JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 62, No. 5, September–October 2025

Check for updates

Optimal Capture of Spinning Spacecraft via Deep Learning Vision and Guidance

Alexander Crain,* Kirk Hovell,† Courtney Savytska,‡ and Steve Ulrich§
Carleton University, Ottawa, Ontario K1S 5B6, Canada

https://doi.org/10.2514/1.A36277

This paper addresses the problem of robotic capture of an uncooperative spinning target spacecraft. To do so, a computationally lightweight and real-time implementable guidance, navigation, and control architecture that relies on deep learning as well as pseudospectral optimization is proposed and experimentally validated. Specifically, a convolutional neural-network-driven stereovision pose determination system is first combined with a deep-reinforcement-learning-based guidance algorithm and pose tracking controller to cancel the relative motion between a chaser platform and an uncooperative spinning target platform in real time. Then, real-time tracking of a pseudospectral-based optimal guidance law generated offline deploys a robotic arm while minimizing the overall attitude corrections required to keep the target in view. The integrated experiment carried out using Carleton University's Spacecraft Proximity Operations Testbed (a state-of-the-art planar air bearing facility, introduced in this work) demonstrates the performance of the developed deep learning architecture.

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

I. Introduction

SUCCESSFUL demonstrations of (semi-)autonomous robotic on-orbit servicing of a cooperative target include Engineering Test Satellite VII (1998) [1] and Orbital Express (2007) [2], the term *robotic* referring to the use of a robotic manipulator to grapple and capture the target spacecraft. On the horizon are more challenging robotic missions to capture *uncooperative* (i.e., not equipped with fiducial markers and typically spinning at a constant and unknown rate) target spacecraft: Astroscale's Cleaning Outer Space Mission through Innovative Capture¶ (2026) to remove two defunct satellites, Northrop Grumman's Mission Robotic Vehicle [70] (2025) as a part of the Defense Advanced Research Projects Agency's Robotic Servicing of Geosynchronous Satellites program to deploy Mission Extension Pods and service satellites in geostationary orbit, and ClearSpace-1 [3] (2026) as a part of the European Space Agency's Clean Space initiative which will capture Proba-1 using four robotic arms and remove it from orbit. The On-orbit Servicing, Assembly, and Manufacturing 1 mission [4] (formerly Restore-L) which would have been the first to demonstrate on-orbit refueling of an uncooperative target was recently canceled. However, technology demonstrations of fluid storage and robotic transfer were successfully completed by the Robotic Refueling Mission 1 through 3 programs [71] hosted by the International Space Station (ISS) in 2011, 2015, and 2018, respectively.

The on-orbit servicing industry is still growing, and there is a need to develop more advanced (yet simple) guidance, navigation, and control (GNC) technologies to enable proximity operations, especially with uncooperative target objects. As evidenced by the previously described proposed mission profiles, specific capabilities must include relative pose (i.e., relative position and orientation) determination and tracking as well as robotic deployment and capture maneuvers. The approach and capture must be performed autonomously, in real time, so that the spacecraft is adequately equipped to react to unforeseen circumstances that may otherwise pose a safety risk. Several advancements have been made in each of the aforementioned areas of relative pose determination, pose tracking, and robotic manipulator deployment, as will be discussed in subsequent sections of this paper.

State-of-the-art test beds have similarly been developed over the years to validate such advancements (see, e.g., Refs. [5,6] for a review of spacecraft simulator technologies). Indeed, the use of grounded hardware simulators has always been an important component during the planning and development stages of any spaceborne mission. All hardware and software present in a spacecraft of any size must be validated to the greatest extent possible, as this helps to ensure not only the safety of the spacecraft itself but also the success of the mission. Most spacecraft simulators use air bearings and a smooth surface upon which the bearings can float, providing a near-frictionless environment in which spacecraft GNC algorithms, dynamics, and hardware can all be validated together. Many simulators are already in use at various research institutes. Examples of full six-degree-of-freedom spacecraft simulators include the California Institute of Technology's Multi-Spacecraft Testbed for Autonomy Research (M-STAR) [7], Florida Institute of Technology's Orbital Robotic Interaction, On-Orbit Servicing, and Navigation (ORION) [8], and University of Florida's Advanced Autonomous Multiple Spacecraft (ADAMUS) [9], now at Embry-Riddle Aeronautical University. These facilities use planar air bearings over a granite or epoxy floor for planar translational motion, spherical air bearings for rotational motion, and either a vertical kinematics stage or a counterbalance system for out-of-plane motion enabling the validation of spacecraft proximity operation GNC algorithms. ORION's chaser spacecraft model is additionally equipped with a number of robotic manipulators to simulate autonomous or telerobotic capture. These types of research facilities are not entirely new. NASA's Jet Propulsion Laboratory developed the Formation Control Testbed (FCT) [10] in 2004 to support the Terrestrial Planet Finder (TPF) mission. The FCT was built to simulate and demonstrate six-degree-of-freedom autonomous formation flying and reconfiguration for TPF. Five-degree-of-freedom facilities do not include out-of-plane motion and only use planar and spherical air bearings. The DLR, German Aerospace Center's (DLR) Test Environment for Applications of Multiple Spacecraft (TEAMS) [11], Yonsei University's Autonomous Satellite Test Environment for Rendezvous in Proximity

Received 2 October 2024; accepted for publication 28 February 2025; published online 21 May 2025. Copyright © 2025 by Alexander Crain, Kirk Hovell, Courtney Savytska, and Steve Ulrich. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-6794 to initiate your request. See also AIAA Rights and Permissions https://aiaa.org/publications/publish-with-aiaa/rights-and-permissions/.

*Research Associate, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Member AIAA.

†Research Associate, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive; also Co-Founder and CTO, Obruta Space Solutions. Member AIAA.

‡Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive; formerly published under Courtney Bashnick. Student Member AIAA.

§Associate Professor, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive; steve.ulrich@carleton.ca. Associate Fellow AIAA (Corresponding Author).

¶Astroscale, "COSMIC," https://astroscale.com/missions/cosmic/ [retrieved 27 April 2024].

(ASTERIX) [12], and Georgia Institute of Technology's Autonomous Spacecraft Testing of Robotic Operations in Space (ASTROS) [13] are examples of such facilities that test formation flying techniques, on-orbit servicing, and autonomous rendezvous and docking maneuvers. Three-degree-of-freedom facilities abide by double integrator dynamics and use only planar air bearings on a flat floating surface to achieve translational and rotational motion. Examples of such facilities include the Naval Postgraduate School's Proximity Operation of Spacecraft: Experimental Hardware-in-the-Loop Dynamic (POSEIDYN) test bed [14], Stanford University's Formation Flying Testbed [15], the Polish Academy of Sciences' Micro-gravity Simulator [16], and the Naval Research Laboratory's Gravity Offset Table [17]. Additional capabilities to test close-range rendezvous, docking, and berthing using three-degree-of-freedom air bearing facilities are provided by incorporating six-degree-of-freedom robotic arms mounted on walls, floors, or rail systems for an extra degree of freedom. Examples of such facilities include the University of Luxembourg's Zero-G Lab [18] and the European Space Agency's Orbital Robotics and GNC Laboratory (ORGL) [19] consisting of the Orbital Robotics Bench for Integrated Technology (ORBIT) flat floor and the GNC Rendezvous, Approach and Landing Simulator (GRALS) robotic arm installation. Not all simulators use air-bearing-based systems. The Synchronized Position Hold Engage Re-Orient Experimental Satellites (SPHERES) facility [20] was designed to demonstrate the viability of autonomous proximity operations algorithms. The facility provided six degrees of freedom to evaluate the spacecraft dynamics of multiple simultaneous satellite systems in a microgravity environment. Unlike most other simulators, SPHERES was designed specifically for operation in the Japanese Experiment Module of the ISS. This decommissioned test bed paved the way to Astrobee [21], the successor proximity operations test bed onboard the ISS.

Despite these existing test beds and facilities, the full integration of autonomous computer vision, robotics, guidance, and control into a meaningful proximity operations and capture scenario has yet to be experimentally demonstrated. Still, an early cooperative integrated docking experiment conducted with POSEIDYN [22] has since been followed by notable uncooperative integrated experiments reported in the literature. In Ref. [23], pose tracking was demonstrated using the European Proximity Operations Simulator (EPOS 2.0) at DLR, whereby the servicing spacecraft was equipped with a monocular camera and an image segmentation method based on edge detection. Constant distances of 5 and 20 m were held between the servicer and target. Muralidharan et al. [24] used a convolutional neural network (CNN) based on Mask R-CNN to output keypoints and bounding boxes for target pose detection by a perspective-*n*-point solver at Zero-G Lab. A linear quadratic regulator was used to track an optimal reference trajectory generated offline to a final approach of 0.5 m from the target. Finally, Mahendrakar et al. [25] used a CNN-based on You Only Look Once (YOLOv5) to detect and identify features of a satellite mockup at the ORION laboratory in real time. The relevant information was transmitted to an Artificial Potential Function guidance algorithm that generated an approach trajectory toward identified capture features while avoiding collisions with solar panels. None of the aforementioned demonstrations used a robotic manipulator. Experimental demonstrations of capturing an uncooperative target have thus far been limited to tentacle robots [26] and a custom grasping tool [27] (with a stationary target). Neither technique was validated in a fully integrated GNC architecture.

In this context, the main original contribution of this work is the development and laboratory demonstration of an integrated deep-learning-based vision and guidance approach enabling the relative pose tracking and robotic capture of an *uncooperative* spinning target. Note that the term uncooperative refers to a target that is passively spinning at a constant angular rate, with no communication means to transmit its own dynamical states to the active and that is not equipped with fiducial markers [e.g., circular patterns, ArUco markers, and light-emitting diodes (LEDs)] that help the computer vision software.

The proposed experiment builds upon previously published work discussed earlier in this paper by integrating real-time computer vision, advanced path planning, and control methods, providing a more realistic, yet complex, framework for capturing a spinning target. Furthermore, the approach leverages state-of-the-art deep learning techniques, highlighting the ability to adapt to highly dynamic environments. To carry out the experimental demonstration, Carleton University's Spacecraft Proximity Operations Testbed (SPOT) (a new state-of-the-art planar test bed equipped with a three-link robotic arm, an array of computer vision sensors, and embedded computing resources capable of executing, in real time, modern deep-learning-based algorithms and software) is also introduced in this paper. The laboratory test bed used in this work not only showcases cutting-edge computing and sensing capabilities but also emphasizes the advanced robotics integration required to perform the autonomous capture of a spinning target in realistic conditions.

The remainder of this paper is organized as follows. First, Carleton University's planar air bearing SPOT is introduced. Next, the integrated GNC experiment to demonstrate the advanced capabilities of the SPOT facility is described. The subsequent three sections review the relative pose determination, pose tracking, and optimal robotic manipulator deployment algorithms of the proposed GNC architecture, respectively; Finally, the experimental results are presented before this paper is concluded.

