# On-ground validation of orbital GNC: Visual navigation assessment in robotic testbed facility

**Vivek Muralidharan[1,*] (✉), Mohatashem Reyaz Makhdoomi[1,*], Augustinas Žinys[2], Bronislovas Razgus[2], Marius Klimavičius[2], Miguel Olivares-Mendez[1], and Carol Martinez[1]**

1. *Space Robotics Research Group (SpaceR), Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855, Luxembourg*
2. *Blackswan Space, Vilnius LT01121, Lithuania*

## ABSTRACT

CubeSats have become versatile platforms for various space missions (e.g., on-orbit servicing and debris removal) owing to their low cost and flexibility. Many space tasks involve proximity operations that require precise guidance, navigation, and control (GNC) algorithms. Vision-based navigation is attracting interest for such operations. However, extreme lighting conditions in space challenge optical techniques. The on-ground validation of such navigation systems for orbital GNC becomes crucial to ensure their reliability during space operations. These systems undergo rigorous testing within their anticipated operational parameters, including the exploration of potential edge cases. The ability of GNC algorithms to function effectively under extreme space conditions that exceed anticipated scenarios is crucial, particularly in space missions where the scope of errors is negligible. This paper presents the ground validation of a GNC algorithm designed for autonomous satellite rendezvous by leveraging hardware-in-the-loop experiments. This study focuses on two key areas. First, the rationale underlying the augmentation of the robot workspace (six-degree-of-freedom UR10e robot + linear rail) is investigated to emulate relatively longer trajectories with complete position and orientation states. Second, the control algorithm is assessed in response to uncertain pose observations from a vision-based navigation system. The results indicate increased control costs with uncertain navigation and exemplify the importance of on-ground testing for system validation before launch, particularly in extreme cases that are typically difficult to assess using software-based testing.

**KEYWORDS**

guidance, navigation, and control (GNC)
rendezvous
proximity operations
vision-based navigation (VBN)
software-in-the-loop (SIL)
hardware-in-the-loop (HIL)
robot manipulation

**Research Article**
Received: 6 October 2023
Accepted: 31 December 2023

## 1 Introduction

The increasing number of government and commercial space missions to near-Earth orbits, particularly the low-Earth orbit (LEO), medium-Earth orbit (MEO), and geostationary orbit (GEO), has resulted in a push towards an increase in the autonomy of satellites. Proximity operations, including rendezvous and docking/berthing, are crucial in carrying out various tasks such as on-orbit servicing (OOS) to restore or enhance the functionality of operational satellites, extend the mission life span through refueling, capturing, and de-orbiting defunct satellites, and capturing space debris. Numerous rendezvous operations have been successfully facilitated in the past, beginning with the Gemini space program, followed by the Apollo program, the assembly of the Mir and International Space Station, and several subsequent others [1]. Several past and present rendezvous and docking processes were supervised by humans on board [2, 3]. These rendezvous operations involving humans on

[*] Vivek Muralidharan and Mohatashem Reyaz Makhdoomi contributed equally to this work.
✉ vivek.muralidharan@uni.lu

board are expensive and risky, making them inconceivable for large-scale operations, particularly considering the number of satellites launched in the current era and the large amount of space debris that poses threats to operating satellites. In contrast, CubeSats are the most suitable option for such large-scale operations owing to their limited cost and ease of manufacture and launch. The recent success of Mission Extension Vehicle (MEV) 1 and 2 for autonomous docking demonstrates the advancement in autonomous operations. Recently, researchers have explored innovative approaches for autonomous navigation in proximity operations, including sensor fusion with a laser range finder and monocular camera [4], the custom integrated XACT-50 guidance, navigation, and control (GNC) system by Blue Canyon Technologies [5], a combination of radar and LiDAR technologies [6], and a dual-inertial state filter that processes GPS, intersatellite link range, and optical sensor measurements [7].

A rendezvous and docking procedure involves a significantly close approach and potential coupling of two objects in space, specifically an active satellite (also referred to as the "chaser") and a "target" satellite [1, 8]. In a wider context, the targets include satellites that are not ideally designed for rendezvous and docking operations, are partly or entirely non-functional, or are debris objects. Consequently, the target is considered to be "passive". Here, only the chaser satellite can execute the required controlled orbital maneuvers to approach the target in its orbit, i.e., the chaser is "active". Additionally, for controlled autonomous operations, a satellite may possess supportive features, such as transponders, reflectors, artificial visual markers, and dedicated fixtures, which enable harmonious cooperation with another body that has sensory devices to detect certain features. For any proximity operations in space, if a target satellite is designed with external markers, they are labeled "cooperative". A few applications of these have been discussed in the literature [9–13]. Typically, the target satellite may not have specific markers available for rendezvous and docking operations and, hence, is classified as "non-cooperative". Moreover, the dynamics of the target may result in tumbling or non-tumbling motion. Nevertheless, the target satellite may have an active controller and be cooperative. In this investigation, however, the target is considered to be non-cooperative and tumbling, enabling the testing of GNC algorithms for more extreme and challenging cases.

On-ground tests of algorithms for space missions are essential for validating systems before they are launched. Only a few research test beds are available for the on-ground validation of orbital GNC systems and algorithms, particularly those that rely on emulating motions in space, such as the European Proximity Operations Simulator (EPOS) at the German Aeronautics Centre (DLR) [14] consisting of two robot systems with six and seven degree-of-freedom (DoF) motion capabilities, the GNC Rendezvous and Landing Simulator (GRALS) at the European Space Research and Technology Centre (ESTEC) in the Netherlands [15] that includes a 7-DoF robot and rail system with floating platforms, and the Testbed for Robotic Optical Navigation (TRON) in Stanford University [16] with two 7-DoF robot systems. The Zero-G Lab at the University of Luxembourg is a newly developed facility for performing spatial motion. These facilities offer testing capabilities for proximity operations, debris capture and removal, validation of computer vision systems, etc. When testing a GNC algorithm designed for autonomous satellite rendezvous in challenging space environments, hardware-in-the-loop (HIL) experiments are used to stress-test their reliability for actual spatial operations. The HIL tests in this investigation involve three main components: a robotic test-bed facility consisting of ceiling and wall robots that emulate target and chaser trajectories to replicate the rendezvous operation, a vision-based navigation system to deliver relative satellite poses, and a GNC algorithm that accounts for orbital dynamics. Preliminary studies have validated the use of software-in-the-loop (SIL) experiments to validate GNC algorithms; however, the performance of vision-based navigation under extreme lighting conditions necessitates the use of actual cameras and trajectories.

This investigation focuses on the adoption of a 7-DoF robotic testbed in the Zero-G Lab at the University of Luxembourg to emulate rendezvous scenarios. Herein, we discuss the setup of HIL tests, integration of the visual GNC, and real-time testing of the combined systems. The remainder of this paper is organized as follows: Section 2 presents the overall GNC architecture and emphasizes the infrastructure requirements of an on-ground test facility to validate the feasibility of space operations. Section 3 presents the dynamics of the satellite motion and the GNC strategy. The operational dynamics of the integrated robotic testbed are discussed in Section 4. Section 5 presents the experimental setup and the results of HIL experiments. Finally, important takeaways and remarks are presented in Section 6.

# 2 Problem statement

## 2.1 GNC architecture

The aim of this investigation is the ground validation of an autonomous GNC technique for orbital rendezvous in near-Earth orbits that exploits coupled orbit and attitude dynamics. In the context of the orbital rendezvous described here, both the target and chaser satellites are modeled as rigid bodies that require a minimum prescribed separation to avert potential collisions between them. Furthermore, in an eventual docking scenario, a predetermined face of the satellite is structured to facilitate contact. In either scenario, the center of mass of the target and chaser satellites must never overlap but maintain a fixed separation from each other during the entirety of rendezvous operations. Navigation is performed using a custom system built by Blackswan Space Ltd. onboard a chaser satellite, which delivers the pose of a potential target satellite. The applicability of the GNC algorithm is initially validated using the Mission Design Simulator (MDS) software (also developed by Blackswan Space Ltd.) [17]. An outline of the GNC architecture is shown in Fig. 1; it relies on key packages such as orbital dynamics simulator (ODS), vision-based navigation (VBN), and the software/hardware infrastructure, similar to the one adopted in Ref. [18]. The ODS is a Python package that outputs waypoints for hardware motion or control outputs for software application programming interfaces (APIs) while accepting pose observations from the VBN system. The ODS package incorporates equations for relative orbital dynamics and attitude motion. Relevant features are incorporated to identify the guidance path through a nonlinear optimizer and identify control maneuvers to overcome the predicted deviations. Furthermore, if the controller feature is turned off, a natural ballistic motion of these satellites is achieved, which does not include any midcourse trajectory corrections. Similarly, the VBN system is incorporated using a different Python package configured with the hardware infrastructure within the software, and it identifies the satellite pose as the output. More rigorous HIL tests are subsequently required to assess the durability of the guidance and control algorithms, particularly under the limiting cases of VBN. The VBN system is deliberately exposed to

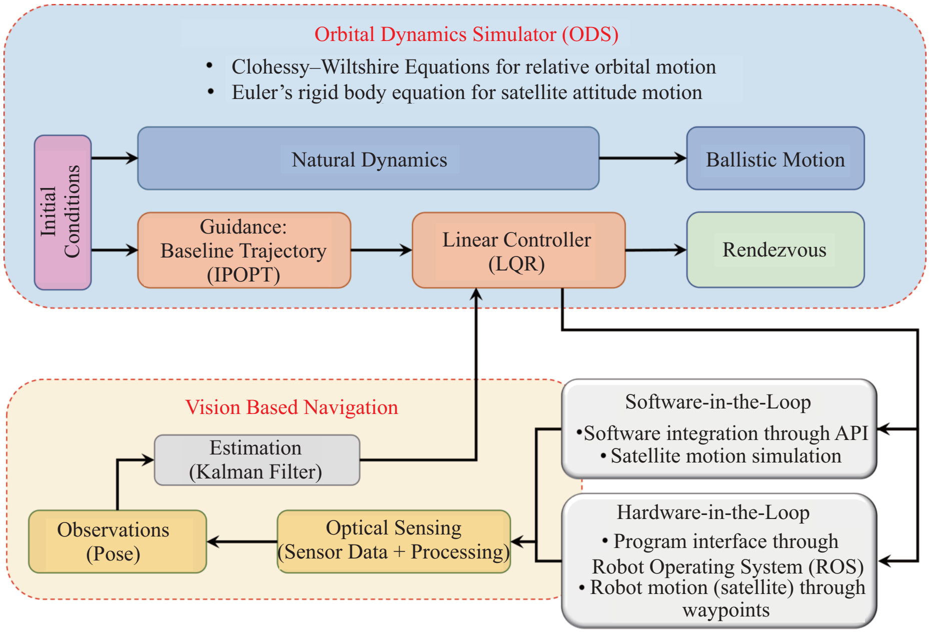

Orbital Dynamics Simulator (ODS)
• Clohessy–Wiltshire Equations for relative orbital motion
• Euler's rigid body equation for satellite attitude motion
Initial Conditions
Natural Dynamics
Ballistic Motion
Guidance: Baseline Trajectory (IPOPT)
Linear Controller (LQR)
Rendezvous
Vision Based Navigation
Estimation (Kalman Filter)
Observations (Pose)
Optical Sensing (Sensor Data + Processing)
Software-in-the-Loop
•Software integration through API
• Satellite motion simulation
Hardware-in-the-Loop
• Program interface through Robot Operating System (ROS)
• Robot motion (satellite) through waypoints

**Fig. 1** Overview of the GNC architecture. The ODS block integrates equations governing both orbital and attitude motion and evaluates guidance and control algorithms. The VBN block houses algorithms for pose estimation. The architecture flexibly adapts to either SIL or HIL testing through suitable infrastructure and programs.

shadows, insufficient lighting, partial visibility, lens flares, and other challenging conditions. These tests are crucial for achieving space-level readiness.

Furthermore, this study includes a non-cooperative target for HIL experiments, i.e., the satellite by itself does not have built-in features for active navigation. This study was an indirect extension of prior tests conducted at the Zero-G Lab at the University of Luxembourg, which focuses on cooperative rendezvous in the cislunar system and utilizes ArUco markers on the target satellite fixture for pose observations [13].

### 2.2 On-ground validation platform

Space missions are expensive and challenging. On-ground test facilities enable the testing of various subsystems before they are launched. One such facility is the Zero-G Lab at the University of Luxembourg. Through this study, the capabilities of the Zero-G Lab for on-ground validation tests are enhanced by integrating a linear motion along a rail to deliver an additional DoF for robots to emulate satellite motion, eventually enabling the testing of a wider range of motion. Consolidating a linear rail into a robot control block to obtain a combined interface is vital. The challenges and motivations for augmenting an additional DoF motion for the robotic test-bed facility are presented herein. This study focuses on evaluating workspace accessibility, limiting joint conditions, and safe operation of the augmented system by avoiding potentially risky and unstable joint motions. To evaluate the GNC algorithm, the ceiling and wall robots at the Zero-G laboratory are mounted with satellite mockups and a camera, and subsequently mimick the target and chaser trajectories (in position and orientation) to replicate a rendezvous operation. Previous research demonstrates the use of a 6-DoF robot system to perform rendezvous motion but offers only a limited range of motion [13].

## 3 Orbital rendezvous with vision-based navigation

### 3.1 Dynamics

In close-proximity operations, the precise relative positions and orientations of a chaser satellite and its target are crucial. In particular, for navigation that relies on capturing images of the target satellite using cameras on the chaser satellite, control over all six DoFs (three in position and three in orientation) is of paramount importance. Therefore, a complete coupled orbit and attitude dynamics are considered for close-proximity operations.

The relative nonlinear dynamics of a chaser body with respect to a target body in a circular orbit are highlighted in the Clohessy–Wiltshire (CW) model (defined in Fig. 2) [19]. The CW model has been exploited for numerous rendezvous and formation flying applications [18–25] in circular orbits in near-Earth space, and subsequently used for modeling orbital dynamics in this work. Furthermore, the application of the CW equations is suitable for a large number of artificial objects in nearly circular LEOs [26]. The CW equations accommodate additional control variables that enable us to describe the dynamics of controllable objects, such as active satellites, and non-controllable objects, such as defunct satellites and debris. The motion of the chaser relative to the target is defined as

$$\ddot{x} = 3\Omega^2 x + 2\Omega\dot{y} + u_1 \tag{1}$$

$$\ddot{y} = -2\Omega\dot{x} + u_2 \tag{2}$$

$$\ddot{z} = -\Omega^2 z + u_3 \tag{3}$$

where $x$, $y$, and $z$ define the position vector ($\bar{\rho}$) of the chaser relative to the target, i.e., $\bar{\rho} = x\hat{x} + y\hat{y} + z\hat{z}$. Here, the orbital angular velocity of the target body is defined as

$$\Omega = \sqrt{\frac{\mu}{a^3}} \tag{4}$$

where $a$ is the radius of the target object, and $\mu$ is the standard gravitational parameter for the Earth. The

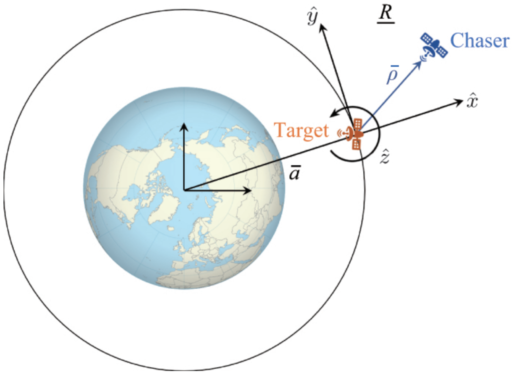


**Fig. 2** Rotating coordinate frame ($\underline{R}$) representing the CW model. The frame is centered on the target body. Direction $\hat{x}$ points radially away from the Earth, $\hat{y}$ points in the direction of orbit velocity, and $\hat{z}$ completes the right-hand coordinate system and points in the direction of the positive angular momentum vector [18].

quantities $u_1, u_2$, and $u_3$ are the external accelerations that the chaser satellite can deliver along each of the principal directions, respectively.

