2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Time-delayed teleoperation for interaction with moving objects in space

Ryder C. Winck, Sean M. Sketch, Elliot W. Hawkes, David L. Christensen,
Hao Jiang, Mark R. Cutkosky, and Allison M. Okamura

***Abstract*—Telerobotics has the potential to facilitate the repair of satellites in geosynchronous orbit by allowing human operators to interact naturally with remote objects. Time delays on the order of seconds make it difficult to provide immersive feedback to the operator, motivating the use of predictive visual and haptic displays of the robot and environment. A teleoperation framework developed for this scenario invokes a two-part environment model that predicts motion of objects in the environment, both in free space and during contact with the robot. When objects in the environment are in free space, a propagated model using delayed data provides predictive feedback to the operator. However, when the robot interacts with the environment, a local environment model that does not propagate delayed data is used. This reduces computational load and ensures stability during robot-environment interactions. Two experiments were carried out to test the teleoperation system. Results demonstrate the ability of the prediction algorithm to provide reliable feedback and improve operator performance before, during, and after robot-environment interactions.**

I. Introduction

THIS paper is motivated by the application of robotic servicing of satellites in geosynchronous orbit, as defined by the USA's Defense Advanced Research Project Agency (DARPA) project known as "Phoenix" (http://darpa.mil/Our_Work/TTO/Programs/Phoenix.aspx). We propose that teleoperation of the remote robot provides the flexibility of human-in-the-loop control needed for many of the tasks required for satellite servicing, such as installing equipment and reacting to unexpected circumstances (e.g. retrieving floating parts or tools). We present a framework for ground-based teleoperation of a remote robot. To compensate for communication delays, virtual robot and environment models are displayed to the operator. These models predict the motion of the remote robot and environment so that the operator perceives no delay. For the Phoenix project, remote manipulation will be accomplished using the FREND arm [1], shown in Fig. 1 along with a teleoperator interface developed by our team. However, the work presented in this paper uses a laboratory test-bed.

This work was supported in part by DARPA award #D13PC00008, Jet Propulsion Laboratory RSA 1473616, the National Science Foundation Graduate Research Fellowship Program, National Science Foundation grant #1227406, and Stanford University. Any opinions, findings, and conclusions or recommendations in this material are those of the authors and do not necessarily reflect the views of the sponsors.

R.C. Winck, S. Sketch, E.W. Hawkes, D.L. Christenson, H. Jiang, M.R. Cutkosky, and A.M. Okamura are with the Dept. of Mech. Eng., Stanford University, Stanford, CA 94305, e-mail: ryder@stanford.edu.

Previous work on robotic satellite servicing has demonstrated teleoperation of robots in low Earth orbit [2]-[4]. In early experiments, the communication delays considered were on the order of seconds [2], [3]. Since then, successful teleoperation has been demonstrated in low Earth orbit, but using communication through geosynchronous orbit with communication delays as low as half a second [4]. However, we expect factors such as end-to-end data-link processing to add to the half-second communication delay due to the distance from Earth to geosynchronous orbit. This will likely result in round-trip time delays of two or more seconds, similar to previous work teleoperating in low Earth orbit.

Early work on space teleoperation expressed the need for predictive displays [4] to overcome communication delays exceeding 1 second. Such displays have been used to provide visual feedback of robot [6] and environment motion [2], as well as force feedback [3]. While free space prediction of a moving object and model-mediated haptic feedback have been explored separately, we present a unified teleoperation framework that both predicts environment motion and permits bilateral interactions with the environment. This framework combines predictive robot and environment models with a model-mediated approach to haptic interaction. Our approach to environment modeling expands on the concept of using two environment models put forth by Brady and Tarn [7].

To validate this combined approach, we present experimental results of catching and pushing a free-floating object under communication time delay. The experiments demonstrate the need and effectiveness of the predictive, model-mediated approach. We also present the application of a novel robotic gripper based on directional adhesion that is robust to orientation errors [8]. The gripper is shown to be beneficial in catching a free-floating object because it compensates for orientation constraints of the robotic hardware.

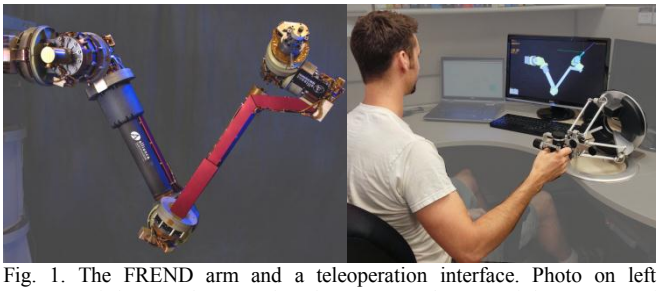

Fig. 1. The FREND arm and a teleoperation interface. Photo on left courtesy of the Naval Research Lab, © 2009; photo on right courtesy of Millennium Engineering and Integration Company.

