Augmented Reality-based Robotic System for In-space Servicing

Francesco Garcia-Luna, *Member, IEEE,* Alma Rodriguez-Ramirez, *Member, IEEE,* Manuel Nandayapa, *Member, IEEE,* and Angel Flores-Abad, *Member, IEEE*.

***Abstract*—Space assets like meteorological stations, communication satellites, and microgravity research stations are essential for human knowledge development. Nevertheless, space assets are subject to failure, damage, or obsolescence in their life cycle. A space asset's failure, damage, or obsolescence is typically addressed using teleoperated space robot systems to diagnose, repair, or upgrade it. In these activities, the region of interest or components to be serviced may be visually occluded by light or a physical obstacle, complicating the teleoperated maneuver and compromising the mission's success and safety as a collision may occur due to blind spots generated by visual occlusions. This paper presents an Augmented Reality (AR)-based robotic framework that allows the robot to perform an on-orbit servicing task despite visually occluded areas. It allows the user to dynamically obtain the best view in a 3D (three-dimensional) model. Besides, for the robot tool to accurately reproduce the natural motion of a human operator's hand, a direct hand-presence device is used to directly map the avatar's hand motion to the robot's end-effector (EE) motion. The system is validated in an AR environment with virtual and physical entities to repair a spacecraft's solar panel in a visually occluded area for the robot. The experimental results demonstrate that because the virtual environment can be manipulated in real-time to show the best perspective to the human operator, the repair trajectory was generated without compromising the safety and operations, even though the robot's EE and the camera-in-hand were not able to observe the area of interest directly.**

***Index Terms*—Visual Occlusion, Teleoperation, Augmented Reality, Low Earth Orbit, Direct Hand-Presence.**

I. Introduction

ROBOTS are key supporters and enablers of different endeavors in space, such as On-orbit Servicing, Assembly, Manufacturing [1] and the Lunar Gateway program [2]. An ultimate goal in space robotics is to have the robots working fully autonomously; however, existing technical limitations and risk tolerances keep this as a long-term goal [3]. For instance, the Space Station Remote Manipulator System (SSRMS), known as "Canadarm2," performs frequent teleoperated maneuvers to provide station maintenance, support extravehicular activities, and grapple visiting vehicles to berth them to the International Space Station (ISS) [4]. A common challenge in the remote manipulation of space robots is providing sufficient situational awareness to the human operator [5]. Teleoperators must choose an appropriate split between available fixed cameras mounted on the servicing satellite and tools mounted close to the robot's end-effector (EE), which does not always provide an appropriate view of the work site. Often, the components or regions that need to be accessed are partially or entirely occluded, making it difficult or impossible to perform a servicing operation.

Different systems have been introduced to increase situational awareness during a robotic servicing mission. Tian Xia et al. [6] proposed an augmented reality (AR) framework where a virtual fixture gives the human operator immediate visual feedback and motion guidance. Kazanzides et al. [7] [8] developed an augmented reality-based approach that combines real information in the local and remote environment so that the operator can identify the Region of Interest (RoI) in the virtual model to feed spatial information to the robot and facilitate the repair task in areas with physical and or visual occlusion.

It has been found that AR is a powerful tool for reconstructing RoI. For instance, in a medical context, AR has been used to overlay middle ear cleft structures and potential middle ear targets over a visually occluded endoscopic view [9], allowing the surgeon to perform a minimally invasive cranial-base surgery; in data visualization, AR has been used to represent multi-variate and multi-dimensional data visually with a 3D radar chart while avoiding data occlusion [10]; or in a robotics context, to dynamically monitor a digital twin in AR by superimposing the virtual data and images onto the camera, when the camera is pointing at the physical twin [11], [12].

Here, we present an AR-based robotic teleoperation approach to provide the human operator with a 3D (Three-dimensional) dynamical view of the target spacecraft to select the best perspective and view of the RoI so that the repair path can be planned appropriately despite the presence of occluded zones. Therefore, the teleoperator is able to manipulate the 3D model in first person to guide the robot and avoid undesired collisions or excessive force exertion. Besides, typically, teleoperation systems feature joysticks-like interfaces [13] to manipulate the robot's EE or tool. However, such a mechanism does not correctly map the motions of the hand's operator and the robotic EE, leading to a misunderstanding of the robot's tool orientation and a reduced immersion experience [14]. This project uses a direct hand-presence device to map more directly between the human's and robot's motion. Besides, these devices estimate the hands' pose by combining external and internal sensors. The device used in this project was a

F. Garcia-Luna, A. Rodriguez-Ramirez and M. Nandayapa are with the Department of Industrial Engineering and Manufacturing, Autonomous University of Ciudad Juarez, Juarez, CHI, 32310 MX e-mail: francesco.garcia@uacj.mx, alma.rodriguez.ram@uacj.mx, and mnandaya@uacj.mx, respectively.

A. Flores-Abad is with the Aerospace and Mechanical Engineering Department and with the Aerospace Center of the University of Texas at El Paso, El Paso, TX 79968 USA e-mail: afloresabad@utep.edu.

Meta Quest 2 Controller, which is reported to have an accuracy of 4.36 ± 2.91 mm [15], [16]. The overall proposed framework here is validated in an AR environment with real and virtual entities for the case of repairing a structurally damaged solar panel using a free-floating space robotic system.

Based on the current literature and technology, the contributions of the paper are identified as follows:

1) A baseline framework using virtual reality tools to support in-space robotic teleoperated tasks.
2) An augmented reality-based system that enables robots in space to work in visually occluded zones despite the region of interest is not visible for the robot to work.
3) The real-time implementation and evaluation of the servicing tasks employing virtual and physical entities.
4) The introduction and evaluation of a hand-presence device to enhance the mapping between the avatar's hand motion and the robot's end-effector.

This paper is structured as follows: In Section I, a brief introduction is presented, where all the related concepts are described, along with the identified problems, and the relevant works are also included. In Section II, the methodology used to solve the problem through the design of a novel AR system that integrates the human's direct hand motions, a robust communication network, and the teleoperation task are described. Later, in Section III, the teleoperation and performance results are shown. Lastly, in Section V, the conclusions and future work are described.

II. Technical Approach

A. Augmented Reality-Based Robotic Servicing System

Traditional teleoperation systems consist of three main components: a human operator that manipulates a primary hand-presence device and a secondary robot designed to act in response to the primary device that captures the motion premises from the human operator [17]. Thus, a teleoperation setup allows an operator to remotely control a robot and perform tasks in remote and sometimes hazardous environments. For instance, in outer space, where the direct presence of the human represents a risk or is not always feasible, a teleoperation setup is required.

Accordingly, as depicted in Figure 1, the AR-based servicing system proposed in this work is separated into local, virtual, and remote environments. The local environment (primary system) corresponds to that where the human operator is located and includes the operator with a head-mounted display (HMD) and a hand-presence device; the virtual environment is a virtualized model of the target spacecraft; and the remote environment includes the entities residing in orbit, namely, the servicing space robot and the real (not virtual) target satellite. In this study, the real servicing space robot and the target satellite are simulated entities. The remote environment shares the estimated target's pose with respect to the service robot frame, as well as the EE's pose. Then, the local environment receives and depicts the information in the AR setup. Subsequently, the local environment commands the servicing space robot (living in the remote environment) based on the input provided by the human operator via the hand-presence device.

