2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Augmented Reality Environment with Virtual Fixtures for Robotic Telemanipulation in Space

Tian Xia†, Simon Léonard, Anton Deguet, Louis Whitcomb, and Peter Kazanzides

***Abstract*— This paper presents an augmented reality framework, implemented on the master console of a modified da Vinci® surgical robot, that enables the operator to design and implement assistive virtual fixtures during teleoperation. Our specific goal is to facilitate teleoperation with large time delays, such as the delay of several seconds that occurs with ground-based control of robotic systems in earth orbit. The virtual fixtures give immediate visual feedback and motion guidance to the operator, while the remote slave performs motions consistent with those constraints. This approach is suitable for tasks in unstructured environments, such as servicing of existing on-orbit spacecraft that were not designed for servicing. We conducted a pilot study by teleoperating a remote slave robot for a thermal barrier blanket cutting task using virtual fixtures with and without time delay. The results show that virtual fixtures reduce the time required to complete the task while also eliminating significant manipulation errors, such as tearing the blanket. The improvement in performance is especially dramatic when a simulated time delay (4 seconds) is introduced.**

I. Introduction

With aging fleets of spacecraft, several space agencies and commercial collaborators are turning their sights toward servicing missions to extend the lifespan of satellites. The successful missions to repair and upgrade spacecraft such as the Hubble Space Telescope and the International Space Station have demonstrated the value of servicing valuable assets in Low Earth Orbit (LEO) by astronauts [1]. Beyond LEO, aging satellites occupy valuable orbital slots while defective ones represent hazards for other spacecraft in Geostationary Earth Orbits (GEO) where human presence is currently not possible. Furthermore, most decommissioned GEO satellites were not designed for servicing, making them less suitable for servicing by autonomous systems. One solution to address this challenge is to provide servicing capabilities with teleoperated robotic spacecraft. For satellites in GEO, however, teleoperation from earth involves time delays due to signal transmission and routing that can vary between one and six seconds and time delays of this magnitude are known to impede teleoperation performance [16].

This paper proposes a framework to address the problem of telerobotic execution of tasks during satellite servicing, subjected to perceptible time delays. Examples of such tasks include the removal of the multi-layer insulation (MLI) blanket that covers the service port, the removal and reattachment of fasteners and the reapplication of the MLI flap. In this paper, we concentrate on the task of removing the MLI blanket flap that covers a fuel receptacle by cutting the tape around the flap on three sides, as illustrated in Fig. 1, without cutting any part of the MLI or causing damage to the spacecraft. As our main contribution, our approach combines elements of augmented reality for visualization with virtual fixtures for guiding the motion during the teleoperation.

†T. Xia, A. Deguet, L. Whitcomb, P. Kazanzides are with the Laboratory for Computational Sensing and Robotics, Johns Hopkins University, Baltimore, MD. S. Léonard is at the Children's National Medical Center, Washington, DC. T. Xia can be reached at txia1@jhu.edu.

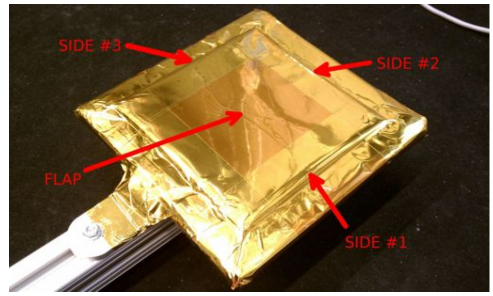


Fig. 1. MLI flap covering a fuel receptacle.

II. Background and Related Work

In a typical teleoperation task, the master arm (which the operator holds) and the remote arm (which mimics the operator's actions) are connected via a continuous closed loop control. For bilateral teleoperation, forces sensed at the slave site are fed back to the master arm, which gives the operator the telepresence of working directly at the remote site. However, even small to medium delays of tens to hundreds of milliseconds can cause force feedback teleoperation systems to become unstable [8]. Furthermore, the typical operator adopts a "move-and-wait" strategy where the operator repeatedly makes small motions and relies primarily on delayed visual feedback from the remote environment to determine the effects of each motion, thus significantly increasing task completion time [4]. Control methods based on maintaining passivity under delay [3] or wave variable transformations of force and motion [13] can achieve stability but are useful for delays under one second. For delays of several seconds, the supervisory control approach proposes that the operator assume a supervisory role, issuing high-level goal specifications and intervening to correct errors, while the remote robot implements these high-level commands by closing a loop local to itself with no effects of time delay and reports back the task completion status (success or error) at sparse time intervals. This approach suffers from difficulties in automating low-level physical interaction between the robot and the environment at the remote site [16].

