A Virtual Reality Planning Environment for High-Risk, High-Latency Teleoperation

Will Pryor, Liam J. Wang, Arko Chatterjee, Balazs P. Vagvolgyi, Anton Deguet, Simon Leonard
Louis L. Whitcomb and Peter Kazanzides

***Abstract*— Teleoperation of robots in space is challenging due to high latency and limited workspace visibility. Previously, the Interactive Planning and Supervised Execution (IPSE) and Augmented Virtuality systems were developed to reduce failure risk. These tools were visualized on a 3D da Vinci surgical console and operated using the da Vinci manipulators or visualized on conventional monitors and operated with a keyboard and mouse. Experimental studies indicated operator preference for the latter. In this work, we develop a 3D virtual reality (VR) interface for IPSE, implemented on a Meta Quest 2 head-mounted display (HMD), and evaluate it against the prior 2D, keyboard-and-mouse-based interface. The results demonstrate improved operator load with the 3D VR interface, with no decrease in task performance, while also providing cost and portability benefits compared to the conventional 2D interface.**

I. Introduction

On-orbit servicing of spacecraft in low earth orbit (LEO) represents a singular challenge. Servicing missions have inherent high risk and only a few high value spacecraft (e.g., the Hubble telescope) can justify a manned mission, while most spacecraft can be preferably serviced by robotic systems. However, most spacecraft in LEO were not designed to be serviced at all, let alone by robots. As such, several of the manipulations during servicing cannot easily be automated and instead require ground-based teleoperation. However, ground-based teleoperation of on-orbit robots is challenging due to high latency communications, with telemetry delays of several seconds, and difficulties in visualizing the remote environment due to limited camera views.

The high cost of failure discourages direct teleoperation, where the operator's motion directly causes motion of the remote robot, especially for large motions, and operators often instead use smaller motions with a "move and wait" strategy. One alternative, especially attractive for large motions, is to carefully plan and evaluate robot motions on the ground before the robot executes those motions on orbit. This is similar to offline programming [8], except that it is performed within the context of teleoperation and therefore must be interactive to enable operators to adjust to changes in the environment, including changes due to the robot actions.

The conventional operator interface for teleoperating robots in space (and in other extreme environments) relies on traditional input devices, such as a keyboard and mouse, and multiple displays for visualizing remote camera views, robot telemetry, and simulated robot motion. While this approach has worked for decades, the recent proliferation of mixed reality hardware, in particular head-mounted displays (HMDs), offers the possibility for a more compact, intuitive and immersive environment.

Johns Hopkins University, Baltimore, MD, USA,
{willpryor, wwang136, balazs, sleonard, llw, pkaz}@jhu.edu

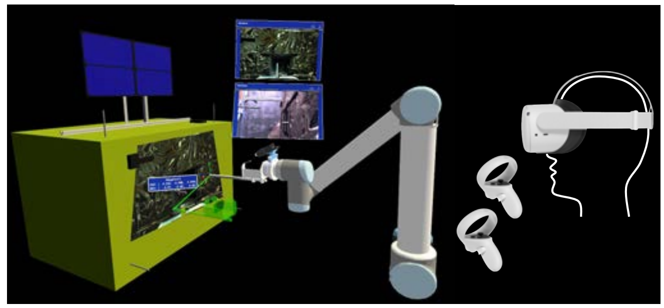

Fig. 1. Virtual reality planning environment using Meta Quest 2 headset.

In this manuscript, we present a mixed reality interface (Fig. 1) that facilitates interactive planning as a mechanism for teleoperation of technically challenging and high-risk robotic operations, such as satellite servicing. Our goals are to alleviate the burden of these procedures for the ground-based operator and to decrease the risk of failure and errors. As an illustrative task, we focus on a critical step of refueling of a spacecraft where the remote manipulator combines large motions to reach its tool staging area with fine motions to insert the nozzle into the filler neck.

Our first contribution is the development of a mixed reality implementation, using a virtual reality (VR) headset with six degrees of freedom (DOF) controllers, of our Interactive Planning and Supervised Execution (IPSE) method [9]. IPSE previously relied on mouse and keyboard inputs to manipulate interactive markers in an augmented virtual environment and visualize the results on multiple monitors (an alternate 3D implementation, using the stereo display and 3D input devices of a da Vinci surgical console, was less preferred). Our second contribution is a multi-user study that compares the mixed reality implementation of IPSE to the prior system.