II. Background

The teleoperation scenario relevant to our work consists of a human operator on Earth using a master manipulator to control a robot in space. The remote robot acts on delayed commands from Earth and transmits sensor data, which is further delayed, back to Earth. The delayed data is received by the operator through visual and haptic feedback. These time delays can reduce the operator's sense of immersion, resulting in the adoption of a "move-and-wait" strategy, and lead to instability when haptic feedback is present [4].

For delays on the order of seconds, predictive robot models have been used to remove visual delays during operation of a robot in a static environment [6]. In these cases, graphical, geometric models of static environments were used to present the environment with the effects of delay removed [2], [3], [9]-[12]. In moving environments, predictive models that propagate delayed sensor data forward in time were shown to be necessary for controlled interaction with the environment [2], [12].

If the remote robot is to physically interact with its environment, then dynamic predictions of contact forces are needed in addition to prediction of environment geometry and motion. A contact model improves the prediction of robot interactions with its environment, and can be used for model-mediated haptic feedback [11], [13], [14].

While a number of researchers have investigated model-mediated haptic feedback for static environments [3], [9], [10], [13], [14], less work has been done to integrate dynamic motion prediction with bilateral force feedback. Kikuchi [12] used haptic feedback with dynamic motion prediction to compensate for delays of up to 0.5 seconds. However, he was able to use scattering methods to provide stable haptic feedback instead of a model-mediated approach due to the relatively small delay. Brady and Tarn [7] suggested a high-level framework for integrating model-based bilateral teleoperation with predictive motion models. They suggested combining (via a weighted sum) a propagated model based on delayed sensor data and a local dynamic model including contact mechanics. In this paper, the two models are called the propagated model and local model. We significantly extend the ideas presented in [6], providing an implementation and experimental validation.

In our implementation, visual and haptic feedback of the predicted virtual robot and environment are supplied to the operator based on dynamic and geometric models. Unlike previous work, we combine a nonlinear robot model, a nonlinear environment model, and model-mediated haptic feedback. This framework enables operators controlling a remote robot under large delays to physically interact with moving environments.

III. Teleoperation Framework

The main components of our teleoperation framework are shown in Fig. 2. The components on Earth communicate with each other with very small time delays between them (order of milliseconds). The communication delays between the Earth and space components are on the order of seconds. For the implementation described in Section III.C, the

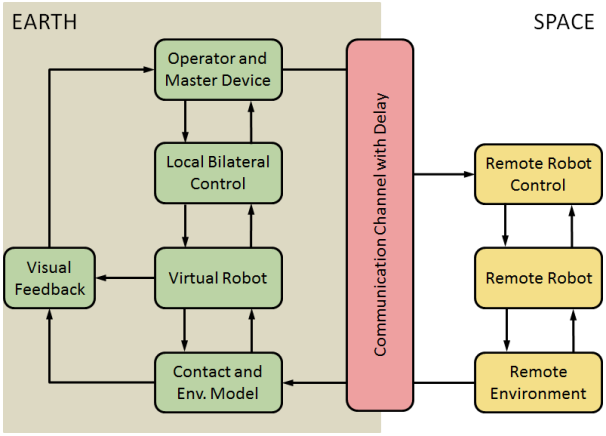

Fig. 2. Teleoperation framework. The components on Earth are in green. The components in space are in yellow. The communication channel adds time delay on the order of seconds.

signals sent over the communication channel are the master states and the states of the remote environment. In future implementation other information will be sent, such as the state of the remote robot. The components of the framework that are of primary interest in this paper are the virtual environment model (Contact and Env. Model block) and dynamic robot model (Virtual Robot block), discussed in the following two subsections.

A. Virtual Environment Model

The virtual environment model is used to provide visual feedback to the operator and to compute contact forces between the environment and robot. It makes use of two models: a propagated model used during free space motion that uses delayed sensor data from the remote environment; and a local model used during environment and robot interaction that includes contact mechanics. The virtual environment switches between the two models as described below and in Fig. 3. Both models provide visual feedback, but only the local model provides haptic feedback. For working with satellites in space, it is assumed that a priori geometric models are available, but if not, these properties could be estimated online [15]. Therefore, the state of the environment completely defines the pose of the geometric models in the visual display.

The need for two models is due in part to the challenge of computing contact points and forces fast enough to use them within the propagated model. The propagated model must simulate the dynamics of the environment over the entire round-trip delay during each time step of the local bilateral controller, which must run at a high rate to provide high-fidelity haptic feedback.