Teleprogramming [15][5] extended earlier supervisory control techniques and insulates the operator from the communication delays by automatically issuing a sequence of elementary motion commands, based on the operator's action in the simulated environment, to the remote slave robot. Several free space motions and several contact, sliding and pivoting motions can be generated from parsing the operator's interaction with the simulated environment. However, the complexity of motion commands is limited by the difficulty of understanding operator intent and of accurately modeling the environment. The model-mediated approach [12] attempts to build and update a simplified model of the unknown remote environment, and allow the operator to interact with this simplified model with force feedback. Our approach is related to these.

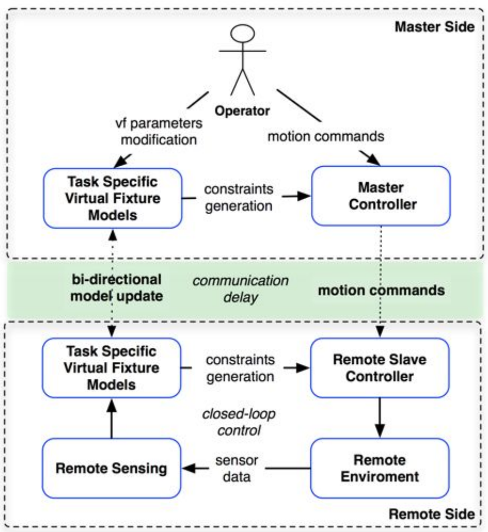


Fig. 2. Control Architecture Concept

We propose the approach of enabling an operator to interactively define complex motion constraints specific to the task, also known as virtual fixtures [10], and to modify the parameters of the virtual fixture models, by using an augmented environment composed of the delayed visual feedback, the graphical models of the remote scene and the virtual fixture models used for the task, as shown in Fig. 2. In effect, we assume that the remote environment does not change or deform and that the registration between the remote slave and the remote environment is accurately known, as described in section IV-C. However, in cases where the environment is slowly or infrequently changing, Fig. 2 also shows the use of remote sensing to dynamically update the virtual fixture models, first on the slave side and then (after the unavoidable time delay) on the master side, where the operator can both see and feel the results without time delay. This is consistent with model-mediated teleoperation, though we note that our approach will support bi-directional model updates (e.g., by the operator via the augmented reality interface and by the remote system via sensor feedback). We are currently integrating a force sensor on the slave robot to enable these capabilities.