II. Related Work

Advances in robotics hardware have enabled robots to execute increasingly complicated tasks in a broad variety of environments requiring equally complex programming. The benefits of using mixed reality in robotics and, in particular, for interactive robot programming have been researched for several years but the recent development of immersive and affordable HMDs has increased the amount of research in this area [3], [20]. Yet, few of these immersive robot programming technologies have been designed for robots operating in non-engineered remote environments where feedback is subject to latencies of several seconds.

A large portion of the research in this area is aimed at industrial applications, with the objective of making robot programming accessible to operators without experience [4]. In [11], a user can plan paths as a series of waypoints in an augmented reality (AR) environment, preview and edit the paths, and then execute them either autonomously or by allowing the user to control progress through the path. Likewise, path-planning within an AR environment is presented in [2] where the user is able to visualize the free space and select configurations by using interactive markers. In [7], instead of moving a virtual robot, input devices are used to directly define paths in an AR workcell. In [5], dynamic constraints were also included to improve the preview of the trajectory of the robot. As is often the case for industrial applications, the aforementioned methods all assume knowledge of the robot's operating environment. This is primarily achieved by accurate CAD models and careful calibration or by the operator physically sharing the workcell with the robot. As the operator is physically present near the robot in these applications, the operator can view the physical robot to monitor execution and compensate for any inaccuracies in the motion plan.

Other research focuses on enabling HMD-based teleoperation in more dynamic environments, using deep learning pose estimation to localize objects. The system described in [19] maps the operator's VR hand controller pose directly to the end effector. In [18], operators plan and preview robot paths by placing a series of waypoints, but the interface lacks methods for precise waypoint positioning. Neither system was designed for high-latency teleoperation.

Drone applications have a natural need for immersive environments for First-Person View (FPV) drone flying [6], [12], building 3D immersive environments [1] or monitoring and surveillance [14], [21]. Similar to the industrial applications, the proximity between the drones and the operators makes the sensed data readily available to the operator. Additionally, few drone applications require precise close-proximity movements, so slight inaccuracies in virtual environments are inconsequential.

Our application requires an interface that allows precise teleoperation of a remote robot where direct observation is not possible. Erroneous motion must be avoided, as operator reaction time is limited due to the communication time delay of several seconds. We aim to improve upon these prior systems by integrating methods to visualize the remote environment with the available limited sensor data, to tolerate inaccuracies in the CAD model of the operating environment, and to allow confident execution in situations with large round-trip time delay.

III. Background and Motivation

Satellite servicing will require robots to perform multiple tasks, with different tools designed for these tasks. It is expected that multiple tool changes will be necessary during the servicing task, which necessitates navigation of the robot arm between the worksite on the target satellite and the tool stowage area on the servicing spacecraft. This must be accomplished while avoiding collisions between the robot arm and structures on either spacecraft. Thus, a typical satellite servicing task will require both coarse motion to transit to/from the tool stowage area and fine motion to perform proximity operations.

In [16], [17], we proposed methods for modeling the remote environment to overcome the visualization challenges due to the limited camera views. Specifically, the servicing robot first performs an image survey by moving its tool-mounted camera through a number of positions around the target satellite. Features on these 2D images are manually identified and used to either register to a known model (for example, the satellite frame) or to reconstruct an unknown or imprecisely known object, such as soft structures attached to the satellite. The resulting 3D model enables a VR visualization of the remote environment. However, the approach proposed in our prior work [16], [17] went beyond VR by projecting the received camera images onto the 3D models. This projection of real content onto the virtual environment is referred to as augmented virtuality (AV). A user study with trained robot operators [10] revealed that the AV visualization led to improved task performance and higher operator satisfaction when compared to the traditional visualization interface (i.e., camera views that are unmodified, or with minimal augmented reality overlays).

In [9] we developed a system for Interactive Planning and Supervised Execution that leverages the constructed 3D models to provide an intuitive environment to specify a plan involving both coarse and fine motion, verify its accuracy to the operator's satisfaction, and execute it with as much human-in-the-loop supervision as can be afforded in the satellite servicing scenario. We evaluated this system using two user interfaces: one with keyboard-and-mouse interaction on a traditional 2D computer monitor (with occasional use of a 3D monitor), and one using the da Vinci surgical robot control console for 3D interaction. The results of this study clearly indicated that the 2D user interface led to better outcomes, however, users indicated that their lack of success with the 3D interface may be attributable to implementation details rather than inherent properties of the interaction. Specifically, the 3D interface lacked some key features, which included the ability to (1) numerically specify precise motions with respect to a selected coordinate frame in the model, and (2) view the augmented virtuality visualization during the supervised execution phase. Thus, in this work we implement an alternative 3D interface to the same system using a VR headset and evaluate its success compared to the previously-indicated best interface.

