JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 22, No. 8, August 2025

Computer-Vision-Driven Artificial Potential Function Guidance and Adaptive Control for Spacecraft Proximity Operations

Natasha Reid,* Ryan Goodridge,* Parker Stewart,* Shae Simonson,*
Austin Dahlseide,* Kristian Delaney,* Alexis Valdez Cascajares,*
Courtney Savytska,† Alexander Crain,‡ and Steve Ulrich§
Carleton University, Ottawa, Ontario K1S 5B6, Canada

https://doi.org/10.2514/1.I011560

This research addresses the growing issue of space debris by developing advanced computer vision, guidance, and control techniques for autonomous docking in proximity operations. Specifically, this work develops these technologies to present an experiment where a chaser platform autonomously docks with a cooperative spinning target while avoiding an uncooperative obstacle. A stereovision system using ArUco markers tracks the target's pose in real-time, while an unscented Kalman filter processes the data. The obstacle is detected through bounding box manipulation and stereo disparity principles. A novel artificial potential function guidance law, herein adapted for spinning targets, calculates a collision-free trajectory, which is tracked using a real-time adaptive control law. Experimental validation at Carleton University's Spacecraft Proximity Operations Testbed confirms the effectiveness of the proposed system.

I. Introduction

SINCE the 1960s, scientists have recognized the threat posed by the growing debris belt in Earth orbit that endangers current and future satellite operations. In 1978, Kessler and Cour-Palais [1] warned that collisions among debris could lead to an exponential increase in orbital debris, a scenario known as *Kessler syndrome*. As of 2023, over 34,000 tracked objects are in orbit, with approximately 75% being classified as debris, including defunct satellites and discarded rocket stages [2]. Furthermore, the European Space Agency (ESA) estimated in 2016 that around 130 million smaller debris fragments, ranging from 1 mm to 10 cm, also populate space [3]. Notably, ESA reported that removing just five large debris objects annually could help mitigate the exponential growth of space debris [4]. One promising solution involves deploying intelligent spacecraft to autonomously rendezvous with and de-orbit debris, alongside on-orbit servicing activities to extend the life of existing satellites and reduce the need for new launches.

The autonomous capture of a cooperative spacecraft in orbit while reliably avoiding uncooperative obstacles presents several challenges that require advanced guidance, navigation, and control (GNC) techniques. While various solutions have been proposed and tested in orbit and in GNC-integrated lab experiments, a comprehensive integration of computer-vision-informed GNC has yet to be experimentally validated in a significant proximity operations scenario with obstacle avoidance. Nevertheless, several notable experiments have laid the groundwork for this development. Successful unmanned rendezvous and docking missions, such as ETS-VII [5] and Orbital Express [6], demonstrated the feasibility of autonomous and semi-autonomous proximity operations. Space agencies continue to invest substantial resources in upcoming autonomous spacecraft missions. ESA's ClearSpace-1 servicer aims to be the first intelligent spacecraft to autonomously rendezvous and dock with a large piece of space debris for removal by 2026 [7,8]. Meanwhile, the Japan Aerospace Exploration Agency has partnered with Astroscale to launch Active Debris Removal by Astroscale-Japan (ADRAS-J) in 2024, marking the first phase of a multistage mission aimed at autonomously removing space debris [9]. The ADRAS-J mission will collect visual data through rendezvous and proximity operations with large debris (greater than 10 mm in diameter), paving the way for the mission's second phase.

Academic institutions also support the development and experimental validation of innovative rendezvous and proximity operations technologies using on-ground laboratory facilities that simulate a microgravity environment. In 2007, Romano et al. [10] used the POSEIDYN testbed at the Naval Postgraduate School to experimentally validate integrated GNC techniques for autonomous rendezvous and docking with a stationary collaborative target. The relative pose of the target was determined through the fusion of data from a single-camera vision sensor and an inertial measurement unit via Kalman filtering. The European Proximity Operations Simulator at the German Aerospace Centre was used to validate an integrated GNC system involving a monocular camera and advanced Kalman filtering techniques for relative navigation of an uncooperative target in hardware-in-the-loop (HIL) autonomous rendezvous simulations [11]. In 2021, using the ORION simulator at the Florida Institute of Technology, Mahendrakar et al. [12] trained a convolutional neural network (CNN) to identify object features using bounding boxes. The machine vision data was integrated with an artificial potential function (APF) guidance law to experimentally test autonomous rendezvous and docking operations with an uncooperative target. More recently, Muralidharan et al. [13] used the 7-degree-of-freedom (7-DoF) robotic testbed in the Zero-G Lab at the University of Luxembourg to validate a GNC system involving a CNN-based vision pose detection algorithm of an uncooperative target. Tracking of an optimal reference trajectory generated offline was completed using a linear quadratic regulator (LQR).

The work reported herein builds upon these previous laboratory experiments by developing a fully integrated GNC architecture that includes a computer vision system, a real-time docking and collision-avoidance guidance algorithm, and an adaptive control law. Unlike in the previously reported literature, this work also considers a separate, uncooperative obstacle that must be tracked by the relative navigation system and avoided during maneuvering. To summarize, the contributions of this work are 1) the development

Received 3 October 2024; accepted for publication 30 March 2025; published online 13 June 2025.