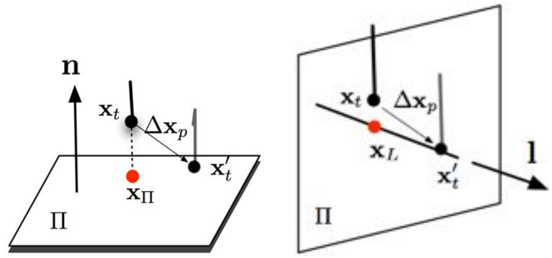
III. Technical Approach

As with most robotics applications that involve a robot interacting with its environment, it is imperative to avoid damaging the robot, its tool or the spacecraft by exerting excessive forces or generating erratic motions. For teleoperation in medical robotics, a strategy often used in surgeries to avoid similar problems is to use virtual fixtures [6][11][18]. A virtual fixture performs a function analogous to that of a physical fixture and can be adjusted in a virtual environment at run time. For example, a virtual fixture that constrains a robot to move along a straight line is analogous to the use of a physical ruler to draw a straight line. Thus, the purpose of virtual fixtures is to help an operator to perform a robot-assisted manipulation task by limiting its movement to a given workspace (Forbidden Region Virtual Fixture) and/or influencing its motion along a desired path (Guidance Virtual Fixture)[2]. Using virtual fixtures in the context of a teleoperation task results in a system that combines the accuracy of a robotic system and the expertise of human operators with improved safety and efficiency. This synergy is especially important for systems with large time delays because the virtual fixture can either be applied on the master, where there is no time delay [14] or it can be uploaded to the slave robot's controller, in which case it is not affected by the time delay. At the master console, the operator receives immediate simulated visual and kinesthetic feedback and has an enhanced telepresence and localization of the remote site. Kapoor, et al. [10] presented a general constrained motion control form for virtual fixture based geometric tasks, such as "stay above a plane", "move along a line", and "rotate about a point", that can be combined using different operators to provide assistance for complex manipulation tasks. However, the system focuses on the integration of virtual fixtures in the robot control loop but lacks libraries of geometric primitives and a graphical user interface to interactively instantiate and configure each virtual fixture.

Our approach allows the human operator to instantiate and compose virtual fixtures from primitives contained in a library within an interactive environment. The environment is rendered in a master console that is composed of a stereo display and a back-drivable robot arm capable of force feedback.

A. Virtual Fixtures for Teleoperation with Time Delay

As the operator performs a small motion in the virtual environment, the virtual fixture controller computes an optimal incremental motion $\Delta\mathbf{x}_c$ based on the operator's desired incremental motion $\Delta\mathbf{x}$, which has rotational and translational components, $\Delta\mathbf{x}_r$ and $\Delta\mathbf{x}_p$, respectively. The computation of $\Delta\mathbf{x}_c$ is expressed as a quadratic constrained optimization problem where each virtual fixture is represented by a linear



(a) Plane constraint (b) Line constraint

Fig. 3. Virtual fixtures geometry models. $\mathbf{x}_t'$ is the desired tool tip position at time step $t + \Delta t$. $\Delta\mathbf{x}_p$ is the operator's desired incremental motion at the master.

inequality constraint[10].

$$\min_{\Delta\mathbf{x}_c} \left\|\Delta\mathbf{x}_c - \Delta\mathbf{x}\right\|^2 \tag{1}$$

$$s.t. \quad \begin{array}{c} h_1(\Delta\mathbf{x_c}) < 0 \\ \vdots \\ h_N(\Delta\mathbf{x_c}) < 0 \end{array}$$

where $h_1, \dots, h_N$ represent the constraints of N virtual fixtures.

The optimal incremental motion $\Delta\mathbf{x}_c$, which is subject to the defined virtual fixture constraints, is then executed remotely on the remote slave arms. The operator receives immediate visual and haptic feedback from the model from the virtual fixture constrained model-based simulation.

B. Augmented Reality for Virtual Fixture Modeling

In our blanket cutting task, the 3D augmented environment is composed of the delayed visual feedback, the graphical models of the satellite and the virtual fixtures used for the task. The 3D model of a satellite can be obtained by either using data provided by its manufacturer or by using a range sensor from the docking spacecraft. The model is registered in the coordinate system of the remote arm and overlaid in the augmented reality environment. Using the virtual environment, the operator instantiates the necessary virtual fixtures and adjusts their geometric parameters. In this "master-as-mouse" mode, the remote slave arm is "clutched" from the master arm and the operator only interacts with the virtual environment.

For the planar virtual fixture, the nominal behavior is to keep the remote robot's tool frame position, $\mathbf{x}_t$, above the plane Π that has the normal direction $\mathbf{n}$ pointing to the free half space and passing through point $\mathbf{p}$, see Fig. 3(a). The tool is free to move in the free half space, proportional to the operator incremental motion, $\Delta\mathbf{x}$, in position and orientation. For the line virtual fixture, the behavior is to keep the tool position $\mathbf{x}_t$ on line L which has the direction $\mathbf{l}$ and passes through point $\mathbf{p}$, while allowing the tool to move along L proportional to the operator input $\Delta\mathbf{x}$, as shown in Fig. 3(b). For each primitive, the operator also defines a small positive tolerance value, ϵ, that serves as an absolute bound for deviation from the nominal behavior. For our experiments, we used values between 1e-2 and 1e-4.