The attitude dynamics for the chaser and target bodies are expressed using Euler's equations for rigid-body dynamics. For ease of operation, they are modeled in different frames. During rendezvous, the orientation of the chaser must be consistent with that of the target. Hence, for simplicity, the chaser's orientation is defined relative to the target body frame. Therefore, the quantities for the angular velocity and orientation quaternions are expressed as

$$ {}^{\mathrm{C}}\dot{\omega}_1 = \frac{1}{{}^{\mathrm{C}}I_1}\left[T_1 - ({}^{\mathrm{C}}I_3 - {}^{\mathrm{C}}I_2){}^{\mathrm{C}}\omega_2{}^{\mathrm{C}}\omega_3\right] \tag{5}$$

$$ {}^{\mathrm{C}}\dot{\omega}_2 = \frac{1}{{}^{\mathrm{C}}I_2}\left[T_2 - ({}^{\mathrm{C}}I_1 - {}^{\mathrm{C}}I_3){}^{\mathrm{C}}\omega_1{}^{\mathrm{C}}\omega_3\right] \tag{6}$$

$$ {}^{\mathrm{C}}\dot{\omega}_3 = \frac{1}{{}^{\mathrm{C}}I_3}\left[T_3 - ({}^{\mathrm{C}}I_2 - {}^{\mathrm{C}}I_1){}^{\mathrm{C}}\omega_1{}^{\mathrm{C}}\omega_2\right] \tag{7}$$

$$ {}^{\mathrm{C}}\dot{\epsilon}_1 = \frac{1}{2}\left({}^{\mathrm{C}}\omega_1{}^{\mathrm{C}}\epsilon_4 - {}^{\mathrm{C}}\omega_2{}^{\mathrm{C}}\epsilon_3 + {}^{\mathrm{C}}\omega_3{}^{\mathrm{C}}\epsilon_2\right) \tag{8}$$

$$ {}^{\mathrm{C}}\dot{\epsilon}_2 = \frac{1}{2}\left({}^{\mathrm{C}}\omega_1{}^{\mathrm{C}}\epsilon_3 + {}^{\mathrm{C}}\omega_2{}^{\mathrm{C}}\epsilon_4 - {}^{\mathrm{C}}\omega_3{}^{\mathrm{C}}\epsilon_1\right) \tag{9}$$

$$ {}^{\mathrm{C}}\dot{\epsilon}_3 = \frac{1}{2}\left(-{}^{\mathrm{C}}\omega_1{}^{\mathrm{C}}\epsilon_2 + {}^{\mathrm{C}}\omega_2{}^{\mathrm{C}}\epsilon_1 + {}^{\mathrm{C}}\omega_3{}^{\mathrm{C}}\epsilon_4\right) \tag{10}$$

$$ {}^{\mathrm{C}}\dot{\epsilon}_4 = \frac{1}{2}\left(-{}^{\mathrm{C}}\omega_1{}^{\mathrm{C}}\epsilon_1 - {}^{\mathrm{C}}\omega_2{}^{\mathrm{C}}\epsilon_2 - {}^{\mathrm{C}}\omega_3{}^{\mathrm{C}}\epsilon_3\right) \tag{11}$$

where ${}^{\mathrm{C}}\omega \equiv \omega(\text{Chaser} \rightarrow \text{Target})$ and ${}^{\mathrm{C}}\epsilon_i \equiv \epsilon_i(\text{Chaser} \rightarrow \text{Target})$, with the left superscript C symbolizing the chaser. The moments of inertia of the chaser body along each principal axis are given by ${}^{\mathrm{C}}I_1, {}^{\mathrm{C}}I_2$, and ${}^{\mathrm{C}}I_3$. Here, the quantities $T_1$, $T_2$, and $T_3$ are the components of the external torques delivered by the chaser to deliberately change its course during rendezvous. For convenience, control variables $T_1$, $T_2$, and $T_3$ are occasionally referred to as $u_4, u_5$, and $u_6$, respectively.

However, the target is modeled relative to the inertial frame for ease of incorporating external effects such as the Earth's gravity gradient torque. The angular velocity and orientation quaternions are denoted by the left superscript T to represent the target. Thus, the equations for the attitude dynamics of the target are expressed as

$$ {}^{\mathrm{T}}\dot{\omega}_1 = \frac{{}^{\mathrm{T}}I_3 - {}^{\mathrm{T}}I_2}{{}^{\mathrm{T}}I_1}\left(3\mathbb{C}_{12}\mathbb{C}_{13}\Omega^2 - {}^{\mathrm{T}}\omega_2{}^{\mathrm{T}}\omega_3\right) \tag{12}$$

$$ {}^{\mathrm{T}}\dot{\omega}_2 = \frac{{}^{\mathrm{T}}I_1 - {}^{\mathrm{T}}I_3}{{}^{\mathrm{T}}I_2}\left(3\mathbb{C}_{11}\mathbb{C}_{13}\Omega^2 - {}^{\mathrm{T}}\omega_1{}^{\mathrm{T}}\omega_3\right) \tag{13}$$

$$ {}^{\mathrm{T}}\dot{\omega}_3 = \frac{{}^{\mathrm{T}}I_2 - {}^{\mathrm{T}}I_1}{{}^{\mathrm{T}}I_3}\left(3\mathbb{C}_{11}\mathbb{C}_{12}\Omega^2 - {}^{\mathrm{T}}\omega_1{}^{\mathrm{T}}\omega_2\right) \tag{14}$$

$$ {}^{\mathrm{T}}\dot{\epsilon}_1 = \frac{1}{2}\left({}^{\mathrm{T}}\omega_1{}^{\mathrm{T}}\epsilon_4 - {}^{\mathrm{T}}\omega_2{}^{\mathrm{T}}\epsilon_3 + {}^{\mathrm{T}}\omega_3{}^{\mathrm{T}}\epsilon_2\right) \tag{15}$$

$$ {}^{\mathrm{T}}\dot{\epsilon}_2 = \frac{1}{2}\left({}^{\mathrm{T}}\omega_1{}^{\mathrm{T}}\epsilon_3 + {}^{\mathrm{T}}\omega_2{}^{\mathrm{T}}\epsilon_4 - {}^{\mathrm{T}}\omega_3{}^{\mathrm{T}}\epsilon_1\right) \tag{16}$$

$$ {}^{\mathrm{T}}\dot{\epsilon}_3 = \frac{1}{2}\left(-{}^{\mathrm{T}}\omega_1{}^{\mathrm{T}}\epsilon_2 + {}^{\mathrm{T}}\omega_2{}^{\mathrm{T}}\epsilon_1 + {}^{\mathrm{T}}\omega_3{}^{\mathrm{T}}\epsilon_4\right) \tag{17}$$

$$ {}^{\mathrm{T}}\dot{\epsilon}_4 = \frac{1}{2}\left(-{}^{\mathrm{T}}\omega_1{}^{\mathrm{T}}\epsilon_1 - {}^{\mathrm{T}}\omega_2{}^{\mathrm{T}}\epsilon_2 - {}^{\mathrm{T}}\omega_3{}^{\mathrm{T}}\epsilon_3\right) \tag{18}$$

where the effects of the gravity gradient torque along different principal directions are mapped by the elements of the rotation matrix $\mathbb{C}$, which defines the transformation between the target body frame and rotating coordinate frame ($\underline{R}$). Note that the quantity $\mathbb{C}_{ij}$ is the element in the $i$-th row and $j$-th column. The terms specifically used in the aforementioned equations correspond to the orientation of each of the target body axes in the direction of the gravitational pull, i.e., the $\hat{x}$ direction in the CW frame. The moments of inertia of the target body along each of its principal axes are given by ${}^{\mathrm{T}}I_1, {}^{\mathrm{T}}I_2$, and ${}^{\mathrm{T}}I_3$. Appropriate initial conditions for the target body are selected to achieve a non-tumbling motion balanced by the gravity torque. Other sets of initial conditions produce the corresponding tumbling motions.

The chaser and target satellites are rigid bodies; therefore, the rendezvous path must be gradual and without collisions. During this process, the chaser satellite must remain at a fixed distance from the target body (defined by $\bar{\varrho}$ and labeled as "approach site") to avoid a collision (see Fig. 3). For simplicity, the entire system is described by the state vector $\boldsymbol{x}$:

$$\begin{aligned} \boldsymbol{x} = [&x, y, z, \dot{x}, \dot{y}, \dot{z}, {}^{\mathrm{C}}\omega_1, {}^{\mathrm{C}}\omega_2, {}^{\mathrm{C}}\omega_3, {}^{\mathrm{C}}\epsilon_1, {}^{\mathrm{C}}\epsilon_2, {}^{\mathrm{C}}\epsilon_3, {}^{\mathrm{C}}\epsilon_4, \\ &{}^{\mathrm{T}}\omega_1, {}^{\mathrm{T}}\omega_2, {}^{\mathrm{T}}\omega_3, {}^{\mathrm{T}}\epsilon_1, {}^{\mathrm{T}}\epsilon_2, {}^{\mathrm{T}}\epsilon_3, {}^{\mathrm{T}}\epsilon_4, x_\varrho, y_\varrho, z_\varrho]^{\mathrm{T}} \end{aligned} \tag{19}$$

which includes information such as the chaser's position

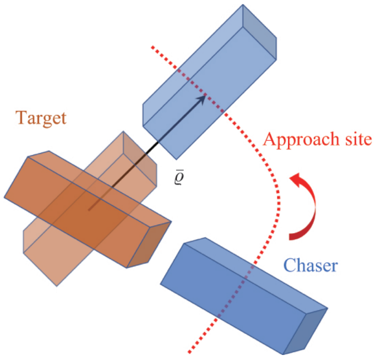


**Fig. 3** Approach site ($\bar{\varrho}$), offering the potential location of the center of mass of the chaser relative to the target satellite during rendezvous [18]. This parameter is defined to ensure a safe operating distance without collision between the satellites.

$(x, y, z)$ and velocity $(\dot{x}, \dot{y}, \dot{z})$ relative to the target, the chaser's angular velocity $({}^{\mathrm{C}}\omega_1, {}^{\mathrm{C}}\omega_2, {}^{\mathrm{C}}\omega_3)$ and orientation $(\epsilon_1, {}^{\mathrm{C}}\epsilon_2, {}^{\mathrm{C}}\epsilon_3, {}^{\mathrm{C}}\epsilon_4)$ relative to the target, the target's angular velocity $({}^{\mathrm{T}}\omega_1, {}^{\mathrm{T}}\omega_2, {}^{\mathrm{T}}\omega_3)$ and orientation $({}^{\mathrm{T}}\epsilon_1, {}^{\mathrm{T}}\epsilon_2, {}^{\mathrm{T}}\epsilon_3, {}^{\mathrm{T}}\epsilon_4)$ relative to the inertial frame, and the location along the approach site $(x_\varrho, y_\varrho, z_\varrho)$. Similarly, the control input vector $\boldsymbol{u}_k$ is defined as

$$\boldsymbol{u}_k = [u_1, u_2, u_3, T_1, T_2, T_3]^{\mathrm{T}} \tag{20}$$

and includes the translational accelerations $u_1$, $u_2$, and $u_3$ along the $x$, $y$, and $z$ directions required for the chaser to progress toward the target. Control torques $T_1$, $T_2$, and $T_3$ (also denoted as $u_4$, $u_5$, and $u_6$, respectively) are included to alter the orientation of the chaser. If $\boldsymbol{u}_k$ is a null vector, i.e., all elements are zero, then the motion of the chaser satellite evolves under natural dynamics.

## 3.2 Guidance, navigation, and control (GNC)

The GNC of orbital rendezvous involves the application of mathematical principles to plan and execute the trajectory and attitudes of satellites during the rendezvous process. A two-layer guidance-and-control strategy is adopted in this investigation [13]. Guidance and control are achieved using features built within the ODS package, whereas the vision-based navigation package offers feedback to the ODS.

### *3.2.1 Guidance: optimal reference motion*

An open-source nonlinear optimizer toolkit called CasADi is employed with an interior-point optimization (IPOPT) method to generate a baseline guidance path that serve as an ideal trajectory for rendezvous. To compute an optimal path, a cost function $J$ is minimized, such that

$$\begin{aligned} J = \int_0^\tau & [(\bar{\rho} - \overline{\varrho^R})^{\mathrm{T}} \boldsymbol{Q}_1 (\bar{\rho} - \overline{\varrho^R}) + {}^{\mathrm{C}}\bar{\omega}^{\mathrm{T}} \boldsymbol{Q}_2 {}^{\mathrm{C}}\bar{\omega} \\ & + {}^{\mathrm{C}}\bar{\epsilon}^{\mathrm{T}} \boldsymbol{Q}_3 {}^{\mathrm{C}}\bar{\epsilon} + \boldsymbol{u}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u}] \mathrm{d}t \end{aligned}$$

where each weighting matrix is a parameter that can be altered depending on the scenario. The weighting matrices $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$, and $\boldsymbol{Q}_3$ penalize the deviations in position states from the approach site, relative angular velocity of the chaser relative to the target, and relative change in orientation between the two spacecraft, respectively. The control inputs $\boldsymbol{u}$ are weighted by the matrix $\boldsymbol{R}$. As mentioned previously, the chaser and target satellites are rigid bodies; consequently, any collisions must be prevented. A collision avoidance constraint is added during orbital rendezvous, represented by $\|\bar{\rho}\| \geqslant r_{\mathrm{collision}}$ and maintains the position of the chaser satellite outside a spherical volume of radius $r_{\mathrm{collision}}$ from the center of the target. The guidance algorithm delivers a sequence of control accelerations and torques to facilitate the coupled orbit and attitude motion of the chaser as it progresses toward the target satellite. The obtained reference path represents an ideal and precise guidance path without fluctuations or uncertainties. However, in reality, actual paths are subject to inherent errors. To address these errors, state estimation and control algorithms operate concurrently throughout proximity operations.

### *3.2.2 Navigation: vision-based pose*

A proprietary VBN system from Blackswan Space is used to detect, recognize, and estimate the position of the target object. This vision system utilizes a deep-learning algorithm based on a Mask-RCNN [27] and consists of several building blocks, as shown in Fig. 4. The captured image is processed by a convolutional neural network, HRNet [28], followed by a region proposal network that outputs relative regions in the feature space, and is eventually pooled using "ROIAlign" [27]. Ultimately, two downstream task heads are built: keypoint head and bounding box (BBox) head. The keypoint head is based on a convolutional auto-encoder to predict keypoints on a 2D image plane, which are then used by a perspective-n-point (PnP) solver to output the relative 3D position and orientation. The BBox head is based on simple linear deep-learning layers that output a BBox and the category class of an object. The algorithm was trained using a synthetic dataset from a simulated orbital environment consisting of a digital twin of a target CubeSat on the MDS software. A small sample dataset (scenario A) obtained from a laboratory emulation was also incorporated into the learning model. The details of the cameras used for the object identification, image resolution, and sample size are listed in Table 1. Additional image datasets were provided for further testing of the algorithm and were classified by dataset in scenarios B and C. The images of the target CubeSat mockup were captured in the laboratory using an Intel Real-Sense D435i camera at varying distances ranging between 0.5 and 3 m and at different light intensities.