Delayed data from the remote environment comes from vision sensing and is used to estimate the state of the environment. An arbitrary object floating in space can be modeled using the general Euler equations of motion with the angles expressed as quaternions. The resulting dynamic model is nonlinear. The states from vision data can be estimated using an extended or unscented Kalman filter [2], [15]. The delayed environment state is used as an initial

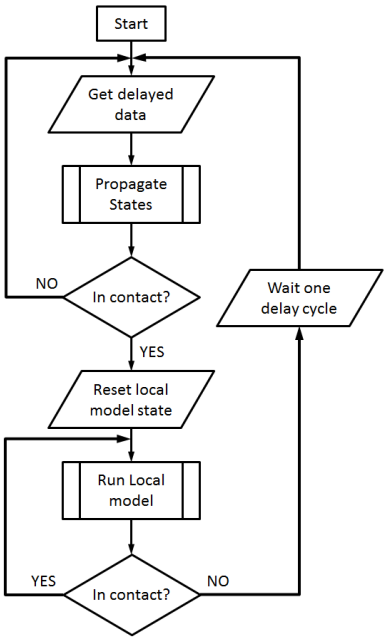

Fig. 3. Flowchart of the environment model. At the start, when the environment is in free space, the propagated model is used as the feedback environment model. When contact is made, the local model states are reset to the current state of the propagated model, and the environment model is switched to the local model. When contact is removed, the environment model is switched back to the propagated model after one round-trip delay cycle is complete.

condition for the propagated model.

The propagated model then computes ahead in time what the state of the environment will be after the round-trip delay. This propagated state is visually displayed to the operator. In addition, the propagated state is used to detect initial contact between the environment and robot. When contact with the robot is detected, the algorithm switches to the purely local model that does not use delayed sensor data but does include a contact mechanics model.

The local model initializes to the state of the propagated model at the instant that contact is detected, thereby ensuring a smooth transition. Although it uses the same nonlinear dynamic model as the propagated model, the local model is computed only once per time step because there is no need to propagate delayed data. The local model also computes contact forces based on compliance properties of the environment and robot. The computed forces are used to predict the environment and robot motion and to provide haptic feedback to the operator. Because the local model is based only on local information, the communication delay does not affect the haptic feedback. Thus the stability of the bilateral control is based only on the local model, virtual robot, and human.

When contact between the robot and the environment is removed, there is not an immediate switch back to the propagated model. Because the propagated model does not include contact mechanics, it diverges during contact. However, after one round-trip delay, when new data is received from the remote environment, it will converge back to the correct free space prediction. Therefore, the algorithm switches back to the propagated model only after contact is removed and one round-trip delay has passed. This transition is not necessarily a smooth one, like the transition from the propagated model to the local model. However, the smoothness of this transition is less important because the environment is no longer in contact with the robot; it is more important that the operator receives the most accurate feedback of the remote environment as soon as possible.

B. Virtual Robot Model

The virtual robot is controlled directly by the operator via a master manipulator. It is not propagated from delayed data like the propagated environment model. Because the master commands are sent to both the virtual and remote robot controllers, the combined virtual robot and controller must reflect the response of the real robot and its controller. Assuming that both the virtual and remote robots are able track the master without steady-state error, the state of the virtual robot will converge to that of the remote robot.

C. Implementation

As shown in Fig. 4, the laboratory implementation of the teleoperation system consisted of a master manipulator, remote robot with gecko-inspired gripper, overhead vision system, and graphical operator interface. The remote environment was an air bearing-mounted "pod." The functionality of each component is described below.

MATLAB Simulink code controlled the system as a whole at 500 Hz. Data from the vision system was obtained at approximately 30 Hz, and the graphics were updated at 60 Hz. The environment and robot models were implemented as MATLAB function blocks. Between these components were the necessary coordinate-frame transformations, control laws, and time delays. The top-level Simulink blocks follow