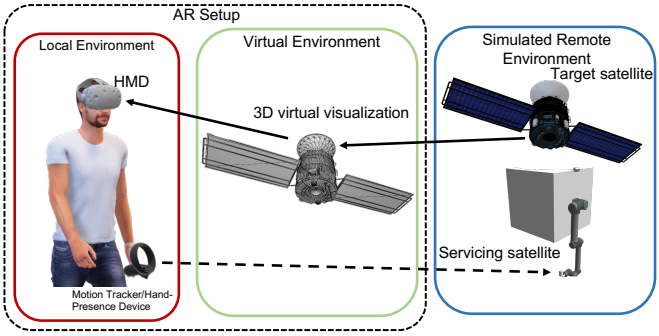

Fig. 1. AR Servicing System consisting of an AR local environment where the target satellite's 3D virtual representation is rendered and a simulated remote environment where the servicing satellite receives the desired trajectory points.

In the local environment, a 3D virtual model of the target satellite is always projected in front of the human operator. The human has the luxury of obtaining the best perspective of the RoI in the virtual model to design the repairing trajectory. Once the teleoperator identifies the trajectory, the occlusion-free position and orientation waypoints can be stored in an array for semi-autonomous pose control. The teleoperator can add, remove or clear all waypoints to improve the trajectory at any moment. When the teleoperator is satisfied with the trajectory, the teleoperator sends the trajectory to the remote robot. During the execution, the teleoperator receives AR visual feedback informing the robot's EE pose with respect to the target satellite.

B. Visual Occlusion

According to [18], a visual occlusion happens when an object interferes entirely or partially in the line of sight between the observer and the RoI, as seen in Figure 3.

In space servicing, a visual occlusion between a camera´s Point of View (POV) and the RoI may occur when sunlight does not directly hit the RoI´s surface, making the RoI invisible or when the RoI is nos visibly accessible due to the physical presence of an object between the camera's POV and the RoI. . This research considers the case where the RoI is visually inaccessible as it is located behind a solar panel.

C. Methodology

According to the methodology diagram proposed in Fig. 2, the simulation environment is launched using a physics-based engine in the Gazebo robotic simulation environment. Then, space environment parameters like microgravity, lighting, interaction physics, background color, and viewports are loaded into the physics engine. The target satellite and the servicing space robot's 3D model representations are loaded and positioned with respect to an inertial frame. The target satellite is then re-positioned with respect to the robot satellite.

Once the remote environment representation is set, the avatar node is launched. This node allows the physics engine to render the estimated hand representation pose. Then, a node that dictates the rigid transformations between all agents is launched. The node also keeps track of the relative transformations between all the frames in the environment.

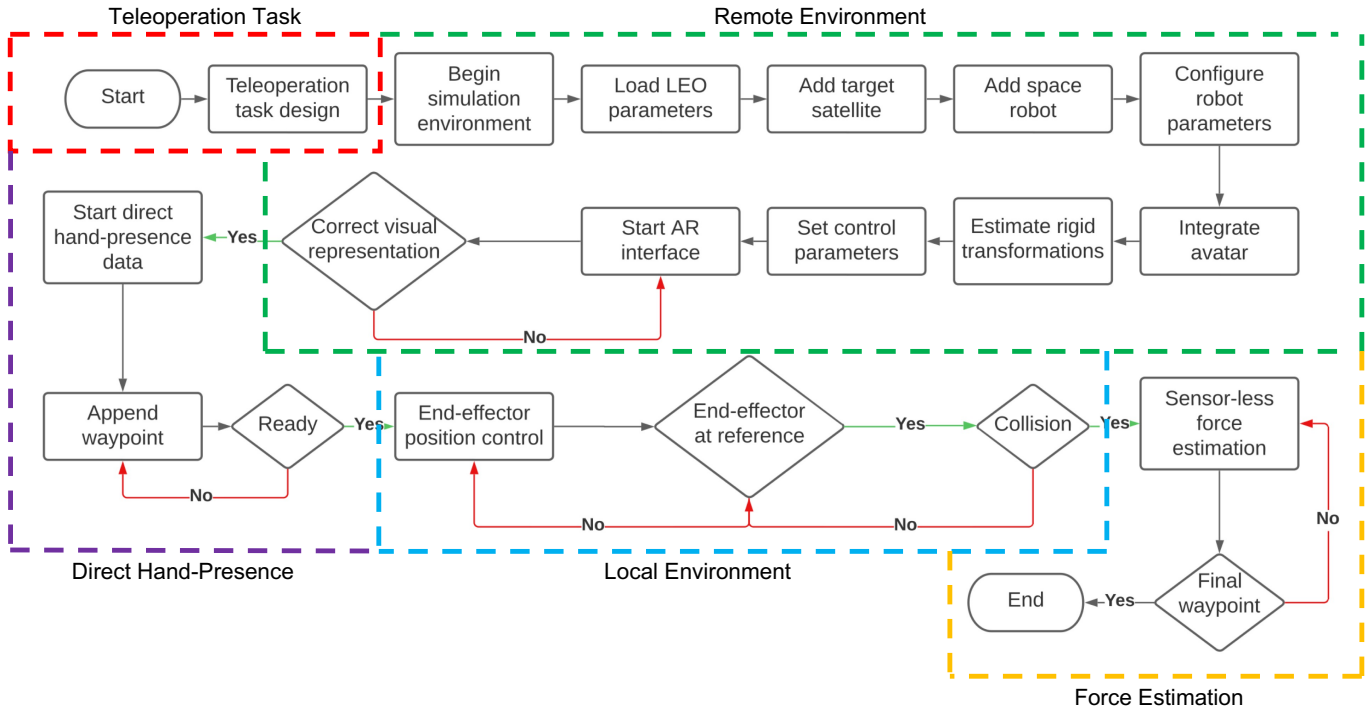

Fig. 2. Methodology used for dealing with occlusions using an AR system with direct hand presence. It is divided into five steps: Teleoperation Task, where the task constraints are designed; Remote Environment, where the simulated actors are located; Direct hand presence, where the control algorithm maps the user's hand movement directly to the end-effector's relative movements; Local Environment where the AR setup enhances the user's perspective by means of rendering the simulated data; and Force Estimation where the contact force is estimated without a sensor.

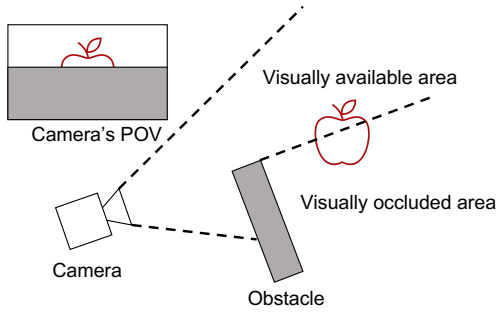

Fig. 3. Representation of partial visual occlusion.

With the avatar and transformations nodes running, the control node is launched. This node uses a basic PID pose controller that receives a set of waypoints and designs a valid trajectory. Once the trajectory is set and the teleoperator considers it correct, the node sends the control commands to the robot's end-effector.

Meanwhile, the AR interface is rendered in the teleoperator HMD in the local environment. Here, crucial mission information like an occlusion-free target satellite 3D model representation, end-effector's pose, RoI highlight, estimated trajectory, and an estimated contact force are rendered. Then, the reference pose is published over the network.

