negative the control law is as above. However, when $\delta_d^T\vec{\boldsymbol{f}} < 0$ the control law becomes just

$$\dot{\vec{\boldsymbol{x}}}_p = k_S k_\delta k_\tau\vec{\boldsymbol{f}}_{\text{cm}} \tag{21}$$

This change reflects the choice that motions opposite the preferred direction should be considered non-preferred and attenuated accordingly. Moreover, they are equally as non-preferred as motions perpendicular to δ_d. Figure 1 illustrates the attenuation of different motions relative to the preferred direction δ_d.

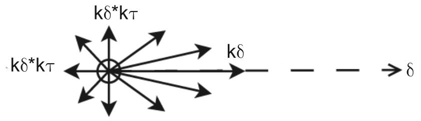

Fig. 1. Resulting velocity magnitude by direction. The preferred direction δ_d is shown along the horizontal to the right. The magnitude is attenuated in the direction opposite δ_d to $k\delta * k\tau$.

The directional controller is multiplied by speed guidance controller output. The value of k_S is scaled by the speed controller independent of directional guidance.

2) Speed guidance: The speed controller is implemented as a proportional controller on the estimated speed of the proxy, $v_p = ||\dot{\vec{\boldsymbol{x}}}_p||$. The controller drives v_p to the desired speed, $v(s)$, or drives the error, $v_e = v(s) - v_p$, to zero.

The guidance is provided in the form of an amplification of the admittance gain. When $v_p < v(s)$, the gain should be large to encourage faster motion, and when $v_p > v(s)$, it should be small to slow the user down.

We convert the velocity error to a gain through exponentiation:

$$k_S = k_{\text{sv}}e^{k_p v_e} \tag{22}$$

Again, a passive controller without speed guidance can be easily obtained by choosing $k_S = k_{\text{sv}}$ instead.

This passive fixture and the active fixture were chosen for implementation because the same pseudoadmittance framework can be used for both fixtures.

V. SIMULATION RESULTS

For clarity, we present a controller analysis for a 1-DOF system. This analysis can be extended to higher degrees of freedom. We consider the device dynamic model to be a pure inertia, m. Forces are considered to be applied directly to the device, ignoring any complex dynamics associated with joints or motor actuation. We also ignore all sampling and discretization effects.

The parameters used in simulation and later in implementation are shown in Table I. The mass of the master was determined by finding the force necessary to achieve steady-state gravity compensation, then dividing by gravitational acceleration. In order to maximize the responsiveness of the system, the spring and damping constants were tuned to be as large as possible without any perceptible oscillation under normal inputs in an implemented system. The magnitude of the step was chosen to be 0.4 N, since it corresponded to

TABLE I

SIMULATION PARAMETERS

Parameter name	Value	Description
v_d	0.00	Desired velocity
k_p	5.00	P-controller coefficient
M	0.06	Master mass
k_r	0.20	Spring constant
b	0.35	Damping constant
k_{sv}	1.00	Velocity scaling constant
k	0.40	Step size

a typical human input and operating velocity for the implemented device. Due to the nonlinearity of the system, these parameters are unlikely to be optimal. They do, however, produce a stable and responsive system.

A. Active Controller Simulation

A state variable system for the active controller is

$$\begin{aligned} p_1 &= x_p \\ p_2 &= x_m \\ p_3 &= \dot{x}_m \\ u_1 &= f_h \\ u_2 &= V_d \end{aligned}$$

with dynamics $q_k = \dot{p}_k$

$$\begin{aligned} q_1 &= -\frac{k_r}{1+b}p_1 + \frac{k_r}{1+b}p_2 + u_2 \\ q_2 &= p_3 \\ q_3 &= \frac{k_r}{m}p_1 - \frac{k_r}{m}p_2 + \frac{1}{m}u_1 \\ y &= p_3 \end{aligned}$$

Using the traditional notation, $\vec{\boldsymbol{q}} = \boldsymbol{A}\vec{\boldsymbol{p}} + \boldsymbol{B}\vec{\boldsymbol{u}}$, and letting $\alpha = s^3 + \frac{k_r}{1+b}s^2 + \frac{k_r}{m}s$, the system has characteristic equation, $\phi\left(s\right) = \left(sI - A\right)^{-1}$, given by

$$\phi\left(s\right) = \begin{vmatrix} \frac{s^2 + \frac{k_r}{m}}{\alpha} & \frac{s\frac{k_r}{1+b}}{\alpha} & \frac{\frac{k_r}{1+b}}{\alpha} \\ \frac{\frac{k_r}{m}}{\alpha} & \frac{s^2 + s\frac{k_r}{1+b}}{\alpha} & \frac{s + \frac{k_r}{1+b}}{\alpha} \\ \frac{s\frac{k_r}{m}}{\alpha} & -\frac{s\frac{k_r}{m}}{\alpha} & \frac{s^2 + s\frac{k_r}{1+b}}{\alpha} \end{vmatrix}$$