II. Spacecraft Proximity Operations Testbed

The Spacecraft Proximity Operations Testbed at Carleton University has the following main components: a large gravity offset table and three spacecraft simulator platforms (chaser, target, and obstacle) (see Fig. 1). Note that the obstacle platform is not used in the proposed experiment. This obstacle platform could, for example, represent other debris while operating on a defunct satellite. This is particularly true for target vehicles which have been damaged in some way, where the resulting smaller debris may still be present around the target and must be safely navigated around. Also shown in Fig. 1 is the three-link robotic arm installed on the chaser platform, the female docking port on the target platform, and the overhead PhaseSpace motion capture cameras. The platforms use onboard compressed air tanks and air bearings to create a near-frictionless environment. These platforms are self-contained and completely autonomous during experiments. An off-board computer is used to upload the GNC software, initiate experiments, and download the experimental data at the end of each test.

Inspired from recent advances in space-hardened computing and artificial intelligence (AI) graphics processing unit (GPU) capabilities, [72] the platforms used NVIDIA Jetson Xavier NX Development boards, allowing for the execution of guidance and control commands, navigation filters, and the logging of data while also providing an interface for the various instruments onboard. The GNC software is developed in MATLAB® Simulink, where simulations can be performed before running an experiment. Following simulations, the Simulink diagram is compiled to the onboard computer. Once compiled, the executable can be initiated using a custom graphical user interface. Any instrumentation on the platform is integrated into the GNC software using custom Simulink driver blocks. A high-level overview of the test bed is shown in Fig. 2.

The platforms are capable of performing a multitude of different experiments such as tethered stabilization of noncooperative targets, robotic manipulation tasks, collision avoidance, rendezvous and docking maneuvers, and vision-based tracking. Several experiments have already been performed using the facility, including tethered capture of a spinning space debris [28], pose tracking control using the Udwadia–Kalaba framework [29], rendezvous and docking via model predictive control [30], and the adaptive control of a tendon-driven manipulator [31]. The following subsections provide further details on the components of the SPOT facility. A summary of relevant specifications for select components can be found in Table 1.

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

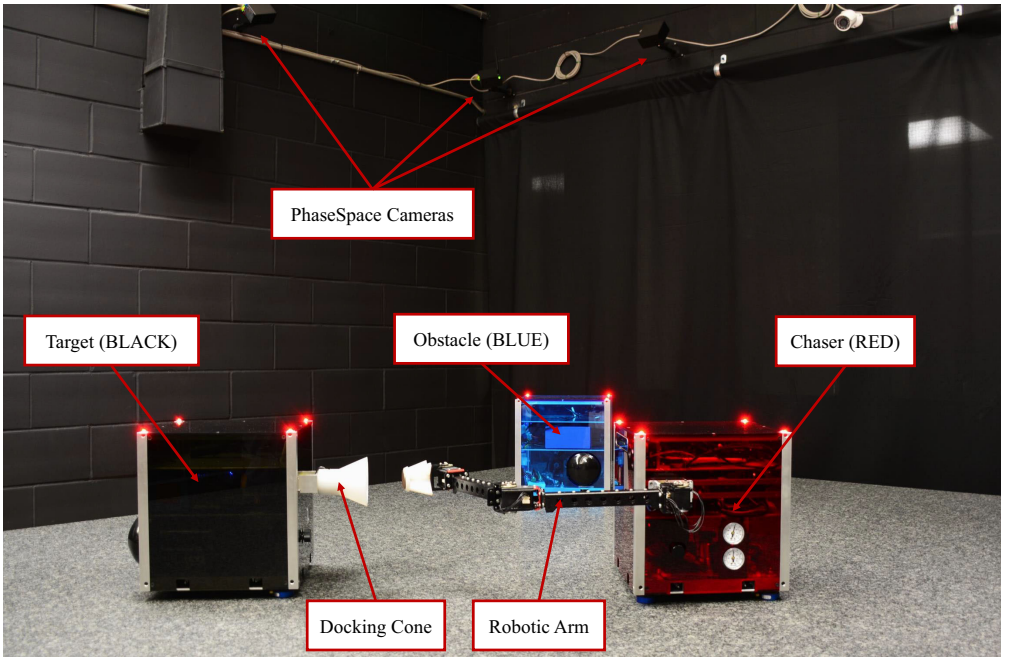


Fig. 1 The spacecraft proximity operations test bed platforms, robotic manipulator, gravity offset table, and overhead motion capture cameras.

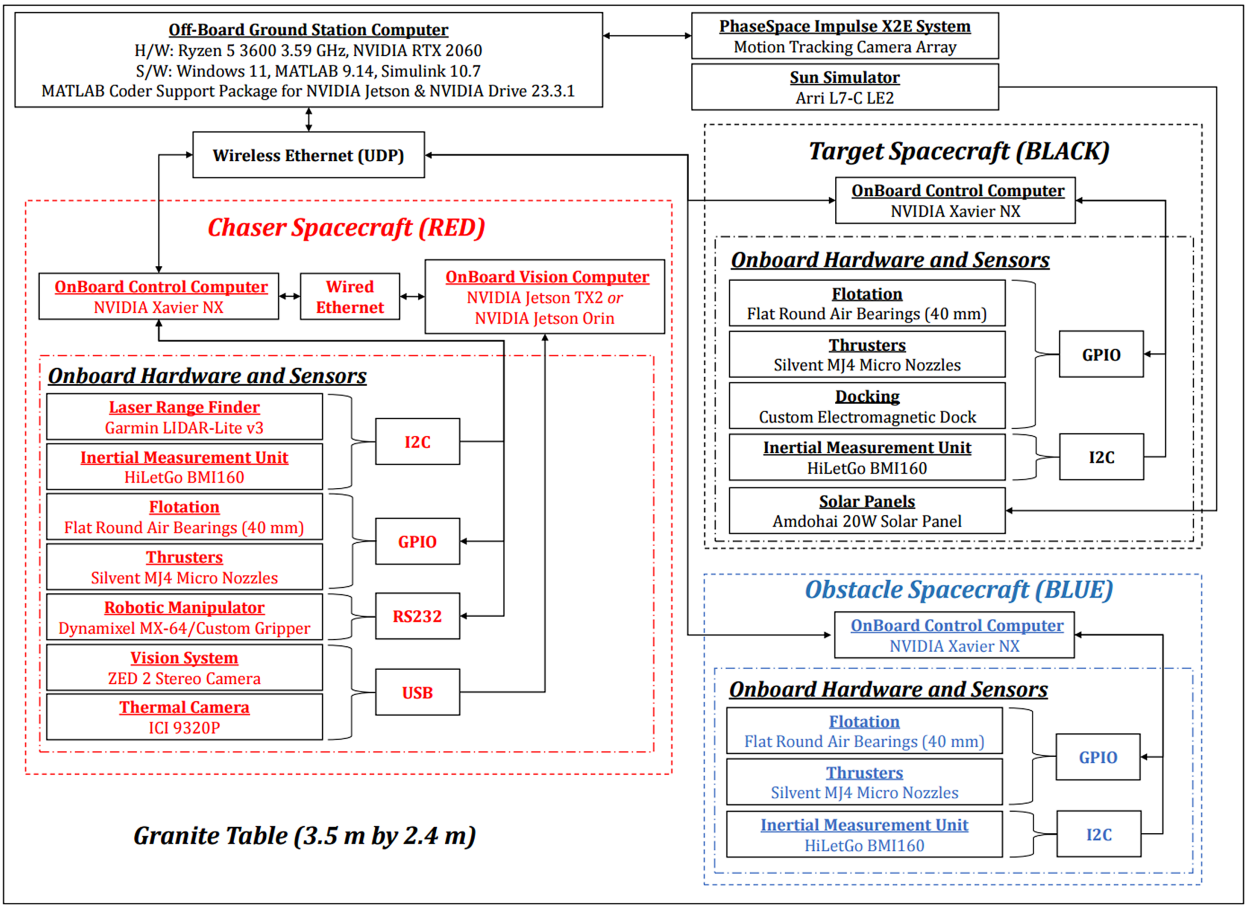


Fig. 2 Block diagram of the spacecraft proximity operations test bed.

A. Floating Surface

Honed by Rock-of-Ages Corporation to U.S. federal "A" specifications, the 3.5 m × 2.4 m table made from a single slab of solid impala black granite is within +/— 0.0007 in. flat across its surface and serves as the interface upon which all spacecraft platforms float. If left uncontrolled on the table, the platforms would slightly drift due to residual gravitational accelerations caused by near negligible surface slope angles, which perturb the dynamics of the platforms in both directions.

B. Spacecraft Simulator Platforms

Each platform has dimensions of 0.3 m × 0.3 m × 0.3 m and consists of three easily removable levels. All propulsion-related

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

Table 1 Specifications of Spacecraft Proximity Operations Testbed components [G = gripper; RM = robotic manipulator (with gripper); DP = docking port (electromagnetic); SP = solar panels]

Category	Property	Chaser (red)	Target (black)	Obstacle (blue)
Platform geometry	Length, width, and height	0.3 m × 0.3 m × 0.3 m	0.3 m × 0.3 m × 0.3 m	0.3 m × 0.3 m × 0.3 m
	Weight[a]	12.852 kg	11.896 kg	11.418 kg
	Inertia[a]	0.222 kg · m^2	0.208 kg · m^2	0.198 kg · m^2
	Center of mass[a] (x, y)	(0.0089 m, −0.0110 m)	(0.0116 m, −0.0141 m)	(0.0066 m, −0.0148 m)
	Accessory add-ons	G, RM, SP	DP, SP	SP
Platform propulsion and air bearings propellant	Propellant	Air	Air	Air
	Number of thrusters	8	8	8
	Maximum thrust (single axis)	0.25 N	0.25 N	0.25 N
	Thruster operating pressure	80 psig	80 psig	80 psig
	Number of air bearings	3	3	3
	Air bearing operating pressure	60 psig	60 psig	60 psig
	Actuators	Solenoid valves	Solenoid valves	Solenoid valves
Dynamixel MX-64	Stall torque (12 V)	6 Nm	— —	— —
	No-load speed	63 rpm	— —	— —
	Weight	126 g	— —	— —
	Resolution	0.088°	— —	— —

Category	Property	Value
Vision system (ZED 2)	Output resolution	Max. 2 × (2208 × 1242) @ 15 frames/s
	Field of view	Max. 110°(H) × 70°(V) × 120°(D)
	Depth range	0.3 to 20 m
	Depth accuracy	<5% up to 15 m
	Weight	166 g
PhaseSpace Impulse X2E motion capture	Number of cameras	10
	Capture rate	Max. 960 frames/s
	Tracking precision	0.01 mm
MATLAB® Simulink	Diagram base rate	Min. 20 Hz–Max. 960 Hz

[a]Platform properties include a 1.352 kg air tank and exclude all docking hardware and accessories.
N·m: a measure of torque.

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

hardware, including the high-pressure compressed air tank, is located on the bottom and sublevels. The battery and required electronics, the inertial measurement unit (IMU), and the onboard computer are housed on the center level. The chaser platform contains the vision system on its top level. The target and chaser platforms further support docking hardware, including a robotic manipulator and a docking interface. Solar panels can be mounted on any platform for additional collision avoidance complexity.

The propulsion system is responsible for the translation and attitude control of the platforms. Air is delivered to the eight thrusters located below the lower deck and is passed through nozzles to produce thrust. Air pressure through the nozzles is 80 psig, which allows the thruster to produce a peak theoretical thrust of 1 N. However, the actual thrust from each thruster is greatly diminished by the pressure loss through the piping, resulting in a thrust closer to 0.25 N.