A Kalman filter is introduced to estimate the states from the available navigation data. The Kalman filter incorporates the observed data to iteratively update the

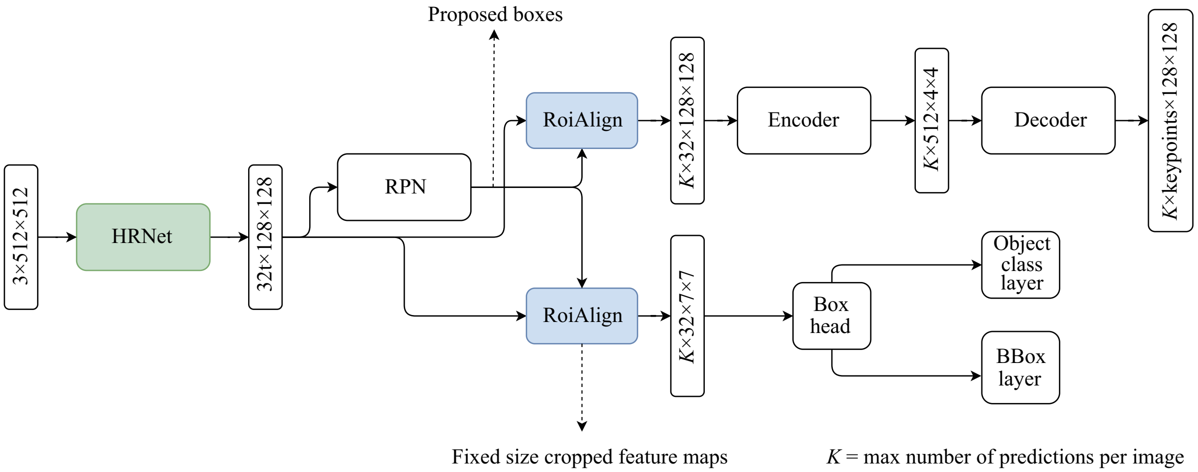


**Fig. 4** Pose estimation deep-learning model based on Mask-RCNN.

**Table 1** Datasets used for training and testing. Training dataset: synthetic and scenario A. Validation datasets: scenarios B and C

| Dataset | $N$ samples | Distance | Image resolution | Camera type | Light intensity | Object |
|---|---|---|---|---|---|---|
| Synthetic dataset | 1000 | 2–5 m | 512 × 512 | Virtual Camera (FOV 30 deg) | — | CubeSat |
| Lab scenario A | 42 | 0.5–3 m | 1280 × 720 | Intel Real-Sense D435i | 35% | CubeSat |
| Lab scenario B | 1021 | 0.5–3 m | 1280 × 720 | Intel Real-Sense D435i | 25% | CubeSat |
| Lab scenario C | 2789 | 0.5–3 m | 1280 × 720 | Intel Real-Sense D435i | 30% | No object |

covariance matrix and state estimates using the system dynamics in Eqs. (21) and (22):

$$\delta\boldsymbol{x}_{j+1} = \boldsymbol{A}_j\delta\boldsymbol{x}_j + \boldsymbol{B}_j\Delta\boldsymbol{v}_j + \boldsymbol{w}_j \tag{21}$$

$$\delta\boldsymbol{y}_{j+1} = \boldsymbol{H}_j\delta\boldsymbol{x}_j + \boldsymbol{e}_j \tag{22}$$

where $\boldsymbol{x}$ is the vector of the state deviations from the baseline path, $\boldsymbol{y}$ is the observed pose, and $\boldsymbol{H}_j = \partial\boldsymbol{y}/\partial\boldsymbol{x}$. Moreover, $\boldsymbol{w}_j$ and $\boldsymbol{e}_j$ are stochastic noises assumed to be Gaussian, with covariances $E[\boldsymbol{w}_j\boldsymbol{w}_j^{\mathrm{T}}] = \mathbb{Q}$ and $E[\boldsymbol{e}_j\boldsymbol{e}_j^{\mathrm{T}}] = \mathbb{R}$. The sampling frequency of the pose data is larger than that of maneuver delivery; hence, the value of $\Delta\boldsymbol{v}_j$ may be zero at certain sampling time $t_j$. Furthermore, note that the target's position and orientation relative to the Earth's inertial frame develop independently of the dynamics of the chaser satellite. Subsequently, a chaser's pose observations or maneuvers have no impact on the motion of the target satellite. Consequently, the number of partial derivatives of the states relative to the observed pose is zero.

### 3.2.3 *Control: linear quadratic regulator*

Navigation feedback offers state deviations from the baseline path, which are typically small, and the linearized dynamics evaluated along the baseline path are adequate for rapidly forecasting an appropriate control output. Linear dynamics are governed by

$$\delta\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\delta\boldsymbol{x}_k + \boldsymbol{B}_k\delta\boldsymbol{u}_k \tag{23}$$

where $\boldsymbol{A}_k$ is the $23\times23$ state transition matrix, and $\boldsymbol{B}_k$ is a $23\times6$ matrix corresponding to the partials of the state vector at the final time ($\boldsymbol{x}_{k+1}$) to a control vector $\delta\boldsymbol{u}_k$, which is assessed along the baseline route. Because the motion of the target and location of the approach site are independent of the motion of the chaser, several partials in matrix $\boldsymbol{A}_k$ are zero. A discrete linear quadratic regular (LQR) then offers a cost-optimal sequence of control maneuvers to minimize the cost functional $\tilde{J}$, such that

$$\tilde{J} = \delta\boldsymbol{x}_N^{\mathrm{T}}\tilde{\boldsymbol{P}}_N\delta\boldsymbol{x}_N + \sum_{k=0}^{N-1}\delta\boldsymbol{x}_k^{\mathrm{T}}\tilde{\boldsymbol{Q}}_k\delta\boldsymbol{x}_k + \delta\boldsymbol{u}_k^{\mathrm{T}}\tilde{\boldsymbol{R}}_k\delta\boldsymbol{u}_k \tag{24}$$

and regulates any position, orientation, and angular velocity deviation in the state vector estimated from the baseline path. It also modulates the size of the control output. The solution to this linear quadratic regulator problem is given by [29, 30]:

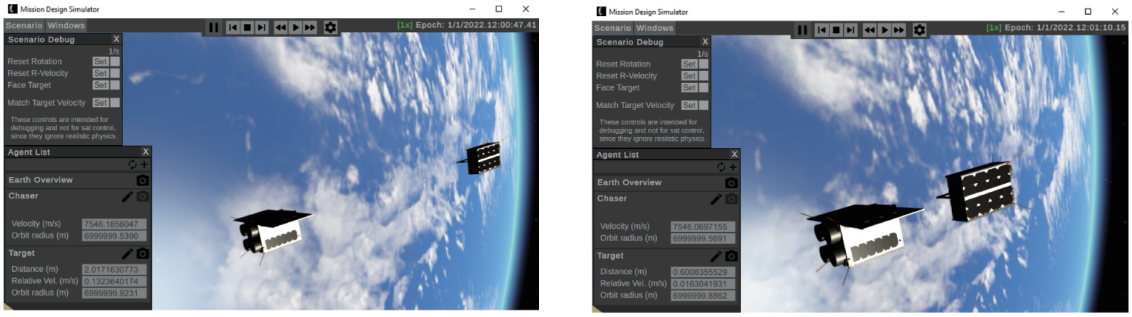
$$\delta\boldsymbol{u}_k = -\tilde{\boldsymbol{K}}_k\delta\boldsymbol{x}_k \tag{25}$$

where $\tilde{\boldsymbol{K}}_k$ is the gain matrix that satisfies the discrete algebraic Ricatti equation to iteratively solve for the coefficient matrices and eventually the gain matrix.

### 3.3 Initial assessment of GNC algorithm

The primary focus of this investigation is to test the feasibility of the algorithm under realistic extreme conditions in an HIL setup. However, the effectiveness and adaptability of an algorithm in a safe environment, such as software and numerical simulations, should be understood before demonstrating it on actual hardware. A previous study demonstrated the integration of a portion of the GNC algorithm as a Python API for SIL testing using the MDS toolkit [18]. The MDS application enables the generation of a variety of orbital scenarios involving the target and chaser satellites [17]. Additional test cases were executed using synthetic pose values to assess the performance of the algorithm further. Learning from the standalone controller and SIL tests aids in understanding and adapting parameters appropriate for higher fidelity HIL tests.

The SIL tests were conducted for a wide range of orbits spanning from 7000 to 36,000 km radius, resembling orbits from an LEO to a GEO. In these cases, slightly aggressive yet prudent rendezvous tests were conducted, achieving rendezvous between the target and chaser, with an initial separation distance ranging between 6 and 240 m and a process duration ranging from 60 to 240 s. The details of the parameters used for optimization and control are listed in Table 2. The selection of the appropriate weighting matrices was crucial for these diverse cases. Notably, the simulated cases exhibited a final displacement error within 0.4 m. Additionally, the chaser satellite adhered to the demanding maximum acceleration level of 0.1 m/s$^2$. Snapshots of the MDS simulator during the SIL tests are shown in Fig. 5, both during the initial phase of rendezvous and as the chaser satellite arrives toward the approach site. The results obtained from the SIL tests facilitated the tuning of the weighting parameters used in rendezvous path generation



These controls are intended for debugging and not for sat control, since they ignore realistic physics.

(a)

[1x] Epoch: 1/1/2022.12:00:47.41

Chaser: Velocity (m/s) 7546.1856047; Orbit radius (m) 6999999.5390

Target: Distance (m) 2.0171630773; Relative Vel. (m/s) 0.1323640174; Orbit radius (m) 6999999.9231

(b)

[1x] Epoch: 1/1/2022.12:01:10.15

Chaser: Velocity (m/s) 7546.0697155; Orbit radius (m) 6999999.5691

Target: Distance (m) 0.6008355529; Relative Vel. (m/s) 0.0163041931; Orbit radius (m) 6999999.8862

**Fig. 5** Phases of chaser motion during proximity operations. (a) Chaser and target during the initial phases of the approach. (b) Chaser arriving at the approach site.

**Table 2** Sample parameters for guidance and control

| Parameter | Software-in-the-loop (simulation) | Hardware-in-the-loop (experiment) |
|---|---|---|
| Radius of target orbit (km) | 7000–36,000 | 7000 |
| Initial separation in CW frame $(x, y, z)$ (m) | Varying | [0.11, −2.81, 0.03]<br>(from actual observation) |
| Distance at rendezvous (m) | 0.4–1.0 | 0.5 |
| Collision avoidance radius (m) | 0.4–1.0 | 0.5 |
| Total time of propagation (s) | 60–240 | 480 |
| Control segments | 30–120 | 60 |
| Max. control acceleration (m/s$^2$) | 0.025–0.1 | $5.0 \times 10^{-5}$ |
| Max. control torque per kg (Nm) | $1.0 \times 10^{-4}$ | $1.0 \times 10^{-4}$ |
| $\boldsymbol{Q}_1$ (diagonal in $\log_{10}$ scale) | Varying | [1, 0.1, 1] |
| $\boldsymbol{Q}_2$ (diagonal in $\log_{10}$ scale) | [0, 0, 0] | [0, 0, 0] |
| $\boldsymbol{Q}_3$ (diagonal in $\log_{10}$ scale) | [0, 0, 0] | [0, 0, 0] |
| $\boldsymbol{R}$ (diagonal in $\log_{10}$ scale) | [−4, −4, −4, −6, −6, −6] | [−4, −4, −4, −6, −6, −6] |
| Pose observation interval (s) | 0.1–5.0 | 0.5 |
| Errors in pose ($3\sigma$) | NA (default software pose values) | NA (based on real pose values) |
| Incorrect orientation or pose flipping | NA (default software pose values) | NA (based on real pose values) |

across diverse scenarios. These parameters achieved a balance by avoiding overly aggressive responses to state changes and excessive inclination toward propellant conservation or maneuver delivery. The values for the weighting matrices that rendered reliable performance are listed in Table 2 and were therefore adopted in HIL tests.

Software-based testing facilitates the assessment of several rendezvous scenarios, particularly those that include significantly longer trajectories that are difficult to simulate in a laboratory environment. However, actual camera performance in scenarios involving challenging lighting conditions cannot be realistically explored via software-based testing and remains a major focus of this investigation.

## 4 Robotic testbed

### 4.1 Laboratory setup

The Zero-G Lab at the University of Luxembourg is a facility dedicated to the real-time emulation of on-orbit servicing missions. The laboratory features a 5 m × 3 m area and incorporates various components to replicate on-orbit scenarios. The laboratory contains two 6-DoF UR10e robotic manipulators mounted on Cobotracks rails, enabling an additional DoF for satellite motion emulation [31]. Figure 6 shows the laboratory setup used in this study. For the rendezvous scenario, the target satellite mockup is positioned on the ceiling arm, whereas the chaser satellite mockup, represented by an optical camera, is mounted on the wall arm. The setup includes the following components:

- **Chaser camera:** The Kurokesu C1 Pro X20 camera with variable exposure (AEC), white balance (AEB), gain, backlight compensation, and low light compensation was used. The camera has a field of view of 66.5 degrees. During the experiments, exposure settings ranged from 50 to 300 ms. Images of 1920 px × 1080 px were captured at 30 fps. Before the experiments, the camera was calibrated using the MoveIt Hand-eye calibration tool with OpenCV libraries.
- **Image processor:** Nvidia Jetson TX2 was used as a reference image processing platform for the chaser to deploy and run deep-learning models during the experiments. The TX2 has a quad-core Cortex A57 + Nvidia Denver processor coupled with an Nvidia Pascal 256-core video card capable of 1.33 TFLOPS and has 8 GB of RAM.
- **Satellite mockup:** A 3D printed mockup of a 0.6× scaled CubeSat resembling a 6U Nanosatellite Bus M6P from Nanoavionics is used in the experiments [32]. These mockups incorporated the same features used to train the deep-learning model.
- **OptiTrack - Motion Capture System (MCS):** Six Primex 13 W cameras are installed in the Zero-

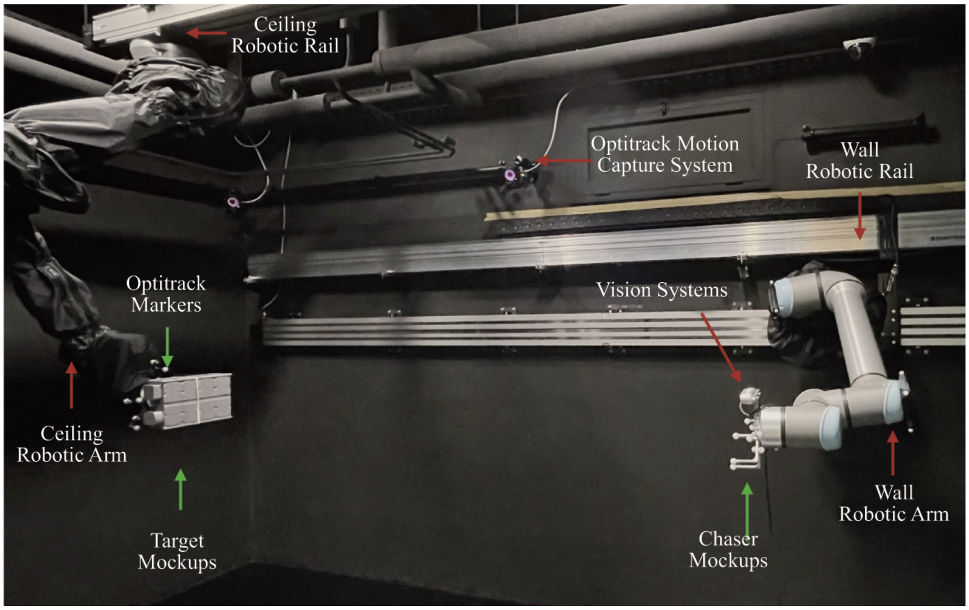


**Fig. 6** Zero-G Lab at the University of Luxembourg.

G Lab facility, pointing in predefined directions for an almost complete coverage of the laboratory area. These cameras track passive (unpowered) and active (powered) markers with positional errors up to ±0.20 mm and rotational errors within 0.5 degrees, and at a frequency up to 240 Hz [33].