The output is then $\vec{\boldsymbol{y}} = \boldsymbol{C}\vec{\boldsymbol{p}} + \boldsymbol{D}\vec{\boldsymbol{u}}$, with $\vec{\boldsymbol{y}} = \dot{\vec{\boldsymbol{x}}}_m$,

$$C = \begin{vmatrix} \frac{\frac{k_r}{m}s}{\alpha} & -\frac{\frac{k_r}{m}s}{\alpha} & \frac{s^2 + \frac{k_r}{1+b}s}{\alpha} \end{vmatrix}$$

and

$$D = \begin{vmatrix} 0 & 1 \\ 0 & 0 \\ \frac{1}{m} & 0 \end{vmatrix}$$

so for input f_h, the transfer function is $G(s) = \frac{s\left(s + \frac{k_r}{1+b}\right)}{s\left(s^2 + \frac{k_r}{1+b}s + \frac{k_r}{M}\right)}$. The response to a step input is shown in Figure 2.

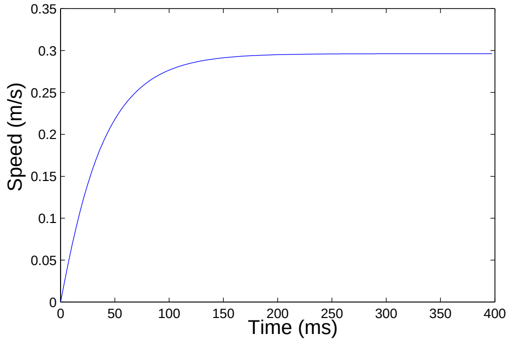

Fig. 2. Simulated active fixture master response to a step input of magnitude 0.4

B. Passive Controller Simulation

Since the exponentiation and product make the passive system strongly non-linear, we investigated its response through simulation rather than analytically. The response of the passive controller to a step function using the parameters in Table I is shown in Figure 3. These parameters were chosen by manually adjusting the implemented system to be both stable and responsive under typical force inputs.

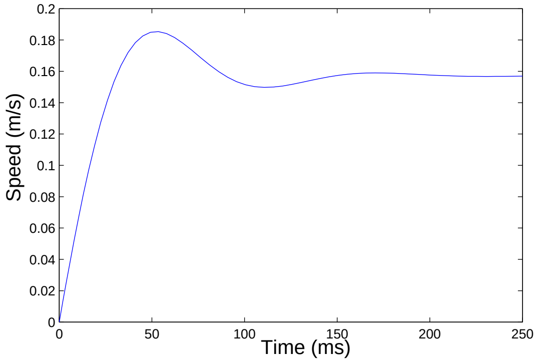

Fig. 3. Passive controller master response to step input of magnitude 0.4

The passive controller's response appears underdamped. The key result is that the passive controller is stable under a step input despite its nonlinearities.

The biggest qualitative difference between the active and passive controllers occurs when the user lets go of the master. In the zero input response of the passive controller, not shown but verified in simulation and implementation, when f_h goes to zero, so does the master and proxy velocity. The master therefore stays still when the user lets go. For the active controller, however, when the user lets go, the proxy continues to feel a force and drags the master behind it at some speed less than v_d in the preferred direction.

VI. IMPLEMENTATION RESULTS

We have implemented the pseudo-admittance framework and the two controllers on a PHANToM Omni [13], using

TABLE II

GRAPHICS AND HAPTICS SPEED MATCHING TEST RESULTS

Fixture	Speed Guidance?	Mean	Std.Dev
Freehand	no	13.23	3.08
Passive	yes	7.17	2.18
	no	11.06	4.12
Active	yes	6.25	1.51
	no	7.36	2.33

the Open Haptics Academic Edition 2.0 software library in C++ [12]. The system behaves stably with all controllers.

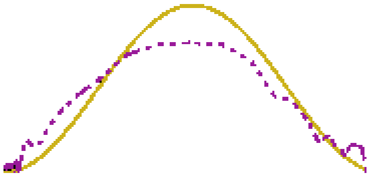

Fig. 4. A screen capture of a user following a velocity profile from left to right with the assistance of the passive fixture. The solid yellow sinusoid is the desired velocity profile $v_d(x)$. The dashed purple path is the user's motion $v_u(x)$ while trying to replicate the desired velocity.