C. Motion Tracking System

To determine the ground-truth position of the platforms during an experiment, an array of 10 PhaseSpace motion capture cameras, some of which are shown in Fig. 1, track the inertial positions of LEDs, each with their own unique pulse frequency. The reported resolution of this camera system is 0.01 mm, and position measurements are made in the x, y, and z directions. Each platform has LEDs located at the corners of its top panel, as shown in Fig. 1. Using these LEDs, the PhaseSpace software can derive the quaternions of the rigid body to provide a measurement of the attitude as well as track the center of geometry positions of the platforms in inertial space.

D. Computer Vision

The vision sensor suite is located on the top level of the chaser platform and provides various sensors for relative navigation capabilities. A ZED 2 stereo camera connected to the onboard vision computer, an NVIDIA Jetson TX2 development board, via Universal Serial Bus (USB) is used alongside vision-based algorithms. The camera is equipped with a wide-angle eight-element all-glass dual lens with optically correct distortion. The field of view is 110°(H) × 70°(V) × 120°(D). Videos can be recorded at a maximum resolution of 2208 × 1242 pixels per lens at a rate of 15 frames/s. An ICI 9320P thermal camera provides infrared imaging capabilities to detect temperature variations and thermal signatures. This thermal camera enhances the platform's ability to operate in low-visibility conditions and aids in target identification and tracking [32–34]. In addition to the two aforementioned camera sensors, the vision sensor suite also includes a Garmin LIDAR-Lite v3 Laser Rangefinder. This sensor is a high-performance, compact light detection and ranging (LIDAR) that provides precise distance measurements. It operates with a range of up to 40 m and a resolution of 1 cm and can thus provide reliable distance data for real-time path planning and collision avoidance.

E. Robotic Manipulator System

The robotic manipulator as shown in Fig. 1 consists of a commercial-off-the-shelf robotic arm, modified to ensure compatibility with the spacecraft platform as well as to improve the overall rigidity of the linkages. The completed manipulator consists of two linkages attached in series followed by an electromagnetic gripper attached to the distal link. Once the gripper is inserted into the electromagnetic docking port on the target platform, a rigid capture is achieved. Each joint is actuated using a Dynamixel MX-64 actuator. The physical specifications of the manipulator arm are given in Table 2, in which a_i is the distance from joint i to the center of mass (COM) of linkage i and b_i is the distance from the COM of linkage i to joint $i + 1$. A diagram of these parameters can be found in Ref. [35].

F. Inertial Measurement System

All three platforms are equipped with a BMI160 IMU connected to the onboard computer through general purpose input/output (GPIO) pins with communication achieved using inter-integrated circuits (I2C). These IMUs are used primarily to measure the angular velocity of the platforms. Validation of the sensors

Table 2 Parameters of the robotic manipulator and red platform

Body	Body number i	ϕ, deg	a_i, m	b_i, m	m_i, kg	I_i, kg · m^2
Base platform (red)	0	68.284	— —	0.2304	Refer to Table 1	
Link #1	1	— —	0.1933	0.1117	0.3377	3.750×10^{-3}
Link #2	2	— —	0.1993	0.1057	0.3281	3.413×10^{-3}
Link #3	2	— —	0.0621	0.0159	0.0111	5.640×10^{-5}

kg·m^2: a measure of inertia.

was performed through experimentation, and the precision of the sensors was found to exceed the performance of the PhaseSpace system. The noise density of the BMI160 gyroscope measurement was calculated to be $0.0008°/s/\sqrt{Hz}$, which is magnitudes lower than the noise density of the PhaseSpace system (calculated to be $0.0146°/s/\sqrt{Hz}$). This increased noise is due to the PhaseSpace measurement being a derived value obtained from taking the derivative of the angular position data.

G. Control Actuation

The opening and closing of the thrusters is controlled by normally closed solenoid valves. Valves are opened in less than 10 ms through the application of 12 volts of direct current (VDC). Each solenoid is driven by a power metal-oxide-semiconductor field-effect transistor (MOSFET), which is controlled using the onboard computer. Through intelligent on/off cycling of valves, it is possible to achieve any force or torque within the feasible limits of the actuators. To convert control forces and torques into an on/off command sequence, a control mixer was developed. The control mixer translates desired control inputs into individual commands for each thruster using pulse-width modulation (PWM). The implementation of PWM involves generating a high-frequency signal that switches the solenoids on and off to control the average power delivered to the thrusters.

Calculating the required duty cycle for the PWM command is done within the main Simulink diagram. The desired duty cycle is then sent to the thruster actuators through a Python script, which converts the duty cycle to the required on/off command. The control mixer starts by generating the thruster allocation matrix $\boldsymbol{H}$, which maps the thruster configuration to the control forces and torques. This matrix is computed based on the thruster positions and their maximum force outputs. Given that $\boldsymbol{H}$ is generally nonsquare, a pseudoinverse of $\boldsymbol{H}$ is used to ensure feasible solutions for actuator commands. The thrust decay factor, which accounts for the decrease in thruster output when there are multiple thrusters open simultaneously, is also included in the calculations and is derived from the average duty cycle of the thrusters. The system adjusts the thruster commands to ensure all fall within operational limits, achieved by scaling the PWM values and removing negative values. This process involves saturating the thruster outputs and calculating the effective operational time for each thruster.

The SPOT facility as described in this section is used to validate the integrated GNC architecture presented in the remainder of this paper.

III. Deep-Learning-Based Architecture for Autonomous Spacecraft Capture

In this section, a deep-learning-based robotic capture maneuver of an uncooperative spinning target is developed to highlight the advanced capabilities of the SPOT facility in the areas of optimization and deep learning algorithms. More specifically, the experiment involves the use of optimal robotic manipulator deployment strategies for minimal attitude displacements of the chaser platform and deep learning methods for both the real-time pose determination of the target platform and the pose tracking guidance of the chaser platform.

A. Deep-Learning-Based Robotic Capture Maneuver

As illustrated in Fig. 3, the proposed experiment can be divided into four phases, defined as follows:

1) During the preexperiment step, the target platform is commanded to move to its initial position located on the center of the floating surface. During this phase, the robotic manipulator on the chaser is stowed, and the chaser holds its initial condition 1.43 m away from the target with its stereo camera pointing toward the target.

2) Next, the experiment proper begins where the attitude of the target platform is actively controlled to replicate the behavior of a spinning spacecraft rotating at 2.86°/s, slightly more than the nominal spin rate for Envisat [36]. At the same time, the chaser uses its computer vision system to determine, onboard and in real time, the pose of the target. Using this determined pose, the deep-reinforcement-learning-based guidance strategy outputs desired acceleration signals for an onboard controller on the chaser to track. These acceleration signals autonomously and safely guide the chaser to circle around the target with a relative offset of 0.9 m, while in alignment with the target's docking port.

3) Once the chaser is properly aligned with the spinning docking port of the target, the robotic manipulator is deployed from its stowed configuration into its docking configuration. During the deployment, the active attitude control via the deep-reinforcement-learning-based guidance strategy is disabled; instead, the platform attitude is indirectly controlled via the motion of the manipulator. The robotic manipulator follows a trajectory generated by a Gauss pseudospectral optimal guidance and control technique that minimizes the angular acceleration of the chaser platform during the deployment and also ensures the final pose of the chaser platform after deployment is aligned with the target.

4) Once the manipulator is deployed and well-aligned with the target's docking port, active attitude control via the deep-reinforcement-learning-based guidance strategy is restored and

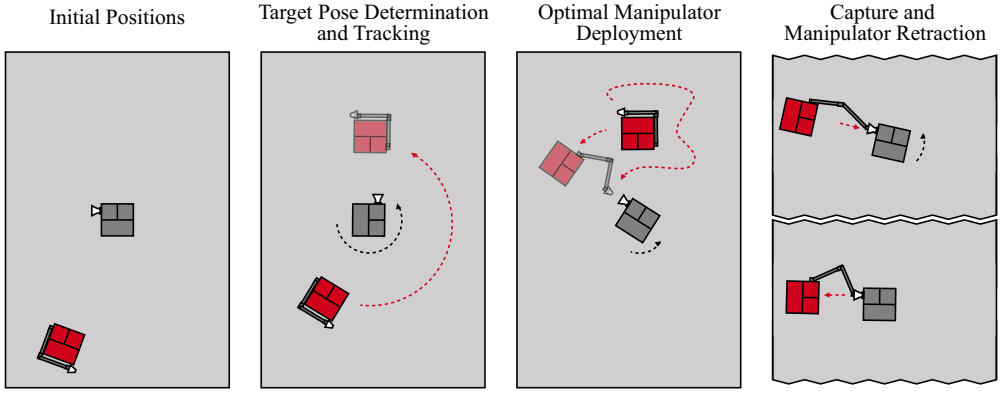


Fig. 3 Proposed experiment: the preexperiment initial positions (left), deep learning target pose determination and pose tracking (center left), optimal manipulator deployment (center right), and target capture (right).

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

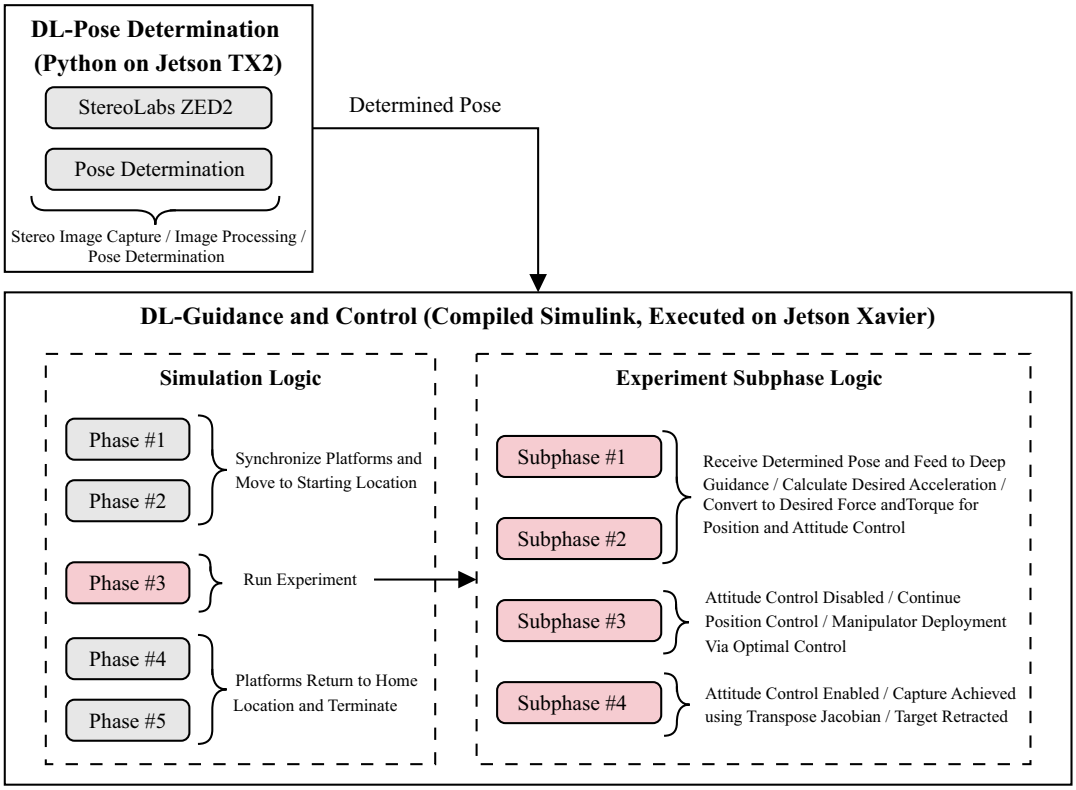


Fig. 4 General overview of the software architecture (DL, deep learning).