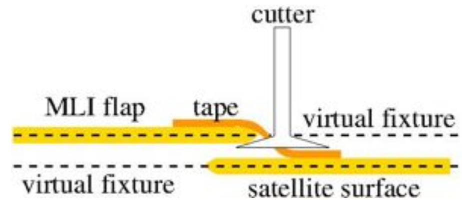


Fig. 4. Cross section view of the satellite surface. The cutter must make an incision and slide along the flap without cutting the MLI blanket or damaging the satellite. Planar virtual fixtures ensure that the cutter remains within a reasonable cutting height.

For a task such as cutting the tape around an MLI flap, the operator can avoid applying excessive force on the spacecraft and pulling on the MLI by constraining the motion of the slave between two planes parallel to the surface that are defined by two planar virtual fixtures as illustrated in Fig. 4. The operator can drag and rotate the 3D interactive manipulators (IMs), or control markers associated with the geometry object, thereby applying rigid body transformations to the geometry associated with each virtual fixture (see Fig. 5). The associated geometric parameters, such as plane normal $\mathbf{n}$ and point $\mathbf{p}$ can be extracted. The operator can add or remove the specifications for virtual fixtures on the fly. The virtual fixtures are formulated mathematically as inequality constraints in an optimization problem.

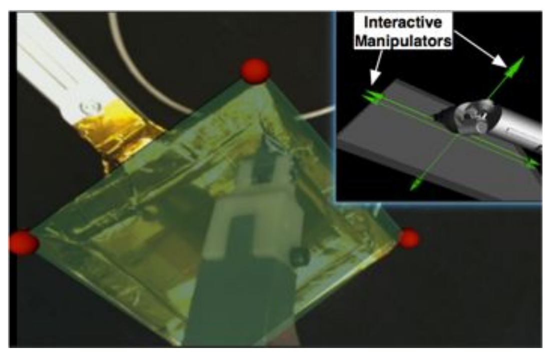


Fig. 5. Virtual fixture modeling interface showing delayed video overlaid on top of registered model, and IMs for manipulating virtual fixtures in the modeling environment.

For the planar virtual fixture defined by a plane Π, we define the inequality constraint

$$h_\Pi(\Delta\mathbf{x}) = \mathbf{n}^T \cdot ((\mathbf{x}_t - \mathbf{x}_\Pi) + \Delta\mathbf{x}_p) - \epsilon < 0 \tag{2}$$

where $\mathbf{x}_\Pi$ is the closest point on plane Π to $\mathbf{x}_t$. For the line virtual fixture defined by a reference line in 3D space, given by $L : L(s) = \mathbf{p} + \mathbf{l} \cdot s, s \in \mathbb{R}$, we define the inequality constraint

$$h_L(\Delta\mathbf{x}) = \left\|((\mathbf{x}_t - \mathbf{x}_L) + \Delta\mathbf{x}_p)) \parallel \Pi_L\right\| - \epsilon < 0 \tag{3}$$

where $\mathbf{v} \parallel \Pi_L$ represents the projection of vector $\mathbf{v}$ onto the plane Π_L that is perpendicular to line L. $\mathbf{x}_L$ is the closest point on the line.

After the parameter specifications are completed, the VF algorithm generates constraints and then combines them in an optimization problem, and solves for $\Delta\mathbf{x}_c$, the optimal incremental motion at each iteration, as in equation (1).

IV. System Implementation

A. Telerobotics System

We have created a testbed for exploring telerobotic on-orbit servicing of spacecraft. The testbed consists of a modified da Vinci ® robot system master console [7] and a slave manipulator, using a Whole Arm Manipulator (WAM) robot at the Johns Hopkins University's Laboratory for Computational Sensing and Robotics (LCSR). The WAM robot is a 7 degree-of-freedom serial robot from Barrett Technologies, Inc.