Though a formal study has not yet been done, preliminary tests indicate that the fixtures allow the user to follow a velocity profile more easily than freehand.

A. Graphics and Haptics Speed Matching Test

In one test, the user was presented with a visual representation of a velocity profile and asked to follow that profile either freehand or with the assistance of the passive or active virtual fixture. Figure 4 illustrates the task. The graph shows the 1-D velocity profile of a user overlaid with a sinusoidal target trajectory. The abscissa shows position x from x_0 to x_f and the ordinate shows velocity at that point, for both the user, $v_u(x)$, and the target trajectory, $v_d(x)$. The figure is a screen capture of an actual trial, so it shows the same visual feedback the user received. An error metric was defined as

$$Err = \int_{x_0}^{x_f} |v_d(x) - v_u(x)| dx \qquad (23)$$

and evaluated in 10 consecutive trials each of freehand, passive fixture guidance, and active fixture guidance, with or without speed guidance from each fixture. Table II shows the results of each set of trials. Additionally, in both of the fixture guidance series with speed guidance, the error can be seen to decrease over the course of the series. Such a trend is not evident in the trials without speed guidance.

B. Haptics-Only Speed Matching Test

In another test, a user was instructed to move along a trajectory with an unknown velocity profile and only haptic feedback from the passive fixture. This test evaluated a user's ability to discover an initially unknown velocity profile within a single trial while exploring with the passive fixture. The active fixture was not analyzed as well since this exploration could be accomplished by simply observing the zero-input response. The curve traced by the trajectory was always a circle, but the speed was chosen randomly from four possible functions, two constant speeds and two sinusoidal patterns. Figure 5 shows the user's speed in two such trials overlaid with the sinusoidal desired speed. The user's speed converges with the desired speed over time.

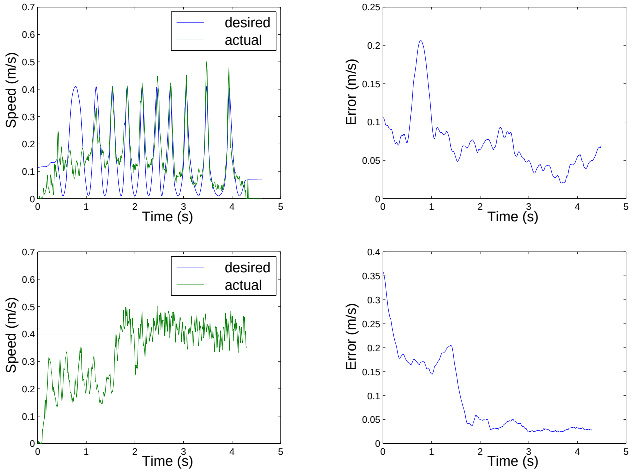

Fig. 5. A user with passive fixture guidance attempting to follow an unknown velocity profile, either a sinusoid (top) or a constant speed (bottom). The user's speed is shown overlaid with the desired speed (left), and the corresponding error (right) can be seen to decrease over time.

As a result of two effects, the desired speed does not appear quite sinusoidal. First, the guidance speed, $v(\theta) = 0.21 + 0.2cos(\theta)$, is defined in terms of position, not time. Also, the desired speed shown is the speed to which the controller is attempting to guide the user at that time, so it is a function of the user's position.

C. Haptics-Only Trajectory Matching Test

In a third test, an "expert" demonstrated a motion using the haptic device and the user attempted to reproduce the same motion with only haptic feedback. During the motion demonstration phase, motion was recorded with no feedback provided. The user then attempted to follow the same trajectory either freehand or with the assistance of the passive or active virtual fixture, each with or without speed guidance. Guidance was provided by the fixtures in the following way: The demonstrated motion was sampled at approximately 100 Hz to generate a discretized version of $T(s)$. The guidance velocity was generated by searching for the closest tool position in the series, $\vec{x}_g(s)$, by Euclidean distance. Matching was restricted to within a sliding window over the range of s, preserving some continuity of matching and allowing the trajectory to be self-intersecting. It should be noted, however, that this method does not handle well motions in which the instantaneous velocity goes to zero, since the guidance velocity then goes to zero. Because the time for the reproduced motion does not necessarily progress at a constant

TABLE III

HAPTICS-ONLY TRAJECTORY MATCHING TEST RESULTS

Fixture	Speed Guidance?	Err_p	Err_s
Freehand	no	73.6	152.9
Passive	yes	10.1	83.1
	no	10.1	185.3
Active	yes	29.21	100.6
	no	9.8	161.5

rate with respect to the demonstrated trajectory, there is no way to distinguish between a momentary pause and a long stop. Additionally, since the guidance provided depends on both sampling of the original demonstrate motion and the results of this matching process, the provided guidance is not always smooth, particularly when multiple matches within the trajectory are plausible.