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

a final manipulator extension accomplished using a transpose Jacobian control law is performed to magnetically capture the target. Once a rigid connection is established, the manipulator retracts to complete the docking maneuver by pulling the target into a captured position.

B. Integrated Architecture

As indicated in Sec. II, the different algorithms introduced in the work are integrated into a custom MATLAB® Simulink diagram. Broadly speaking, the Simulink diagram is subdivided into *phases* and *subphases*. A high-level view of what is included in each of these is presented in Fig. 4.

To be more specific, the core algorithms in this work are located within Phase #3 and follow the layout described in Sec. III.A. The exception is the stereovision-based pose determination algorithm, which is not executed within the MATLAB® Simulink software; rather, it is executed using a Python script located on a standalone embedded computer, the Jetson TX2 which is labeled as the *On-Board Vision Computer* in Fig. 2. The determined pose is sent to Jetson Xavier, labeled as the *On-Board Control Computer* via user datagram protocol (UDP) to the deep reinforcement learning guidance algorithm to calculate the desired accelerations. The control input is then obtained from the known mass matrix for the chaser. The pose determination and guidance algorithms are always running throughout the subphases, except during Subphase #3 where the manipulator is deployed. During this subphase, the attitude control is disabled so that the effect of the manipulator deployment on the chaser attitude can be observed.

IV. Stereovision-Based Uncooperative Pose Determination

Before an autonomous chaser spacecraft can plan and execute, in real time, its approach and capture actions, it must have knowledge of the pose of the target object. To determine the pose of the target object, the chaser spacecraft uses data collected from its onboard navigation sensors, e.g., visual, infrared, LIDAR, and so on devices. Today, cameras are quickly becoming the preferred sensor as compared to traditionally expensive LIDAR devices, enabling advancements in machine vision algorithms and space-grade computation.

Recently, advancements in machine learning, and specifically CNNs, have revolutionized the field of computer vision beyond the capabilities of traditional vision processing techniques. The availability of large data sets and powerful embedded processors has naturally led to CNNs being at the forefront of modern vision processing, especially as CNNs exhibit increased robustness under adverse lighting conditions and an overall reduction in computational complexity as compared to traditional feature-based methods [37–39]. The model-free pose determination problem of an uncooperative spacecraft as seen using a monocular camera has been tackled in the literature using heat-map-derived feature covariance representations [40] and, in a more traditional manner, feature recognition [41–43] using pretrained CNNs. Using a pretrained network as a baseline network, and further training the model using synthetic images of small spacecraft platforms, aids in overcoming the limited amount of space image training data available.

A well-known problem suffered by monocular cameras is scale invariance, which is solved through the use of stereovision cameras that can additionally resolve depth information at a fraction of the cost of LIDAR systems. To this end, a novel stereovision-based CNN was proposed by Despond and Ulrich [44] and was successfully demonstrated to achieve real-time target pose determination using Carleton University's SPOT. As a result, this CNN-driven stereovision pose determination system is implemented in the GNC architecture developed in this paper.

In this section, the stereovision-based CNN that determines the relative planar pose (x, y, and θ) of the uncooperative target platform with respect to the chaser platform is summarized. In brief, stereo images captured using a camera onboard the chaser platform are used as inputs into the trained CNN. The output of the network is a confidence score indicating the certainty that the target is within the view of the camera. Additionally, the network provides the position and attitude of the target relative to the chaser when the confidence score is sufficiently high.

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

A. Network Architecture

The stereovision-based CNN, as shown in Fig. 5, consists of four major sections. The first is the image capture and preprocessing step, which takes a pair of stereo images from the camera, resizes them to be a square 320×320 pixels, and converts them to grayscale. By reducing the dimensions of the stereo images in this way, the resulting CNN can be relatively smaller and faster, which is necessary for real-time implementation on an embedded platform. Additionally, the grayscale images are stacked one on top of the other to create a single two-channel image, a novel feature in the area of stereovision-based pose determination introduced in Ref. [44]. By doing so, the disparity between the two stereo images that compose a single two-channel image is considered in the architecture of the CNN without any additional work.

The second section of the CNN shown in Fig. 5 is a multikernel layer that processes the two-channel image using three different sized filters. This multikernel layer aims to extract additional spatial information from the raw image, thereby reducing the overall network size. A similar architecture is used by the Google Inception network [45]. For each of the 1×1, 3×3, and 5×5 kernels, there are 32 filters that operate on the image with appropriate stride and padding values such that the output image retains a 320×320 size. The output of all filters is concatenated to a single tensor of dimension $320 \times 320 \times 96$.

The third, and core, section of the CNN shown in Fig. 5 takes inspiration from the YOLOv2 [46] architecture and consists of 10 convolutional layers with 1×1 filters that compress the feature map between 3×3 convolutions [47]. Similar to YOLOv2, the filter number doubles after each pooling layer; however, the final three convolutional layers have a restricted number of filters in order to limit the size of the network.

Finally, the fourth section of the CNN shown in Fig. 5 contains four separate dense (fully connected) networks, each with two layers, that calculate the relative planar pose and a target detection confidence score quantifying whether the target platform is within the field of view of the stereo camera or not. To obtain the relative position components, x and y, the dense network containing 128 nodes is connected to the second layer containing only one node. The second layer is paired with a linear activation function to generate an unbounded output suitable for resolving positions not seen in training. To obtain the relative attitude component θ, the first layer of 256 nodes is connected to a second layer of 128 nodes, which is paired with a softmax activation function to output the probability of the relative orientation of the target with a resolution of $2\pi/128$ rad. Increasing the number of nodes in these layers has the potential to improve accuracy, but at the expense of a larger network. To obtain the target detection confidence score, the first layer of 32 nodes feeds into the second layer consisting of a single node paired with a sigmoid activation function that outputs a bounded value between 0 and 1. If the target is in frame, a 1 is expected; otherwise, a value closer to 0 will be output.

In all sections of the network, convolutional layers employ a leaky Rectified Linear Unit (RELU) activation function, which is used to tackle the well-known vanishing gradient problem [48]. All dense layers use the Exponential Linear Unit activation function, unless otherwise specified. The CNN was developed in Python using TensorFlow 2.0 [49]. The resulting CNN is of a similar size to existing end-to-end CNN pose determination method [50]; it contains approximately 29 million trainable parameters, which is comparable to the current range of existing CNNs (which have a range of approximately 7 to 52.5 million parameters).

B. Custom Data Set and Network Training

Details on the data collection methods and network training can be found in Ref. [44]. In brief, the stereovision-based CNN was trained in two stages using a custom spacecraft stereo imagery data set created for SPOT. The first stage focused on training the three relative pose outputs using images containing the target platform while holding the dense layers associated with the target detection confidence score constant. The target detection confidence layers were trained in a like, but opposite, manner using images both with and without the target platform in frame. To improve generalization of the network, both stages of training implemented dropout layers in between each dense layer. To better generalize the network to different lighting conditions, the brightness of the images was randomly adjusted by a maximum of 25% of their original values. Network training continued until the validation error no longer improved after two iterations.

V. Deep Reinforcement Learning Guidance

The chaser spacecraft uses the target object's pose information obtained from its navigation system in its online guidance algorithm. For spacecraft proximity operations and docking, the guidance algorithm's objective is to reduce the relative motion between the two bodies, typically subject to constraints on the dynamical motion, available thrust, and when applicable, obstacle keep-out zones.

A key challenge in developing a guidance system is balancing the ability to handle complex tasks, dynamics, and/or environments whilst being restricted to a tight computational budget. As a consequence, researchers have turned to machine learning to train neural networks (the universal function approximators) to complete difficult guidance tasks. Evaluating neural networks, once trained, is as simple as evaluating a series of multiplication and addition operations, which offers an appropriate computational load for spacecraft embedded computers.

The use of deep reinforcement learning (DRL) to train guidance policies for spacecraft proximity operations has remained largely unexplored. DRL attempts to discover a policy that maps observations to actions. The observation can be the system state, or it may be any signal within which the state of the system can be inferred. The action is the chosen decision that is relayed to the environment at each time step. The environment executes the action and returns a new observation to the policy. The environment also returns a scalar reward signal, with positive rewards for good outcomes and

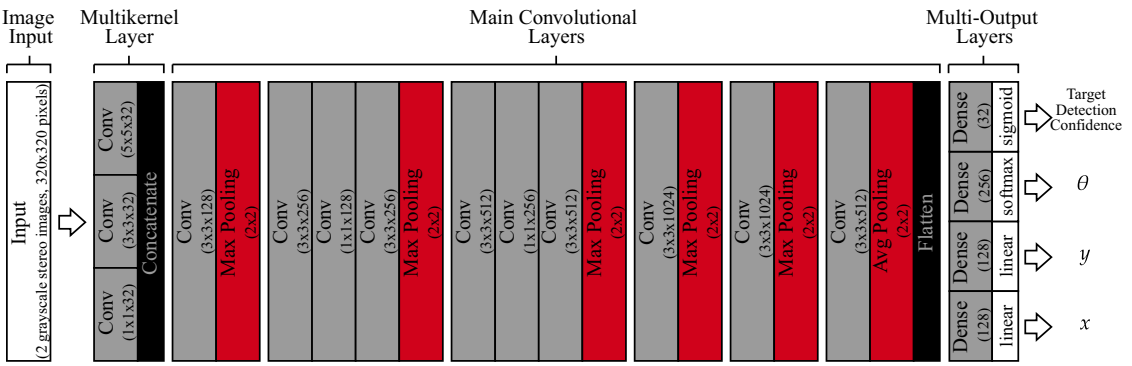


Fig. 5 Visual representation of the stereovision-based CNN (Conv., convolutional).

negative rewards for poor outcomes, which the policy uses to infer the desired actions to take from a given observation. Through trial and error, the policy can learn the desired behavior that is dictated by the reward function. Using this technique, the designer only needs to specify a reward function that encourages the desired behavior; the designer does not need to handcraft the behavior themselves. This simultaneously removes some engineering effort associated with designing behaviors for robots and allows for complex, unbiased behaviors to be learned entirely from scratch.

Guidance policies that directly output thrust commands for three- [51] and six-degree-of-freedom [52] rendezvous and docking maneuvers were trained using Proximal Policy Optimization [53]. The first experimental demonstration of a DRL-based guidance algorithm for spacecraft proximity operations was presented by Hovell and Ulrich [54], who developed a guidance policy trained via the Distributed Distributional Deep Deterministic Policy Gradient (D4PG) algorithm [55]. Additional experiments by the authors demonstrating manipulator-based capture [56] further showed the network's ability to handle significant uncertainties in the states of both the chaser and target platforms during runtime.