IV. System Description

The system we present in this paper builds upon the Interactive Planning and Supervised Execution (IPSE) system presented in [9], which was implemented using the Robot Operating System (ROS) [13]. The VR interface is implemented

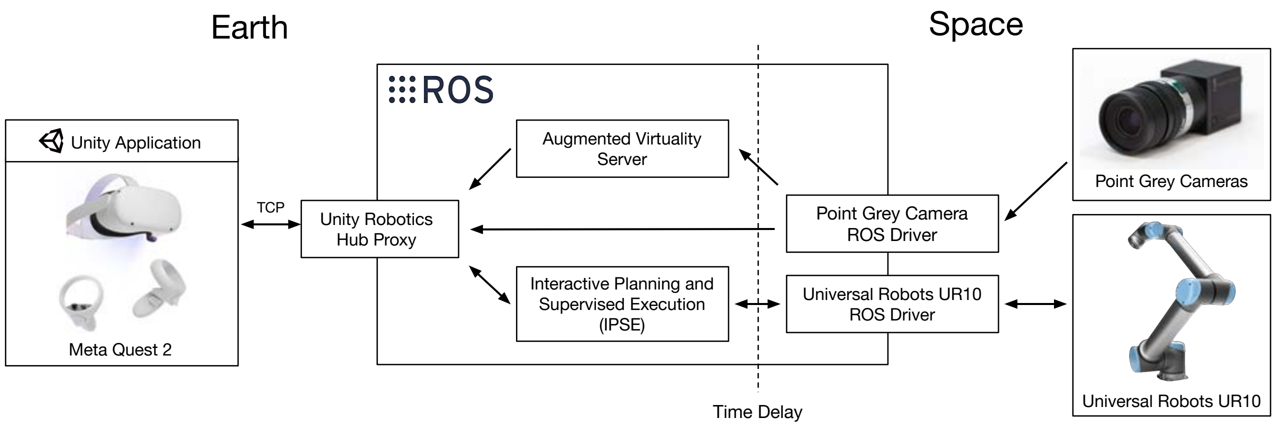

Fig. 2. Overview of virtual reality planning interface architecture. The core IPSE planning system and augmented virtuality server is implemented using ROS. The ROS network is distributed across ground-based computers and space-side computers. The Unity Robotics Hub TCP proxy is used for bidirectional communication between ROS nodes and the VR Unity application.

using the Unity 3D game engine, which is the predominant development platform for mixed reality. We used the Unity Robotics Hub [15] ROS-TCP-Connector package to interface the Unity application with IPSE and the AV server using ROS topics over a local Wi-Fi network (Fig. 2). Because Unity supports a number of different hardware platforms, including most (if not all) commercially-available AR and VR headsets, it also provides the advantage of portability to different hardware. In this paper, we report the development of a VR interface using the Quest 2 (Meta, Menlo Park, CA), a standalone wireless headset with 6-DOF tracked hand controllers, with the following section presenting the rationale for this choice.

A. Selection of Mixed Reality HMD

Our development effort initially targeted the Microsoft HoloLens 2 (HL2) augmented reality headset due to our prior experience with this platform for other projects. However, we concluded that the HL2 hand tracking capability was not sufficient for precise motion specification and therefore integrated a Xencelabs Quick Keys handheld scroll wheel controller for reduced-DOF waypoint adjustment. Later in development we compared the HL2 to the Meta Quest 2 and we found that the improved field of view and visual clarity of the Quest 2 enabled a more immersive operating experience. Additionally, the Quest 2 hand controllers offered greater precision than the built-in hand tracking of the HoloLens 2 and were more ergonomic than the Quick Keys controller. Since the VR interface was developed with Unity, minimal code changes were required to deploy to Quest 2.

B. Interactive Planning: Camera Visualization

The virtual world includes the AV camera feed overlays (Fig. 3) which augment the virtual models with tool camera images. This is implemented using the previously-developed OpenGL-based software [10], running on an external PC, that projects video frames captured by the tool camera onto texture maps covering the 3D model of the satellite. Texture maps are calculated real-time using ray casting to account