*Research Assistant, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Student Member AIAA; Formerly Published Under Courtney Bashnick.

†Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Member AIAA.

‡Research Associate, Department of Mechanical and Aerospace Engineering. 1125 Colonel By Drive.

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

of a simple yet efficient cooperative stereovision system for relative pose determination that relies on ArUco [14] markers and a real-time unscented Kalman filter (UKF), 2) the development of an uncooperative stereovision-based obstacle detection approach based on photogrammetry principles that is also combined to a UKF, 3) the original development of a modified APF approach applicable to spinning target objects and that decouples guidance and control functionalities, and 4) the original integration and laboratory experimental validation of the proposed vision system and modified APF with an existing direct adaptive control (DAC) method for a spacecraft rendezvous and docking maneuver with collision avoidance capabilities.

The remainder of this paper is organized as follows: First, a brief overview of existing computer vision, guidance methods, and adaptive control for proximity operations is provided. Then, the problem statement addressed by this work is introduced, followed by an overview of the experimental testbed used for validation of the techniques discussed. The subsequent three sections discuss the computer vision techniques used for cooperative relative pose determination and uncooperative obstacle detection, the UKF algorithm and line-of-sight tracking control implemented to filter sensor noise to improve navigation data, the implementation of APF guidance to calculate the collision-free rendezvous and docking trajectory for the chaser platform, and the DAC law implemented to track the desired rendezvous and docking trajectory. Finally, the results of the experiments that validate the proposed vision-driven APF guidance and adaptive control system are presented before this paper is concluded.

II. Related Work

This section provides a brief overview of existing notable work in areas related to this paper, i.e., computer vision and relative state estimation schemes, guidance methods, and relative motion adaptive tracking control.

A. Computer Vision and State Estimation

As mentioned earlier, the computer-vision-based system responsible for determining the cooperative target pose relative to the chaser spacecraft and detecting the uncooperative obstacle is based on photogrammetry principles. By extracting three-dimensional (3D) information from two-dimensional (2D) images, photogrammetry-based cooperative vision systems have provided critical data enabling accurate relative navigation in space.

Previous on-orbit applications of cooperative photogrammetric techniques include the Space Vision System by Neptec Design Group for the Canadian Space Agency [15], the Advanced Video Guidance Sensor by NASA's Marshall Space Flight Center for the Defense Advanced Research Projects Agency's Orbital Express mission [16], and the ULTOR Relative Navigation System developed by Advanced Optical Systems for NASA's Hubble Space Telescope Robotic Servicing Mission [17]. Other notable cooperative vision systems that were flight tested are the light-emitting diode (LED)-based system by the PRISMA mission [18] and the concentric contrasting circles approach first used on the International Space Station (ISS) for docking purposes. More recently, the use of ArUco markers was investigated as a possible fiducial for relative navigation systems. Vela et al. [19] proposed a visual-based pose determination algorithm for spacecraft close-proximity operations in which ArUco markers are used as markers placed on the target platform. Numerical simulations demonstrated the applicability and robustness of this approach against several sources of visual disturbance. A similar relative pose determination method was verified in a hardware-in-the-loop robotic facility in the context of cis-lunar orbits for NASA's Gateway [20]. ArUco markers have also been successfully tested in space robotics operations on NASA's Perseverance Rover.[¶]

[¶]NASA, "Rock Sampled by NASA's Perseverance Embodies Why Rover Came to Mars," April 2024, https://www.nasa.gov/missions/mars-2020-perseverance/perseverance-rover/rock-sampled-by-nasas-perseverance-embodies-why-rover-came-to-mars/.

A known problem with photogrammetry-based approaches for relative navigation is that they only determine the relative pose; i.e., they do not provide linear and angular velocities. For this reason, computer-vision-based relative navigation systems are often combined with recursive estimation techniques. While several approaches have been developed and validated in numerical simulations, fewer experimental studies have been conducted. Tweddle and Saenz-Otero [21] combined a multiplicative extended Kalman filter (MEKF) to stereovision relative navigation using four concentric contrasting circles as fiducial markers. Planar free-floating experimental validation on the SPHERES platforms demonstrated the applicability of the approach. Valverde et al. [22], using the dual quaternion framework, developed a MEKF for proximity operations that fused measurements from a rate gyroscope, an inertial measurement unit, and VICON ground truth data. Experiments at Georgia Tech's 5-DoF ASTRO facility validated the performance of their approach. More recently, Rizzolo [23] proposed four different Kalman filtering strategies in combination with stereovision based on ArUco markers. Experiments with a stationary stereo camera and a tumbling target platform were successfully performed.

B. Guidance Methods

In the literature, several guidance methods for proximity operations have been proposed over the years. Glideslope is a traditional rendezvous and docking guidance technique that prescribes an exponentially decreasing velocity as the chaser approaches the target [24,25]. This simple algorithm, however, only applies to a straight-line approach and does not apply for spinning targets or obstacle avoidance.