The teleoperator visualizes the occlusion-free target satellite 3D model representation, and the RoI is highlighted. Then, it hovers the hand over the RoI to append the waypoint, and when there are at least two waypoints, a linear trajectory is calculated. Once the trajectory meets the task requirements, the teleoperator sends it to the control node.

Besides, a control node executes a pose regulation for every point in the trajectory. When a waypoint is detected as a collision, a sensorless force estimation-based algorithm estimates the contact force and displays it in AR.

1) Teleoperation Task: The first step in the methodology consists of defining the teleportation task. This work considers the scenario where the solar panel of a satellite has suffered structural damage, a crack, for instance, which, if not repaired, may spread over time and affect other satellite subsystems. Then, an adhesive repair paste needs to be deposited on the panel to fix it and avoid a more significant problem and failure propagation.

The proposed robotic servicing mission consists of driving the robot's EE with an eye-in-hand camera through a repair trajectory defined by a human operator. The human will be in a remote environment but will have the best repair perspective to guide the robot due to the VR setup of the target satellite.

The end-to-end servicing mission consists of four phases, as depicted in Figure 4. In the preparation phase, the robot's EE safely approaches a position close to the initial point and orientates the tool perpendicular to the panel to get ready. Then, in the approaching phase, the operator commands the robot's EE to move to the first waypoint over the RoI and make the first contact, rendered in an AR environment. Then, in the servicing phase, a trajectory control algorithm is used to command the robot's EE through a set of human-drive

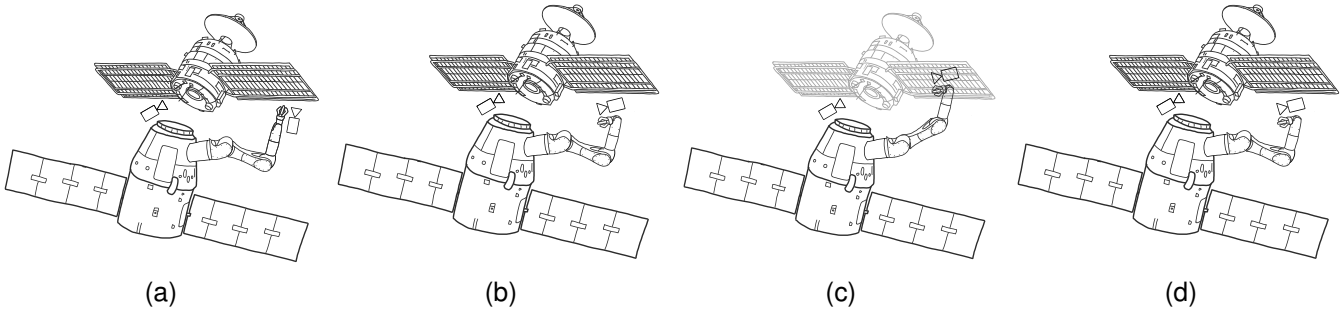

Fig. 4. Space task. Preparation Phase (4a): The end-effector moves from the home position to the initial pose. Reorientation Phase (4b): The end-effector moves and makes a first contact. Servicing Phase (4c): The end-effector moves along the user-defined trajectory. Separation Phase (4d): The end-effector moves from the last waypoint to the initial pose.

waypoints on an occluded region. Finally, in the separation phase, the robot removes the end-effector from the RoI, moves to the initial articular configuration, and renders the visual information in the AR environment.

2) Remote Environment Design: The remote subsystem is intended to reside in outer space; therefore, the physics of the outer space environment, such as reduced gravity, lighting conditions, and visual representations, are used to include the space servicing robot and the target satellite (see Fig. 5).

The serviced satellite is a communication satellite with two solar cell arrays. Meanwhile, the service robot operates in free-floating mode with a 6-DOF robotic arm mounted on a base satellite and two RGB-D cameras.

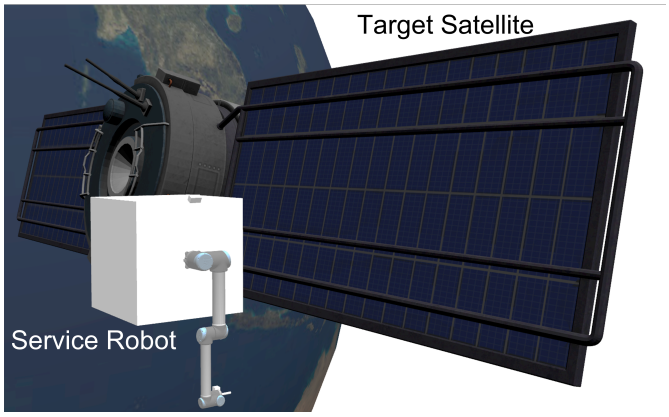

Fig. 5. Simulated Remote Environment depicting the target satellite and the service robot.

3) Kinematics: A multi-body diagram of the servicing system is shown in Figure 6, where $\vec{p}_0$ is the vector from the base satellite's CM to the Inertial Frame, $\vec{p}_1$ is the vector from the inertial frame to the robot's EE. Besides, Σ_w, Σ_b, and Σ_m are the coordinates frame of the world, the base satellite, and the robot's EE, respectively.

Also, the EE's position vector $\vec{p}_e \in \mathbb{R}^3$ can be written in inertial space as

$$\vec{p}_e = \vec{p}_0 + \vec{p}_1 + \sum_{i=0}^{n} \vec{p}_i \tag{1}$$

where $\vec{p}_0 \in \mathbb{R}^3$, $\vec{p}_1 \in \mathbb{R}^3$ and $\vec{p}_i \in \mathbb{R}^3$ are the position vector from the CM (Center of Mass) of the base satellite to the inertial frame, the CM of the base satellite to joint 1, and from joint i to joint $i+1$, respectively.

Differentiating (1) with respect time, the relationship between the EE's linear velocity $\vec{\nu}_{ee} \in \mathbb{R}^3$ and joint velocities $\dot{q}_i$ is obtained (described in eq. 2), which is also a function of space robot's angular velocity $\vec{\omega}_0 \in \mathbb{R}^3$ (described in eq. 3) and its linear velocity $\vec{\nu}_{ee} \in \mathbb{R}^3$, the joint axis of rotation $\vec{z}_i \in \mathbb{R}^3$ and the vectors $\vec{\rho}_i \in \mathbb{R}^3$ from the inertial frame to the i-th joint CM.

$$\vec{\nu}_e = \vec{\nu}_0 + \vec{\omega}_0 \times (\vec{p}_e - \vec{p}_0) + \sum_{i=1}^{n} \left[\vec{z}_i \times (\vec{p}_e - \vec{p}_i)\right] \dot{q}_i \tag{2}$$

$$\vec{\omega}_e = \vec{\omega}_0 + \sum_{i=1}^{n} \vec{z}_i \dot{q}_i \tag{3}$$

Besides, the EE's orientation is given by the Euler angles vector $\gamma = [\phi, \theta, \psi]^T$ in the x-y-z convention.