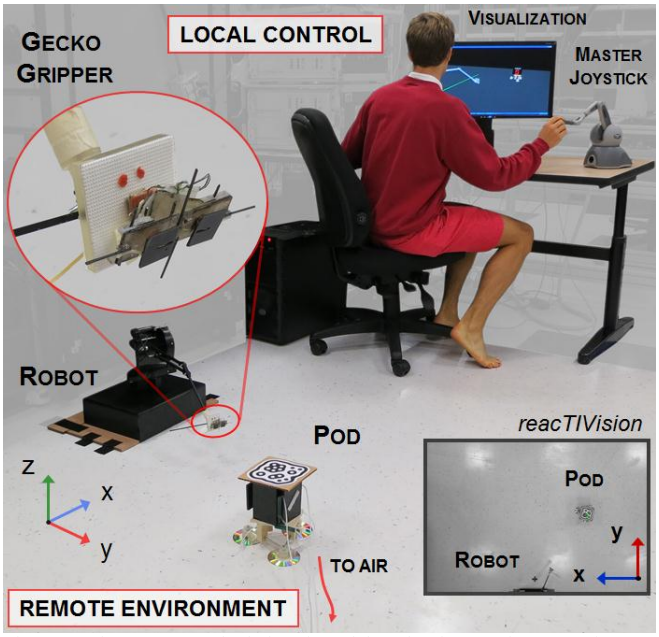

Fig. 4. Laboratory test-bed for time-delayed teleoperation experiments. During testing, the operator faced away from the remote environment, seeing only the on-screen visualization. The camera used for visualization (not shown) was mounted on the ceiling. An image from the camera, taken using reacTIVision, is shown in the inset.

the diagram in Fig. 2.

1) Operator Interface – A PHANTOM Omni was the master input device and provided haptic feedback. Virtual fixtures were applied immediately in front of the base of the device and to either side prevent the master from commanding the robot to a singular position. The Omni was also constrained to a plane at the height necessary for the gripper mounted on the robot to contact the pod.

The Chai3D (http://chai3d.org) library was used to provide visual feedback of the state of the pod and robot to the operator based on a priori CAD models. When using prediction feedback, only the predicted state was displayed. The operator could adjust the view angle and zoom.

2) Local Robot and Control – A Sensable PHANTOM Premium 1.5A was used as the remote robot. The dynamic model of the Premium comes from Abbott [16]. This model is expressed in joint coordinates and PID control was used for each joint. The control gains were tuned to match the performance of the actual PHANTOM Premium used as the remote robot. The control gains for each joint, using the joint label convention from [16], were: $K_{p1} = 0.55$, $K_{I1} = 0$, $K_{D1} = 0.04$; $K_{p2} = 0.4$, $K_{I2} = 0.05$, $K_{D2} = 0.025$; and $K_{p3} = 0.5$, $K_{I3} = 0.01$, $K_{D3} = 0.01$. In addition to the PID control gains, a static torque offset of -0.1 N·m was added to joint 2 and an open-loop gain of 0.83 was added to the desired angle of joint 3. The controller was not designed for performance, but to match the behavior of the remote robot. The contact forces generated by the local environment model were applied to the robot model as joint torques.

3) Virtual Environment – For the experiments discussed in Section IV, air bearings constrained the pod to a horizontal plane, reducing its degrees of freedom to two in translation and one in rotation. The nonlinear dynamic model is

$$\begin{bmatrix}\dot{x}_x(t)\\ \dot{x}_y(t)\end{bmatrix} = \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}^T \begin{bmatrix}v_{x'}(t)\\ v_{y'}(t)\end{bmatrix}$$

$$\dot{\theta}_z\,(t) = \omega_z(t)$$

$$\begin{bmatrix}\dot{v}_{x'}(t)\\ \dot{v}_{y'}(t)\end{bmatrix} = \begin{bmatrix}(F_{x'}/m)\\ (F_{y'}/m)\end{bmatrix} + \begin{bmatrix} v_{y'}(t)\omega_z(t)\\ -v_{x'}(t)\omega_z(t)\end{bmatrix} - \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}\begin{bmatrix}a_x\\ a_y\end{bmatrix}, \text{ and} \quad (1)$$

$$\dot{\omega}_z(t) = M_z/I_{zz}\,.$$

In (1), the subscripts $x'y'z'$ represent a body-fixed coordinate frame and xyz represent a world frame. Unlike previous work [2], [7], we did not linearize the model, but used the nonlinear model for prediction. The pod's mass was $m = 1.3$ kg with an $I_{zz} = 0.0026$ kg·m^2 moment of inertia about its vertical axis. When released from rest, an uneven floor caused the pod to accelerate. The acceleration was modeled as $a_x = -0.035$ cm/s^2 and $a_y = 0.06$ cm/s^2 in (1).

Ralston's Second Order Runge-Kutta method was used to propagate the model. For a round-trip time delay of T_d and a step size of T_s, T_d/T_s iterations of the model were solved in one sample time of the local bilateral controller.

When using camera data to determine the state of the pod, the camera's update rate may be lower than the sampling rate of the local bilateral controller. For this experiment, vision data was processed at 30 Hz, while the controller sampling rate was 500 Hz. Therefore, in addition to propagating state of the environment to compensate for the communication delay, the propagated model also predicted the state of the environment in between the packets of vision data. To do this, the propagated model recorded both the round-trip delay prediction and a one-sample-ahead prediction. In the next sample time, if no new data had been received from the vision system, then the one-sample-ahead prediction from the previous sample-time was used as the initial value for the propagated model.