In this work, the deep-learning-based guidance algorithm is inspired by the approach developed by Hovell and Ulrich [54]. A guidance policy is trained that takes as input the chaser absolute position and velocity; target relative position and absolute angular rate; and the desired relative position, velocity, and orientation of the chaser with respect to the target and generates acceleration commands tracked by a conventional controller to cancel the relative motion between the spinning target and chaser spacecraft at a standoff distance of 0.9 m.

A. Deep Guidance

The use of DRL enables a spacecraft to learn problem-solving abilities without being explicitly programmed, completing tasks that were previously unsolvable. Even with recent advances in DRL algorithms, however, it still takes hundreds or thousands of attempts before the trial-and-error nature of the learning succeeds. Therefore, it is not feasible to train onboard the spacecraft platform due to time considerations. Instead, training must occur in simulation and the resulting policy be transferred to the spacecraft. This technique often fails due to the simulation-to-reality gap [57], which was addressed by the SPOT facility by developing the Deep Guidance [54] technique.

Briefly, a DRL policy, when applied to robotics, typically receives an observation and calculates a control effort that is commanded to the onboard actuators. However, the simulation-to-reality gap states that models trained in simulation often fail once deployed to reality because the simulated dynamics can never perfectly model the real-world dynamics. For this reason, the Deep Guidance method was developed [54] by taking inspiration from Harris et al. [58] to restrict the DRL policy to perform guidance only. However, unlike Ref. [54], which used a velocity-based proportional-derivative (PD) control law to track the desired velocity generated by the Deep Guidance policy, this work proposes a simpler acceleration-based feedforward strategy. The guidance policy is fed an observation $\boldsymbol{o}$ at time step t and generates an acceleration command $\boldsymbol{a}$, which is used by a conventional feedforward controller to calculate a control effort $\boldsymbol{u}$, via $\boldsymbol{u} = \boldsymbol{M}\boldsymbol{a}$ where $\boldsymbol{M}$ is a diagonal matrix containing the mass and inertia properties of the chaser spacecraft. This control effort is executed on the dynamics to produce the next observation.

B. Reward Function

To encourage the chaser to learn to autonomously track a spinning target at a standoff distance of 0.9 m from which the robotic manipulator may be deployed for capture, a sparse reward function is used. Sparse reward functions only deliver rewards to the agent upon the satisfactory completion of the behavior; rewards are not given at each time step to encourage the learning of a specific behavior.

The desired inertial position of the chaser $\boldsymbol{x}_d$ is

$$\boldsymbol{x}_d = [x_t \quad y_t \quad \theta_t]^T + [-\sin(\theta_t) r_d \quad \cos(\theta_t) r_d \quad -\pi/2]^T \tag{1}$$

where x_t and y_t are the Cartesian inertial coordinates of the target, θ_t is the target's orientation, and r_d is the desired standoff distance (0.9 m in this example). The sparse reward function at time step t outputs a scalar reward r following

$$r = [12.5 \quad 12.5 \quad 5][(\boldsymbol{x}_c - \boldsymbol{x}_d) < [0.05\ \text{m} \quad 0.05\ \text{m} \quad 2\ \text{deg}]^T] \tag{2}$$

where $\boldsymbol{x}_c = [x_c \quad y_c \quad \theta_c]^T$ is the actual chaser's inertial position and orientation; the logical statement along each axis evaluates to a 1 or 0, thereby enabling/disabling that specific reward in the expression.

C. D4PG Implementation

Among the many learning algorithms available, the D4PG [55] algorithm was implemented because it is deterministic and continuous and can be trained in a distributed manner. During training, 10 agents run simulations in parallel to generate training data. The policy and value neural networks are equipped with 400 and 300 neurons in their first and second hidden layers, respectively. The RELU activation function is used in all hidden layers, except a tanh function is used as the output of the policy to ensure the commanded accelerations are bounded and a softmax function is used as the output of the value network to ensure its output is a valid probability distribution. Fifty-one evenly spaced bins are used, distributed across the range $[-100, 750]$ for the value network output. The policy and value networks are trained using the Adam [59] stochastic gradient-descent optimization routine with learning rates of 0.0001. Observations input to the policy are normalized to avoid the vanishing gradients problem. The replay buffer holds 10^6 samples, of which batches of 256 data points are drawn from during each training iteration. A discount factor of 0.99 is used, along with an N-step return of 5. The standard deviation of the noise applied to the output of the policy to encourage exploration is one-third of the action range. After every five training simulations, an additional simulation is run without any exploration noise applied to the policy in order to allow for policy performance during training to be properly evaluated. The neural networks were generated and trained in Python using TensorFlow 1.0.

VI. Optimal Robotic Manipulator Deployment

Using a robotic manipulator to grapple and capture a target object is a complicated problem because the robotic manipulator and chaser spacecraft base form a nonlinear coupled system. Because of conservation laws, extending a manipulator arm produces an attitude disturbance to the base, which may cause communication failures to ground stations or destabilization of the spacecraft. The base spacecraft can be actively controlled through reaction wheels or thrusters to remain stationary throughout the manipulator motion (free-flying scenario); however, the aerospace community generally agrees that the base of the chaser spacecraft should be allowed to move freely (free-floating scenario) during robotic capture maneuvers so as to reduce the risk of unexpected responses of the attitude control system as well as reduce unnecessary actuation and propellant usage [60]. See Ref. [60] for a survey on guidance and control techniques of spacecraft robotic manipulators.

The primary concern facing free-floating robots that has been addressed in the literature is to minimize the attitude disturbances on the base spacecraft caused by the robotic manipulator motion. This concept of disturbance minimization can also be applied to free-flying robots, which reduces the required actuation and propellant usage applied to the spacecraft base to maintain its orientation. The most common approach to solving this issue is through trajectory planning of the robotic manipulator. Indeed, the final orientation of the base spacecraft depends on the path taken by the robotic manipulator. Thus, through carefully considered robotic manipulator motion, the orientation of the base spacecraft may be indirectly controlled. Several approaches to solving the free-floating/free-flying trajectory planning problem have been proposed, including a virtual manipulator [61], a

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

Generalized Jacobian Matrix approach [62], and a reaction null space theory [63,64]. Optimal control approaches have also been proposed that typically minimize the operation time, joint actuation cost, base disturbance throughout or at the maneuver termination, and/or length of the manipulator motion path [65–67]. The experimental validation of optimization-based methods is uncommon due to their high computational demand; however, Crain and Ulrich [35] successfully validated a pseudospectral-based method using Carleton University's SPOT facility. Contrary to similar works [65,66], the endpoint attitude of the base is minimized rather than a running cost, allowing for a greater range of potential trajectories and thus reducing the overall computational effort required to converge to a solution.

In this work, the optimal manipulator deployment algorithm developed by Crain and Ulrich [35] is modified for the integrated GNC scenario. As previously described, the manipulator will be deployed during a tracking maneuver, presenting a new challenge that is not present in the original work. As such, the Mayer and Lagrange components of the optimal control problem must be modified. The Mayer component remains similar to the work by Crain and Ulrich; however, the final target attitude is not the same as the initial attitude before deployment. Instead, the final attitude is offset to account for the change in attitude which occurs during the tracking maneuver. Similarly, the Lagrange component, which was not used in the original work, is used to encourage a trajectory that minimizes changes in the angular rate of the spacecraft. Both of these combined result in a deployment which minimizes the number of required adjustments from the actuators.

A. Dynamics Model

As previously discussed, a space robot is a coupled system consisting of the spacecraft base and manipulator. The nonlinear dynamics equation of a rigid multilink robot can be derived using the system's kinetic and potential energies defined in terms generalized coordinates $\boldsymbol{q}$ (see, e.g., Ref. [68]). The dynamics for any n-link serial manipulator can be written in the intuitive matrix form $\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau}$, where $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the centrifugal or Coriolis matrix, and $\boldsymbol{\tau}$ is the vector of generalized forces and torques. The equations are further partitioned according to Ref. [69]; note that the form of this model that is specific to the SPOT facility will not be presented here but can be found in Ref. [35].

B. Optimal Trajectory Problem

In its general form, an optimal guidance/control problem is formulated so as to minimize a cost function subject to system dynamics and boundary conditions. The optimization problem for this work is written as follows: minimize

$$J(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\tau}) = (q_0(t_f) - q_0(t_0))^2 + \int_{t_0}^{t_f} (\ddot{q}(t))^2 \, \mathrm{d}t \tag{3a}$$

subject to

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{3b}$$

$$\begin{cases} \boldsymbol{x}(t_0) = [0.8 \text{ m}, 1.2 \text{ m}, 2\pi \text{ rad}, \pi/2 \text{ rad}, \pi/2 \text{ rad}, 0 \text{ rad}]^T \\ \dot{\boldsymbol{x}}(t_0) = [6 \text{ mm/s}, -46 \text{ mm/s}, 0.05 \text{ rad/s}, 0 \text{ rad/s}, \\ \qquad 0 \text{ rad/s}, 0 \text{ rad/s}]^T \\ x_{ee}(t_f) = -0.08 \text{ m} \\ y_{ee}(t_f) = 0.70 \text{ m} \\ x_w(t_f) = -0.08 \text{ m} \\ \dot{\boldsymbol{x}}(t_f) = [21 \text{ mm/s}, -29 \text{ mm/s}, 0.05 \text{ rad/s}, 0 \text{ rad/s}, \\ \qquad 0 \text{ rad/s}, 0 \text{ rad/s}]^T \end{cases} \tag{3c}$$

$$t_f = 30 \text{ s} \tag{3d}$$

$$\begin{cases} |u_x| \le 0.5 \text{ N} \\ |u_y| \le 0.5 \text{ N} \\ \tau_0 = 0 \text{ N} \cdot \text{m} \\ |\tau_i| \le 0.05 \text{ N} \cdot \text{m} \ \forall i = 1, 2, 3 \end{cases} \tag{3e}$$

$$|q_i| \le \pi/2 \text{ rad} \ \forall i = 1, 2, 3 \tag{3f}$$

The cost function in Eq. (3a) for the optimal deployment of the manipulator is defined to accomplish two goals. The first goal is to minimize the error in the attitude of the base spacecraft q_0 from its initial position at t_0 at the end of the maneuver (at t_f), knowing that the final attitude of the spacecraft must be $q_0 + \dot{q}_0(t_f - t_0)$. The second goal uses the Lagrange component to maintain a constant angular velocity $\dot{q}_0$. The system is subject to the dynamic constraints Eq. (3b) for the free-flying robot, and kinematic constraints Eq. (3c) on the inertial states $\boldsymbol{x} = [x_0 \ y_0 \ q_0 \ q_1 \ q_2 \ q_3]^T$ and their derivatives at time t_0, on the final end effector and wrist pose (x_{ee}, y_{ee}, x_w), and additionally on the final state derivatives to ensure that the system is moving at the desired velocities by the end of the maneuver, defined in Eq. (3d). The control and path constraints are specified in Eqs. (3e) and (3f), respectively. The maximum inertial forces u_x and u_y on the body of the spacecraft are bounded by the physical limits of the thrusters. The controls τ_i and states q_i for the manipulator links $i = 1, 2, 3$ are bounded by knowing the physical limits of the manipulator. The allowable torque τ_0 on the body is fixed to be $0 \text{ N} \cdot \text{m}$ to keep the attitude uncontrolled. Any states and controls not listed in the constraints section are to be considered unconstrained. The optimal trajectory problem is solved within the MATLAB® Simulink environment using the commercially available optimization tool Sparse Nonlinear Optimizer (SNOPT) via a direct transcription method called the Gauss pseudospectral method.