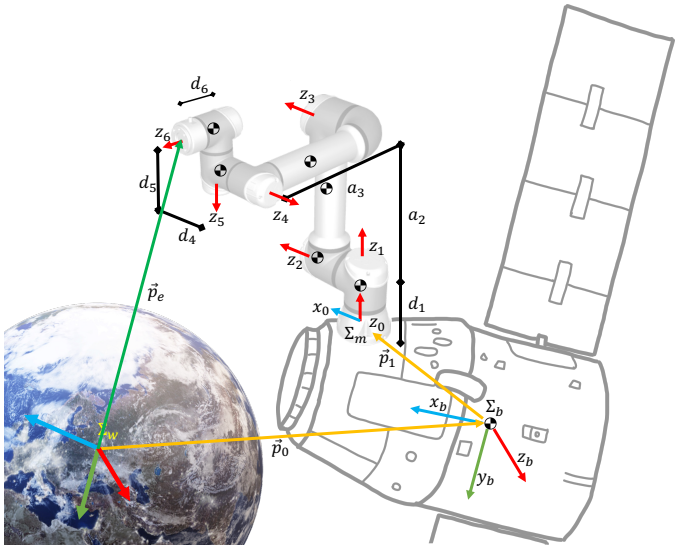

Fig. 6. Rigid Multibody Dynamic System of a remote base satellite for On-Orbit Servicing, where Σ_i is the i-th reference frame, $\vec{p}_0 \in \mathbb{R}^3$ is the base satellite's position vector with respect an inertial frame, and $\vec{p}_e \in \mathbb{R}^3$ is the end-effector position's vector with respect the inertial frame.

TABLE I
NOMENCLATURE DESCRIPTION

Symbol	Description
$M_b \in \mathbb{R}^{m\times m}$	Base Satellite's Inertia Matrix
$M_m \in \mathbb{R}^{n\times n}$	Robot Manipulator's Inertia matrix
$M_{bm} \in \mathbb{R}^{m\times n}$	System's Inertia Matrix
$\ddot{\vec{\phi}} \in \mathbb{R}^m$	Base Satellite's angular accelerations vector
$\ddot{\vec{\theta}} \in \mathbb{R}^n$	Robot Manipulator's angular accelerations vector
$C_b \in \mathbb{R}^{m\times m}$	Base Satellite's Coriolis Matrix
$C_m \in \mathbb{R}^{n\times n}$	Robot Manipulator's Coriolis Matrix
$C_{bm} \in \mathbb{R}^{m\times n}$	System's Coriolis Matrix
$m \in \mathbb{R}$	Base Satellite's rotational DoF
$n \in \mathbb{R}$	Robot Manipulator's rotational DoF
$\vec{f_b} \in \mathbb{R}^m$	Base Satellite's control forces exerted on its CM
$\vec{\tau_c} \in \mathbb{R}^n$	Robot Manipulator's joint torques
$\vec{f_e} \in \mathbb{R}^3$	Forces and moments exerted on the end-effector
$J_b \in \mathbb{R}^{n\times m}$	Base Satellite's Jacobian Matrix
$J_m \in \mathbb{R}^{n\times n}$	Robot Manipulator's Jacobian Matrix

4) Dynamics: In the absence of gravitational and non-conservative forces, the general form of a free-floating system in space can be written as Eq. (4)

$$M(\vec{q})\ddot{\vec{q}} + C(\vec{q},\dot{\vec{q}})\dot{\vec{q}} = \vec{\tau} \tag{4}$$

where $M(\vec{q}) \in \mathbb{R}^{(n+3)\times(n+3)}$ is the inertia matrix, $C(\vec{q},\dot{\vec{q}}) \in \mathbb{R}^{(n+3)\times(n+3)}$ is the Coriolis matrix which includes the centripetal and centrifugal forces, $\vec{q} \in \mathbb{R}^{n+3}$ is the vector of generalized coordinates that include the attitude of the free-floating base $\vec{\phi} \in \mathbb{R}^3$, $\vec{\theta} \in \mathbb{R}^n$ the joint displacements, and $\vec{\tau}$ is the vector of generalized forces including torques and axial forces.

The general equation (4) can be decomposed to separate the influence of the robot and the base satellite in the system as follows [19],

$$\begin{bmatrix} M_b & M_{bm} \\ M_{bm}^T & M_m \end{bmatrix} \begin{bmatrix} \ddot{\vec{\phi}} \\ \ddot{\vec{\theta}} \end{bmatrix} + \begin{bmatrix} C_b & C_{bm} \\ C_{bm} & C_m \end{bmatrix} \begin{bmatrix} \dot{\vec{\phi}} \\ \dot{\vec{\theta}} \end{bmatrix} = \begin{bmatrix} \vec{f_b} \\ \vec{\tau_c} \end{bmatrix} + \begin{bmatrix} J_b^T \\ J_m^T \end{bmatrix} \vec{f_e} \tag{5}$$

which nomenclature is described in TABLE I

5) Direct Hand Presence: The integration of a motion-tracking technology that accurately captures the hand's range of motion and augmented reality into the primary device allows the human to know at every moment the approximate EE position and orientation in space.

Here, the primary device consists of an AR HMD and a direct hand-presence device that measures relative movements between the marker and the HMD. The remote servicing robot has two cameras: a primary (main) camera with a fixed perspective on the front top of the robot and an eye-in-hand camera mounted on top of the end effector. Hence, the kinematics model between the remote servicing robot and the human arm can be mapped seamlessly because the base satellite's pose is mapped to the human shoulder pose, the remote end-effector pose maps to the human hand pose, and the HMD pose maps to the remote robot main camera pose (depicted in Fig. 7). Also, these maps mean that a coincident frame exists on each relation.

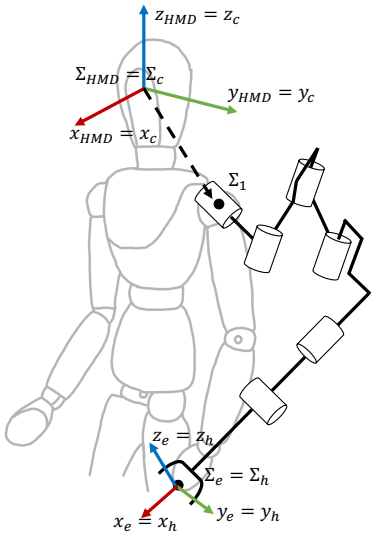

Fig. 7. Kinematic relations between a human and the remote servicing robot.

A control algorithm was developed for the interfaces, allowing the operator to control both the position and orientation of the servicing robot's end-effector in all six axes, mapping the interface events to a different motion, as described in Table II.

TABLE II
TELEOPERATION PRIMARY DEVICE MOVEMENT MAPPING

End-effector reference	**Proposed system***
x_d	Hand Forward/Backward
y_d	Hand Left/Right
z_d	Hand Up/Down
ϕ_d	Wrist rotation over X axis
θ_d	Wrist rotation over Y axis
ψ_d	Wrist rotation over Z axis

6) Local Environment Design: The local environment was designed using a modified Unreal Engine 4 version that integrates the Stereolabs SDK into a depth-aware AR setup. It consists of a virtual camera attached to the HMD movements, a communication node that manages the bidirectional data, a 3D model of the target satellite, a visual representation of the occluded RoI that is always visible in the virtual model, and an auxiliary text placeholder that displays useful information about the task (see Fig. 8).

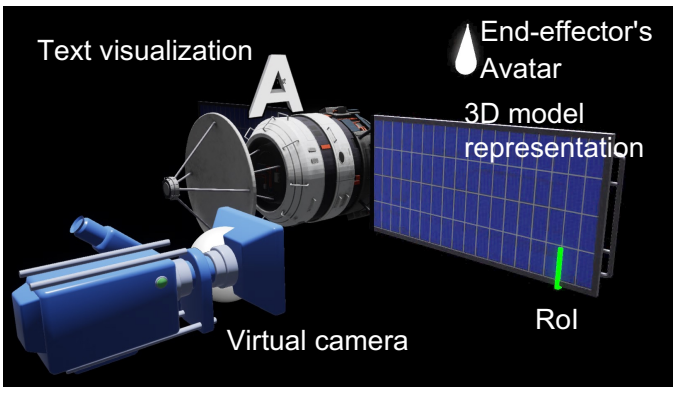

Fig. 8. Local environment for the AR setup. Where the virtual camera, the 3D model representation, RoI, the avatar, and a text visualization are rendered.