- **Light source:** A 60 W LED light with a color temperature rating of 5600 K (similar to peak sunlight) is used as the lighting source while vision-based navigation is active [34]. The position of the light source was varied according to the requirements of the HIL experiments, as discussed later.

The OptiTrack MCS is employed to determine the ground truth for the satellite position and orientation. OptiTrack Motive software provides an interface for viewing OptiTrack data and calibration through the wanding process. As shown in Fig. 6, two sets of abstract 3D body/coordinate frames are created using multiple passive markers with known fixed distances from each other. Marker frames are affixed to each robotic arm alongside the satellite mockups, and a well-defined transformation exists between the markers and mockups. The states of individual markers are detected by the MCS through triangulation, and the pose of the satellite body is subsequently evaluated with submillimeter accuracy. The output from the OptiTrack data is the ground truth value, which is used for computing the guidance path, identifying true satellite states, robot manipulation, and retrieving appropriate transformations between different frames relevant to HIL tests.

The open-source Robot Operating System (ROS) toolkit is used as a robotics middleware to control and communicate with the robots in the experimental setup. The system consists of interconnected computers running ROS nodes to enable HIL simulations. Robot motion is controlled by servoing using feedback from sensors, such as cameras and motion capture systems. The chaser satellite follows a rendezvous path provided as Cartesian set-points, with updates based on visual feedback and pose estimation. MoveIt Servoing, a plugin in the MoveIt motion planning framework, enables real-time control of manipulators, with the chaser and target satellite integrated into the kinematic chain. This setup enables an accurate simulation of the rendezvous process. Concise values of the Zero-G Lab specifications are listed in Table 3.

**Table 3** Features of the Zero-G Lab

| | |
|---|---|
| Target DoFs | 6+1 |
| Simulation range | Rendezvous/proximity |
| Orbital mechanics | Included through ODS package |
| Spacecraft (mockup) payload | 10 kg (maximum) |
| Operating range | 3.2 m (longest direction) |
| Sensing modality | OptiTrack MCS/RGB-D Cam |
| Ground truth error/tracking error | 0.576 mm (based on calibration) |
| Simulation latency | 100–280 ms |
| ROS support | Yes |
| Customizable mockups | Yes |

## 4.2 Motion control of robotic manipulators for spacecraft emulation

### *4.2.1 Inverse kinematics: jacobian pseudo-inverse formulation*

Consider an emulated satellite (mockup) whose center of mass (COM) coincides with the tool center point (TCP) of the robot end-effector. A satellite is defined as an additional link to the UR10e robot system. Any position or velocity imparted to this satellite is conceived as the rotation and translation of its COM. Because of the coincidence of the satellite COM with the TCP of the robot, the velocities and positions of the emulated satellite are the same as those of the robot TCP (i.e., robot end-effector). Based on the assumptions that the rendezvous scenario involves no interactions (i.e., no force and torque transfer) and the manipulator can achieve highly accurate motion control, the velocity of the mockup is related to the joint velocities of the manipulator as Eq. (26):

$$\dot{\mathbb{x}} = \mathbb{J}\dot{\mathbb{q}} \tag{26}$$

where $\dot{\mathbb{x}} \in \mathbb{R}^m$, $\dot{\mathbb{q}} \in \mathbb{R}^n$, and $\mathbb{J} \in \mathbb{R}^{n\times m}$ is the Jacobian matrix that maps the joint velocities to Cartesian velocities and depends on the frame in which the end-effector velocities are modeled. For a non-redundant manipulator, $m = n$, and $\mathbb{J}$ is a square matrix, whereas for a redundant manipulator, $m < n$, the Jacobian is a non-square matrix. The Jacobian $\mathbb{J}$ for an isolated 6-DoF robot (i.e., no rails included in the system) is a square matrix. If the Jacobian does not lose rank, the inverse relationship holds true:

$$\dot{\mathbb{q}} = \mathbb{J}^{-\mathbb{1}}\dot{\mathbb{x}} \tag{27}$$

and enables the identification of the corresponding joint states to achieve certain Cartesian positions and/or velocities.

For a redundant manipulator, as for an augmented 7-DoF robot (i.e., rails are included in the system), the Jacobian is no longer a square matrix but a rectangular one, thus offering at least one degree of redundancy. In such cases, the inverse solution is calculated using the pseudo-inverse of the Jacobian matrix:

$$\dot{\mathbb{q}} = \mathbb{J}^{+}\dot{\mathbb{x}} \tag{28}$$

to identify the corresponding joint states. Considering this mapping between the Cartesian and joint spaces in terms of differential motion, the relation is described as

$$\mathrm{d}\mathbb{q} = \mathbb{J}^{+}\mathrm{d}\mathbb{x} \tag{29}$$

thereby relating the incremental positions between the Cartesian and joint spaces. Despite being computationally efficient, the Jacobian pseudo-inverse $\mathbb{J}^{+}$ suffers from instabilities in the singular and near-singular configurations. The utilization of this method for robot motion control involves a trade-off between computational efficiency and instability. Other possible solutions include the use of the damped pseudo-inverse and selectively damped least-squares solutions. Although inherently stable for singular configurations, unlike the Jacobian pseudo-inverse method, these methods require dynamic decoupling and exhibit a loss of accuracy with increased damping [35]. A practical solution to successfully achieve Jacobian inverse-based control without adverse effects is to avoid the singularity altogether. However, the overall workspace of the robot manipulator is not used efficiently. The motivation for using the Jacobian inverse method for motion control is twofold. First, it is computationally efficient and highly accurate. Second, because of the availability of an off-the-shelf implementation along with singularity and collision avoidance [36], the implementation has been adapted to the laboratory setup for the on-ground validation of on-orbital scenarios.

The limited workspace of the manipulator, combined with high joint velocities in singular configurations, also results in discrepancies between the simulated motion states and the motion states emulated by the TCP manipulator. To mitigate these effects, velocity scaling is implemented as

$$\dot{\mathbb{q}} = \begin{cases} \dot{\mathbb{q}}, & \lambda(\mathbb{J}) \leqslant \lambda_{\mathrm{l}} \\ \dot{\mathbb{q}} \cdot \left(1 - \dfrac{\lambda(\mathbb{J}) - \lambda_{\mathrm{l}}}{\lambda_{\mathrm{u}} - \lambda_{\mathrm{l}}}\right), & \lambda_{\mathrm{l}} < \lambda(\mathbb{J}) < \lambda_{\mathrm{u}} \\ \mathbf{0}, & \lambda(\mathbb{J}) \geqslant \lambda_{\mathrm{u}} \end{cases} \tag{30}$$

and scale down the velocities as the satellite (mockup) reaches kinematic singularities. Here, the condition number $\lambda(\mathbb{J})$ is used to assess the sensitivity of the inverse kinematic solution. A high value of $\lambda(\mathbb{J})$ implies that a small change in the input may cause a large amplification of the output (as observed for robot singularities when small Cartesian velocities cause large joint velocities). It reflects the proximity to a singular configuration. The definition of $\lambda(\mathbb{J})$ is approximated from the eigenvalues of $\mathbb{J}$ as Eq. (31):

$$\lambda(\mathbb{J}) \approx \frac{\sigma_{\max}(\mathbb{J})}{\sigma_{\min}(\mathbb{J})} \tag{31}$$

where $\sigma_i$ denotes the singular values of the Jacobian. If the condition number of the Jacobian matrix exceeds the lower threshold $\lambda_{\mathrm{l}}$, the joint velocities begin to decelerate. In contrast, if the condition number exceeds the upper threshold $\lambda_{\mathrm{u}}$, the joint velocities are set to zero, preventing any motion. For the UR10e configuration in the Zero-G Lab, the lower limit $\lambda_{\mathrm{l}}$ is set to 17, and the upper limit $\lambda_{\mathrm{u}}$ is 30.

### *4.2.2 Control architecture*

The HIL experiments involve the systematic and safe operation of a robot and rail hardware, timely data acquisition for navigation, and solving orbital dynamics for appropriate satellite motions. An elaborate representation of the implementation is presented as a block diagram in Fig. 7. The prominent aspects of the implementation are the Jacobian pseudo-inverse control method highlighted under the green block and the ODS, which numerically solves the satellite motion to identify waypoints for the robots to follow. A combined ROS interface is implemented to operate two robotic hardware (robot and rail) as one. The end-effector motion (as a result of the combined motion of the joints) is obtained via a forward transformation. In the ground-truth data acquisition block, based on the ROS TF2 package [37] combined transformations are obtained not only from the robotic hardware but also from pose estimation from cameras and motion capture system. Any mockup added becomes part of the serial kinematic chain, whose transformation relative to the robot control frames (e.g., base-link, tool link, etc.) is easily obtained via the ground-truth data acquisition block. This block diagram explains both the chaser and target motions. For the motion-actuated robots used in this study, the emulated satellite (mockup) motion was specified using the desired

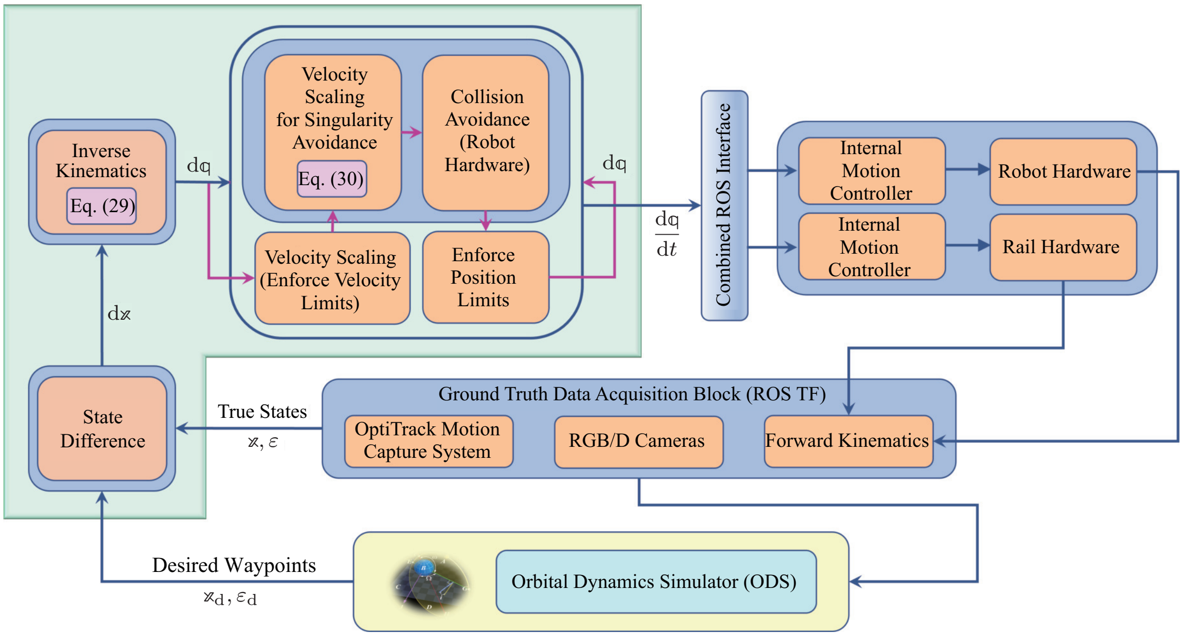

Inverse Kinematics
Eq. (29)
$d\mathbb{q}$
Velocity Scaling for Singularity Avoidance
Eq. (30)
Collision Avoidance (Robot Hardware)
Velocity Scaling (Enforce Velocity Limits)
Enforce Position Limits
$d\mathbb{q}$
$\frac{d\mathbb{q}}{dt}$
Combined ROS Interface
Internal Motion Controller
Robot Hardware
Internal Motion Controller
Rail Hardware
$d\mathbb{x}$
State Difference
True States
$\mathbb{x}, \mathcal{E}$
Ground Truth Data Acquisition Block (ROS TF)
OptiTrack Motion Capture System
RGB/D Cameras
Forward Kinematics
Desired Waypoints
$\mathbb{x}_d, \mathcal{E}_d$
Orbital Dynamics Simulator (ODS)

**Fig. 7** Overall control block diagram.

waypoints $\mathbb{x}_{\text{d}}$ in Cartesian space. The positional error term $\mathbb{e} = \mathbb{x}_{\text{d}} - \mathbb{x}$, where $\mathbb{x}$ is the actual satellite Cartesian position. Note that when emulating the target satellite motion, $\mathbb{x}$ includes the position and orientation of the target satellite from Eq. (19). Similarly, when emulating the chaser motion, $\mathbb{x}$ includes the position and orientation of the chaser satellite from Eq. (19). The orientation of the ground truth is expressed in terms of a quaternion. Consequently, the orientational error $\omega_{\mathbb{e}}$ is defined as Eq. (32):

$$\omega_{\mathbb{e}} = \mathcal{E}_{\text{error}} = \begin{bmatrix}\epsilon_x\\ \epsilon_y\\ \epsilon_z\\ \epsilon_{\text{w}}\end{bmatrix} = \mathcal{E}_{\text{desired}} \cdot \mathcal{E}^{-1}_{\text{current}} \tag{32}$$

which is then expressed in the angle-axis format as

$$\mathbb{a}_{\text{v}} = \begin{bmatrix}\alpha_x\\ \alpha_y\\ \alpha_z\end{bmatrix} = \begin{bmatrix}\frac{\epsilon_x}{\sqrt{1-\epsilon^2_{\text{w}}}}\\ \frac{\epsilon_y}{\sqrt{1-\epsilon^2_{\text{w}}}}\\ \frac{\epsilon_z}{\sqrt{1-\epsilon^2_{\text{w}}}}\end{bmatrix} \tag{33}$$

$$\boldsymbol{\Theta}_{\mathbb{e}} = 2 \cdot \arccos(\epsilon_{\text{w}}) \tag{34}$$

where $\mathbb{a}_{\text{v}}$ is the equivalent axis of rotation, and $\boldsymbol{\Theta}_{\mathbb{e}}$ is the angle of rotation. Considering the linear and angular terms of $d\mathbb{x}$ in Eq. (29) described in Section 4.2.1 separately as $d\mathbb{x} = \begin{bmatrix}\mathbb{v}\\ \omega\end{bmatrix} dt$, the translational terms are mapped into the differential terms via a PID controller as

$$\mathbb{v} = \mathbb{K}_{\text{p}} \cdot \mathbb{e} + \mathbb{K}_{\text{i}} \cdot \int \mathbb{e} dt + \mathbb{K}_{\text{d}} \cdot \dot{\mathbb{e}} \tag{35}$$

and the angular terms are mapped as

$$\boldsymbol{\omega}_{\text{m}} = \mathbb{K}_{\text{p}} \cdot \omega_{\mathbb{e}} + \mathbb{K}_{\text{i}} \cdot \int \omega_{\mathbb{e}} dt + \mathbb{K}_{\text{d}} \cdot \dot{\omega}_{\mathbb{e}} \tag{36}$$

where $\omega_{\text{m}}$ is an intermediate term used to obtain the final differential angular motion along each of its constituent directions:

$$\boldsymbol{\omega} = \begin{bmatrix}\omega_{\text{m}} \cdot \alpha_x\\ \omega_{\text{m}} \cdot \alpha_y\\ \omega_{\text{m}} \cdot \alpha_z\end{bmatrix} \tag{37}$$