Genetic algorithms have better odds of finding global optima in nonconvex problems and can be useful in generating trajectories offline when computation time is not a concern. Despite this, they can still get caught in local minima, especially if optimization space is limited, and the computational costs are often too high to implement genetic algorithms in a real-time docking scenario [26]. Another optimal guidance approach is model predictive control (MPC). MPC generally solves, but is not limited to, convex optimization problems to minimize a cost function that includes weighted terms for state error and commanded control effort over an ever-receding discrete time horizon. Similar to genetic algorithms, MPC is computationally expensive but, given the correct hardware, appropriate horizon lengths, and discretization times, can be run in real-time [27]. Sliding mode laws include N-DoF state feedback and guarantee linear convergence to an artificially created "sliding surface" of order $N-1$ and exponential convergence once the sliding surface is reached. Although sliding mode controllers are computationally inexpensive, they are nonoptimal and do not readily accept path constraints as is required for obstacle avoidance and docking [24].

Despite also being nonoptimal, the APF is capable of accepting path constraints and provides good convergence to the desired state. There are no guarantees on constraint adherence, but APF requires no knowledge of the system dynamics and is extremely computationally efficient. APF has been combined with an LQR and validated in close proximity operations using the SPHERES facility [28]. This guidance and control integrated system used the LQR to attract the chaser to the desired position and only used APF for collision avoidance. Unfortunately, this technique introduces the possibility of undesired local minima when the control input from the APF is exactly equal and opposite to that of the LQR. Further developments include fuel optimization in the LQR and integrating the LQR/APF controller with a wall-following algorithm to address the issue of local minima [29]. An adaptive APF (AAPF) method that implements a time-varying attractive potential shaping matrix to improve the fuel efficiency of the nominal APF method was compared in experiment to APF [30]. These experiments, however, were limited to only considering a stationary target. Recently, APF was implemented with a machine vision-aided navigation system to

command a swarm of chasers to safely rendezvous with a single rotating object [31]. This paper builds upon these recent works in the area of APF guidance for proximity operations by experimentally validating APF with a vision-based relative pose determination system, an adaptive closed-loop control law, and a gradient descent method to not only achieve docking with a rotating target but to do so in the presence of an obstacle.

C. Adaptive Trajectory Tracking Control

Various control techniques have been employed for spacecraft relative motion tracking during proximity operations, with adaptive control methods being particularly notable for their ability to manage dynamic uncertainties and unknown perturbations. Adaptive controllers can be either direct or indirect. Direct controllers adjust control gains based on instantaneous tracking errors, while indirect controllers, such as $\mathcal{L}_1$ [32] and immersion and invariance [33] methods, estimate unknown parameters and external perturbations to adjust control parameters. However, indirect methods often require precise plant dynamics models and are computationally intensive [34]. In contrast, DACs offer a simpler approach by modifying control gains directly in response to tracking errors. Ulrich et al. [35] first applied DAC theory to spacecraft motion tracking, demonstrating effective performance in experiments with the SPHERES testbed despite significant mass uncertainties. More recently, simple adaptive control (SAC) trajectory tracking controllers were designed using various particle swarm optimization schemes and successfully validated in experiments with SPOT [36]. Other successful examples of SAC-based controllers for spacecraft relative motion tracking include formation-flying studies by Ulrich [37] and Chihabi and Ulrich [38]. The main limitation of SAC is related to its relatively complex adaptation mechanisms and the need for an onboard ideal model.

As an alternative, a simpler DAC method—which does not rely on the onboard implementation of an ideal model—based on the work of Hong and Bernstein [39,40] is proposed in this work. This particular DAC has been used in several motion control experiments, including a mass–spring system with electromagnetic actuation and a servopneumatic actuator [39].

III. Problem Statement

Three spacecraft exist in a coplanar environment: a chaser spacecraft with a male docking port is actively controlled, an obstacle spacecraft is passive and spinning at a constant angular rate, and a target spacecraft equipped with a female docking cone is collaborative, spinning at a constant angular rate. The goal of this work is twofold: first, to enable the chaser to successfully dock with a collaborative target equipped with ArUco markers; second, to demonstrate the docking maneuver, yet, while avoiding an uncooperative obstacle.

Two reference frames define the planar components of the relative and inertial positions and orientations of the platforms on the testbed. In the lab, the inertial frame $\mathcal{F}_I$ has its unit vectors $\{\boldsymbol{I}_x, \boldsymbol{I}_y, \boldsymbol{I}_z\}$ defined with respect to the floating surface, while the body-fixed reference frames of the chaser $\mathcal{F}_{B_c}$, target $\mathcal{F}_{B_t}$, and obstacle $\mathcal{F}_{B_o}$ platforms have unit vectors $\{\boldsymbol{B}_{cx}, \boldsymbol{B}_{cy}, \boldsymbol{B}_{cz}\}$, $\{\boldsymbol{B}_{tx}, \boldsymbol{B}_{ty}, \boldsymbol{B}_{tz}\}$, and $\{\boldsymbol{B}_{ox}, \boldsymbol{B}_{oy}, \boldsymbol{B}_{oz}\}$, respectively. Due to the planar nature of the problem, all unit vectors in the z-axis point vertically upward.