The master console is built around two Master-Tele-Manipulators (MTMs) and a stereo display console from a da Vinci Classic Surgical System provided by Intuitive Surgical, Inc. The commercial version of the da Vinci is a clinically approved telerobotic system for Minimally Invasive Surgery. Each MTM is a cable driven 7-DoF serial robot with a redundant wrist mechanism for Cartesian motion input. Through custom electronics hardware and software (based on cisst/SAW) [9], we have created a modified system which gave us direct access to low-level joint control and we have implemented new features such as haptic feedback. Our control software relies heavily on the concept of component-based system design, where each component has well-defined interfaces for joint-level or Cartesian-level control, for example. As a result, we can quickly prototype new system configurations, by connecting the master arm to any one of the remote slaves at run-time, see Fig. 6.

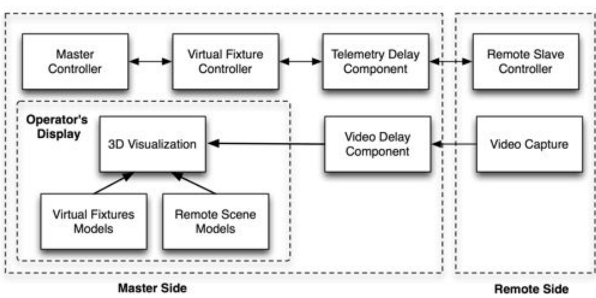


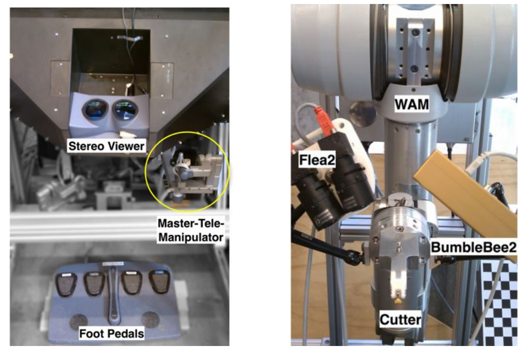
Fig. 6. Software System Components

B. Delay Components

The stereo video is obtained by two Flea cameras (Point Grey, Inc.), in a calibrated stereo-vision system. We used software components to simulate the time delay (typically 2-7 seconds) by buffering the stereo video and telemetry data from the remote slave. The amount of time delay can be adjusted at run time.

C. Registration

We register the coordinate systems of the remote slave to an MLI blanket covered aluminum plate, which serves as the mockup of the fuel receptacle of a satellite. After this registration, the virtual geometrical model of the satellite is shown in the coordinate system of the remote slave, and the operator can use the satellite model as a guide to interactively place plane or line virtual fixtures. We also calibrate the stereo-vision system on the remote slave, in order to perform video-to-model augmented reality overlay.



(a) da Vinci master console. (b) WAM slave with cutter, two Flea2 and one Bumblebee2.

Fig. 7. JHU robots for teleoperation MLI cutting.

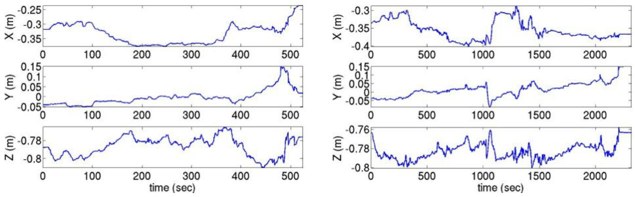
V. Experimental Results

We have implemented and tested the teleoperation system using the da Vinci master console (Fig. 7(a)) and the Barrett WAM robot (Fig. 7(b)) described in Section IV. The vision system mounted on the WAM consists of two Point Grey Research Flea2 cameras and one Bumblebee2 stereo camera. The blade of the cutter is mounted on the end effector of the WAM, with one rapid prototyped part designed to act as a "mechanical fuse" to avoid damaging the blade. The satellite surface is represented by a 20 cm^2 aluminum plate that is covered with MLI. A small 15 cm^2 MLI flap is taped on top of the plate to represent the fuel hatch. To evaluate the performance of our system we compare the results of the following experiments:

1) Unassisted teleoperation, with no time delay.
2) Unassisted teleoperation, with 4 second delay.
3) Virtual fixture-based teleoperation, with no time delay.
4) Virtual fixture-based teleoperation, with 4 second time delay.