The expressions in Eqs. (35) and (37) allow obtaining the differential motion in the joint space, as shown in Eq. (29). Depending on the control mode in which the robot's internal motion controller operates, $d\mathbb{q}$ is either provided as a set point to the position controller while respecting the joint position limits or converted to a velocity term to serve as a set point to the velocity controller while respecting the velocity scaling terms defined by Eq. (30). Table 4 lists the set of parameters used in the experiments performed in this study. Note that although a full PID controller can be used, the choice

**Table 4** Parameter table for Jacobian pseudo-inverse control

| Parameter | 6-DoF case | 7-DoF case |
|---|---|---|
| P, I, D ($x$-axis) | 2.0, 0.021, 0.0 | 5.0, 0.021, 0.0 |
| P, I, D ($y$-axis) | 2.0, 0.021, 0.0 | 5.0, 0.021, 0.0 |
| P, I, D ($z$-axis) | 2.0, 0.021, 0.0 | 5.0, 0.021, 0.0 |
| P, I, D (angle-axis) | 0.5, 0.0, 0.0 | 5.0, 0.0, 0.0 |
| $\lambda_{\text{l}}$, $\lambda_{\text{u}}$ | 17, 30 | 17, 30 |

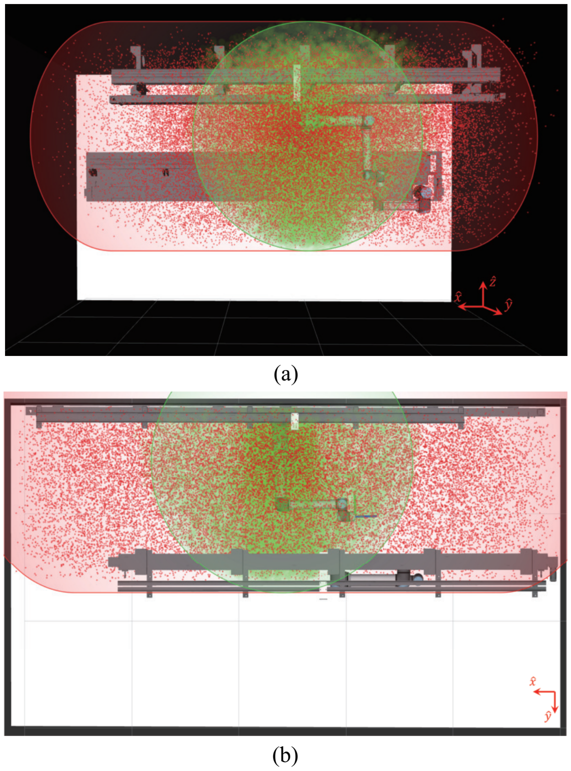
of gains in this study was sufficient to provide accurate tracking results. Although the gains for terms P, I, and D may depend on various factors, selecting higher P gains was sufficient for a successful 7-DoF implementation. The identification of the optimal gains was beyond the scope of this study.

# 5 Experiments and results

## 5.1 Robotic testbed assessment

### *5.1.1 Workspace augmentation for enhanced motion emulation*

As described in the previous section, although avoiding singularity guarantees stable robot performance, restrictions on the usable workspace persist. In such a scenario, the incorporation of an additional joint, such as a linear axis, increases the redundancy of the overall system and augments the volume of the workspace. This enhanced workspace volume is critical for simulating longer rendezvous trajectories. In this research, maximal workspace reachability was also considered. Figure 8 depicts the workspace augmentation concept. A 6-DoF system that comprises only robotic arms can access a limited workspace and is marked by a green point cloud for both ceiling and wall robots. The green point cloud almost forms a spherical sector volume, in which the delivered waypoints are reached using the robot alone. By comparison, the red point cloud indicates an enhanced workspace, thus enabling the emulation of longer rendezvous trajectories. Among the possible configurations in the point cloud, the addition of boundary constraints and obstacles in the vicinity of robotic hardware makes the workspace more hemispherical. Furthermore, as demonstrated in Fig. 9, the augmented workspace of the robots partially overlaps and enables the emulation of various interaction scenarios such as docking and berthing.


(a)

(b)

**Fig. 8** Visualization: robot workspace with 6 DoFs (green) vs. augmented workspace with the usage of rail, i.e., 7 DoFs (red). In these demonstrations, the actual workspace is indicated by a point cloud, and its limits are approximately marked with a geometrical volume. (a) Ceiling robot (side view of the laboratory). (b) Wall robot (top view of the laboratory).

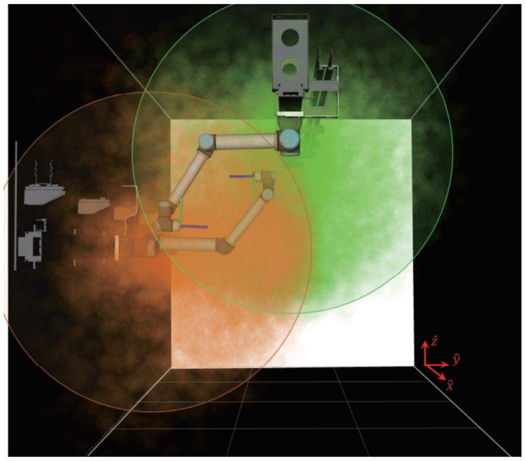


**Fig. 9** Overlap of augmented workspaces of ceiling (green) and wall robots (orange). View: front view of the laboratory. The actual workspace is indicated by a point cloud, and its limits are approximately marked with a geometrical volume.

### *5.1.2 Workspace utilization*

Understanding the reachability and workspace overlap, albeit intuitively, serves two functions: (1) setting up a feasible position of the robot base for each experiment and (2) effective utilization of the workspace [38]. An augmented workspace enables more configurations that robots can achieve; however, different configurations affect joint state motions differently. The impact of the joint state motion on achieving Cartesian waypoints influences the usefulness of workspace augmentation. Mathematically, this impact can be understood from the manipulability index [39], which is defined by

$$\mathbb{M} = \sqrt{\det(\mathbb{J}\mathbb{J}^{\mathrm{T}})} = \sigma_1\sigma_2\cdots\sigma_m \tag{38}$$

where $\sigma_1, \sigma_2, \cdots, \sigma_m$ are the singular values of the Jacobian matrix. The effects of the Jacobian matrix in physical space are shown in Fig. 10, where the singular values denote appropriate transformation along the principal axes. Matrices $\mathbb{U}$ and $\mathbb{V}$ correspond to the left and right singular vectors that denote the principal directions at the final and initial times, respectively. As the robot approaches a singular configuration, at least one of the singular values collapses to 0, i.e., $\sigma_i \to 0$, indicating that any change in joint states along corresponding principal axes $\mathbb{V}_i$ will have no impact on the final Cartesian velocity in the direction defined by $\mathbb{U}_i$. Consequently, the manipulability index deteriorates as the robot approaches a singular configuration. The emulation of any satellite (mockup) motion causes the manipulability index $\mathbb{M}$ to approach zero when the robot reaches its workspace limit. Beyond this point, joint space motion may not necessarily drive the desired motion in Cartesian space. Recall that the robots may stop before the manipulability index becomes exactly zero because of the safety limits adopted in Eqs. (30) and (31).

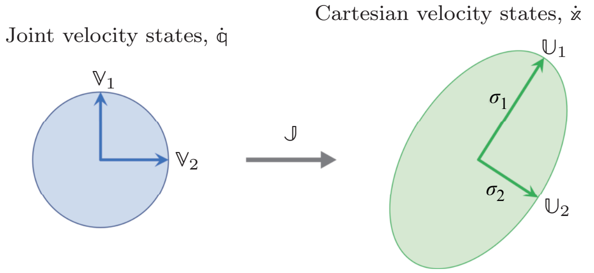


**Fig. 10** State change along principal directions mapped by the Jacobian matrix.

Two sample trajectories were imparted to demonstrate the limitations of the 6-DoF robot system in comparison with the 7-DoF robot–rail system. These trajectories, as shown in Fig. 11, were carefully selected to drive the robots towards two different singular configurations: one that moved towards the "external" limits and the other that remained within the "internal" limits of the geometric volume encompassed by the ceiling arm with a fixed base. The manipulability indices and corresponding joint states for these cases are illustrated in Figs. 12(a) and 12(b), respectively. For convenience, all values for the 6-DoF system are marked with solid lines, and those for the 7-DoF system are marked with dashed lines. Note

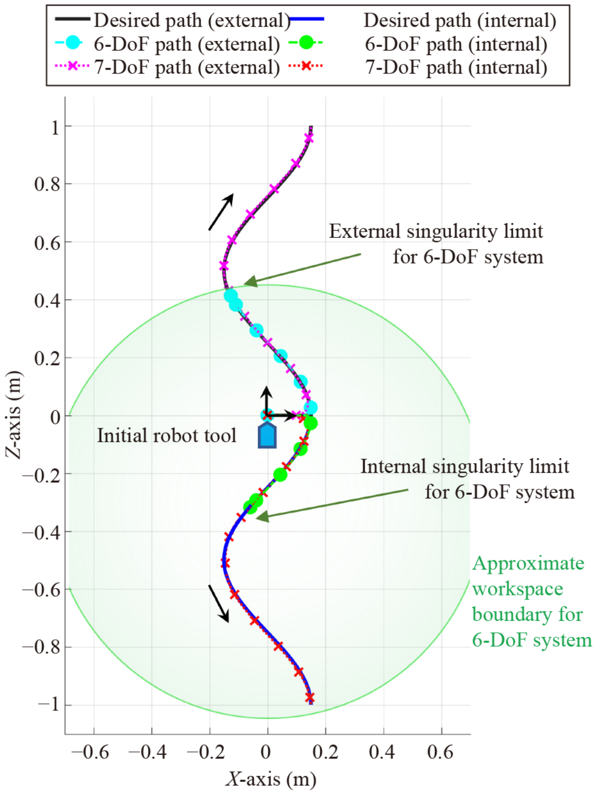
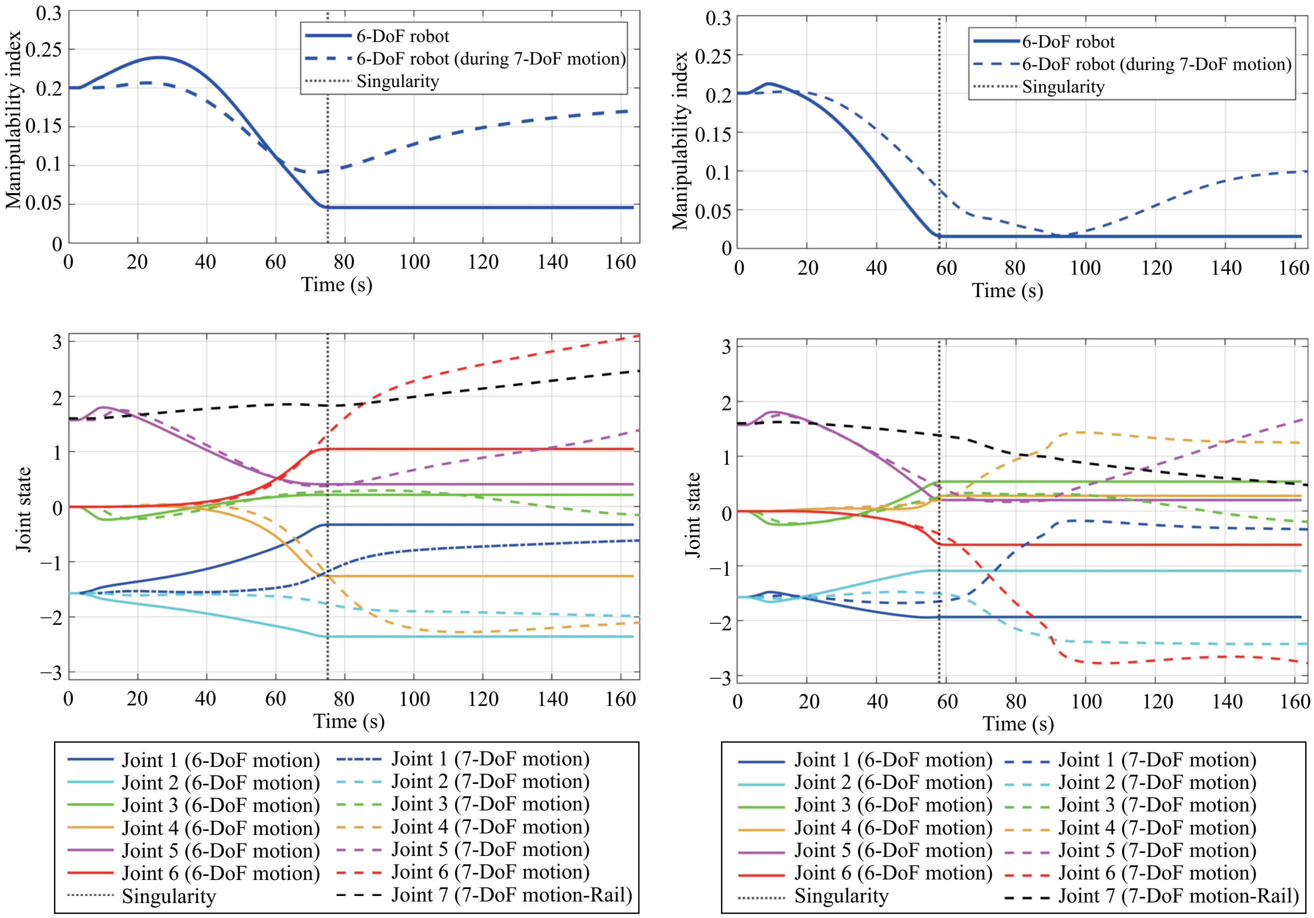


**Fig. 11** Trajectories that include singularity at external and internal boundaries.

that in a 6-DoF system, the Jacobian is a 6 × 6 matrix. However, when the same manipulator is considered as part of a 7-DoF serial kinematic chain in the system, the overall Jacobian is a 6 × 7 matrix. For consistency, the submatrix of $\mathbb{J}$ that corresponds to the same joint states as those of the 6-DoF system is used for calculating $\mathbb{M}$. The effect of the manipulability index on the joint velocities is explained as follows:

- **"External" trajectory**

  In a 6-DoF system, the trajectory that leads the robot into a singularity at the external boundary of its workspace results in an "elbow singularity" (owing to the arm being fully stretched with elbow joint position values close to zero). As the robot approaches the singularity, it cannot maintain the desired satellite trajectory and is compelled to halt close to the boundary (via Eq. (30)) to avoid unstable behavior. In the 7-DoF system augmented by using the rails, the additional linear DoF enables complete emulation of the desired trajectory. As shown in Fig. 12(a), the quantity $\mathbb{M}$ for a 6-DoF system degrades over time (approaching close to zero) as the robot approaches the singularity and eventually flattens out, and no



(a)

(b)

**Fig. 12** Comparison of manipulability for internal and external type trajectories. (a) External trajectory. (Top) Manipulability index for a 6-DoF system, and a 6-DoF system during an augmented 7-DoF motion. (Bottom) Joint state profile for both 6- and 7-DoF systems. (b) Internal trajectory. (Top) Manipulability index for a 6-DoF system, and a 6-DoF system during an augmented 7-DoF motion. (Bottom) Joint state profile for both 6- and 7-DoF systems.

subsequent motion occurs for the remaining duration of the trajectory. In contrast, all the joints in the 7-DoF system are operational for the entire duration, of which a significant portion of the linear motion is performed by the rails. The manipulability index of the 7-DoF system exhibits a consistent evolution and does not plateau with time.