The governing equations of motion derived from Newton's second law allow the calculation of the accelerations of all platforms as follows:

$$\ddot{x} = \frac{f_x}{m} \qquad \ddot{y} = \frac{f_y}{m} \qquad \ddot{\theta} = \frac{\tau}{J} \tag{1}$$

where x and y refer to inertial position along the $\boldsymbol{I}_x$ and $\boldsymbol{I}_y$ directions, respectively; θ refers to the orientation of the body with respect to the inertial lab reference frame measured counterclockwise from $\boldsymbol{I}_x$; f_x and f_y are the forces acting on the body in the $\boldsymbol{I}_x$ and $\boldsymbol{I}_y$ directions, respectively; m is the mass of the body; τ is the torque acting on the body; and J is the moment of inertia about the z-axis. According to standard practice in proximity operations, due to the relatively short time scale and separation distance of the chaser to the obstacle and target, orbital effects are ignored [10,41–43]. The inertial platform position and orientation are combined to form the inertial platform pose, denoted as $\boldsymbol{x}$. Written explicitly, this is

$$\boldsymbol{x}_c = \begin{bmatrix} x_c & y_c & \theta_c \end{bmatrix}^\top \quad \boldsymbol{x}_t = \begin{bmatrix} x_t & y_t & \theta_t \end{bmatrix}^\top \quad \boldsymbol{x}_o = \begin{bmatrix} x_o & y_o & \theta_o \end{bmatrix}^\top \tag{2}$$

where the subscripts c, t, and o refer to the chaser, target, and obstacle platform, respectively.

Figure 1 illustrates the integrated GNC system proposed in this work. In brief, computer vision, assisted by a UKF, estimates the relative pose of the target and obstacle, which is fed to the APF guidance law to calculate the desired docking—and collision avoidance—trajectory in real-time. A DAC is used to actively track the desired trajectory. In further detail, the target pose relative to the chaser, $\boldsymbol{r}_{tc}$, and the obstacle pose relative to the chaser, $\boldsymbol{r}_{oc}$, mathematically described as

$$\boldsymbol{r}_{tc} = \boldsymbol{x}_t - \boldsymbol{x}_c \qquad \boldsymbol{r}_{oc} = \boldsymbol{x}_o - \boldsymbol{x}_c \tag{3}$$

are determined by the computer vision relative navigation system, as shown in Fig. 1. In operational scenarios, the inertial state of the chaser is typically known through star tracker and GPS sensors. In the context of the lab environment, this inertial state $\boldsymbol{x}_c$ is provided via PhaseSpace—the overhead motion capture ground-truth system discussed in the following section. Using knowledge of the chaser's inertial state, the relative target and obstacle poses, $\boldsymbol{r}_{tc}$ and $\boldsymbol{r}_{oc}$, are converted to their respective inertial poses, denoted as $\boldsymbol{z}_t$ and $\boldsymbol{z}_o$, before being fed to UKF estimation filters. Indeed, the noisy data provided by the computer vision system may cause issues with the APF guidance. For data smoothing purposes, a UKF is proposed to filter the inertial poses of the target and obstacle. The UKFs also have the ability to reject outliers in real-time. Together with $\boldsymbol{x}_c$, the outputs of both UKFs—i.e., the estimated inertial pose of the target and obstacle, denoted as $\hat{x}_t$ and $\hat{x}_o$, respectively—are used as inputs to the APF guidance to calculate the desired inertial chaser docking

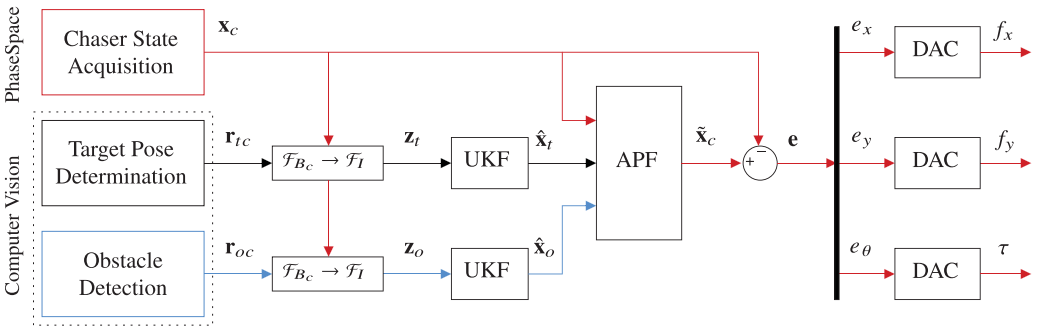


Fig. 1 Integrated GNC system block diagram.

trajectory $\tilde{\boldsymbol{x}}_c$. To quantify the trajectory tracking objective, an output error denoted by $\boldsymbol{e}$ is defined as

$$\boldsymbol{e} \triangleq \tilde{x}_c - \boldsymbol{x}_c \equiv \begin{bmatrix} e_x & e_y & e_\theta \end{bmatrix}^\top \tag{4}$$