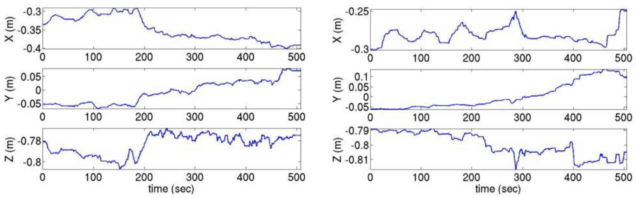
Our virtual fixture-based teleoperation scenarios include a single line or a pair of planes.

The task consists of cutting one side of the plate and comparing results based on the time of completion, deviation from a straight line trajectory and damage done to the MLI. The plate was laid in the X-Y plane of the slave coordinate system and the task consisted of cutting the tape along the Y axis with direction $\mathbf{y} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$. For each experiment the tool was moved close to a start corner on the plate to provide consistent initial conditions for the experiments. From there, the operator had to make an incision in the tape and slide the blade along the surface of the plate to cut the tape. For experiments using virtual fixtures, the operator had to manually adjust the position of each fixture in the virtual environment before starting the cut. The time for adjustments is not included in our results, as it is expected to be minimal with a well-implemented user interface.



(a) No delay. (b) 4 second delay.

Fig. 8. Cutting with no virtual fixture.



(a) No delay. (b) 4 seconds delay.

Fig. 9. Cutting with two planar virtual fixtures.

A. Unassisted teleoperation

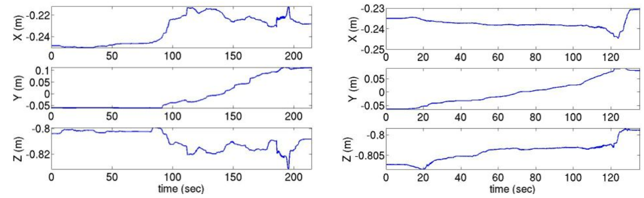
In our first set of experiments, an operator was asked to cut one side of the tape around the MLI flap without any assistance. In the first experiment, no delay was injected in the system other than the unavoidable delays caused by video frame grabbing, processing and local area network communications, which amounts to less than half a second. The resulting trajectory of the experiment is reported in Fig. 8(a). In the second experiment, a four second delay was injected and the resulting trajectory is illustrated in Fig. 8(b).

A striking observation from the figures is the amount of time required to execute the task in the presence of time delay. While the task without time delay took approximately 8 minutes, the same task with time delay took over 30 minutes and resulted in severe damage to the MLI flap (Fig. 11(a)). In fact, most of the cutting during the experiments with time delay resulted in tearing the MLI. We also fit a 3D line through the cutter's trajectory data points. The parameters of the line were estimated by total least squares (TLS) [17] and the direction of the line is reported in Table I. The TLS only considers the 3D position of the cutter such that the orientation of the cutter is not considered by the optimiztion. While the expected direction of the cutting trajectory is $\mathbf{d} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$ these results suggest that the direction of the trajectories deviate by 26° and 38°.

TABLE I

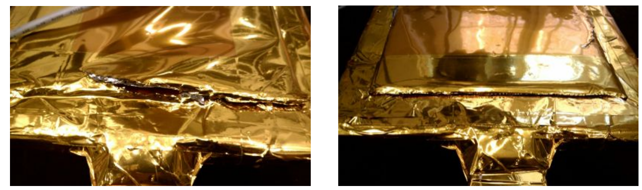
ESTIMATED DIRECTION OF CUTTING TRAJECTORY