- **"Internal" trajectory**

  Evident in Fig. 11, the trajectory that lies within the internal workspace volume of a 6-DoF UR10e system forces the robot into another singular configuration known as "shoulder singularity" (owing to the wrist being close to the shoulder origin in one or more axes). Similar to the previous case, the additional linear DoF in the 7-DoF system enables the execution of the desired trajectory. Figure 12(b) describes that the joint positions in the 7-DoF system evolve to avoid singularity for its entire duration; however, for a 6-DoF system, the quantity $\mathbb{M}$ collapses at a certain time, beyond which robot does not execute the desired waypoints.

  Note that the 7-DoF system suffers from its singularities; however, this is beyond the scope of this paper. The use of an additional rail is sufficient to improve the emulation capability of robots when performing HIL experiments.

## 5.2 Hardware-in-the-loop GNC validation

The HIL tests enable the integration of actual sensors, hardware components, and simulation environments to assess the real-time performance of GNC systems [40]. In this investigation, a visible-light camera is introduced to deliver pose information, whereas the guidance and control scheme operates on an object-oriented Python library and communicates over the ROS to test the real-time feasibility of the entire GNC framework. Finally, satellite mockups mounted on robotic manipulators enable the capture of relative satellite poses and the visualization of rendezvous operations.

Successful execution of the rendezvous motion in the Zero-G Lab requires adaptability to the limited available space and the ability to accommodate different initial conditions. The guidance and control package implemented in this study allows for flexibility by accepting any initial relative position and orientation between the chaser and target satellites. The target satellite mockup is securely mounted on the ceiling end-effector, whereas the chaser satellite, represented by a camera, is mounted on the wall arm. A schematic representation of the Zero-G Lab configured for the HIL experiment, along with the associated frames, is shown in Fig. 13. To ensure consistency, the CW rotating frame is selected at the center of the target and kept it fixed throughout the entire experiment. The prominent direction of the approach, i.e., the $\hat{y}$ direction in the CW frame (the direction of orbital velocity), is set along the larger dimension of the laboratory, one that is parallel to the rails. The relative position and orientation of the chaser are determined either from the initial camera pose readings or from the OptiTrack MCS, which serves as the ground truth. Four distinct scenarios are tested under varying lighting conditions and the initial conditions remain unchanged in each case.

### 5.2.1 *Scenario description*

The HIL experiment focuses on an LEO with a radius of 7000 km. Initially, the chaser and target satellites are separated by approximately 2.81 m. Detailed values for the different axes and weighting parameters are listed in Table 2. The aim of the rendezvous is to achieve a final separation of 0.5 m while maintaining a collision avoidance radius of the same value. The complete rendezvous motion is designed to be completed within 480 s with 60 equally spaced control segments identified for the process. Pose observations are captured at intervals of 0.5 s throughout the proximity operations. Optimization techniques are applied to obtain a guidance path, which serves as a reference for all HIL experiments. Figure 14 shows the guidance path along with snapshots demonstrating the orientation of the target and chaser satellites. Note that, for the physics and visualization in the HIL tests, the target and chaser satellites are modeled as 6U CubeSats. As the chaser approaches the target, pose values were obtained as feedback, and the linear control inputs are adjusted accordingly to keep the chaser near the reference path.

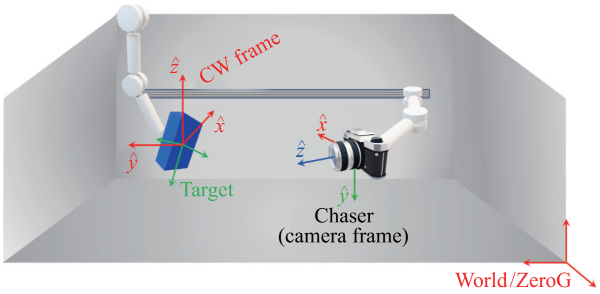


**Fig. 13** Configuration of the Zero-G Lab with associated frames for HIL testing. The camera's optical frame axis is specifically marked in RGB colors.

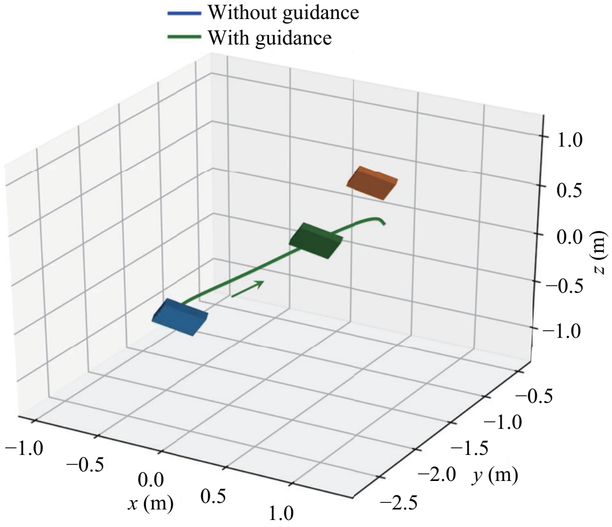


**Fig. 14** Optimal reference path calculated using the guidance strategy. The natural path (without guidance) is obscured by the chaser's body. A snapshot of the orientation of the chaser (in green) and target (in orange) satellite is also provided for reference. Note that the orientation is also compensated with the trajectory.

In satellite operations utilizing vision-based navigation, environmental factors can contribute to significant uncertainties in pose estimation. The intensity and direction of light, as well as the surface reflectivity of the target satellite, are crucial in pose determination. Deliberate attempts are made to test some of these challenging cases in the Zero-G Lab. Four different positions for the light source are considered, as depicted

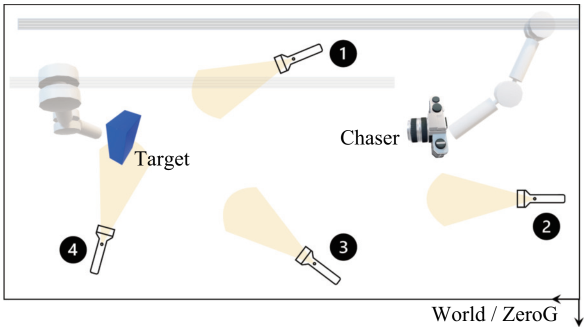


**Fig. 15** Four different positions for light sources considered for the experiments, as identified in the schematic of the laboratory space (top view).

in Fig. 15. For simplicity, these locations are referred to as LP1, LP2, LP3, and LP4, where LP means “light position”, and the number following corresponds to the one in Fig. 15. These specific light positions induce partial and complete shadows being cast by the chaser, and inevitably by the robots, onto the target satellite, as shown in Fig. 16. Light at LP4 also induced lens flare at intermittent periods of flight (see Fig. 16(f)). In addition to environmental factors, operation beyond the calibrated specifications of the camera in terms of separation distances and viewing angles contributes to inaccurate predictions of relative satellite pose. Even when the target satellite is fully visible in the camera frame, instances of inaccurate pose predictions are observed during laboratory experiments, and shown in Fig. 17. Moreover, certain orientations of the chaser

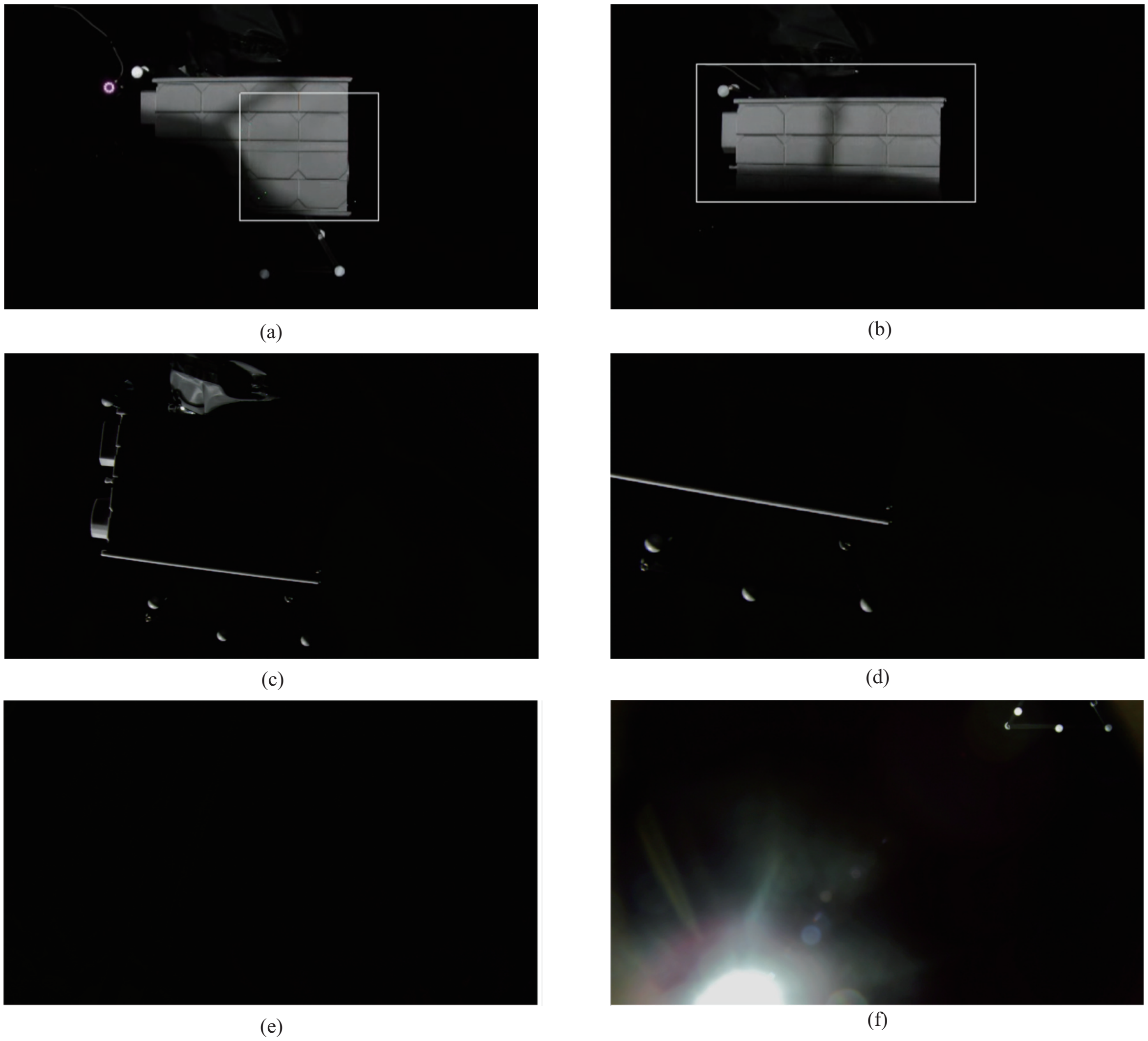
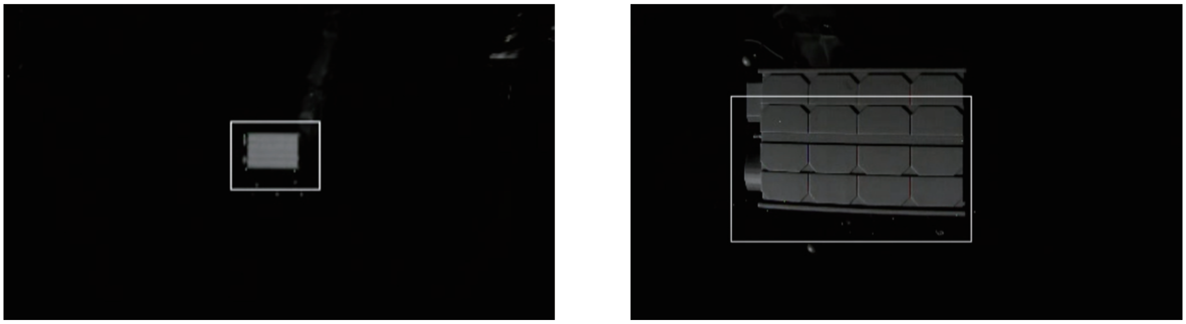


**Fig. 16** High uncertainty in pose estimation as a result of environmental factors. Some observed cases were as follows: (a, b) partial shadow of the chaser or other bodies cast on the target causing incorrect pose prediction; (c) only one edge of the target visible; (d) partial edge of the target visible; (e) complete darkness or no pose detected; and (f) lens flare.

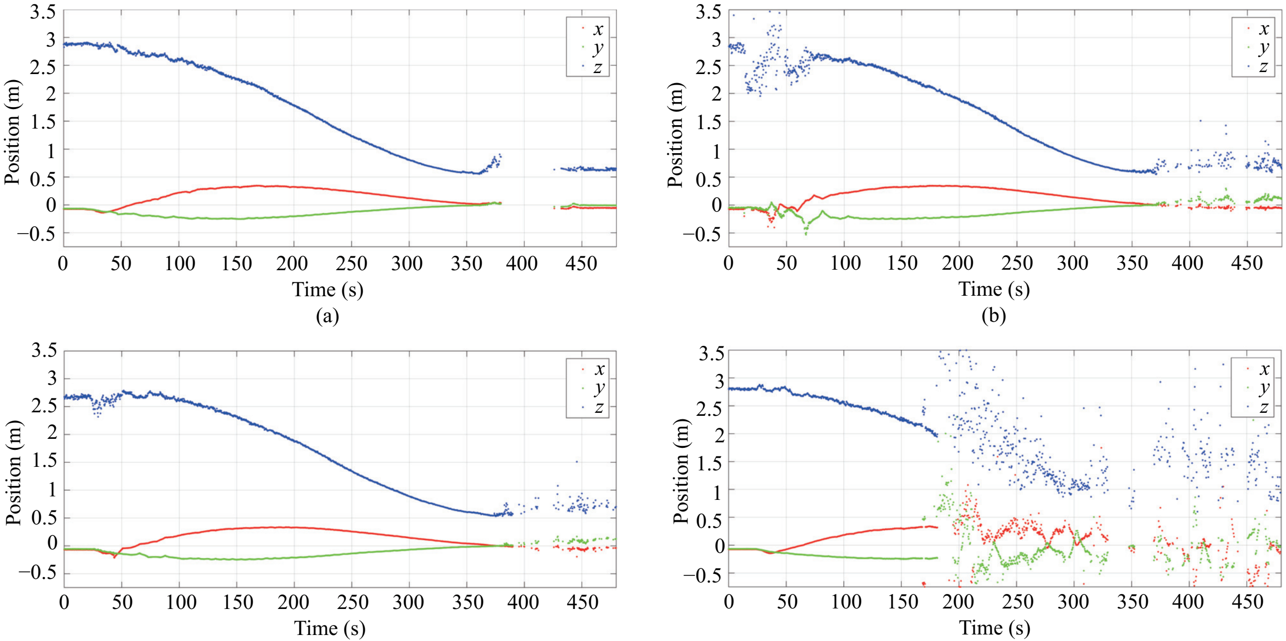
camera relative to the target do not provide favorable conditions for image identification, resulting in frequent periods of flight where no pose is determined.