The augmented local target satellite's 3D model pose is represented as the relative pose between the remote target satellite and the remote servicing robot. Also, the remote end-effector pose takes the pose sent by the AR system within the local environment.

The AR system in the local environment consists of a human-in-the-loop teleoperation primary device that allows a human to use his own body as a position and orientation reference. It uses a VR headset with a stereo camera mounted as an HMD and a direct hand-tracking device that the user grasps with the hand.

The horizontal distance between the cameras is approximately the same distance a human has between its eyes, designed to minimize motion sickness. Also, the direct hand-tracking device allows tracking of both the hand's position and the hand's orientation with respect to the HMD on all three axes and maps them directly to the end EE's reference frame (see Fig. 9).

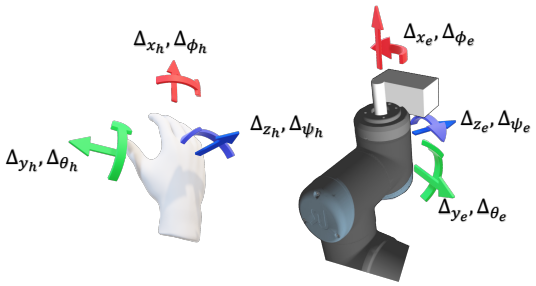

Fig. 9. Direct hand-presence primary device mapping, where the different Δs represent the local variation in position and orientation.

7) Force Determination: The contact force generated by the interaction between the avatar and the RoI is determined using a mass-spring-damper model (see Fig. 10) as

$$f = \begin{cases} kd + b\dot{d}, & \text{if } d < 0, \\ 0, & \text{otherwise.} \end{cases}$$

where $f \in \mathbb{R}$ is the estimated contact force, estimated from a mass-spring-damper model, $k \in \mathbb{R}$ is the spring constant, $b \in \mathbb{R}$ is the damping constant, and $d, \dot{d} \in \mathbb{R}$ are the separation/penetration distance and relative speed between the avatar and the contact surface, respectively.

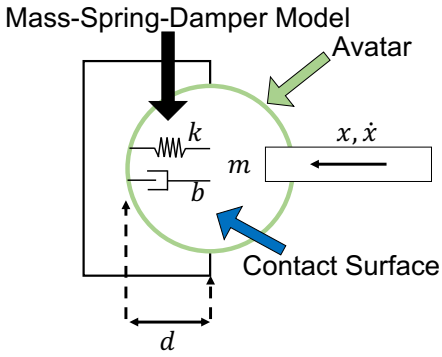

Fig. 10. Contact force estimation graphic representation using a mass-spring-damper model.

III. Experimental Setup

A. Actors

To validate the performance of the proposed approach, we consider the case of a damaged solar cell, whose technical specs are shown in Table III. Then, a robotic servicing mission is proposed to perform the deposition of a repair paste.

TABLE III
Solar Cell Array Technical Specifications

Characteristic	Specification
Length	$3m$
Width	$0.2m$
Height	$1m$
Surface Hooke Coefficient	40
Surface Damper Coefficient	0.2

The space robot (see Fig 11) is modeled as a cube-shaped base satellite with a fixed RGB camera on top and a rigid 6-DOF serial manipulator, with mass, geometric, and instrumentation aspects depicted in Table IV).

TABLE IV
Space Robot Technical Specifications

Characteristic	Specification
Base Satellite	
Shape	$1m^3$ cube
Mass	$163.5Kg$
Inertial Matrix	$I_{xx} = I_{yy} = I_{zz} = 27.25Kgm^2$
Instrumentation	
Principal Camera Sensor	RGB
Eye-in-Hand Camera Sensor	RGB
Field of View	$80H \times 80Vdeg$
Image Resolution	672×376 pixels
Update Frequency	$60Hz$
Robot Arm	
Manipulator Attached	UR10
Degrees of Freedom	6
Mass	$32.7Kg$
End-Effector	Cylinder-shaped custom Probe

Due to the nature of the task, the remote environment was simulated using Gazebo under Ubuntu 18.04 Operating System and ROS Melodic (the specifications are shown in Table V). To simulate outer space conditions, the gravity vector was set to zero, and the atmospheric model was deactivated to emulate a vacuum. At the same time, the lighting conditions include

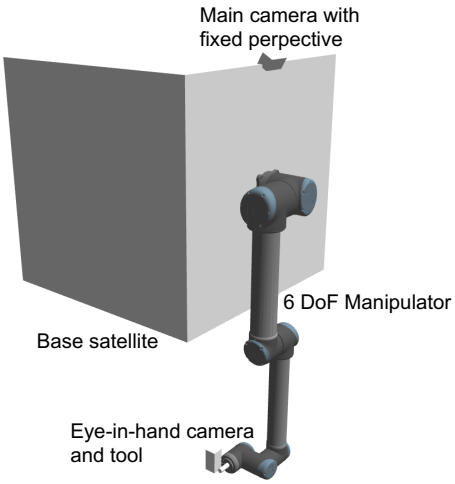

Fig. 11. Space robot design consisting of a base satellite, a rigid manipulator based on the UR10 configuration, a camera as the principal point of view, and an eye-in-hand camera.

a single source of directional light and a black background color.

Furthermore, the base satellite's CM is coincident with the origin at the beginning of the simulation; the target satellite is considered to be already in close proximity; therefore, no significant linear displacements of the servicing systems are required, and the effects of external disturbances such as the solar pressure and gravity gradient are also neglected.

TABLE V
PROCESSING MACHINES TECHNICAL SPECIFICATIONS

Characteristic	Specification
Remote Computer Operating System	Ubuntu 18.04.5 LTS
Remote Computer Processor	ARM64 M1-Max
Remote Computer RAM	32 GB
ROS Version	Melodic
Gazebo Version	9
Local Computer Operating System	Windows 10
Local Computer Processor	Intel i7-8700K
Local Computer RAM	32 GB
Graphics Card	RTX 2080
CUDA Cores	2944
Unreal Engine Version	Custom 4.21
CUDA Version	11.0
Stereolab's SDK Version	3.7

The servicing robot drives the servicing tool across the region of interest. The servicing robot is modeled as a UR10 rigid manipulator with a custom cylinder-shaped probe as the EE (the specifications are shown in Table VI).

TABLE VI
ROBOT ARM UR10 TECHNICAL SPECIFICATIONS

Characteristic	Specification
Weight	$28.9Kg$
Payload	$10Kg$
Reach	$1300mm$
Joint ranges	$\pm 180\,\mathrm{deg}$
Speed	$120\,\mathrm{deg}/s$-$180\,\mathrm{deg}/s$
Repeatability	$\pm 0.1mm$

The RGB cameras mounted in the space robot visualize the task from two different perspectives. Both cameras are modeled emulating a Stereolabs ZED mini RGB-D camera depicted in Table VII. The distance between the lens is approximately the mean distance a human has between the eyes. According to Stereolabs, this minimizes motion sickness and improves immersion in an AR environment.