Based on this error signal, the DAC adapts its gains in real-time and calculates the control input signal

$$\boldsymbol{u} = \begin{bmatrix} f_x & f_y & \tau \end{bmatrix}^\top \tag{5}$$

IV. Spacecraft Proximity Operations Testbed

The Spacecraft Proximity Operations Testbed (SPOT), a part of the Spacecraft Robotics and Control Laboratory at Carleton University, is used for all experiments presented in this work. As pictured in Fig. 2, SPOT features a 3.5 m × 2.4 m granite floating surface and three spacecraft platforms, which act as the chaser (RED), target (BLACK), and obstacle (BLUE). Each platform is equipped with an NVIDIA Jetson Xavier NX Development board to execute guidance and control software and interface with the onboard hardware. The chaser is additionally equipped with an NVIDIA Jetson Orin AGX, which serves as a dedicated embedded computer to execute all vision-based techniques and related software. This onboard computer measures 100 × 87 mm, weighs approximately 200 g, supports a 12 V power supply, and has a typical power consumption around 30–45 W depending on the workload. These characteristics are almost identical to the latest generation of space-hardened computing and AI GPU capabilities.** The stereo camera installed on the chaser is a ZED 2 stereo camera by Stereolabs, capable of capturing video in 2k at 15 frames per second (fps). Each platform is supported by air bearings that, when activated, create a cushion of compressed air that is sufficient to support the platform, allowing it to float approximately 5 μm above the surface of the testbed and glide with negligible friction effects. The translational and rotational motion of each platform is controlled by eight thrusters, allowing it to propel itself across the surface of the testbed during experimentation. The inertial position and orientation of each platform are captured in real-time through the use of a PhaseSpace Impulse X2E motion tracking system. The platforms are equipped with four LEDs mounted to their upper corners, each with a unique pulse frequency that the PhaseSpace cameras track. There are 10 motion capture cameras mounted around the edges of the SPOT facility, enabling position measurement accuracy to approximately 0.01 mm in the $\boldsymbol{I}_x$, $\boldsymbol{I}_y$, and $\boldsymbol{I}_z$ directions and orientation accuracy of 2×10^{-5} rad (0.0014 deg). The SPOT facility also features a sun simulator that generates high-contrast lighting conditions from a single source, which is used to validate computer vision techniques under more realistic lighting conditions.

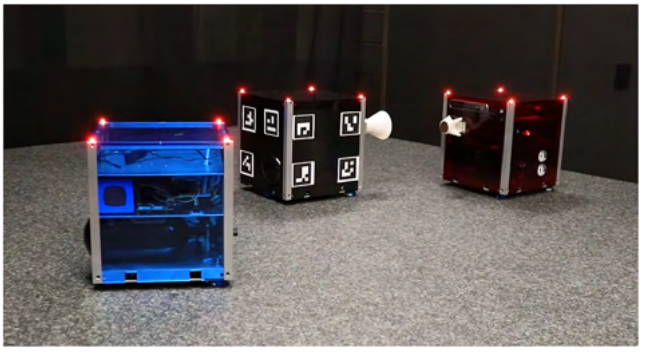


Fig. 2 Carleton University's Spacecraft Proximity Operations Testbed (SPOT).

**Tereza Pultarova, "SpaceX to Launch 1st Space-Hardened Nvidia AI GPU on Upcoming Rideshare Mission," August 2024, https://www.space.com/ai-nvidia-gpu-spacex-launch-transporter-11.

V. Computer Vision

To solve the relative target pose and obstacle detection tasks, this work investigates vision-based pose determination methods founded in photogrammetry. With a collaborative target, a pose determination algorithm is based upon knowledge of a light or beacon configuration [10,44] or marker features [45,46] on the object. When such knowledge of the object is unavailable, alternative methods must be used. For example, the detection of an unknown, uncooperative object was achieved using the disparity of stereo camera images to extract the relative position measurements of its geometric center [47]. This method was validated in an inspection maneuver using the SPHERES facility onboard the ISS.

Building upon these techniques, the work presented herein considers fiducial markers fixed to the target, enabling its position and orientation to be determined in a complex environment with low computational requirements. To create a cooperative target, ArUco [14] markers have been selected for this work because they are supported by an extensive detection and pose estimation library based on OpenCV, provide accurate data at short ranges, and are computationally efficient as compared to other available fiducial marker packages [48]. To determine the pose of an uncooperative obstacle, this work uses stereo disparity principles. Unlike in Ref. [47], where information from a disparity map is immediately exploited, this work detects an object in the image using edge detection and bounding box methods before determining its relative position from a depth map. The remainder of this section details the developed algorithms.