Pose observations are initially identified in the local camera frame (depicted in Fig. 13). Subsequently, these values are transformed into the CW frame, which serves as the reference frame for the primary states within the Kalman filter and controller. Note that the camera frame may not always align with the CW frame because it evolves along with the orientation of the chaser satellite. Figure 18 displays the pose values determined in the camera frame for each case. During intermittent time intervals, the camera fails to capture images of the target, resulting in no available pose values. This occurrence is evident across all cases and is depicted as a gap in the pose data. Among the four light positions tested, LP1 exhibited the most favorable conditions, with minimal shadow regions. The light sources at LP2 and LP3 had more instances of partial and complete shadows, making accurate pose estimation challenging, and resulting in higher uncertainties in certain areas. The presence of a lens flare significantly affected pose observations in LP4.



(a) (b)

**Fig. 17** Limitations in modeling causes of uncertainty in pose estimation. Some recurring cases are the following: (a) misidentification of mounting points on external apparatus as part of the target, and (b) limited images within the training set for the pose algorithm.



(a) (b) (c) (d)

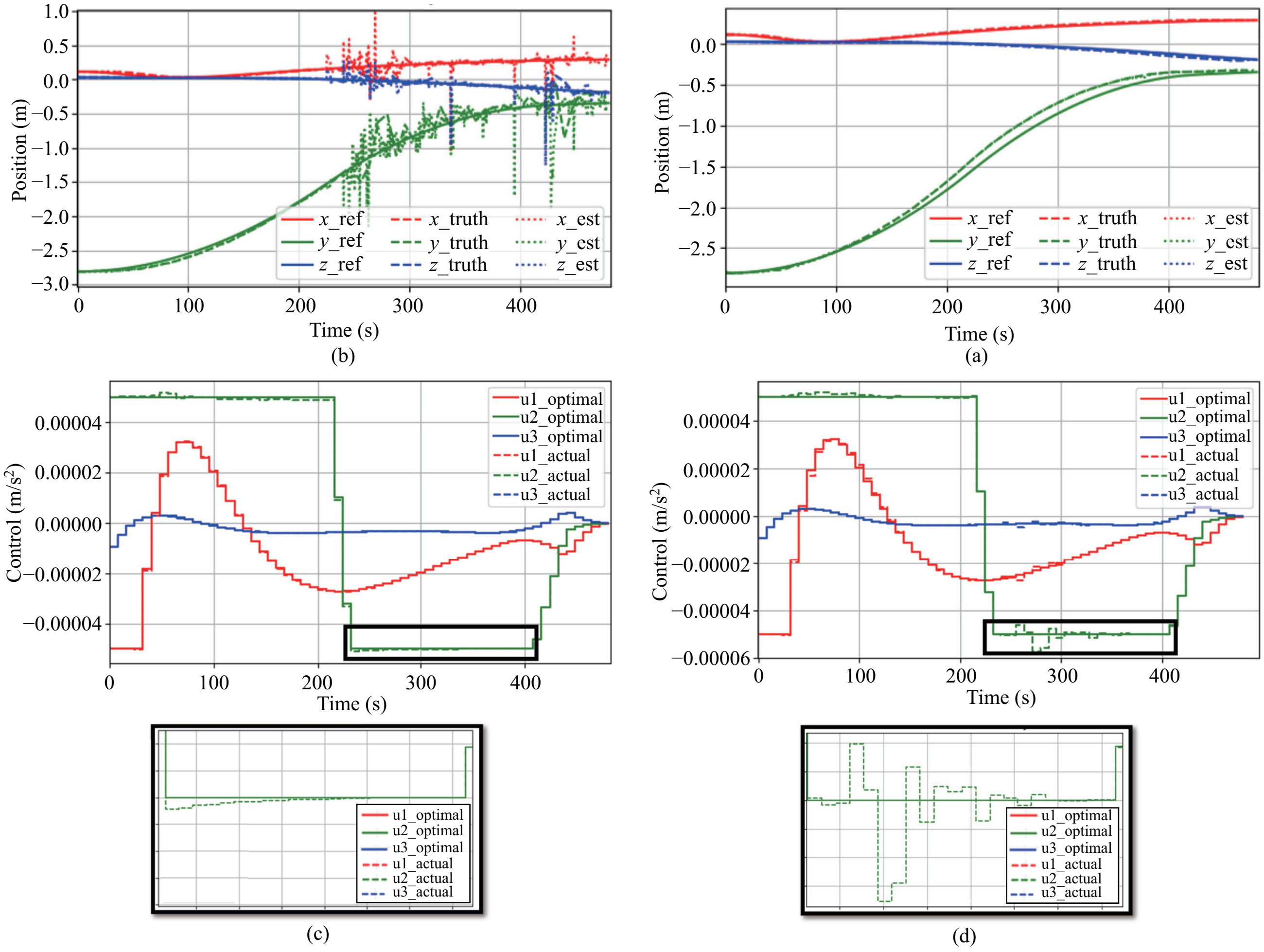
**Fig. 18** Pose estimated through images captured under distinct lighting conditions as the chaser approached the target. Positions for the light source were varied as in Fig. 15. Values are indicated in the camera's optical frame (see Fig. 13). (a) Light source at position 1. (b) Light source at position 2. (c) Light source at position 3. (d) Light source at position 4.

Additionally, imprecise camera parameters contributes to variations in pose quality, yielding accurate observations at specific viewing distances and angles while producing inferior results in other areas.

### *5.2.2 Controller characteristics*

To assess the influence of pose estimation on satellite control during the rendezvous process, the state and control histories of all four cases are analyzed. Among these four cases, the light source at position 1, or case LP1, is observed to be the most favorable for feature detection, thereby offering reliable navigation. In contrast, case LP4 was detrimental to pose detection given the direct illumination towards the camera. Specifically, Figs. 19 and 20 describe the state and control histories for the most conducive case (LP1) and least conducive case (LP4), respectively. Consequently, a comparison between LP1 and LP4 cases is conducted, which highlights the following:

- Figure 19 shows the history of position compensation, the reference path, and the corresponding control acceleration levels. The position history in Figs. 19(a) and 19(b) shows the reference path the satellite is intended to follow in ideal conditions (indicated by the suffix "_ref"), the ground truth or the true path that the chaser satellite follows incorporating actual maneuvers ("_truth"), and the estimated path after processing pose observations through a Kalman filter ("_est"). Similarly, as shown in Figs. 19(c) and 19(d), the optimal control accelerations are calculated for ideal motion along the reference path ("_optimal"), whereas the actual control outputs accounted for state compensation ("_actual"). For LP1, the errors



(b)

(a)

(c)

(d)

**Fig. 19** Position and acceleration history for the chaser satellites while it approached the target as illumination was varied by placing the light source at positions 1 and 4. These lighting conditions deliver edge cases for pose estimations. (a) Position history for LP1. (b) Position history for LP4. (c) Acceleration history for LP1. (d) Acceleration history for LP4.

in determining position states are significantly fewer than that of case LP4, where the pose estimator interprets states with significant deviation owing to lens flare. Poor pose observations results in poor estimates and an inferior maneuver. Figure 19(d) shows erratic maneuvers in regions with poor position observation in comparison with Fig. 19(c), where maneuvers are relatively gradual. Compensating pose errors result in control maneuvers, and hence the true satellite path does not precisely overlap the reference guidance path. A striking observation in Figs. 19(a) and 19(b) is that, for LP1, less uncertainty exists with minor bias in position states, whereas less bias and high uncertainty exists in LP4. Nevertheless, with an active Kalman filter and frequent observation of as much as 2 Hz (two observations every second), the overall control history could achieve a successful rendezvous control. Total control costs vary according to the quality of pose observations.

- Figure 20 illustrates the chronological evolution of the chaser's orientation relative to the target, along with the corresponding control torques applied. The orientation history reveals the reference path, true path, and estimated values (Figs. 20(a) and 20(b)). In both LP1 and LP4 scenarios, a drift is noticeable during the initial stages, indicating a bias in the pose estimation process. Specifically, for LP4, the camera's orientation induces lens flare when it reaches a certain position (close to 180 s); the subsequent pose observations exhibit significant uncertainty, resulting from frequent and abrupt changes. The estimated orientation manifest chaotic behavior, resulting in a significant torque demand for compensation for these deviations. Consequently, Fig. 20(d) for LP4 shows more pronounced fluctuations in the applied torque throughout the experiment compared with Fig. 20(c). Despite such large and chaotic errors, the controller is sufficiently potent to deliver a successful rendezvous at

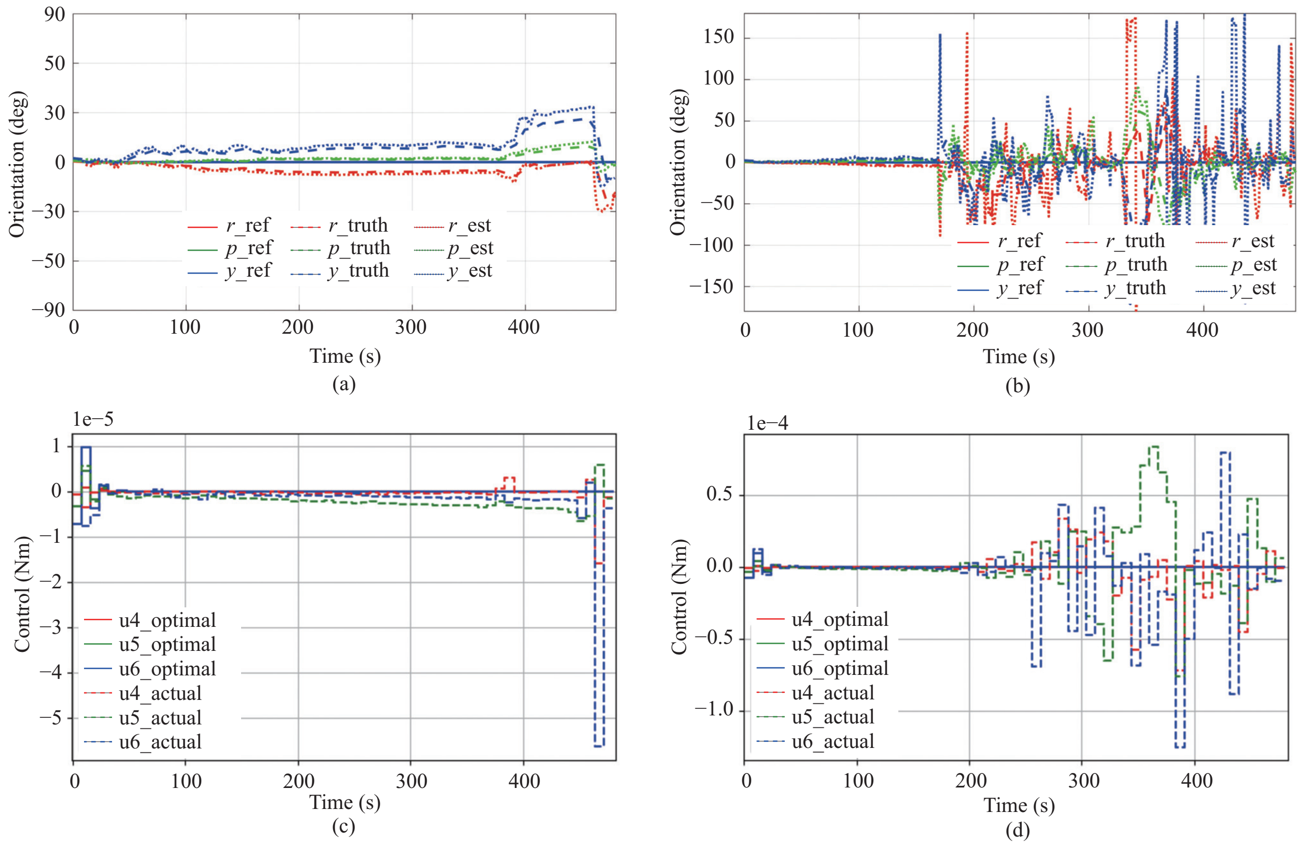


**Fig. 20** Orientation and torque history for the chaser satellites as they approached the target while illumination was varied by placing the light source at positions 1 and 4. These lighting conditions deliver edge cases for pose estimations. (a) Orientation history for LP1 in the roll, pitch, and yaw. (b) Orientation history for LP4 in roll, pitch, and yaw. (c) Torque history for LP1. (d) Torque history for LP4.

the cost of large and rapidly changing control torques. Note that while delivering these control actions might be mathematically feasible, practical implementation is still challenging.

Four scenarios were tested using the same foundational baseline rendezvous path but with varying light positions. As observed in LP1, the navigation reliably estimated the pose, maintaining deviations within 10 cm in the position and 30 degrees in the orientation. Correspondingly, a successful rendezvous resulted in a maximum acceleration of $5 \times 10^{-5}$ m/s$^2$ (equivalent to a 50 μN thrust for a 1 kg satellite) and a maximum thrust of $6 \times 10^{-5}$ Nm torque for attitude correction. Conversely, in LP4, which was characterized by relatively weaker navigation, the predicted position deviations reached up to 1 m, with orientation errors of up to 180 degrees (pose flip). Consequently, trajectory corrections demanded a more rigorous response, necessitating a maximum acceleration of $6 \times 10^{-5}$ m/s$^2$ (equivalent to a 60 μN thrust for a 1 kg satellite) and a maximum thrust for a torque of $1.3 \times 10^{-4}$ Nm for attitude correction. Additionally, a more rapid change in torque was required. Cases LP2 and LP3 demonstrated performance levels between those of LP1 and LP4. Real-world scenarios may require a combination of navigation approaches to mitigate risks, as pure vision-based navigation, as observed in LP4, can be risky. The latency between the hardware and software, ranging from a median of 100–280 ms to a maximum of 1.6 s, caused minor deviations, which were effectively compensated for by the algorithm.

Despite the presence of inherent errors in real-world pose observations, which may not necessarily follow a Gaussian distribution and can vary under different environmental conditions, the estimator and controller demonstrated effective coherence. Navigation accuracy is important in influencing control costs. The experimental findings indicated that the absence of pose observations is less detrimental and preferred over incorrect poses. When pose data are absent, the covariance within the Kalman filter increases, indicating reduced confidence in the estimated states. Nevertheless, maneuvers are still delivered based on the predictions of the *apriori* states. In contrast, frequent and incorrect pose data result in poor estimates with higher confidence levels within the Kalman filter. Thus, a tradeoff exists between the frequency and quality of observations.

## 6 Conclusions

The HIL tests are conducted to validate the feasibility of the visual GNC approach in real-world scenarios. All system components, including the pose-identification camera, robot end-effectors for target and chaser motion, and controller, are interconnected through a common network, ensuring efficient real-time communication. The integration of the linear rail into the robotic system enabled an extended range of motion while overcoming the potential singular configurations of the 6-DoF robot system, both at the internal and external limits of the operable 6-DoF workspace. The mathematical significance of the inverse kinematics solver for configuring appropriate joint motions to deliver the desired Cartesian motion is revealed for both the 6- and 7-DoF systems. With a 7-DoF system, whenever possible, the joint space motions are marginally altered to prevent the manipulability index from collapsing.