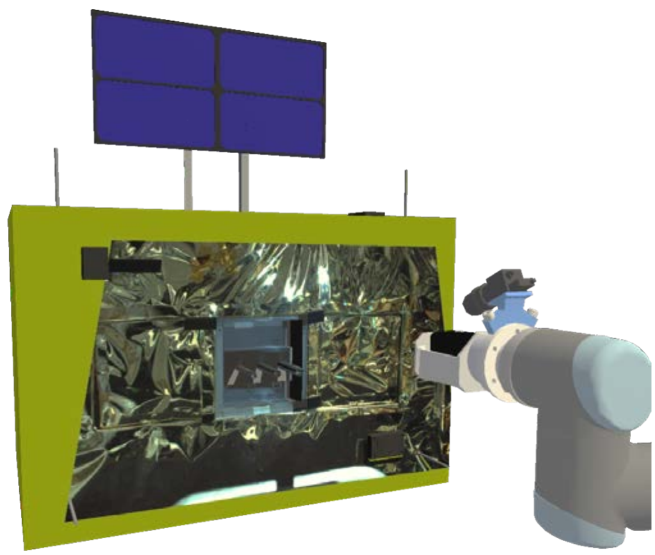

Fig. 3. Augmented Virtuality texture overlays are displayed in the virtual reality planning environment.

for occlusions, then published by the renderer to compressed ROS image topics. The HMD receives the images via the ROS-TCP-Connector and updates the texture maps on the rendered 3D models. In addition, operators can view multiple live camera feeds with virtual 2D windows which can be arbitrarily resized and positioned, as can be seen in Fig. 4.

C. Interactive Planning: Virtual Reality Interface

Using the headset, operators can view the virtual world (Fig. 4) in 3D and create IPSE motion plans, which consist of a series of waypoints. The built-in 6-DOF inside-out tracking on the Meta Quest 2 allows operators to look around the virtual world by moving in physical space. The 6-DOF tracked hand controllers allow operators to interact with IPSE and the virtual world. The hand controllers have joysticks, triggers, and buttons that are mapped to various actions within the planning environment. Using the hand controllers, operators can move, rotate, and scale the virtual world to inspect areas