For the local model, the nonlinear dynamic model in (1) was implemented using the Bogacki-Shampine method for the numerical integration. Multi-point contact mechanics were implemented using the Bullet physics (http://bulletphysics.org) library. The force generated between the environment and robot was applied to the virtual pod and robot, as well as haptic feedback to the operator using the Omni.

4) Remote Robot and Controller – The PHANTOM Premium 1.5A was controlled by the master manipulator via position control in world coordinates. The PID control gains were: $K_{px} = 10$, $K_{Ix} = 0$, $K_{Dx} = 1$; $K_{py} = 40$, $K_{Iy} = 0$, $K_{Dy} = 2$; and $K_{pz} = 50$, $K_{Iz} = 10$, $K_{Dz} = 2$.

A collapsing truss grasper [8] (a.k.a. Gecko Gripper), shown in Fig. 4, was attached to a 3D-printed end effector on the Premium for the catching experiment described in Section IV. The Premium's end effector was not actuated, so independent control of the gripper's orientation was not possible. However, the end effector was used to roughly orient the gripper relative to the face of the pod and enable catching. The gripper compensated for minor misalignment with the pod face by using outriggers for dynamic passive alignment. The Gecko Gripper employs directional dry-adhesive pads. Such adhesives adhere to smooth surfaces when a shear load is applied parallel to their surface. In this way, they can be "turned on" by applying shear, and "off" by removing the load. The same pads were used for all the tests. The gripper's mechanism is designed to apply such a shear load to a pair of opposing adhesive pads as the truss collapses when the gripper contacts a surface. Once the truss is fully collapsed, the mechanism magnetically latches and locks the shear loads on the adhesives. Forces and torques exerted on the gripper during rebound are absorbed by a nonlinear spring attached to the back of the truss via a boom arm, increasing the gripper's ability to absorb kinetic energy.

For the contact experiment, a thin piece of foam replaced the Gecko Gripper on the Premium end effector. This allowed the robot to push the pod without gripping it and provided compliance to allow for misalignment due to the lack of control of the gripper's orientation.

5) Remote Environment – The pod was constructed with a lightweight wood base that attached to three 3D-printed air

bearings that received air from a wall supply via a tether, based on the design by Howard et al. [17]. To reduce friction, the test-bed floor was waxed, and Teflon-coated mini-DVDs were affixed to the bottom of each bearing. A rectangular steel tube was mounted on top of the wood base, and acrylic plates were attached to each side to provide a smooth catching surface for the Gecko Gripper.

An overhead web camera and reacTIVision (http://reactivision.sourceforge.net), an open-source two-dimensional fiducial tracker, were used to obtain the states of the pod at approximately 30 Hz. The states were scaled based on calibration fiducials, and the velocities were estimated using a finite difference and a first-order Butterworth low-pass filter with a 0.75 Hz cutoff frequency. The cutoff frequency was low because the acceleration of the pod moving in free space was low. An image from reacTIVision is shown in the inset in Fig. 4.

IV. Experiment

Two experiments were carried out to test the teleoperation framework. The first, a catching experiment, tested the free space prediction in a manner similar to previous studies [2], [12]. An operator attempted to catch a floating pod under time delays of up to 2 seconds. Performance during the use of predictive robot and environment models was compared to that with using only a predictive robot model and with using no prediction. The second, a contact test, involved the operator pushing the pod under time delay. For some trials the pod was initially at rest when the operator made contact, and for others, the pod was moving towards the robot; the operator had to stop the pod and push it away in a controlled fashion. This experiment tested the propagated and local model and the ability to switch between them. It also demonstrated haptic feedback provided by the contact mechanics of the local model.

A. Procedure

The catching experiment was carried out for time delays of 0, 0.5, 1, and 2 seconds with three different types of visual feedback: delayed feedback of the robot and environment; predictive feedback of the robot only; and predictive feedback of both the robot and the environment. For each combination of delay and feedback, five trials were conducted. A trial consisted of the pod being pushed towards the robot from an angle of 50-55 degrees from the x axis and then the operator attempting to catch the pod. The average velocity of the pod for all the trials was approximately 44 cm/sec, varying from 35 cm/sec to 50 cm/sec. The minimum velocity was constrained by the need to push the pod fast enough to reduce the effect of friction between the air bearings and the floor. Two seconds of delay was the longest delay used due to the high pod velocity and the space constraints, but it has been shown that increasing velocity has a similar effect on performance as increasing delay when tracking a constant-velocity object [12].