VII. Experimental Results

For the proposed experiment, Table 3 outlines the initial pose for both the chaser and the target in the inertial frame.

Table 3 Initial conditions of the platforms in experiment

Platform	x, m	y, m	θ, deg
Chaser	3.06	0.61	180
Target	1.75	1.21	−180

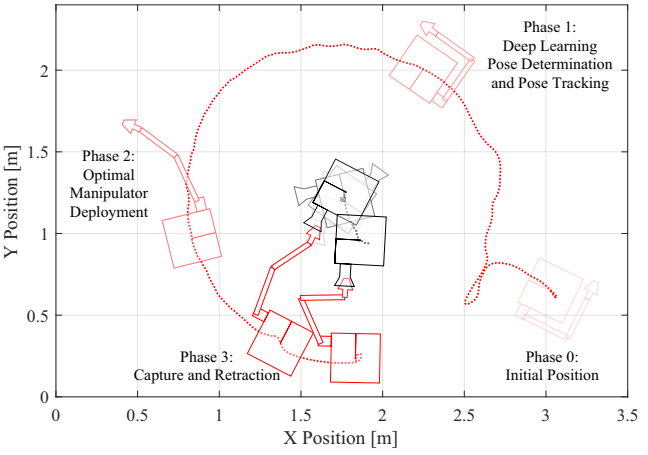
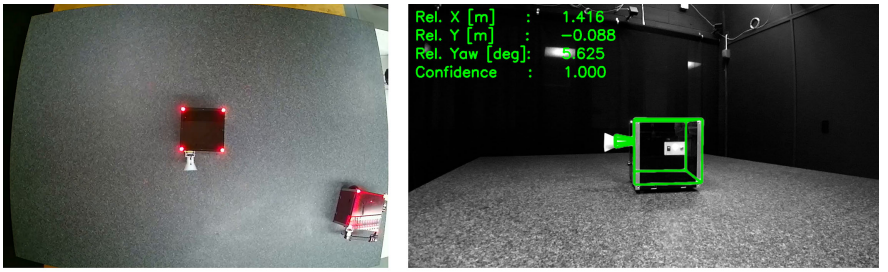
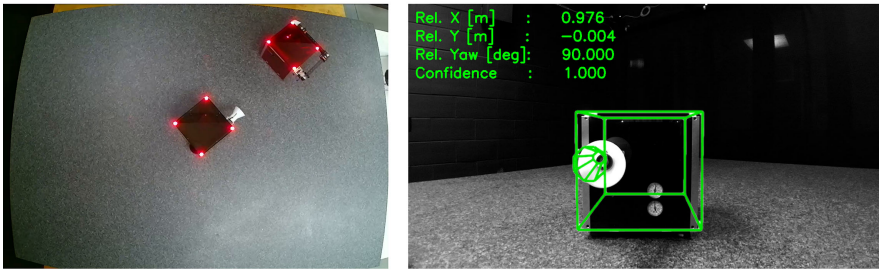


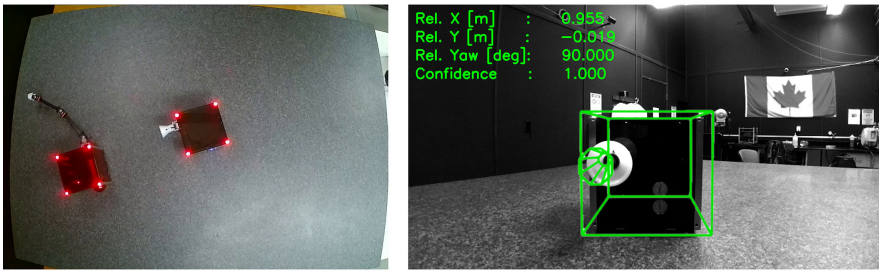
Fig. 6 Visual representation of the successful tracking, deployment, and capture experiment.



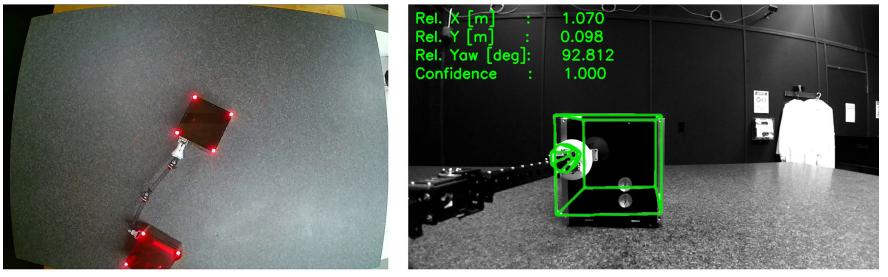
a) Phase 0: Initial position



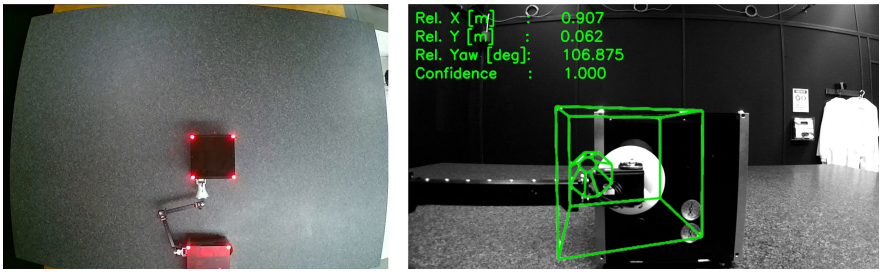
b) Phase 1: Deep learning pose determination and pose tracking



c) Phase 2: Optimal manipulator deployment



d) Phase 3a: Target capture



e) Phase 3b: Manipulator retraction

Fig. 7 Frames captured during the experiment from an overhead camera (left) and the ZED 2 camera (right) with pose determination overlays added in postprocessing using logged experiment data.

A video of the fully integrated experiment can be found online https://youtu.be/egv2pmCyzJM[**]. A visual representation of the experiment, generated using data from the experiment, is provided in Fig. 6. As shown in this figure, the chaser platform successfully completes the capture and retraction maneuver of the spinning target platform through the three-phase approach incorporating deep learning pose determination and tracking and optimal robotic manipulator deployment.

Figure 7 displays images captured during the successful experiment. The left column features images obtained using an overhead camera, while the right column shows images from the ZED 2 camera. Information overlays on the ZED 2 frames were added postexperiment to visually demonstrate the stereovision-based CNN pose determinations.

The error in the relative pose determined from the stereovision-based CNN compared to the PhaseSpace motion capture system

**Video available online at https://youtu.be/egv2pmCyzJM.

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

through all phases of the experiment is presented in Fig. 8. Aside from the errors when the target detection confidence value is low, the largest errors in the pose determination are seen when the chaser begins moving toward the target at the start of Phase 1 and also during the manipulator retraction during Phase 3b, implying the error is a function of the relative separation and velocities of the platforms. The root-mean-squared error (RMSE) of the stereovision-based CNN estimates, with a 95% confidence interval obtained using bootstrapping, are 2.67 ± 0.04 cm, 3.93 ± 0.09 cm, and 2.51 ± 0.09 deg for x, y, and θ, respectively. Measurements collected when the target detection confidence value was below 0.95 were not included in the RMSE.

The distance and orientation of the chaser relative to the target through all phases of the experiment as a function of time are presented in Fig. 9. As previously mentioned, the standoff distance during the experiment was set to 0.9 m to ensure adequate space of the manipulator deployment while staying within the bounds of the gravity offset table. The attitude offset was set to −90 deg to ensure the ZED 2 camera would be pointing toward the target platform when in front of the target's docking port.

During Phase 1, once the chaser reached its nominal tracking position, the standoff distance achieved during experiment was 0.96 ± 0.02 m, and the attitude offset achieved was -89.2 ± 0.7 deg. During Phase 2, the optimal manipulator deployment perturbed the final attitude offset of the chaser by 0.2 deg from its initial attitude offset. The minimum (maximum) attitude offset during Phase 2 was −85.9 deg (−90.8 deg). In Phase 3, the maximum separation of the chaser and target during capture was 1.05 m, and manipulator retraction subsequently brought the separation to 0.72 m.

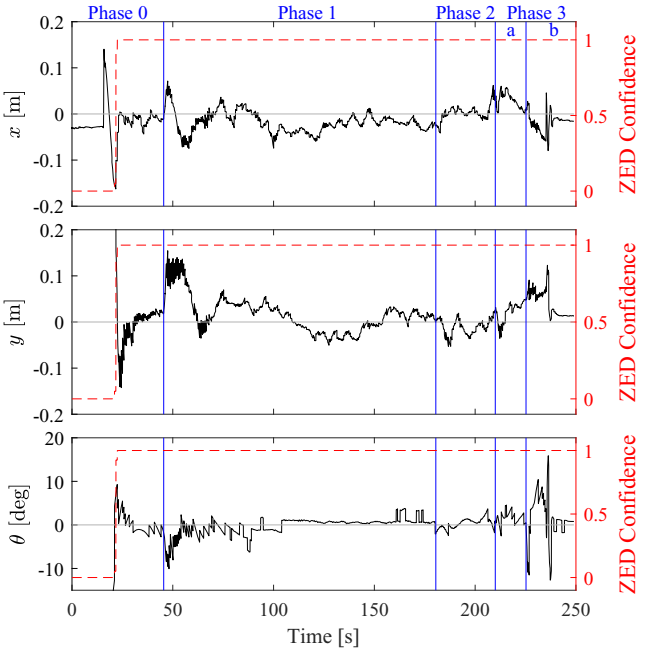


Fig. 8 Relative pose determination error of the stereovision-based CNN and associated target detection confidence value.

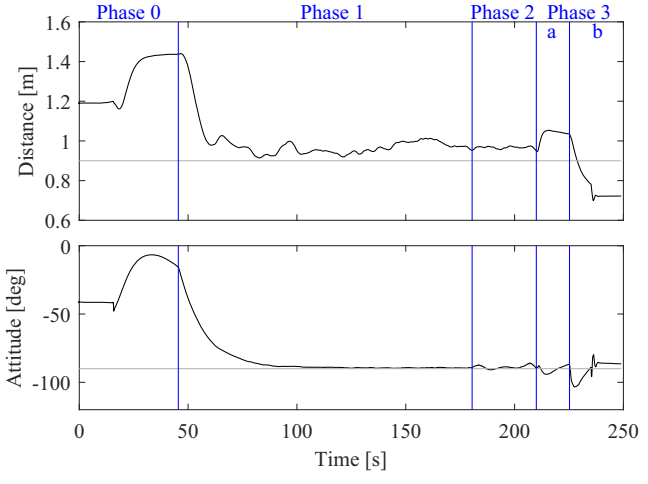


Fig. 9 Distance and orientation of the chaser relative to the target throughout all phases of the experiment.