A. Cooperative Target Relative Pose Determination

The real-time relative pose determination process is conducted entirely in the camera reference frame inherent to the stereovision system and OpenCV framework. Two sets of ArUco markers are used: one larger set of 6.8 cm and one smaller set of 3.5 cm. Smaller markers are used for distances less than 1 m to improve relative pose determination accuracy for final docking due to the occlusion of the larger markers. The relative pose determination requires three measurements, as shown in Fig. 3. First is the position vector of the stereo camera and both of its lenses relative to the chaser's center of geometry, which is known through physical measurement. Second is the real-time pose of the ArUco markers relative to the camera, measured using the ArUco module from the OpenCV library in Python. Third is the position of the target's center of geometry relative to each ArUco marker, which is known by physical measurement a priori. In other words, both the position of the stereo camera relative to the chaser's center of geometry and the position of the target's center of geometry relative to each marker are known information acquired offline before real-time operations.

During runtime, the captured image from the stereo camera is split in half according to its frames from the left and right lenses. The target pose determination is completed on each half of the image. As such, vector 1 (as specified in Fig. 3) is selected for the relevant lens. ArUco markers are detected in each image using OpenCV's `aruco.detectMarkers` function, and their identification numbers are stored. Using the ArUco identification number, vector 3 is known by parsing through a stored dictionary in the vision software. For each ArUco marker, the rotation angles and position components relative to the camera are determined in real-time using the `aruco.estimatePoseSingleMarkers`

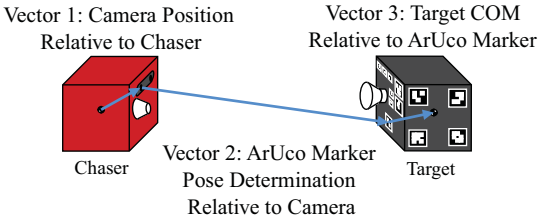


Fig. 3 Relative pose determination of the target platform using ArUco markers.

function. These components provide the necessary information to determine vector 2 in addition to the attitude of the target relative to the chaser, which is used to express vector 3 in the chaser body-fixed reference frame. For each detected ArUco marker in each half of the captured image, the pose of the target relative to the chaser is simply calculated as the vector summation of vectors 1, 2, and 3. Finally, all resulting poses are averaged to provide a better estimate of the relative target pose, which is currently expressed in the body-fixed reference frame of the chaser platform.

For integration with the guidance and control systems, the relative pose of the target must be expressed in the inertial reference frame. To do so, a rotation about the z-axis through an angle corresponding to the chaser's actual inertial orientation is required. This angle, θ_c, is known through the PhaseSpace motion capture system. With the relative pose $\boldsymbol{r}_{tc}$ in the inertial reference frame, the inertial pose of the target can then be determined using Eq. (3), recalling that $\boldsymbol{x}_c$ is known through the PhaseSpace motion capture system. In Fig. 1, the inertial pose of the target is denoted $\boldsymbol{z}_t$ to signify this as the input measurement to the UKF. Further details on the UKF are provided at the end of this section.

B. Obstacle Detection

During proximity operations in orbit, particularly in high-traffic scenarios such as on-orbit assembly, it is crucial to avoid obstacles to prevent collisions and ensure successful docking with the target. Additionally, during space debris removal missions, where the target vehicle may have sustained damage, any resulting smaller debris in the vicinity must be treated as obstacles and navigated around safely. The obstacle in this work is uncooperative in the sense that it does not provide any information to the chaser regarding its dynamic states. As a result, fundamental photogrammetry principles such as edge detection and bounding box manipulations applied to a stereo camera image form the basis of the obstacle detection method proposed in this work. An example image from the stereo camera is shown in Fig. 4. The size of the intake frame is limited by the frame rate and settings of the stereo camera. The camera frame rate is set to 15 fps to maximize the resolution to 4416×1242 pixels. During runtime, the image is split and calibrated in the left and right frames separately, then recombined into a single frame to process. The following subsections describe the steps of the proposed obstacle detection method.

1. *Frame Manipulation*

First, to limit the computational requirement of the obstacle detection task, the height of the intake frame is reduced by 30%. Next, it is converted from RGB to hue, saturation, and value (HSV). The following operations are applied to only the *value* component of the image, as this reduces the amount of image processing to one value per pixel and produces the most distinct result. The resulting frame after these two changes is shown in Fig. 5.

2. *Adaptive Thresholding*

Thresholding provides a binary interpretation of an image by classifying each pixel as either white or black based on a specified threshold. Adaptive thresholding improves this by using a square area around each pixel to determine a unique threshold value. The next step in the obstacle detection method involves blurring the image with OpenCV's `GaussianBlur` function to reduce Gaussian noise and create a continuous pixel range for thresholding. A block size of 15 pixels was found to yield the best detection in laboratory lighting. The image is then processed with OpenCV's `adaptiveThreshold` function, using a Gaussian-weighted sum of neighborhood values minus a constant (empirically set to 13) for optimal results. A comparison of standard and adaptive thresholding is shown in Fig. 6.