For each trial, the same subject with significant teleoperation experience was the operator and the same individual pushed the pod. The result was a catch, a hit, or a miss. A catch was considered a successful trial and a miss an unsuccessful trial. A hit occurred when the gripper made contact with the pod but did not catch the pod. This occurred during testing for a number of reasons, such as over-rotation of the pod, such as could not be compensated for by the gripper's dynamic passive alignment, or an improper contact caused by the operator. As the speed of the tests made it difficult to ascertain the cause of the hit without catching, these hits were noted and the trials were repeated until either a catch or miss occurred. Data was also recorded to evaluate the prediction accuracy of the robot and environment models. For the catching experiment, the local model with contact mechanics was not implemented and only the propagated environment model was used.

During the contact test, the operator pushed the pod with the end effector. With both the propagated and local environment models implemented, the teleoperation system predicted the pod's motion during contact and free space. The pod was first pushed from rest in the $+y$ direction and allowed to float freely. Then the pod was given an initial velocity directly toward the robot $(-y)$ and the operator would contact the pod, stop it, and push it back. Finally, the side of the gripper was used to push the pod in the $-x$ direction. The contact experiment was done with 0.5 seconds of delay to reduce the time after contact was broken before the environment prediction algorithm switched from the local model to the propagated model.

B. Results

The results of the catching experiment are summarized in Table I. The propagated model enabled the operator to catch the pod in each of the 5 trials for all of the delay magnitudes attempted. Without the environment prediction, the operator was only able to catch the pod two times with 1 second of delay and never with 2 seconds of delay. Without the robot or environment prediction, the operator's performance degraded further, catching the pod only once with a 0.5 second delay and never with a 1-second delay. No trials were attempted with a 2-second delay for the latter case with no robot or environment prediction.

Fig. 5 shows an example trajectory of the pod and the prediction with 1 second of delay. The prediction is shifted in time so that it lines up with the delayed data. When the pod is caught, the prediction continues to move for 1 second because the local model was not implemented for the catching tests. The root-mean-square errors of the prediction for the five trials at 0.5, 1, and 2 seconds were: 0.74 cm (x) and 1.11 cm (y); 0.91 cm (x) and 1.62 cm (y); and 3.44 cm (x) and 1.58 cm (y) respectively. Error was computed following the round-trip delay, after prediction converged.

The contact experiment successfully demonstrated the ability of the predictive model to handle robot-environment interaction and provide stable haptic feedback to the operator. The left plot of Fig. 6 shows the prediction of the

pod being pushed from rest in the $+y$ direction. The shaded region shows when the local model is in use, during contact and for one round-trip delay after contact. The average error of the local model prediction over five trials was 0.54 cm in the y direction. The right plot of Fig. 6 shows the prediction of the pod while it is floated towards the robot, stopped by the robot, and pushed back. Fig. 7 shows a push in the $-x$ direction, where the operator uses the side of the end effector to push the pod. During contact, the robot and pod measurements do not overlap because the robot position is measured at the center of its end effector and the pod position is measured at its center of mass. The offset is the thickness of the material between these points. In this example, the force on the virtual pod is also applied to the virtual robot and provides haptic feedback to the operator.

C. Discussion

The catching experiment demonstrates that the free space propagated model enables the operator to reliably catch the floating pod. In addition to enabling the operator to catch the pod every time, the prediction also reduced the number of hits that did not result in a catch. Furthermore, using the predictive model, when a hit did occur due to improper contact between the gripper and the pod, the operator could see that he had hit the pod insufficiently. When such a hit occurred, the operator often called out that the contact was not good before the robot made contact because the predictive interface enabled the operator to see the contact ahead of time. With only robot prediction or no prediction, the operator had to guess when to move the robot before the pod approached. Thus, the operator had no knowledge of the outcome of the trial until after the fact. This was worse for the robot only prediction because the robot and environment, as displayed to the operator, do not line up in time. This resulted in the operator having to be told the outcome of the trial. The task without environment prediction was actually made easier than it would be under normal circumstances because the pod was coming from approximately the same location and with the same velocity. Instead of predicting the motion of the pod in each trial, the operator only needed to learn the location of the pod at which he should move the robot to contact the pod. This enabled the operator to catch the pod twice with 1 second of round-trip delay using only the robot prediction. These two catches were obtained after the operator learned from three misses. This is not sufficient for practical use unless the task is equally consistent.

Without robot and environment prediction, catching the pod was even more difficult, resulting in only one catch with 0.5 seconds of delay. The robot prediction alone compensates for half of the round-trip delay, and the delay for the operator is effectively doubled without it. Thus, the performance with 0.5 and 1 seconds of delay with no prediction is similar to the performance with 1 and 2 seconds of delay and only robot prediction.