VIII. Conclusions

This paper presents a comprehensive approach to the autonomous robotic capture of an uncooperative spinning spacecraft, integrating deep learning and optimal control techniques within a unified experimental framework. The key contribution of this work is the successful demonstration of an integrated guidance, navigation, and control architecture that combines a convolutional-neural-network-driven stereovision system for real-time pose determination, a deep-reinforcement-learning-based guidance algorithm for relative pose tracking maneuvers, and a Gauss pseudospectral method for optimal robotic manipulator deployment.

The integrated GNC architecture was experimentally validated using Carleton University's Spacecraft Proximity Operations Testbed, a state-of-the-art planar test bed formally introduced in this paper. The experiments demonstrated the system's ability to track the relative pose of a spinning target, optimally deploy a robotic manipulator with minimal attitude disturbances, and successfully capture the target. The results showed that the developed deep learning algorithms and control strategies are computationally simple and can operate in real time on embedded hardware, achieving a high degree of accuracy and reliability.

Ultimately, it is expected that experimental results presented in this work will improve the confidence in computer-vision-driven deep learning GNC techniques so that they can be applied in upcoming spacecraft proximity operations. In this context, future work will focus on the six-degree-of-freedom validation of the presented methods as well as evaluate their performance in comparison to classical GNC techniques.

Acknowledgments

This research was financially supported in part by the Natural Sciences and Engineering Research Council of Canada under the Discovery Grant program; the Canadian Space Agency Space Technology Development Program; the Ontario Centres of Excellence Voucher for Innovation and Productivity II award 24053; the Ontario Ministry of Economic Development, Job Creation and Trade through the Early Researcher Awards program; the Ontario Graduate Scholarship program; and Carleton University's Department of Mechanical and Aerospace Engineering.

References

[1] Kawano, I., Mokuno, M., Kasai, T., and Suzuki, T., "Result of Autonomous Rendezvous Docking Experiment of Engineering Test Satellite-VII," *Journal of Spacecraft and Rockets*, Vol. 38, No. 1, 2001, pp. 105–111. https://doi.org/10.2514/2.3661

[2] Dennehy, C. J., and Carpenter, J. R., "A Summary of the Rendezvous, Proximity Operations, Docking, and Undocking (RPODU) Lessons Learned from the Defense Advanced Research Project Agency (DARPA) Orbital Express (OE) Demonstration System Mission," NASA TM 2011-217088, NASA Langley Research Center, 2011, https://ntrs.nasa.gov/citations/20110011506.

[3] Biesbroek, R., Aziz, S., Wolahan, A., Cipolla, S., Richard-Noca, M., and Piguet, L., "The ClearSpace-1 Mission: ESA and ClearSpace Team Up to Remove Debris," *Proceedings of the 8th European Conference on Space Debris (virtual)*, ESA Space Debris Office, Darmstadt, Germany, April 2021.

[4] Naasz, B., "NASA Satellite Servicing and the OSAM-1 Mission," *70th Joint Army-Navy-NASA-Air Force (JANNAF) Propulsion Meeting Space Access Mobility and Logistics Session*, JANNAF Interagency Propulsion Committee, Columbia, MD, May 2023, pp. 1–17, https://ntrs.nasa.gov/api/citations/20230006585/downloads/Naasz%20ISAM%20JANNAF%202023%20v2.pdf.

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

[5] Rybus, T., and Seweryn, K., "Planar Air-Bearing Microgravity Simulators: Review of Applications, Existing Solutions and Design Parameters," *Acta Astronautica*, Vol. 120, March–April 2016, pp. 239–259. https://doi.org/10.1016/j.actaastro.2015.12.018

[6] Wilde, M., Clark, C., and Romano, M., "Historical Survey of Kinematic and Dynamic Spacecraft Simulators for Laboratory Experimentation of On-Orbit Proximity Maneuvers," *Progress in Aerospace Sciences*, Vol. 110, Oct. 2019, Paper 100552. https://doi.org/10.1016/j.paerosci.2019.100552

[7] Nakka, Y., Foust, R., Lupu, E., Elliot, D., Crowell, I., Chung, S.-J., and Hadaegh, F., "A Six Degree-of-Freedom Spacecraft Dynamics Simulator for Formation Control Research," *2018 AAS/AIAA Astrodynamics Specialist Conference*, American Astronautical Soc., Paper 18-476, Springfield, VA, 2018, p. 20.

[8] Wilde, M., Kaplinger, B., Go, T., Gutierrez, H., and Kirk, D., "ORION: A Simulation Environment for Spacecraft Formation Flight, Capture, and Orbital Robotics," *2016 IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2016, pp. 1–14. https://doi.org/10.1109/AERO.2016.7500575

[9] Gallardo, D., Bevilacqua, R., and Rasmussen, R., "Advances on a 6 Degrees of Freedom Testbed for Autonomous Satellites Operations," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2011-6591, 2011. https://doi.org/10.2514/6.2011-6591

[10] Regehr, M., Acikmese, A., Ahmed, A., Aung, M., Clark, K., MacNeal, P., Shields, J., Singh, G., Bailey, R., Bushnell, C., et al., "The Formation Control Testbed," *IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2004, pp. 557–564. https://doi.org/10.1109/AERO.2004.1367641

[11] Daitx, H., Schlotterer, M., Whidborne, J., and Sagliano, M., "Development of a Combined Attitude and Position Controller for a Satellite Simulator," *67th International Astronautical Congress*, International Astronautical Federation, Guadalajara, Mexico, 2016, pp. 1–15.

[12] Eun, Y., Park, C., and Park, S.-Y., "Design and Development of Ground-Based 5-DOF Satellite Formation Flying Testbed," *AIAA Modeling and Simulation Technologies Conference*, AIAA, Reston, VA, 2016. https://doi.org/10.2514/6.2016-1668

[13] Tsiotras, P., "ASTROS: A 5DOF Experimental Facility for Research in Space Proximity Operations," *Astrodynamics Specialist Conference*, AIAA, Reston, VA, 2014, p. 14.

[14] Zappulla, R., Virgili-Llop, J., Zagaris, C., Park, H., and Romano, M., "Dynamic Air-Bearing Hardware-in-the-Loop Testbed to Experimentally Evaluate Autonomous Spacecraft Proximity Maneuvers," *Journal of Spacecraft and Rockets*, Vol. 54, No. 4, 2017, pp. 825–839. https://doi.org/10.2514/1.A33769

[15] Robertson, A. D., "Control System Design for Spacecraft Formation Flying: Theory and Experiment," Ph.D. Thesis, Stanford Univ., 2001.

[16] Rybus, T., Barciński, T., Lisowski, J., Nicolau-Kukliński, J., Seweryn, K., Ciesielska, M., Grassmann, K., Grygorczuk, J., Karczewski, M., Kowalski, M., et al., "New Planar Air-Bearing Microgravity Simulator for Verification of Space Robotics Numerical Simulations and Control Algorithms," *12th ESA Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, Vol. 8, Noordwijk, The Netherlands, 2013, pp. 1–8, https://robotics.estec.esa.int/ASTRA/Astra2013/Papers/rybus.

[17] Henshaw, G., "Space Robotics Microgravity Simulation at NRL," *Gravity Offload Testbeds for Space Robotic Mission Simulation, Workshop at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, https://wvrtc.com/iros2017/files/IROSMicrogravitySimulations.pdf.

[18] Olivares-Mendez, M., Makhdoomi, M. R., Yalçın, B. C., Bokal, Z., Muralidharan, V., Del Castillo, M. O., Gaudilliere, 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*, Vol. 10, No. 4, 2023, pp. 509–521. https://doi.org/10.1016/j.jsse.2023.09.003

[19] Zwick, M., Huertas, I., Gerdes, L., and Ortega, G., "ORGL—ESA'S Test Facility for Approach and Contact Operations in Orbital and Planetary Environments," *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Vol. 6, European Space Agency, Noordwijk, The Netherlands, 2018, pp. 1–7.

[20] Saenz-Otero, A., and Miller, D. W., "SPHERES: A Platform for Formation-Flight Research," *UV/Optical/IR Space Telescopes: Innovative Technologies and Concepts II*, edited by H. A. MacEwen, Vol. 5899, SPIE, San Diego, CA, 2005, pp. 230–240. https://doi.org/10.1117/12.615966

[21] Smith, T., Barlow, J., Bualat, M., Fong, T., Provencher, C., Sanchez, H., and Smith, E., "Astrobee: A New Platform for Free-Flying Robotics on the International Space Station," *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS)*, European Space Agency, Noordwijk, The Netherlands, 2016, pp. 1–8, https://ntrs.nasa.gov/citations/20160007769.

[22] Romano, M., Friedman, D. A., and Shay, T. J., "Laboratory Experimentation of Autonomous Spacecraft Approach and Docking to a Collaborative Target," *Journal of Spacecraft and Rockets*, Vol. 44, No. 1, 2007, pp. 164–173. https://doi.org/10.2514/1.22092

[23] Benninghoff, H., Boge, T., and Tzschichholz, T., "Hardware-in-the-Loop Rendezvous Simulation Involving an Autonomous Guidance, Navigation and Control System," *1st IAA Conference on Dynamics and Control of Space Systems (DyCoSS 2012)*, American Astronautical Soc., Springfield, VA, 2012, pp. 1–20.

[24] Muralidharan, V., Makhdoomi, M. R., Žinys, A., Razgus, B., Klimavičius, M., Olivares-Mendez, M., and Martinez, C., "On-Ground Validation of Orbital GNC: Visual Navigation Assessment in Robotic Testbed Facility," *Astrodynamics*, March 2024, pp. 1–25. https://doi.org/10.1007/s42064-024-0198-4

[25] Mahendrakar, T., Wilde, M., and White, R., "Use of Artificial Intelligence for Feature Recognition and Flightpath Planning Around Non-Cooperative Resident Space Objects," *2021 ASCEND (Accelerating Space Commerce, Exploration, and New Discovery)*, AIAA Paper 2021-4123, 2021. https://doi.org/10.2514/6.2021-4123

[26] Wilde, M., Walker, I., Kwok Choon, S., and Near, J., "Using Tentacle Robots for Capturing Non-Cooperative Space Debris — A Proof of Concept," *AIAA SPACE and Astronautics Forum and Exposition*, AIAA Paper 2017-5246, 2017. https://doi.org/10.2514/6.2017-5246

[27] Kwok Choon, S. T., Madden, D., Wilde, M., and Go, T., "Capture of a Non-Cooperative Space Object Using a Grasping Tool with Force Sensors," *2018 AIAA SPACE and Astronautics Forum and Exposition*, AIAA Paper 2018-5227, 2018. https://doi.org/10.2514/6.2018-5227

[28] Hovell, K., and Ulrich, S., "Postcapture Dynamics and Experimental Validation of Subtethered Space Debris," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 2, 2018, pp. 519–525. https://doi.org/10.2514/1.G003049

[29] Pothen, A. A., Crain, A., and Ulrich, S., "Pose Tracking Control for Spacecraft Proximity Operations Using the Udwadia-Kalaba Framework," *Journal of Guidance Control and Dynamics*, Vol. 45, No. 2, 2022, pp. 296–309. https://doi.org/10.2514/1.G005169

[30] Bashnick, C., and Ulrich, S., "Fast Model Predictive Control for Spacecraft Rendezvous and Docking with Obstacle Avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 5, 2023, pp. 998–1007. https://doi.org/10.2514/1.G007314