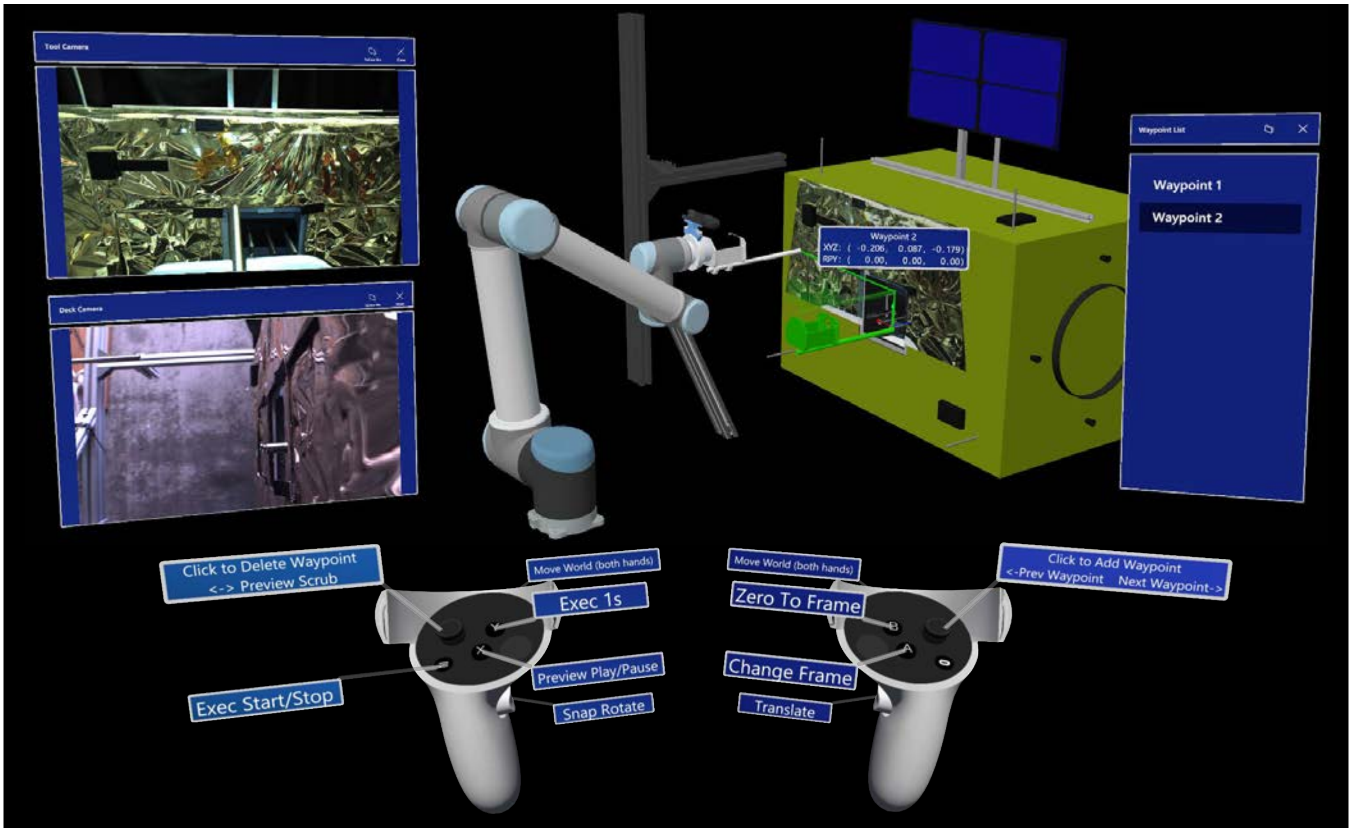

Fig. 4. A screenshot of the operator's perspective in the Meta Quest 2 headset. The 3D virtual planning scene (top center) shows the pose of the space-side robot relative to the satellite and refueling tool station. Operators can place green waypoint markers to create a motion plan using the Meta Quest 2 6-DOF hand controllers, which are displayed virtually with labeled buttons (bottom). Virtual camera windows (top left) display live 2D video from space-side cameras. The waypoint list (top right) displays the waypoints in the motion plan and highlights the currently selected waypoint.

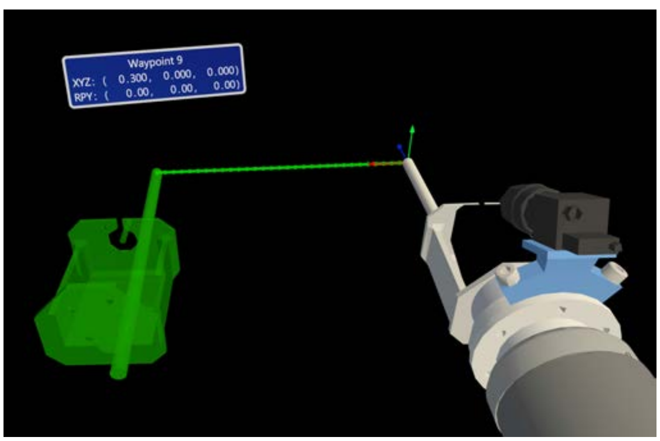

Fig. 5. Motion plans are constructed by placing green waypoint markers (left) in 3D space. A dotted line shows the path of the end effector when moving from the initial pose to the destination pose.

of interest.

Buttons on the controllers allow operators to add and delete waypoints (Fig. 5), change waypoint reference frames, and reset waypoints to reference frame origins. Operators can grab waypoints with the controllers and move them in all six degrees of freedom. Using an alternate grab button, operators can translate waypoints along a single axis at a time relative to the selected reference frame. For precise translational adjustment of waypoint position, operators can scale the world to "zoom in" before grabbing waypoints, which allows large physical motion of the hand controllers to be mapped to an arbitrarily small virtual movement of waypoints. Operators can view the position and rotation of waypoints using the virtual tooltip which appears above the selected waypoint. The sequence of waypoints in the motion plan can be viewed in the waypoint list window. The ability, described above, to precisely move each waypoint with respect to a selected reference frame solves the first limitation of our prior implementation on the da Vinci console (Section III).

D. Interactive Planning: Motion Preview and Execution

After creating an IPSE motion plan, operators can use buttons and joysticks on the hand controllers to preview the robot motion and execute the motion plan on the remote robot. The VR environment always displays the robot model at its actual location, based on (time-delayed) telemetry feedback. The preview option moves a second robot model (the preview robot) through each waypoint in the motion plan, enabling operators to determine whether the motion plan is acceptable or whether the plan must be revised. The interface also includes a "scrubber" bar, enabling operators to scroll through the plan at any speed. Plans are calculated and stored in robot joint space, so operators can be confident that the actual robot motion will match the previewed motion.

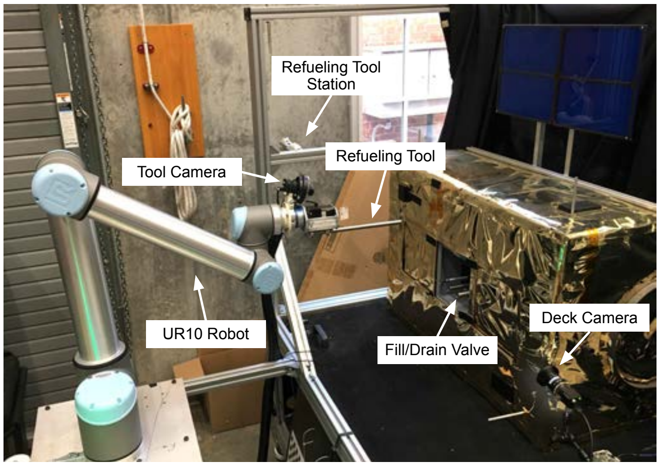

Fig. 6. The space-side setup includes a mock satellite (right), mock servicing robot (left), and refueling tool station (center).