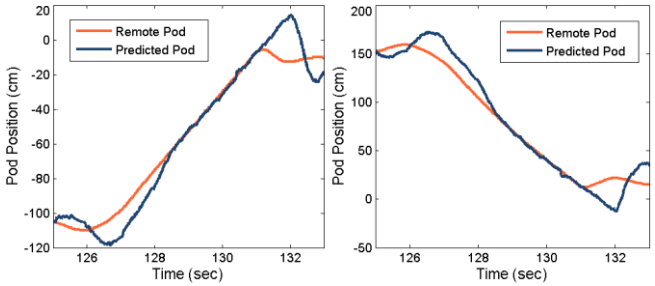

Fig. 5. Trajectories of the remote and predicted pods during the 1-second delay catching experiment in both the x (*left*) and y (*right*) directions. After the pod is pushed, the prediction converges after one round-trip delay. When the pod is caught, the predicted pod position continues moving for one round-trip delay because contact mechanics not implemented during the catching experiment.

TABLE I
RESULTS OF POD CATCHING TRIALS

Prediction	Delay	Catches	Misses	Hits
No Prediction	0	5	0	2
	0.5	1	4	0
	1	0	5	1
Only Robot Prediction	0.5	5	0	4
	1	2	3	4
	2	0	5	4
With Prediction	0.5	5	0	0
	1	5	0	3
	2	5	0	3

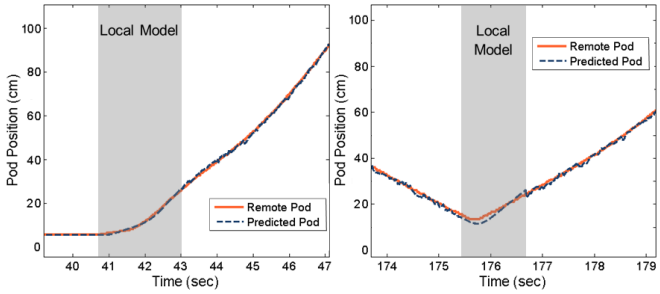

Fig. 6. Trajectories of the remote and predicted pods during the contact experiment in the y direction. On the left, the pod is pushed from rest. On the right, the pod is stopped and pushed away. When the robot makes initial contact, the algorithm switches from the propagated model to the local model. The algorithm switches back to the propagated model 0.5 seconds after contact is broken.

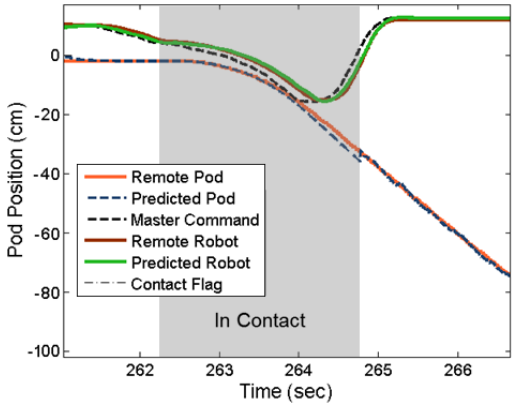

Fig. 7. Trajectories of the remote and predicted pods, master joystick, and remote and virtual robot during the contact experiment in the x direction. The pod is pushed from rest. Because the virtual force applied to the virtual pod is also applied to the virtual robot, the virtual robot does not follow the master command but instead predicts the resulting motion of the real robot.

The prediction accuracy decreased with increasing time delay. However, this decrease in accuracy was less significant than the increase in time required for the prediction to converge. This was why the largest delay in the experiment was 2 seconds, though the pod was even successfully caught repeatedly with a 2.5 second delay. Given the aforementioned velocity and space constraints, larger delays prevented detection of the pod's motion before the pod reached the robot. For 2 seconds of delay, the prediction barely converged at the time of contact, so the errors are significantly higher. Prediction errors after convergence, for instance with 0.5 and 1 seconds of delay, were due to the imperfections of the floor and forces exerted by the air tubing on pod. The resulting unmodeled accelerations introduced error into the prediction.

While the catching experiment demonstrated the effectiveness of the propagated model, the contact test verified the local model's ability to predict the robot and pod interaction and the capability of the environment model to seamlessly transition between free space motion and contact. The local model prediction is invariant to the time delay because it does not make use of delayed data, except when initializing to the state of the propagated model during first contact and waiting to switch back to the propagated model after contact is broken. Therefore, even though the contact test was performed with 0.5 seconds of delay, the local prediction accuracy is applicable to arbitrarily long delays.