[31] Kernot, J., and Ulrich, S., "Adaptive Control of a Tendon-Driven Manipulator for Capturing Non-Cooperative Space Targets," *Journal of Spacecraft and Rockets*, Vol. 59, No. 1, 2022, pp. 111–128. https://doi.org/10.2514/1.A34881

[32] Shi, J.-F., Ulrich, S., and Ruel, S., "Unsupervised Method of Infrared Spacecraft Image Foreground Extraction," *Journal of Spacecraft and Rockets*, Vol. 56, No. 6, 2019, pp. 1847–1856. https://doi.org/10.2514/1.A34449

[33] Shi, J.-F., Ulrich, S., and Ruel, S., "Real-Time Saliency Detection for Greyscale and Colour Images," *Visual Computer*, Vol. 37, No. 6, 2021, pp. 1277–1296. https://doi.org/10.1007/s00371-020-01865-x

[34] Shi, J.-F., and Ulrich, S., "Uncooperative Spacecraft Pose Estimation Using Monocular Monochromatic Images," *Journal of Spacecraft and Rockets*, Vol. 58, No. 2, 2021, pp. 284–301. https://doi.org/10.2514/1.A34775

[35] Crain, A., and Ulrich, S., "Experimental Validation of Pseudospectral-Based Optimal Trajectory Planning for Free-Floating Robots," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1726–1742. https://doi.org/10.2514/1.G003528

[36] Sharf, I., Woo, P., Nguyen-Huynh, T.-C., and Misra, A., "System Rigidization and Control for Post-Capture Maneuvering of Large Space Debris," *2016 IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2016, pp. 1–12, http://ieeexplore.ieee.org/document/7500800/. https://doi.org/10.1109/AERO.2016.7500800

[37] Carlbaum, E., Mansouri, S. S., Kanellakis, C., Koval, A., and Nikolakopoulos, G., "Towards Robust Localization Deep Feature Extraction by CNN," *46th Annual Conference of the IEEE Industrial Electronics Society*, Inst. of Electrical and Electronics Engineers, New York, 2020, pp. 807–812. https://doi.org/10.1109/IECON43393.2020.9254941

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277

[38] Bello-Cerezo, R., Bianconi, F., Di Maria, F., Napoletano, P., and Smeraldi, F., "Comparative Evaluation of Hand-Crafted Image Descriptors vs. Off-the-Shelf CNN-Based Features for Colour Texture Classification under Ideal and Realistic Conditions," *Applied Sciences*, Vol. 9, No. 4, 2019, p. 738. https://doi.org/10.3390/app9040738

[39] Remmen, F., Dubbelman, G., and de With, P. H. N., "Evaluating Image Features in Real-World Scenarios," *5th IEEE International Conference on Signal and Image Processing*, Inst. of Electrical and Electronics Engineers, New York, 2020, pp. 535–541. https://doi.org/10.1109/ICSIP49896.2020.9339363

[40] Pasqualetto Cassinis, L., Fonod, R., Gill, E., Ahrns, I., and Gil Fernandez, J., "CNN-Based Pose Estimation System for Close-Proximity Operations Around Uncooperative Spacecraft," *AIAA Scitech 2020 Forum*, AIAA Paper 2020-1457, 2020. https://doi.org/10.2514/6.2020-1457

[41] Shi, J.-F., Ulrich, S., and Ruel, S., "CubeSat Simulation and Detection Using Monocular Camera Images and Convolutional Neural Networks," *2018 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2018-1604, 2018. https://doi.org/10.2514/6.2018-1604

[42] Sharma, S., Beierle, C., and D'Amico, S., "Pose Estimation for Non-Cooperative Spacecraft Rendezvous Using Convolutional Neural Networks," *IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2018, pp. 1–12. https://doi.org/10.1109/AERO.2018.8396425

[43] Sharma, S., and D'Amico, S., "Neural Network-Based Pose Estimation for Noncooperative Spacecraft Rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 6, 2020, pp. 4638–4658. https://doi.org/10.1109/TAES.2020.2999148

[44] Thomas Despond, F., and Ulrich, S., "Real-Time Stereovision-Based Spacecraft Pose Determination Using Convolutional Neural Networks," *Journal of Spacecraft and Rockets*, 2025, pp. 1–11. https://doi.org/10.2514/1.A35973

[45] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going Deeper with Convolutions," *IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2015, pp. 1–9.

[46] Redmon, J., and Farhadi, A., "YOLO9000: Better, Faster, Stronger," *IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2017, pp. 7263–7271.

[47] Lin, M., Chen, Q., and Yan, S., "Network in Network," arXiv, 2014, arXiv:1312.4400v3. https://doi.org/10.48550/arXiv.1312.4400

[48] Glorot, X., Bordes, A., and Bengio, Y., "Deep Sparse Rectifier Neural Networks," *14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR Workshop and Conference Proceedings, JMLR, Cambridge, MA, 2011, pp. 315–323.

[49] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," arXiv, 2016, arXiv:1603.04467. https://doi.org/10.48550/arXiv.1603.04467

[50] Pauly, L., Rharbaoui, W., Shneider, C., Rathinam, A., Gaudillière, V., and Aouada, D., "A Survey on Deep Learning-Based Monocular Spacecraft Pose Estimation: Current State, Limitations and Prospects," *Acta Astronautica*, Vol. 212, Nov. 2023, pp. 339–360. https://doi.org/10.1016/j.actaastro.2023.08.001

[51] Broida, J., and Linares, R., "Spacecraft Rendezvous Guidance in Cluttered Environments via Reinforcement Learning," *29th AAS/AIAA Space Flight Mechanics Meeting*, American Astronautical Soc., Springfield, VA, Jan. 2019, pp. 1–15.

[52] Oestreich, C. E., Linares, R., and Gondhalekar, R., "Autonomous Six-Degree-of-Freedom Spacecraft Docking Maneuvers via Reinforcement Learning," *Journal of Aerospace Information Systems*, Vol. 18, No. 7, 2021, pp. 417–428. https://doi.org/10.2514/1.I010914

[53] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal Policy Optimization Algorithms," arXiv, 2017, arXiv: 1707.06347. https://doi.org/10.48550/arXiv.1707.06347

[54] Hovell, K., and Ulrich, S., "Deep Reinforcement Learning for Spacecraft Proximity Operations Guidance," *Journal of Spacecraft and Rockets*, Vol. 58, No. 2, 2021, pp. 254–264. https://doi.org/10.2514/1.A34838

[55] Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., D., T. B., Muldal, A., Heess, N., and Lillicrap, T., "Distributed Distributional Deterministic Policy Gradients," arXiv, 2018, arXiv: 1804.08617. https://doi.org/10.48550/arXiv.1804.08617

[56] Hovell, K., and Ulrich, S., "Laboratory Experimentation of Spacecraft Robotic Capture Using Deep-Reinforcement-Learning–Based Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 11, 2022, pp. 2138–2146. https://doi.org/10.2514/1.G006656

[57] Kober, J., Bagnell, J. A., and Peters, J., "Reinforcement Learning in Robotics: A Survey," *International Journal of Robotics Research*, Vol. 32, No. 11, 2015, pp. 1238–1274. https://doi.org/10.1177/0278364913495721

[58] Harris, A., Teil, T., and Schaub, H., "Spacecraft Decision-Making Autonomy Using Deep Reinforcement Learning," *AAS/AIAA Space Flight Mechanics Meeting*, AAS Paper 2019-0447, Springfield, VA, 2019, http://hanspeterschaub.info/Papers/Harris2019.pdf.

[59] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, ICLR, Appleton, WI, 2015, pp. 1–15, http://arxiv.org/abs/1412.6980.

[60] Moghaddam, B. M., and Chhabra, R., "On the Guidance, Navigation and Control of In-Orbit Space Robotic Missions: A Survey and Prospective Vision," *Acta Astronautica*, Vol. 184, July 2021, pp. 70–100. https://doi.org/10.1016/j.actaastro.2021.03.029

[61] Vafa, Z., and Dubowsky, S., "On the Dynamics of Manipulators in Space Using the Virtual Manipulator Approach," *1987 IEEE International Conference on Robotics and Automation*, Vol. 4, Inst. of Electrical and Electronics Engineers, New York, 1987, pp. 579–585. https://doi.org/10.1109/ROBOT.1987.1088032

[62] Umetani, Y., and Yoshida, K., "Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, 1989, pp. 303–314. https://doi.org/10.1109/70.34766

[63] Yoshida, K., Hashizume, K., and Abiko, S., "Zero Reaction Maneuver: Flight Validation with ETS-VII Space Robot and Extension to Kinematically Redundant Arm," *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 1, 2001, pp. 441–446. https://doi.org/10.1109/ROBOT.2001.932590

[64] Virgili Llop, J., Drew, J., Zappulla, R., and Romano, M., "Autonomous Capture of a Resident Space Object by a Spacecraft with a Robotic Manipulator: Analysis, Simulation and Experiments," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2016-5269, 2016. https://doi.org/10.2514/6.2016-5269

[65] Flores-Abad, A., Wei, Z., Ma, O., and Pham, K., "Optimal Control of Space Robots for Capturing a Tumbling Object with Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 2014–2017. https://doi.org/10.2514/1.G000003

[66] Crain, A., and Ulrich, S., "Nonlinear Optimal Trajectory Planning for Free-Floating Space Manipulators Using a Gauss Pseudospectral Method," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2016-5272, 2016. https://doi.org/10.2514/6.2016-5272

[67] Rybus, T., Seweryn, K., and Sasiadek, J. Z., "Trajectory Optimization of Space Manipulator with Non-Zero Angular Momentum During Orbital Capture Maneuver," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2016-0885, 2016. https://doi.org/10.2514/6.2016-0885

[68] Crain, A., "Optimal Trajectory Planning and Compliant Spacecraft Capture Using a Space Robot," Master's Thesis, Carleton Univ., 2018, https://curve.carleton.ca/565918a1-d695-471f-a85c-47a66f3cc147.

[69] Yoshida, K., "ETS-VII Flight Experiments for Space Robot Dynamics and Control," *Experimental Robotics VII*, edited by D. Rus, and S. Singh, Springer, Berlin, 2001, pp. 209–218. https://doi.org/10.1007/3-540-45118-8_22

[70] Eisele, A., "Order by Intelsat Completes Manifest for Inaugural Launch in Early 2025," Northrop Grumman, Dulles, VA, 2023, https://news.northropgrumman.com/news/releases/northrop-grummans-spacelogistics-continues-revolutionary-satellite-life-extension-work-with-sale-of-third-mission-extension-pod [retrieved 27 April 2024].

[71] Yan, I. C., "Robotic Refueling Mission 3 (RRM3)," NASA, 2024, https://www.nasa.gov/nexis/robotic-refueling-mission-3/ [retrieved 27 April 2024].

[72] Pultarova, T., "SpaceX to launch 1st space-hardened Nvidia AI GPU on upcoming rideshare mission," Space.com, New York, 2024, https://www.space.com/ai-nvidia-gpu-spacex-launch-transporter-11 [retrieved 26 Sept. 2024].

S. Li
Associate Editor

Downloaded by Carleton University on October 21, 2025 | http://arc.aiaa.org | DOI: 10.2514/1.A36277