3. *Morphological Operations*

To further reduce low-frequency noise in the binary image, background noise (white pixels among majority black regions) is reduced with the "Morph Open" operation, and noise on the object

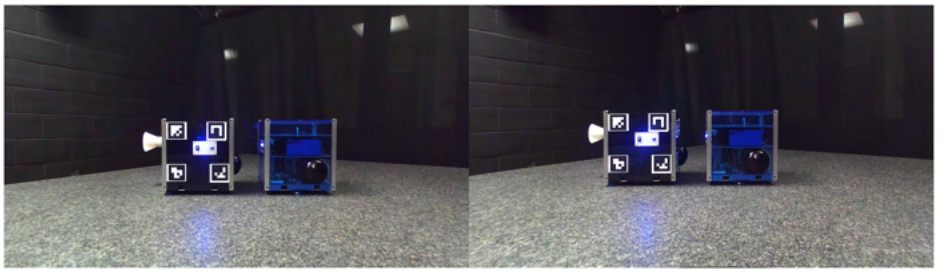


Fig. 4 Image from the ZED 2 stereo camera captured at 15 fps with 4416×1242 pixel resolution.

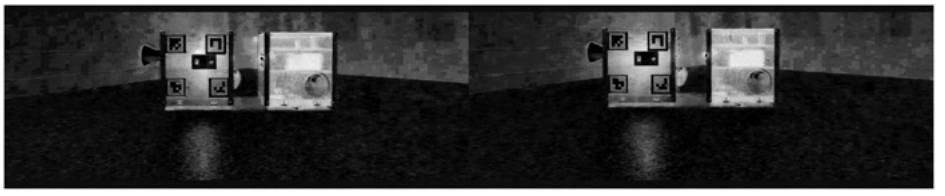
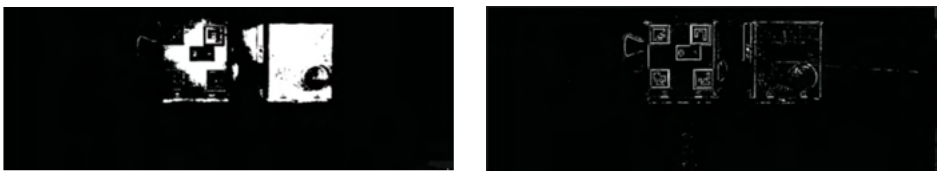


Fig. 5 Intake frame that has been cut down to 70% and converted to HSV color space.



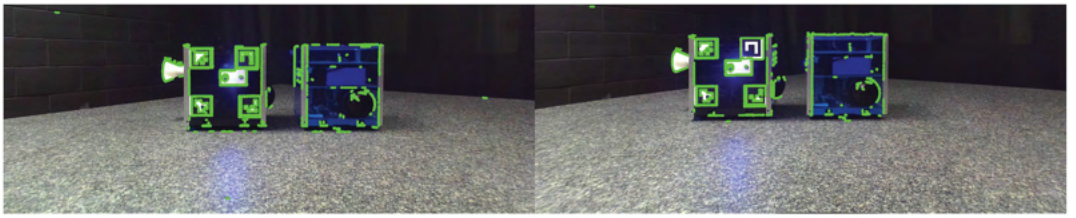
a) Standard thresholding **b) Adaptive thresholding**

Fig. 6 Examples of adaptive thresholding in comparison to standard thresholding.

of interest (black pixels scattered among the white object) is filled in with the "Morph Close" operation.

4. Contour and Bounding Box Generation

The next step is to find the contours tracing the detected edges. The image at this point looks similar to Fig. 6b: binary, with white regions representing edges found in the frame and black in areas without detected edges. To find the edges, contours are drawn around the enclosed white regions using OpenCV's `findContours` function. Small contours of 10 pixels or less are then filtered out to reduce noise and processing time. The output contours of this step drawn on the original sample image are shown in Fig. 7a. The contours, which are made of many points, are then encapsulated within bounding boxes to 1) reduce the number of points stored and 2) generate a larger area around the original contour, resulting in a larger obstacle keep-out zone. OpenCV provides a dedicated function, `boundingRect`, to determine the bounding boxes of a given contour, extending to the maximum limits of the contour in all four directions. Figure 7b shows the resulting bounding boxes superimposed onto the original image.



a) Simple contours

b) Bounding boxes

c) Bounding box size filter

d) ArUco proximity rejection

e) Merged bounding boxes

f) Overlapping boxes

Fig. 7 Obstacle edge detection workflow displayed on the colored sample image.

5. Bounding Box Filtering

Small bounding boxes are present after the previous step due to residual noise in the binary image. In this step, a bounding box size threshold is applied to reject all bounding boxes containing 300 pixels or less, resulting in Fig. 7c.

6. Target Detection and Rejection

As shown in Fig. 7c, the obstacle detection software is detecting both the obstacle and the target. Since the target is cooperative, the ArUco markers may be used to filter out the target from the obstacle detection routine. By detecting the ArUco markers on the target using previously discussed methods, the location of the target within the image can be estimated, and any detected bounding boxes in that region of the image can be eliminated. As a result, the only bounding boxes remaining are those detecting the obstacle, as shown in Fig. 7d.