TABLE VII
STEREOLABS ZED MINI TECHNICAL SPECIFICATIONS

Characteristic	Specification
Video Output	2x (1280x720) @60fps
Output Format	YUV 4:2:2
Field of View	90º (H) x 60º (V) x 100º (D)
RGB Sensor	1/3" 4MP CMOS
Focal Length	2.8mm - f/2.0
Baseline	63mm
Depth Range	0,10 m to 15m
Depth Accuracy	$\leq$ 1.5% up to 3m and $\leq$ 7% up to 15m
Motion Sensors	Gyroscope, Accelerometer @800Hz
6-axis Pose Accuracy	Position: $\pm$1mm, Orientation: 0.1 deg
Pose Update Rate	100Hz

B. Communication

To communicate the remote and the local environments with each other, a ROS network was designed as described in Fig. 12. The network manages the information flow to guarantee the publishing rate of every topic.

Gazebo is used as the ROS-compatible physics engine to simulate the remote environment and publishes into the ROS network information regarding the space robot's configuration. Then, UE4 receives and displays the space robot's configuration information in the HMD.

In the local environment, UE4 is used to receive depth information from the Stereolabs ZED mini, to increase user awareness, and the reference pose from the Oculus motion controllers. Then, UE4 sends the reference pose information to the control node.

The control node receives the reference pose and the base satellite's configuration and estimates a valid joint configuration to drive the simulated or real robot's end-effector to the reference pose.

Several ROS nodes were designed to manage all of the simulation data:

- Communication between local and remote environment
- AR primary device data acquisition
- 3D models manipulation
- Rigid Transforms publication

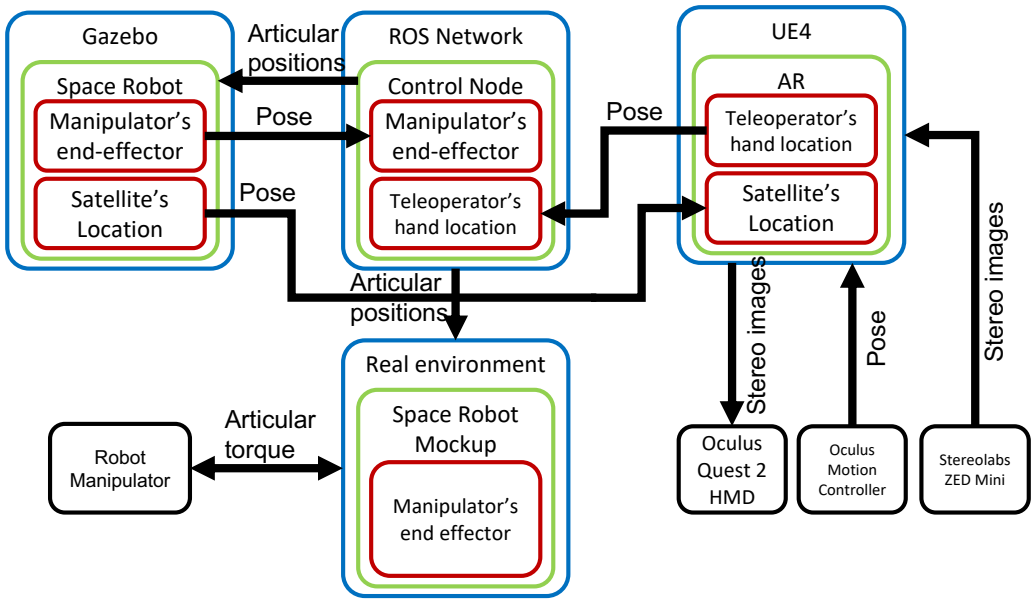

Fig. 12. ROS network architecture, where the information between the local and remote environment flows bilaterally.

- 3D Data visualization and rendering
- Data logging
- Data visualization
- Robot's manipulator control
- Debugging

Meanwhile, the local environment was designed using a custom Unreal Engine 4 (UE4) version that integrates the Stereolabs camera for AR applications in pass-through mode. The virtual camera is positioned relative to the HMD's motion, and an avatar was used to represent the operator's hand in the virtual environment relative to its hand motion. Also, the 3D visual representation of the target satellite is positioned relative to the remote target satellite's motion in an augmented virtuality-like setup.

The modified UE4 engine receives the HMD and 6-axis marker position and orientation and then publishes them in the ROS network to control the virtual camera and the avatar pose, respectively. It subscribes to the end-effector's and target satellite's pose topics and renders them in the AR environment.

IV. Experimental Results

The local environment rendered by the AR teleoperation system with direct hand presence is shown in Figure 13, where a 3D virtual representation of the target satellite, the avatar, and the RoI are depicted in front of the user. The 3D virtual representation of the target satellite receives the location relative to the space robot from the remote environment. Meanwhile, the avatar hand moves relative to the direct hand-presence device, and it is used to select and store the waypoints that the user sends to the remote environment to be interpolated and executed. At all times, the RoI is rendered over the 3D virtual representation of the target satellite for reference.

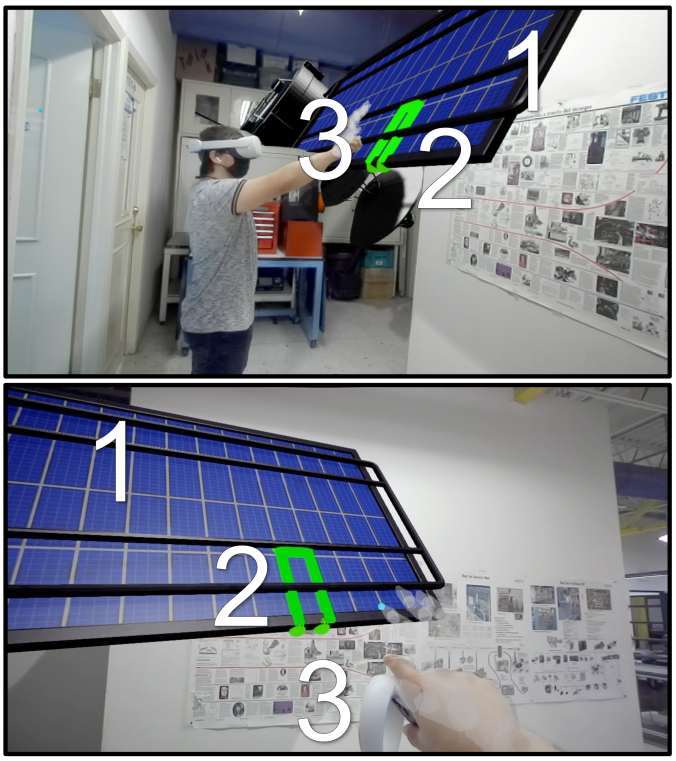

Fig. 13. Augmented local environment of the AR Servicing system, where (1) the target satellite's 3D model, (2) the desired trajectory, and (3) the hand avatar are rendered in front of the teleoperator. Also, the first-person View is depicted for reference below.

Figure 14 depicts the robot performing the servicing operation in a physically occluded zone. This figure can highlight the contribution of the approach, as it is evident that the area of interest is invisible for the robot; consequently, without the aid of a system such as the proposed AR framework, it is simply not possible to execute the task without the high risk

of collision and further damage.