Some sample results are shown in Figure 6. The 3D Euclidean coordinates of the tool for both the demonstrated and user-replicated trajectories are shown, where the demonstrated motion is a spiral about the x-axis. Alongside them is the user's velocity profile overlaid with the speed the fixture was attempting to guide the user toward at that moment. Since the user does not replicate the motion exactly, due to the matching process described above, the desired speed profile is not the same for every trial. To analyze the precision of the user's motion with respect to the reference trajectory, R, an error metric was defined in terms of the matching process. Let $c\left(T, \vec{\boldsymbol{x}}\right)$ be once again the value of parameter s corresponding to the closest point to $\vec{\boldsymbol{x}}$ on trajectory T. Also, let l denote the length in time of the user's trajectory. Then position error, Err_p, was defined

$$Err_p = \frac{1}{l} \int_R \left(R\left(c\left(R, \vec{\boldsymbol{x}}_m \right) \right) - \vec{\boldsymbol{x}}_m \right) ds \qquad (24)$$

The position error is normalized by l to leave it independent of the speed of the user. The speed error, Err_s, is defined

$$Err_s = \int_R \left(v\left(c\left(R, \vec{\boldsymbol{x}}_m \right) \right) - \dot{\vec{\boldsymbol{x}}}_m \right) ds \qquad (25)$$

For the trajectories shown in Figure 6, results are shown in Table III. Both fixtures allowed the user to follow the demonstrated motion much more accurately than freehand, with the passive fixture outperforming the active fixture in position guidance and similar performance in speed guidance. Speed guidance produced much lower values of Err_s.

VII. CONCLUSION

Thus far, we have presented a set of virtual fixtures for velocity guidance and shown them to be stable under standard inputs. The next step is to apply them to a real-world task with multiple users and evaluate their usefulness. The fixtures show promise over freehand in experiential trials, both for guidance of speed and direction, but it is not yet clear which of the fixtures offers the best guidance. The main factor in a choice between an active or passive fixture will likely be the zero-input response. Whether a zero or non-zero zero-input response is preferable may depend on the application, but the passive behavior seems to be what is currently expected by the user of a haptic system.

In the future, we intend to conduct human factors experiments to determine how each fixture affects performance on real-world tasks and how users respond to it.

VIII. ACKNOWLEDGMENTS

This work was funded by the National Science Foundation (NSF) under Grant IIS-0205318. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] A. Bettini, S. Lang, A. Okamura, G. Hager. Vision assisted control for manipulation using virtual fixtures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1171–1176, November 2001.

[2] J. J. Abbott. *Virtual Fixtures for Bilateral Telemanipulation*. PhD thesis, Department of Mechanical Engineering, The Johns Hopkins University, August 2005.

[3] J. T. Feddema and R. W. Simon. Visual servoing and cad-driven microassembly. *IEEE Robotics & Automation Mag.*, pages 18–24, December 1998.

[4] K. Kosuge, K. Takeo, T. Fukuda. Unified approach for teleoperation of virtual and real environment-manipulation based on reference dynamics. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 938–943, 1995.

[5] Fuji Lai and Robert D. Howe. Virtual fixtures for robotic endoscopic surgery. Technical report, Biorobotic Lab, Harvard University, 2000.

[6] S. Payandeh and Z. Stanisic. On application of virtual fixtures as an aid for telemanipulation and training. 10th Int'l. Symp. on Haptic Interfaces for Virtual Env. and Teleop. Sys., pp. 18-23, 2002.

[7] R. J. Adamas, B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, January 1999.

[8] R. Kumar, T. Goradia, A. Barnes, P. Jensen, L. Whitcomb, D. Stoianovici, L. Auer, R. Taylor. Performance of robotic augmentation in common dextrous surgical motions. In *MICCAI*, pages 1108–1115, 1999.

[9] R. Taylor, P. Jensen, L. Whitcomb, et. al. Steady hand robotic system for microsurgical augmentation. *Int. J. Robot Res.*, 18(12):1201–1210, 1999.

[10] L. Rosenberg. *Virtual Fixtures*. PhD thesis, Department of Mechanical Engineering, Stanford University, August 1994.

[11] C. A. Sayers. *Remote Control Robotics*. Springer-Verlag, 1998.