The trials in which the pod was moving before contact provide a good example of the ability of the environment prediction to switch between the propagated and local models. When contact is made, the prediction switches from the propagated model to the local model, but this switch is not visible to the operator because the local model is initialized to the state of the propagated model. After the robot pushes the pod away, the prediction switches back to the propagated model after 0.5 seconds. This switch is visible to the operator because the propagated model will account for any error that accumulated during the use of the local model. This is best seen in Fig. 7.

In addition to providing a prediction of the pod's motion, the local model also applies contact forces to the virtual robot to predict its motion in response to contact with the pod. This is seen in Fig. 7, where the virtual robot predicts the motion of the remote robot rather than tracking the master command as it penetrates the pod. This force is also provided to the operator through haptic feedback.

V. Conclusions

This paper presented a method for bilateral teleoperation of a remote robot, using models of the robot and its environment that can predict motion both in free space and during contact. Experiments demonstrated the benefit of the propagated model for repeatedly catching a floating object with up to 2 seconds of delay. A separate test displayed the ability of the combined propagated and local models to accurately predict both environment and robot motion when transitioning between free space and contact, as well as provide stable haptic feedback to the operator. Future work will include online updating of the model, improved contact models with additional experimental validation, and implementation in space on a space-qualified robot, such as the FREND arm. Additional challenges will be met in this future implementation, such as modeling the inertia of the remote robot base, since it will not be grounded in space.

Acknowledgments

The authors thank Bryce Wallis, Mike Briggs, Edmund dela Cruz, and Thomas Yu from the Millennium Engineering and Integration Company, and Aaron Parness and Evan Hilgemann from the Jet Propulsion Lab (JPL).

References

[1] T. J. Debus and S. P. Dougherty, "Overview and performance of the front-end robotics enabling near-term demonstration (FREND) robotic arm," *AIAA Aerospace Conference*, 2009.

[2] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Sensor-based space robotics—rotex and its telerobotic features," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 649-663, Oct. 1993.

[3] W. K. Yoon et al., "Model-based space robot teleoperation of ETS-VII manipulator,: *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 602-612, June 2004.

[4] C. Preusche, D. Reintsema, K. Landzettel, and G. Hirzinger, "Robotics Component Verification on ISS ROKVISS - Preliminary Results for Telepresence," *IEEE Int. Conference on Intelligent Robots and Systems*, pp. 4595-4601, Oct. 2006.

[5] T. B. Sheridan, "Space teleoperation through time delay: review and prognosis," *IEEE Trans. Robot. Autom.*, vol. 9 no. 5, pp. 592-606, 1993.

[6] A. K. Bejczy and W. S. Kim, "Predictive displays and shared compliance control for time-delayed telemanipulation," *IEEE Int. Workshop on Intelligent Robots and Systems*, vol. 1, pp. 407-412, July 1990.

[7] K. Brady and T. J. Tarn, "Handling latency in internet-based teleoperation," in *Beyond Webcams*, Cambridge, MA: MIT Press, 2002, ch. 10, pp. 171-192.

[8] E. W. Hawkes et al., "Dynamic surface grasping with directional adhesion," *IEEE Int. Conf. on Robots and Systems*, to be published.

[9] T. Kotoku, "A predictive display with force feedback and its application to remote manipulation system with transmission time delay," *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 239-246, July 1992.

[10] C. P. Kuan and K. Y. Young, "VR-based teleoperation for robot compliance control," *J. of Intelligent and Robotic Systems*, vol. 30, pp. 377-398, 2001.

[11] L. Huijun and S. Aiguo, "Virtual-environment modeling and correction for force-reflecting teleoperation with time delay," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 1227-1233, April, 2007.

[12] J. Kikuchi, K. Takeo, and K. Kosuge, "Teleoperation system via computer network for dynamic environment," *IEEE Int. Conf. on Robotics and Automation*, vol. 4, pp. 3534-3539, May, 1998.

[13] P. Mitra, and G. Niemeyer, "Model-mediated telemanipulation," *Int. J. of Robotics Research*, vol. 27, no. 2, pp. 253-262, 2008.

[14] S. V. Velanas and C. S. Tzafestas, "Human telehaptic perception of stiffness using an adaptive impedance reflection bilateral teleoperation control scheme," *IEEE RO-MAN*, pp. 21-26, Sept. 2010.

[15] M. D. Lichter and S. Dubowsky, "State, shape, and parameter estimation of space objects from range images," *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2974-2979, April 2004.

[16] J. J. Abbott, "Virtual fixtures and bilateral telemanipulation," Ph.D. dissertation, Dept. of Mech. Eng., Johns Hopkins Univ., Baltimore, MD, 2005.

[17] I. S. Howard, J. N. Ingram, and D.M. Wolpert, "A modular planar robotic manipulandum with end-point torque control," *J. of Neuroscience Methods*, vol. 181, pp. 199-211, 2009.