	$\mathbf{d}^T$	$\arccos(\mathbf{d}^T \cdot \mathbf{y})$
Unassisted (0 sec)	$\begin{bmatrix} 0.4337 & 0.8978 & -0.0759 \end{bmatrix}$	26.1296°
Unassisted (4 sec)	$\begin{bmatrix} -0.6061 & 0.7829 & 0.1402 \end{bmatrix}$	38.4731°
Planes (0 sec)	$\begin{bmatrix} -0.3828 & 0.9179 & 0.1045 \end{bmatrix}$	23.3790°
Planes (4 sec)	$\begin{bmatrix} -0.0077 & 0.9945 & -0.1048 \end{bmatrix}$	6.0120°
Line (0 sec)	$\begin{bmatrix} 0.1263 & 0.9892 & -0.0743 \end{bmatrix}$	8.4283°
Line (4 sec)	$\begin{bmatrix} -0.0180 & 0.9988 & 0.0448 \end{bmatrix}$	2.8072°



(a) No delay. (b) 4 seconds delay.

Fig. 10. Cutting with one line virtual fixture.



(a) No virtual fixture. (b) With line virtual fixture.

Fig. 11. Cut results with 4 second delay.

B. Virtual Fixture Teleoperation: Planes

The second set of experiments considered using two planar virtual fixtures to constrain the position along the Z axis between both planes. Before executing the task, the operator adjusted the height of the planes to squeeze the tool between the lower and upper planes. To make the initial incision in the tape, the height of the lower plane must allow the tool to push on the surface to avoid gliding over the tape with the cutter. We repeated the experiment of section V-A with these fixtures and present the resulting trajectories in Fig. 9. The first observation is that with no time delay, the plane virtual fixtures only reduced the task completion time by 30 seconds. The direction of the best fit line (Table I) also improved by deviating from the ideal line by 23° instead of 28°. The main improvement, however, occurs in the time delay scenario, where the unaided time of 33 minutes was reduced to less than 9 minutes by the addition of the virtual fixture. Our results indicate that when using virtual fixtures, the completion time is essentially independent of the time delay (at for the time delays we tested). Furthermore, the direction of the line that best fits the trajectory deviates from the ideal trajectory by only 6° instead of 38°. Finally, and significantly, there were no tears of the MLI when using the virtual planes.

C. Virtual Fixture Teleoperation: Line

For the line virtual fixtuer experient, the cutter was placed near the flap where the virtual line was adjusted and activated. The resulting trajectories presented in Fig. 10 illustrate that the task, irrespective of time delays, was accomplished in less than 4 minutes which represents a 57% improvement over the experiments in V-B and, when time delays are injected, a 90% improvement. The virtual lines also improved the accuracy of the trajectory (Table I). The lines that best fit the trajectories only deviate from the ideal line by 8° and 2°. The MLI was left intact in both tests, as illustrated in Fig. 11(b).

VI. Discussion And Conclusion

We have developed a testbed for exploring telerobotic on-orbit servicing tasks on spacecraft, motivated by our ongoing research in telerobotic surgery. Our initial task is to make a precise incision in the tape that fastens a flap of MLI blanket on the satellite, using a tool that bears some resemblance to a scalpel. We further note that most satellites currently in orbit have not been designed for servicing, much as the human body has not been designed to facilitate surgery.

Our testbed consists of a modified da Vinci master console, which we use to teleoperate various robots at JHU, NASA Goddard, and WVU. Compared to a typical surgical scenario, teleoperation of robots in space introduces significant time delays and bandwidth constraints. We implemented software components to simulate time delays of telemetry and video. We do not yet consider bandwidth constraints, which may reduce the quality (resolution) of video feedback unless they can be overcome by image compression methods or future improvements to the communication infrastructure.

The key contribution of our work is the development of an augmented reality framework that enables the operator to design and implement assistive fixtures when confronted with an unknown task in an unstructured environment unsuitable for autonomous systems. Although motivated by our research in medical robotics (where we extensively use virtual fixtures and augmented reality overlays), this framework represents a new development in either domain.

We have conducted a pilot study by teleoperating the WAM robot for an MLI cutting task using virtual fixtures with and without time delay. The results show that virtual fixtures reduce the time required to complete the task while also eliminating significant adverse events, such as tearing the MLI. The improvement in performance is especially dramatic when a simulated time delay (4 seconds) is introduced. These results, however, relied on the perfect placement of the virtual fixtures, which was enabled by an accurate registration of the robot with respect to the satellite mockup. Our current work includes the use of sensor feedback (e.g., force sensing) at the remote site to update the placement of the virtual fixtures, as shown in the lower left of Fig. 2.