Once satisfied with the plans, operators can choose the execution option to send the motion plan to the remote robot. In this case, the robot model will follow the motion plan, subject to the telemetry delay. A heads-up display in the virtual world contains a progress bar to indicate execution progress. During execution, the camera views will be shown as AV projections on the models, as in Figs. 3 and 4, and optionally on separate 2D virtual windows, as in Fig. 4. The availability of the AV visualization in the VR environment solves the second limitation of our prior implementation on the da Vinci console (Section III), which lacked this capability and therefore required operators to leave the console and view the visualization on a 3D monitor with shutter glasses.

V. Experiments

A. Experimental Setup

The experimental setup, as in [9], consists of a mock satellite, mock servicing robot, and operator station. The mock satellite, shown at the right of Fig. 6, is constructed from 80/20 aluminum bars and panels wrapped in a layer of Mylar [16]. Not including its solar panel, the satellite is a box of size $24 \times 24 \times 36$ inches. A set of three pipes of varying diameters is fixed within a cavity on one side of the satellite; one of these pipes is selected to represent the satellite fill/drain valve. A tool stowage area, visible behind the mock servicing robot in Fig. 6, is attached to the same structure as the mock satellite. This setup is identical to the one shown in [9], except for an improved design of the tool holder in the tool stowage area.

The mock servicing robot, shown at the left of Fig. 6, is a Universal Robots UR-10. A custom tool including a camera, light ring, and tool mounting assembly is mounted to the UR-10. The tool mounting assembly includes a motor to which active tools can be mounted (not used in this experiment) and a set of magnets to which passive tools can be attached. The refueling tool, a section of pipe affixed to magnets, is shown attached to the tool mount. The outer diameter of the refueling tool fits snugly into the smallest pipe on the mock satellite. At the base of the servicing robot is a deck camera with a stationary view of both the tool stowage area and the pipe cavity on the mock satellite.

The operator station is located in a different room, on a different floor of the building, to reproduce the remote teleoperation experience. The conventional teleoperation interface consists of four monitors (one of which is a 3D monitor), a keyboard, standard mouse, and a 3D mouse, as in our prior experiments [9]. It includes one new feature, which leverages the software described in Section IV-B to incorporate AV visualization in rviz, in addition to displaying it on the 3D monitor as in our prior implementation. The experimental teleoperation interface consists of the Meta Quest 2 and hand controllers, as described in Section IV. A software-created 5 second delay is added to emulate the multi-second delay expected between the ground station and on-orbit robot.

B. Experimental Task

An experiment begins with the mock satellite already registered using the Vision Assistant software tool described in [9]. The robot begins in a standard location not near the mock satellite or tool stowage area and the refueling tool begins in the tool stowage area. During the experiment, the operator must first command the robot to the tool stowage area and lower it onto the tool to engage the magnetic attachment. Once attached, the operator must then command the robot to move the tip of the refueling tool inside one of the pipes on the mock satellite (which represents the fill/drain valve). The experiment is complete when the operator believes the tool tip is inserted at least 3 cm inside the pipe, or failed if the operator believes it is not possible to insert the tool tip into the pipe (for example, if the tool is knocked off the magnetic mount).

C. Conditions

Each operator performs the experimental task twice, once with each experimental condition. The order of the conditions was randomly selected for each participant.

Both experimental conditions use the same virtual environment, planning and execution engine, and satellite registration. The registration error was estimated to be about 3-5 mm, which is large enough to require operators to visually confirm (on the camera images) alignment of the refueling tool with the pipe, rather than relying on the VR environment. Tool pickup was not adversely affected because the magnetic attachment tolerated that amount of registration error.

In the test condition, the subject interacts with the planning and execution engine using the VR interface described in Section IV. In the control condition, the subject uses the 2D interface previously described in [9].

VI. Results

We evaluated the system with an IRB-approved user study (HIRB00000701) consisting of nine operators all of whom self-reported as “experienced” or “familiar” with robotic teleoperation. This reflects the intended training level of the operators of this system in a real-life environment.