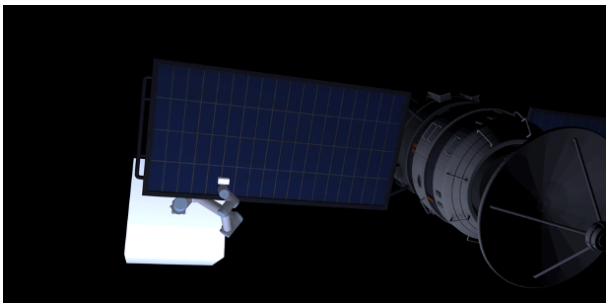

Fig. 14. Robot working in the occluded zone.

The EE's position tracking in 3D space is depicted in Fig. 15 with four control waypoints. At the same time, the position and orientation are shown in Fig. 16 and Fig. 17, respectively. The set-point controller scheme used to control the end-effector in Euclidean space was a PID controller with fixed gains. The gains were tuned such that the EE's pose convergences to the reference in a finite time.

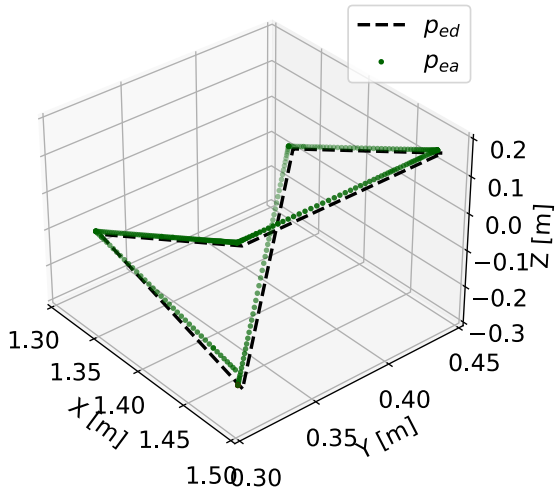

Fig. 15. End-effector's position tracking.

Particularly, in the designed task, the position is the first priority, followed by the orientation. As shown in Fig. 18, the position error over time tends to zero after each reference change. As long as the robot's EE tool touches the solar cell surface, the repair action is being performed.

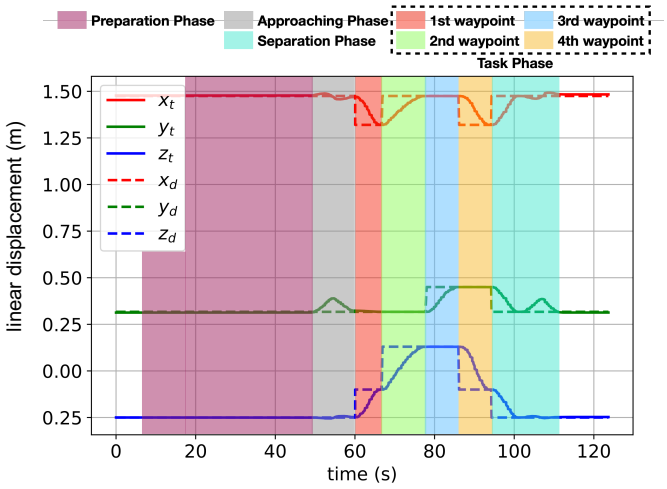

Fig. 16. End-effector's position tracking performance.

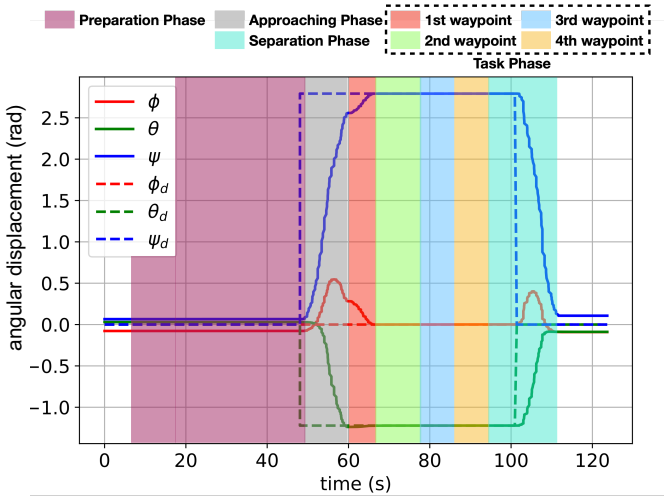

Fig. 17. End-effector's orientation tracking performance.

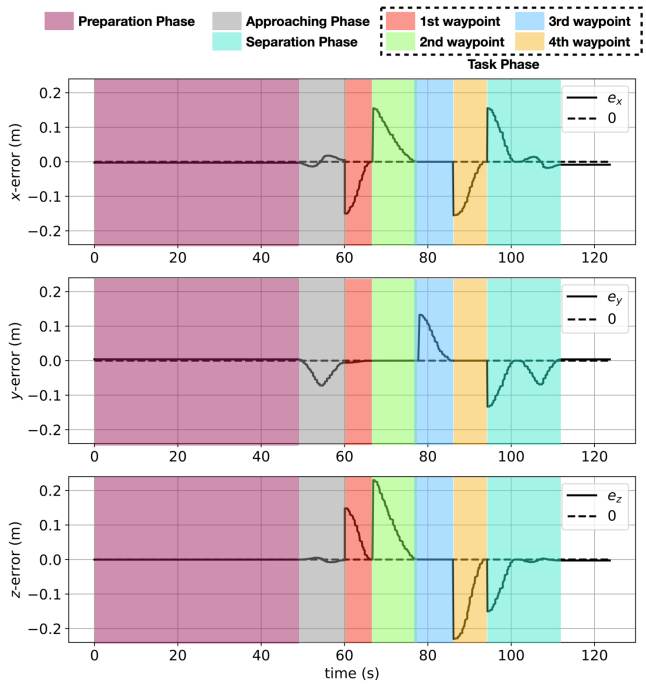

Fig. 18. End-effector's position error in Euclidean space

Orientation is set to be the second priority in the pose controller. As described in Fig. 19, when the pose reference change from the Approaching Phase to the Task Phase, the controller, to maintain the position reference, makes the EE's orientation varies in a magnitude of 0.29 rad around the x-axis, 0 rad around the y-axis, and 0.2 rad over the z-axis. Although 0.29 rad could be interpreted as a large error, the variation is over the x-axis, which is perpendicular to the RoI's surface and does not impact the repair trajectory.

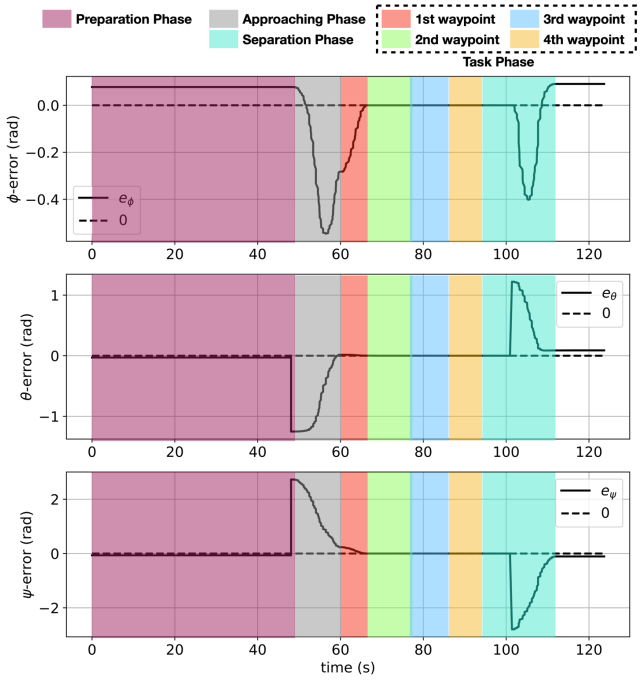

Fig. 19. End-effector orientation error

Besides, considering the stiffness and damping ratio of the surface of the solar panels given in the mass-spring-damper model as $k = 10$ and $b = 0.2$, the estimated contact force between the end-effector and the target satellite's surface in one of the trials is depicted in Fig. 20.