The effect of the pose estimation quality on the rendezvous process is evaluated. Conditions such as inadequate lighting, lens flares, and camera operation beyond its calibrated limits across different distances and viewing angles contribute to errors in the navigation estimates. Subsequently, these errors result in higher control costs. However, despite these challenges, the Kalman filter and controller enable efficient performance in tandem, even when pose errors do not precisely follow a Gaussian distribution. The impact of frequent and incorrect poses on control costs is more significant than that in the absence of pose data. In practical applications, multiple navigation approaches can be combined to improve the pose estimation quality. For example, in addition to optical sensors operating in the visible-light spectrum, other types of sensors, such as infrared or radar-based sensors, can be employed. The integration of diverse sensor technologies can enhance the overall accuracy and reliability of pose observations, resulting in more robust and effective rendezvous operations.

## Acknowledgements

This work is supported by the Luxembourg National Research Fund: INTER20/EUROSTARS/15254521/VBN/Olivares Mendez. The project, E115088 - VBN, has received funding from the Eurostars-2 Joint Programme with cofunding from the European Union's Horizon 2020 Research and Innovation Programme.

## Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

## References

[1] Fehse, W. *Automated Rendezvous and Docking of Spacecraft.* Cambridge University Press, **2003**.

[2] Luo, Y. Z., Zhang, J., Tang, G. J. Survey of orbital dynamics and control of space rendezvous. *Chinese Journal of Aeronautics*, **2014**, 27(1): 1–11.

[3] Zimpfer, D., Kachmar, P., Tuohy, S. Autonomous rendezvous, capture and in-space assembly: Past, present and future. In: Proceedings of the 1st Space Exploration Conference: Continuing the Voyage of Discovery, **2005**: AIAA 2005-2523.

[4] Napolano, G., Vela, C., Nocerino, A., Opromolla, R., Grassi, M. A multi-sensor optical relative navigation system for small satellite servicing. *Acta Astronautica*, **2023**, 207: 167–192.

[5] Lai, P. C., Sternberg, D. C., Haw, R. J., Gustafson, E. D., Adell, P. C., Baker, J. D. Lunar Flashlight CubeSat GNC system development. *Acta Astronautica*, **2020**, 173: 425–441.

[6] Pirat, C., Richard-Noca, M., Paccolat, C., Belloni, F., Wiesendanger, R., Courtney, D., Walker, R., Gass, V. Mission design and GNC for in-orbit demonstration of active debris removal technologies with CubeSats. *Acta Astronautica*, **2017**, 130: 114–127.

[7] Roscoe, C. W. T., Westphal, J. J., Mosleh, E. Overview and GNC design of the CubeSat Proximity Operations Demonstration (CPOD) mission. *Acta Astronautica*, **2018**, 153: 410–421.

[8] Benninghoff, H., Boge, T., Tzschichholz, T. Hardware-in-the-loop rendezvous simulation involving an autonomous guidance, navigation and control system. In: Proceedings of the IAA Conference on Dynamics and Control of Space Systems, **2012**: IAA-AAS-DyCoSS1-09-04.

[9] Gao, X. H., Liang, B., Xu, W. F. Attitude determination of large non-cooperative spacecrafts in final approach. In: Proceedings of the 11th International Conference on Control Automation Robotics & Vision, **2010**: 1571–1576.

[10] Vela, C., Fasano, G., Opromolla, R. Pose determination of passively cooperative spacecraft in close proximity using a monocular camera and AruCo markers. *Acta Astronautica*, **2022**, 201: 22–38.

[11] Yoshimitsu, T., Kawaguchi, J., Hashimoto, T., Kubota, T., Uo, M., Morita, H., Shirakawa, K. Hayabusa-final autonomous descent and landing based on target marker tracking. *Acta Astronautica*, **2009**, 65(5): 657–665.

[12] Kramer, E. L., Parker, W. E., Masterson, R. A. Vision-based spacecraft relative pose estimation in variable lighting conditions. In: Proceedings of the IEEE Aerospace Conference, **2022**: 1–12.

[13] Muralidharan, V., Makhdoomi, M. R., Barad, K. R., Amaya-Mejía, L. M., Howell, K. C., Martinez, C., Olivares-Mendez, M. Rendezvous in cislunar halo orbits: Hardware-in-the-loop simulation with coupled orbit and attitude dynamics. *Acta Astronautica*, **2023**, 211: 556–573.

[14] Benninghoff, H., Rems, F., Risse, E. A., Mietner, C. European proximity operations simulator 2.0 (EPOS) - A robotic-based rendezvous and docking simulator. *Journal of Large-Scale Research Facilities JLSRF*, **2017**, 3: A107.

[15] Cassinis, L. P., Menicucci, A., Gill, E., Ahrns, I., Sanchez-Gestido, M. On-ground validation of a CNN-based monocular pose estimation system for uncooperative spacecraft. In: Proceedings of the 8th European Conference on Space Debris, **2021**.

[16] Park, T. H., Bosse, J., D'Amico, S. Robotic testbed for rendezvous and optical navigation: Multi-source calibration and machine learning use cases. *arXiv preprint*, **2021**, arXiv:2108.05529.

[17] Blackswan Space. Mission design simulator: Digital twin for your space mission. **2022**. Available at https://www.blackswan.ltd/mission-design-simulator/

[18] Muralidharan, V., Martinez, C., Zinys, A., Klimavicius, M., Olivares-Mendez, M. Autonomous control for satellite rendezvous in near-Earth orbits. In: Proceedings of the International Conference on Control, Automation and Diagnosis, **2022**: 1–6.

[19] Clohessy, W. H., Wiltshire, R. S. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, **1960**, 27(9): 653–658.

[20] Jezewski, D. J., Donaldson, J. D. An analytic approach to optimal rendezvous using Clohessy–Wiltshire equations. *Journal of the Astronautical Sciences*, **1979**, 27: 293–310.

[21] Palmer, P. Optimal relocation of satellites flying in near-circular-orbit formations. *Journal of Guidance, Control, and Dynamics*, **2006**, 29(3): 519–526.

[22] Prussing, J. E. Optimal two- and three-impulse fixed-time rendezvous in the vicinity of a circular orbit. *AIAA Journal*, **1970**, 8(7): 1221–1228.

[23] Singla, P., Subbarao, K., Junkins, J. L. Adaptive output feedback control for spacecraft rendezvous and docking under measurement uncertainty. *Journal of Guidance, Control, and Dynamics*, **2006**, 29(4): 892–902.

[24] Vaddi, S. S., Alfriend, K. T., Vadali, S. R., Sengupta, P. Formation establishment and reconfiguration using

impulsive control. *Journal of Guidance Control Dynamics*, **2005**, 28(2): 262–268.

[25] Pagone, M., Boggio, M., Novara, C., Vidano, S. A Pontryagin-based NMPC approach for autonomous rendezvous proximity operations. In: Proceedings of the IEEE Aerospace Conference, **2021**: 1–9.

[26] Boley, A. C., Byers, M. Satellite mega-constellations create risks in Low Earth Orbit, the atmosphere and on Earth. *Scientific Reports*, **2021**, 11: 10642.

[27] He, K., Gkioxari, G., Dollár, P., Girshick, R. B. Mask R-CNN. *arXiv preprint*, **2017**, arXiv:1703.06870.

[28] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., *et al.* Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2021**, 43(10): 3349–3364.

[29] Stengel, R. F. *Optimal Control and Estimation.* Courier Corporation, **1994**.

[30] Muralidharan, V., Weiss, A., Kalabic, U. V. Tracking neighboring quasi-satellite orbits around Phobos. *IFAC-PapersOnLine*, **2020**, 53(2): 14906–14911.

[31] Olivares-Méndez, M., Makhdoomi, M. R., Yalçın, B., Bokal, Z., Muralidharan, V., Del Castillo, M. O., Gaudillière, V., Pauly, L., Borgue, O., Alandihallaj, M., *et al.* Zero-G Lab: A multi-purpose facility for emulating space operations. *Journal of Space Safety Engineering*, **2023**, 10: 509–521

[32] Nanoavionics. 6U Nanosatellite Bus M6P. Available at https://nanoavionics.com/small-satellite-buses/%206u-nanosatellite-bus-m6p/

[33] OptiTrack. Prime$^{\mathrm{X}}$-13 camera. Available at https://optitrack.com/cameras/primex-13/

[34] Pauly, L., Jamrozik, M. L., Del Castillo, M. O., Borgue, O., Singh, I. P., Makhdoomi, M. R., Christidi-Loumpasefski, O. O., Gaudilliere, V., Martinez, C., Rathinam, A., *et al.* Lessons from a space lab – An image acquisition perspective. *arXiv preprint*, **2022**, arXiv:2208.08865.

[35] Scherzinger, S., Roennau, A., Dillmann, R. Virtual forward dynamics models for Cartesian robot control. *arXiv preprint*, **2020**, arXiv:2009.11888.

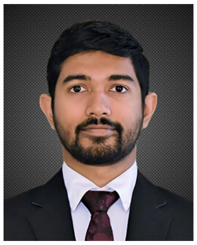
[36] MoveIt. Tutorials - Realtime Arm Servoing. Available at https://ros-planning.github.io/moveit_tutorials/doc/realtime_servo/realtime_servo_tutorial.html

[37] ROS. Wiki: tf2. Available at http://wiki.ros.org/tf2

[38] Makhal, A., Goins, A. K. Reuleaux: Robot base placement by reachability analysis. In: Proceedings of the 2nd IEEE International Conference on Robotic Computing, **2018**: 137–142,

[39] Yoshikawa, T. Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, **1985**, 4(2): 3–9.

[40] Muralidharan, V., Makhdoomi, M. R., Barad, K. R., Amaya Mejia, L. M., Howell, K. C., Martinez Luna, C., Olivares Mendez, M. A. Hardware-in-the-loop proximity operations in cislunar space. In: Proceedings of the International Astronautical Congress, **2022**: IAC–22–C1,4,8,x69397.



**Vivek Muralidharan**, now serving as a flight dynamics engineer at ICEYE in Finland, previously held the position of a research associate at the Interdisciplinary Centre for Security, Reliability & Trust (SnT), University of Luxembourg from 2021 to 2023. He graduated with a bachelor degree in mechanical engineering from the National Institute of Technology Karnataka (NITK), India in 2015. He then received his M.S. and Ph.D. degrees in aeronautics and astronautics from Purdue University, USA, in 2017 and 2021, respectively. While at Purdue University, Dr. Muralidharan worked at the Multi-Body Dynamics Research Group investigating stationkeeping strategies and transfer trajectory design within the framework of cislunar orbits. His research focus includes orbital dynamics, the circular-restricted three-body problem, stationkeeping strategies, orbit determination, as well as guidance, navigation, and control. He has previously worked at the Indian Institute of Space Science and Technology in Thiruvananthapuram, India, and Mitsubishi Electric Research Laboratories (MERL) in Massachusetts, USA. Dr. Muralidharan featured in the 2022 list of “20 under 35” published by Space and Satellite Professionals International (SSPI) and was a finalist for the Luigi G. Napolitano Award at the 73rd International Astronautical Congress (IAC) 2022.

**Mohatashem Reyaz Makhdoomi** is a doctoral researcher at the University of Luxembourg, affiliated with the Space Robotics Research Group. Coming from a background as a robotics engineer, he actively contributes to ongoing research at the ZeroG Lab, particularly focusing on on-orbit emulation using robotic manipulators. Holding a master degree in biomedical engineering and a bachelor degree in mechatronics, he brings a diverse skill set to the field, emphasizing practical applications in robotics and automation. His interests also encompass interaction control of robotics, showcasing a commitment to exploring and contributing to the intersection of engineering and research in the realm of space exploration.

**Augustinas Žinys** received his Master of Science degree in embedded systems with specialization of artificial intelligence from Technology University of Eindhoven, the Netherlands, in 2021. During his studies, his research focus was on object recognition using WiFi CSI data with artificial intelligence. After 4 years, currently, he is an AI specialist with a focus on computer vision and anomaly detection. His current research interests are AI system domain generalization and data efficiency.

**Bronislovas Razgus** received his Master of Science degree in aerospace engineering, spaceflight track from Technology University of Delft, the Netherlands in 2016. Throughout his work and research experience at ESA and DLR, his focus was on AOCS and GNC systems, in particular navigation hardware, algorithms, estimation, and filtering. Currently, he is pursuing a doctoral degree in a joint-Ph.D. program between Vilniustech in Vilnius and Polytechnic University of Milan on GNC algorithms for Active Debris Removal (ADR) missions.

**Marius Klimavičius** received his M.Eng. degree in space systems engineering from the University of Southampton. During the first 10 years of his technical career, he has worked for various organizations in the aerospace sector including General Electric, NanoAvionics, and the European Space Agency where he was developing novel autonomous navigation and space robotics technologies. Over the past 5 years he has been leading Blackswan Space, a space software startup with the goal of automating the current processes and developing better mission design and operation tools used in the space industry. His core expertise lies in space systems engineering, technical leadership, and building global partnerships.

**Miguel Olivares-Mendez** is a professor on space robotics at the Interdisciplinary Centre for Security, Reliability and Trust of the University of Luxembourg. He leads the Space Robotics Research Group (SpaceR), the LunaLab, and the Zero-gravity Lab. In addition, he is the program director of the Interdisciplinary Space Master (ISM) at the University of Luxembourg and the founder and General Chair of the International Conference on Space Robotics (iSpaRo). Dr. Olivares-Mendez received his Eng. degree in computer science from the University of Malaga in 2006, and his M.Sc. and Ph.D. degrees in robotics and automation from the Technical University of Madrid in 2009 and 2013 respectively. During his Ph.D., he was visitor researcher at EPFL (Switzerland) and ARCAA-QUT (Australia). He was awarded with the 2013 Best Ph.D. Thesis award by the European Society for Fuzzy Logic and Technology (EUSFLAT). In May 2013 he joined the Interdisciplinary Center for Security Reliability and Trust (SnT) at the University of Luxembourg (Uni.Lu), as associate researcher in the Automation & Robotics Research Group. In November 2019, he became an assistant professor on space robotics, and in January 2020 head of the SpaceR research group. He is currently the main supervisor of 11 Ph.D. students and 7 PostDocs. He has published over 120 peer-reviewed publications. His main research interests are extreme environment, planetary and orbital robotics for autonomous navigation, situational awareness, perception, machine learning, and multi-robot interaction in autonomous exploration, inspection, and operations.

**Carol Martinez** received her M.Sc. and Ph.D. degrees in robotics and automation from Universidad Politécnica de Madrid (UPM), in 2009 and 2013, respectively, with a focus on computer vision for unmanned aerial vehicles (visual tracking, pose estimation, and control), for which she received the outstanding Ph.D. thesis award by UPM. As a Ph.D. candidate, she was a visiting researcher with the Queensland University of Technology and the University of Bristol, UK, where she developed algorithms for tracking and pose estimation using cameras on-board aerial vehicles. She held positions as a postdoctoral researcher with UPM, and an assistant professor with PUJ, Bogotá, Colombia, from 2015 to 2020. She has led and conducted interdisciplinary research in computer vision, machine learning, and deep learning for process automation (industry and health) and robotics (aerial, industrial, and space). Since 2020, she has been a research scientist with SnT, Space Robotics Research Group, University of Luxembourg (UniLu), where she leads projects in orbital robotics (space debris removal and vision-based navigation), robotic manipulation (control, perception, and learning), and on-ground testing environment (ZeroGLab facility). She is currently a lecturer of the Interdisciplinary Space Master (ISM) Program with UniLu and provides guidance and supervision to master and Ph.D. researchers (as a co and a main supervisor). She is a mechatronics engineer with UPM. Her research interests include perception approaches for the autonomous operation of robots in space and multi-purpose manipulation tasks for planetary and orbital robotics applications.