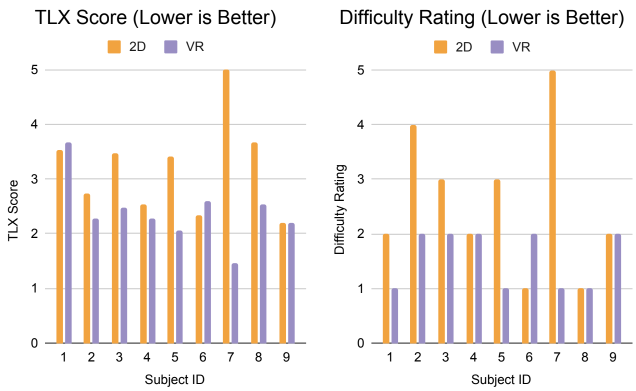

Fig. 7. NASA TLX results (left) and operator difficulty ratings (right) show that most operators preferred the VR interface, and those who did not only had a slight preference against it.

The most important metric for a teleoperation system in a high-risk environment with a high cost of failure is task success. All users successfully completed both portions of the experimental task under both interfaces. Two users of the proposed VR interface and one user of the baseline 2D interface misaligned the tool after contact with the environment, but all successfully performed the task using the misaligned tool. All three users who misaligned the tool did so on their first trial, suggesting that additional practice mitigated this outcome.

We also recorded the results of a NASA TLX survey administered after each experimental condition. TLX results are reported in Fig. 7. The TLX showed a preference for the proposed VR interface (p=0.070), with four users rating it as more favorable by at least one full TLX point. Those who did not favor the VR interface rated it either equal (one user) or only slightly worse than the baseline 2D interface. The average TLX score was 3.21 for the 2D interface and 2.39 for the VR interface, with standard deviations 0.87 and 0.58 respectively.

Additionally, users were asked to rate the difficulty of using each interface on a scale from 1 (Very Easy) to 5 (Very Hard). Users reported an average of 1.56 with the VR interface compared to 2.56 with the 2D interface. This also shows a preference for the proposed VR interface (p=0.081).

Task duration, shown in Table I, for the two conditions also favored the VR interface. While task duration is much less important than task success for the application we consider, increased operating time can contribute to the operators' mental and physical load. The decrease in task time by 22% and 25% respectively for the two segments of the task may be partially responsible for the improved TLX score for the VR interface, although this is a weaker result (p=0.242 and 0.214 respectively). Additionally, decreased time for operations without corresponding increases in task failure rate improves the overall productivity.

To further evaluate successful execution of the experimental task, we recorded the robot's position and orientation at the end of each segment of the task. Due to imperfect registration between the robot and the satellite, the absolute position

TABLE I
TASK DURATION (SECONDS)

	2D Mean	2D Std.Dev.	VR Mean	VR Std.Dev.
Tool Pickup	250	158	198	101
Refueling	740	460	575	346

TABLE II
VARIANCE IN POSITION (mm^2) AND ORIENTATION (deg^2)

	2D		VR	
	Pos.	Ori.	Pos.	Ori.
Tool Pickup	0.84	1.53	0.70	1.00
Refueling	0.66	0.29	0.60	0.25

and orientation are not meaningful. However, the variance reflects the consistency operators were able to achieve with each interface. These results, shown in Table II, indicate no significant difference between the two interfaces. Cases where the tool became misaligned are excluded from the calculation as the true tool position is not known.

VII. Discussion and Conclusions

We developed a virtual reality interface for the Interactive Planning and Supervised Execution (IPSE) robotic planning system as an alternative to the previous keyboard and mouse interface. The virtual reality interface was implemented for the Meta Quest 2 standalone headset and 6-DOF hand controllers. While using the headset, operators can create, preview, and execute motion plans on satellite servicing robots while monitoring execution progress with Augmented Virtuality overlays and 2D camera feeds.

The system was evaluated against the baseline keyboard and mouse 2D interface through a user study with nine operators, all of whom rated themselves as "experienced" or "familiar" with teleoperation systems. We found significant indication that the VR interface led to a lower operator workload as compared to the 2D interface presented in [9], without any indication that the VR interface lowers performance. This improves over the VR interface presented in that work (there termed "3D interface"), which was shown to have worse outcomes and higher workload. In addition, the VR interface we implemented requires only a commercial off-the-shelf virtual reality headset and a standard personal computer, whereas the previous 3D interface used a highly specialized da Vinci master console. The 2D interface used previously and in this work contains four monitors, one of which is a stereo display. Thus, the VR interface also offers portability and economy of space. These advantages were not offset by any loss in performance and were accompanied by an improvement in operator workload. Future work may include experiments with alternative headsets and input devices.