In summary, we note that presently it is virtually impossible to perform complex manipulation tasks with robotic manipulators in unstructured environments. While the highly structured known environment of the assembly line has enabled successful robotic automation of structured manipulation tasks in manufacturing, robotics has had comparatively little impact in less structured manipulation tasks such as surgery and machine assembly/disassembly/repair due to the uncertainty arising in these tasks and environments. By combining human strengths in reasoning and high-level strategy with machine capabilities in information fusion, task planning, and simulation, we can manage uncertainty and achieve successful human-robot partnerships to perform complex tasks in uncertain environments that were previously considered impractical or infeasible.

VII. Acknowledgements

This work is supported by a subcontract from West Virginia University (WVU) on NASA NNX10AD17A. The authors gratefully acknowledge the assistance of Thomas Evans and his team at WVU, the Satellite Servicing Capabilities Office at NASA Goddard Flight Center, and our colleagues Greg Hager, Jonathan Bohren, Kelleher Guerin, Isha Kandaswamy, and Zihan Chen at the Laboratory for Computational Sensing and Robotics at the Johns Hopkins University.

References

[1] On-orbit satellite servicing study project report. Technical report, NASA Goddard Space Flight Center, http://ssco.gsfc.nasa.gov/servicing_study.html, 2010.

[2] J. J. Abbott and A. M. Okamura. Analysis of virtual fixture contact stability for telemanipulation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2699–2706, 2003.

[3] R. J. Anderson and M. W. Spong. Asymptotic stability for force reflecting teleoperators with time delay. *International Journal of Robotics Research*, 11(2):135–149, 1992.

[4] W. Ferrell. Delayed force feedback. *Human Factors*, 8(5):449–455, 1966.

[5] J. Funda, T. Lindsay, and R. P. Paul. Teleprogramming: Toward delay invariant remote manipulation. *Presence*, 1(1):29–44, 1992.

[6] J. Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben. Constrained Cartesian motion control for teleoperated surgical robots. *IEEE Transactions on Robotics and Automation*, 12(3):453–465, June 1996.

[7] G. S. Guthart and K. Salisbury. The Intuitive^{TM} telesurgery system: overview and application. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[8] K. Hashtrudi-Zaad and S. Salcudean. Analysis and evaluation of stability and performance robustness for teleoperation control architectures. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3107 –3113, 2000.

[9] M. Y. Jung, A. Deguet, and P. Kazanzides. A component-based architecture for flexible integration of robotic systems. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6107 –6112, 2010.

[10] A. Kapoor, M. Li, and R. Taylor. Constrained Control for Surgical Assistant Robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[11] M. Li, M. Ishii, and R. H. Taylor. Spatial Motion Constraints Using Virtual Fixtures Generated by Anatomy. *IEEE Transactions on Robotics*, 23(1):4–19, 2007.

[12] P. Mitra and G. Niemeyer. Model-mediated telemanipulation. *International Journal of Robotics Research*, 27(2):253–262, February 2008.

[13] G. Niemeyer and J.-J. E. Slotine. Telemanipulation with time delays. *International Journal of Robotics Research*, 23:873–890, 2004.

[14] L. B. Rosenberg. The use of virtual fixtures to enhance operator performance in time delayed teleoperation. Technical Report AL/CR-TR-1994-0139, Wright-Patterson Air Force Base, 1993.

[15] C. P. Sayers, R. P. Paul, L. L. Whitcomb, and D. R. Yoerger. Teleprogramming for subsea teleoperation using acoustic communication. *IEEE Journal of Oceanic Engineering*, 23:60–71, 1998.

[16] T. B. Sheridan. Space teleoperation through time delay: review and prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, 1993.

[17] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1991.

[18] T. Xia, C. Baird, G. Jallo, and K. Hayes. An integrated system for planning, navigation and robotic assistance for skull base surgery. *International Journal of Medical Robotics and Computed Assisted Surgery*, 4(4):321–330, September 2008.