Data transmission delays are common in teleoperated tasks. Major obstacles for such applications include latency, channel corruptions, and bandwidth, which limit teleoperation efficacy [20]. The mitigation or reduction of time delays is of cumbersome importance. The authors plan to address this issue in future work by optimaly selecting and assimilating the information to transmit.

V. Conclusion

This paper introduced an AR-based scheme to cope with the occlusion problem in orbital teleoperation tasks. A direct hand-presence mechanism allowed a direct map between the human operator's hand orientation and the robot's EE tool orientation. Having a dynamic view of the RoI allowed to define an obstacle-free trajectory that passes through the repair path despite the fact the repair zone was in an occluded area. This type of development is required to support several robotic semi-autonomous missions where the local and fixed vision

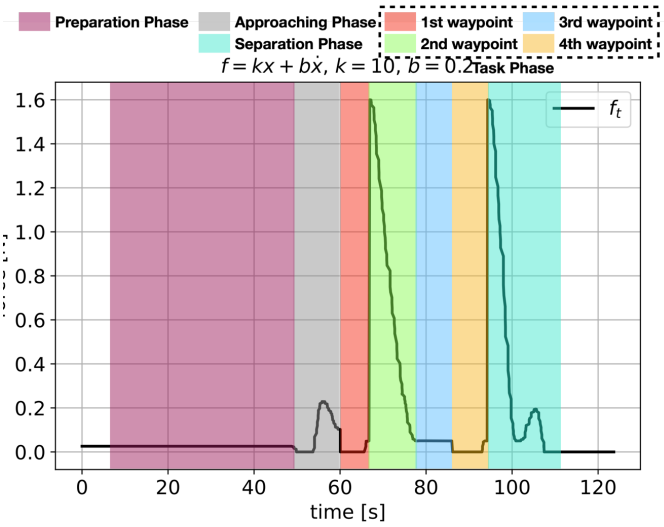

Fig. 20. Sensor-less contact force estimation between the end-effector and the RoI surface in the welding task.

systems are not enough to perceive the RoI properly and, as a result, avoid a defined trajectory with the risk of collision.

In a future effort, the system will be improved by combining virtual and real visual feedback to enhance user immersion.

References

[1] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, march 2014.

[2] E. Ackerman, R. Rao, J. Pepitone, and M. Hampson, "News," *IEEE Spectrum*, vol. 59, no. 5, pp. 7–13, may 2022.

[3] W. J. Gallagher, K. Solberg, G. Gardell G., and B. Roberts, "A survey of enabling technologies for in-space assembly and servicing," *2018 AIAA SPACE and Astronautics Forum and Exposition*, pp. 5116–5126, september 2018.

[4] E. Coleshill, R. R. Layi Oshinowa, D. R. Bardia Bina, and S. Sindelar, "Dextre: Improving maintenance operations on the international space station," *Acta Astronautica 64*, pp. 869–874, december 2009.

[5] A. W. Yew, S. K. Ong, and A. Y. Nee, "Immersive augmented reality environment for the teleoperation of maintenance robots," vol. 61. Elsevier B.V., 2017, pp. 305–310.

[6] T. Xia, S. Léonard, A. Deguet, L. Whitcomb, and P. Kazanzides, "Augmented reality environment with virtual fixtures for robotic telemanipulation in space," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, december 2012, pp. 5059–5064.

[7] P. Kazanzides, B. P. Vagvolgyi, W. Pryor, A. Deguet, S. Leonard, and L. L. Whitcomb, "Teleoperation and visualization interfaces for remote intervention in space," *Frontiers in Robotics and AI*, vol. 8, pp. 1–19, december 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2021.747917

[8] C. D'Ettorre, A. Mariani, A. Stilli, F. R. Y. Baena, P. Valdastri, A. Deguet, P. Kazanzides, R. H. Taylor, G. S. Fischer, S. P. DiMaio, A. Menciassi, and D. Stoyanov, "Accelerating surgical robotics research: A review of 10 years with the da vinci research kit," *IEEE Robotics & Automation Magazine*, vol. 28, pp. 56–78, september 2021.

[9] R. Hussain, A. Lalande, C. Guigou, and A. Bozorg-Grayeli, "Contribution of augmented reality to minimally invasive computer-assisted cranial base surgery," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 7, pp. 2093–2106, 2020.

[10] N. Reski, A. Alissandrakis, and A. Kerren, "Exploration of time-oriented data in immersive virtual reality using a 3d radar chart approach," in *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*, ser. NordiCHI '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3419249.3420171

[11] G. Schroeder, C. Steinmetz, C. E. Pereira, I. Muller, N. Garcia, D. Espindola, and R. Rodrigues, "Visualising the digital twin using web services and augmented reality," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 522–527.

[12] R. Williams, J. A. Erkoyuncu, T. Masood, and R. Vrabic, "Augmented reality assisted calibration of digital twins of mobile robots," *IFAC-PapersOnLine*, vol. 53, no. 3, pp. 203–208, 2020, 4th IFAC Workshop on Advanced Maintenance Engineering, Services and Technologies - AMEST 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320301798

[13] P. Casey, I. Baggili, and A. Yarramreddy, "Immersive virtual reality attacks and the human joystick," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 550–562, 2019.

[14] M. Kim, C. Jeon, and J. Kim, "A study on immersion and presence of a portable hand haptic system for immersive virtual reality," *Sensors*, vol. 17, no. 5, pp. 1–18, may 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/5/1141

[15] D. Eger Passos and B. Jung, "Measuring the accuracy of inside-out tracking in xr devices using a high-precision robotic arm," in *HCI International 2020 - Posters*, C. Stephanidis and M. Antona, Eds. Cham: Springer International Publishing, 2020, pp. 19–26.

[16] T. A. Jost, B. Nelson, and J. Rylander, "Quantitative analysis of the oculus rift s in controlled movement," *Disability and Rehabilitation: Assistive Technology*, vol. 16, no. 6, pp. 632–636, 2021, pMID: 31726896. [Online]. Available: https://doi.org/10.1080/17483107.2019.1688398

[17] A. Akay and Y. S. Akgul, "An end-to-end stochastic action and visual estimation system towards autonomous teleoperation," *IEEE Access*, vol. 10, pp. 16 700–16 719, february 2022.

[18] J. Wen, H. Ma, and X. Zhang, "Optimization of the occlusion strategy in visual tracking," 2016.

[19] A. Flores-Abad, M. A. García Terán, I. U. Ponce, and M. Nandayapa, "Compliant force sensor-less capture of an object in orbit," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 497–505, september 2021.

[20] P. Farajiparvar, Y. Hao, and P. Abhilash, "A brief survey of telerobotic time delay mitigation," *Frontiers in Robotics and AI*, vol. 7, 2020.