[12] SensAble Technologies. *OpenHaptics Toolkit version 2.0 Programmer's Guide*, 2005. Available at http://www.sensable.com/products/phantom_ghost/OpenHaptics_ProgGuide.pdf.

[13] SensAble Technologies. *Specifications for the PHANTOM Omni(TM) haptic device*, 2005. Available at http://www.sensable.com/products/datafiles/phantom_ghost/PHANTOM Omni Specifications1.pdf.

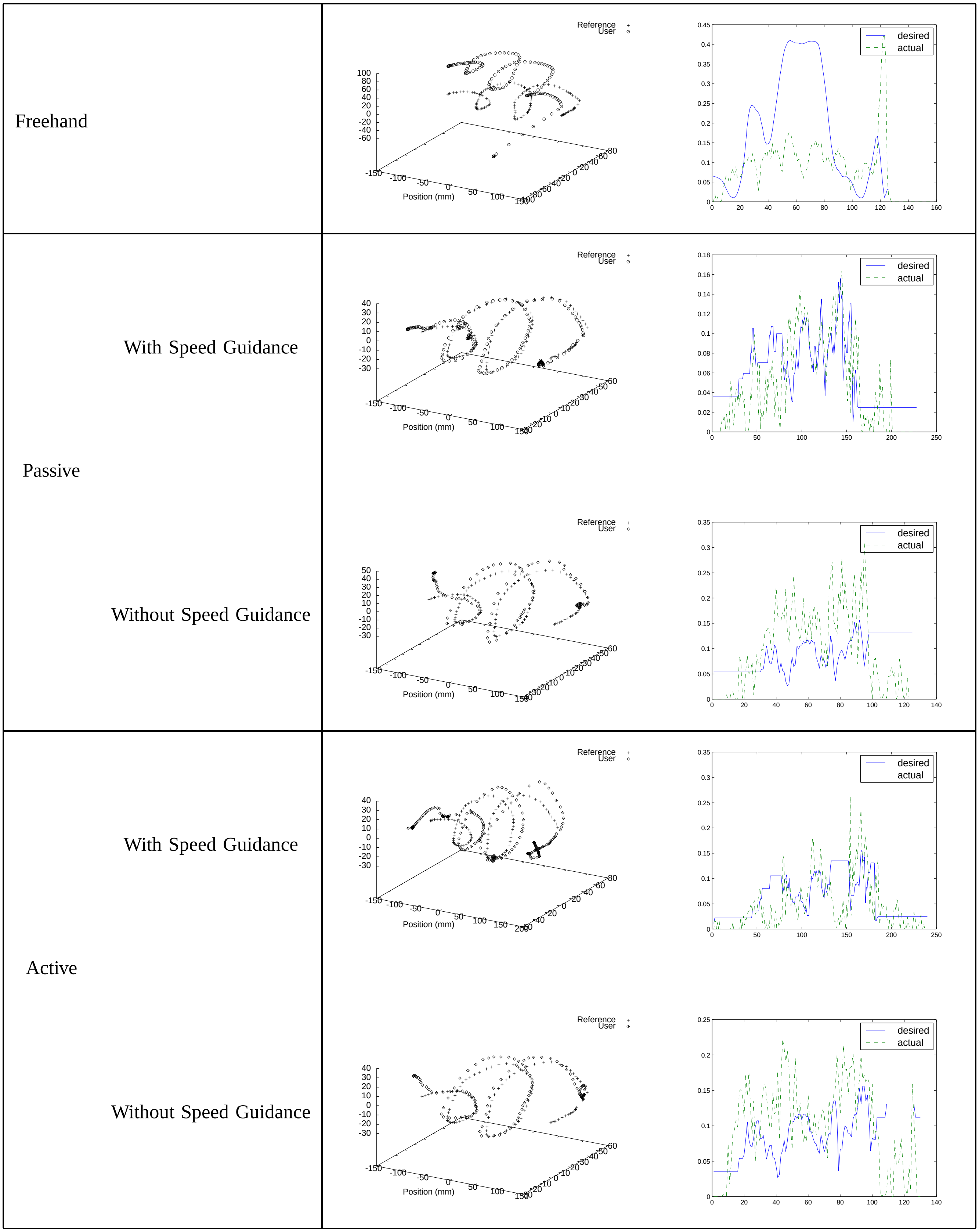

Fig. 6. A user attempting to follow a demonstrated motion in the haptics-only trajectory matching test. The user is working either freehand or with the assistance of a fixture, and the user's position (left) and speed (right) are displayed overlaid with the target trajectory. For each fixture, the results with and without speed guidance are shown. These data show the effectiveness of virtual fixtures with speed guidance at encouraging the user to maintain a desired trajectory.