Acknowledgments

Anna Goodridge designed and built the tool holder in the stowage area. Nicholas Greene, Cynthia Li and Haochen Wei assisted with the experimental setup and user study.

References

[1] Z. Ai, M. A. Livingston, and I. S. Moskowitz, "Real-time unmanned aerial vehicle 3D environment exploration in a mixed reality environment," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 664–670.

[2] J. Chong, S. Ong, A. Nee, and K. Youcef-Youmi, "Robot programming using augmented reality: An interactive method for planning collision-free paths," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 689–701, 2009.

[3] G. d. M. Costa, M. R. Petry, and A. P. Moreira, "Augmented reality for human-robot collaboration and cooperation in industrial applications: A systematic literature review," *Sensors*, vol. 22, no. 7, 2022.

[4] F. De Pace, F. Manuri, A. Sanna, and C. Fornaro, "A systematic review of augmented reality interfaces for collaborative industrial robots," *Computers & Industrial Engineering*, vol. 149, p. 106806, 2020.

[5] H. Fang, S. Ong, and A. Nee, "Interactive robot trajectory planning and simulation using augmented reality," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 227–237, 2012.

[6] D.-H. Kim, Y.-G. Go, and S.-M. Choi, "An aerial mixed-reality environment for first-person-view drone flying," *Applied Sciences*, vol. 10, no. 16, 2020.

[7] S. Ong, A. Yew, N. Thanigaivel, and A. Nee, "Augmented reality-assisted robot programming system for industrial applications," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101820, 2020.

[8] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE, 2010, pp. 1–8.

[9] W. Pryor, B. Vagvolgyi, A. Deguet, S. Leonard, L. Whitcomb, and P. Kazanzides, "Interactive planning and supervised execution for high-risk, high-latency teleoperation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 10 2020, pp. 1857–1864.

[10] W. Pryor, B. P. Vagvolgyi, W. J. Gallagher, A. Deguet, S. Leonard, L. L. Whitcomb, and P. Kazanzides, "Experimental evaluation of teleoperation interfaces for cutting of satellite insulation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2019, pp. 4775–4781.

[11] C. P. Quintero, S. Li, M. K. Pan, W. P. Chan, H. M. Van der Loos, and E. Croft, "Robot programming through augmented trajectories in augmented reality," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 1838–1844.

[12] N. Smolyanskiy and M. Gonzalez-Franco, "Stereoscopic first person view system for drone navigation," *Frontiers in Robotics and AI*, vol. 4, 2017.

[13] Stanford Artificial Intelligence Laboratory et al., "Robot operating system." [Online]. Available: https://www.ros.org

[14] M. Sun, N. Dong, C. Jiang, X. Ren, and L. Liu, "Real-time MUAV video augmentation with geo-information for remote monitoring," in *Fifth International Conference on Geo-Information Technologies for Natural Disaster Management*, 2013, pp. 114–118.

[15] Unity Technologies, "Unity Robotics Hub." [Online]. Available: https://github.com/Unity-Technologies/Unity-Robotics-Hub

[16] B. Vagvolgyi, W. Niu, Z. Chen, P. Wilkening, and P. Kazanzides, "Augmented virtuality for model-based teleoperation," in *IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, Sep 2017.

[17] B. Vagvolgyi, W. Pryor, R. Reedy, W. Niu, A. Deguet, L. Whitcomb, S. Leonard, and P. Kazanzides, "Scene modeling and augmented virtuality interface for telerobotic satellite servicing," *IEEE Robotics & Automation Letters*, vol. 3, no. 4, pp. 4241–4248, Oct. 2018.

[18] M. Wonsick, T. Keleştemur, S. Alt, and T. Padır, "Telemanipulation via virtual reality interfaces with enhanced environment models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 2999–3004.

[19] L. S. Yim, Q. T. Vo, C.-I. Huang, C.-R. Wang, W. McQueary, H.-C. Wang, H. Huang, and L.-F. Yu, "WFH-VR: teleoperating a robot arm to set a dining table across the globe via virtual reality," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2022, pp. 4927–4934.

[20] J. Yu, T. Wang, Y. Shi, and L. Yang, "MR meets robotics: A review of mixed reality technology in robotics," in *6th International Conference on Robotics, Control and Automation (ICRCA)*, 2022, pp. 11–17.

[21] S. Zollmann, C. Hoppe, T. Langlotz, and G. Reitmayr, "FlyAR: augmented reality supported micro aerial vehicle navigation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 4, pp. 560–568, 2014.