7. Bounding Box Manipulation

The final step in locating the obstacle is merging bounding boxes that outline it. Boxes from each half of the image are merged if they lie within a specified distance, iterating until no two remain within that distance. Any remaining boxes smaller than a certain size, due to noise, are eliminated. The resulting bounding box around the obstacle in each half of the image is shown in Fig. 7e. To enhance detection, the boxes from the left and right frames are overlaid with a 66-pixel offset, and the overlapping area is used as the final obstacle position, as shown in Fig. 7f. Note that this process can be challenging to solve under harsh lighting conditions, as illustrated in Fig. 8.

8. Stereo Depth Mapping

The stereo disparity between the left and right boxes in Fig. 7e is insufficient to determine the distance to the obstacle due to the often large difference in box shapes. Instead, the distance to the obstacle is determined using a stereo depth map. First, a stereo disparity map is generated using OpenCV's `StereoSGBM` function, which is tuned to the expected operating lighting conditions, taking approximately 0.5 s to compute onboard the NVIDIA Jetson Orin. A sample stereo disparity map including both the target and obstacle is shown in Fig. 9. Using the stereo disparity map, the known distance between the two cameras L_{baseline} (0.12 m for the stereo camera used in this work), and the focal length L_{focal} (in pixels) of the camera, a depth map can be produced by applying the following transformation to each pixel:

$$D_{ij} = \frac{L_{\text{focal}} L_{\text{baseline}}}{d_{ij}} \tag{6}$$

where d_{ij} is the pixel disparity value and D_{ij} is the pixel depth value. Applied over the entire image, the calculation of the depth map takes 2 s. To avoid this prohibitively long processing time, the stereo depth map is only calculated over the desired region, which brings the compute time to less than 0.1 s. The desired region considered is the area contained within the final obstacle bounding box, 20 pixels from each side, which helps ensure no stereo shadow (i.e., areas of high contrast that are occluded by the object from the perspective of the second camera) is captured in the map. The obstacle distance is calculated as the average over the depth map values.

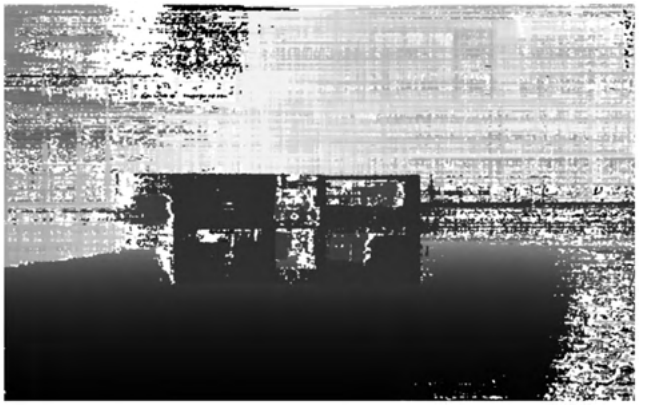


Fig. 9 Stereo disparity map of obstacle and target at ~1.5 m distance.

9. Position Estimation

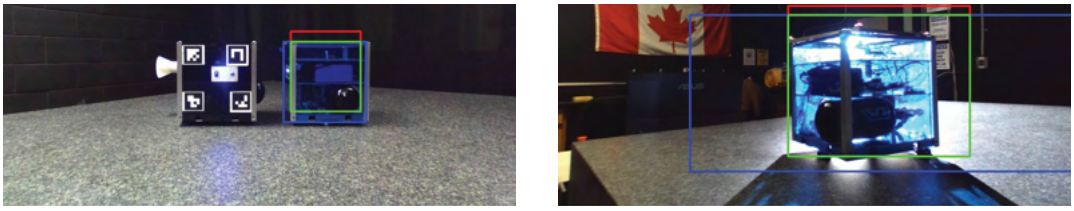
The distance estimation outputs the position of the obstacle in the camera reference frame, which is similarly along the $\boldsymbol{B}_{cx}$ axis. The position of the obstacle along the $\boldsymbol{B}_{cy}$ axis can be determined using the trigonometric relationships between the depth estimation, pixel width of the image, pixel position of the obstacle in the image, and the camera field-of-view angle. For integration with the guidance and control systems, the relative pose of the obstacle must be expressed in the inertial reference frame, which can be accomplished following a similar methodology as previously described for the relative target pose.

C. Unscented Kalman Filtering

In the literature, several stochastic filtering methods are available. An extended Kalman filter (EKF) was first considered; however, it assumes that noise on the states is Gaussian. This would not have been adequate for application in tandem with the computer vision system, which can have noise in multiple forms in addition to Gaussian, such as Poisson, impulse, and uniform noise [49,50]. The MEKF was considered as an alternative, as it was designed to address specific challenges in spacecraft state estimation, specifically quaternion-based representations for attitude estimation [51]. Quaternions, however, were not used for this particular application, rendering the MEKF algorithm identical to the EKF, and thus the same limitations applied. The UKF uses a deterministic sampling technique to capture a better estimate of the mean and covariance of the state distribution, making it robust to nonlinear transformations and non-Gaussian noise [52]. It does not rely on the assumption that noise is Gaussian, making it well-suited for the complex and varied noise environment encountered in spacecraft proximity operations [53].

The UKF algorithm applied in this work uses a modified sigma point method whereby the sigma point weights are calculated as

$$w_0^{(m)} = \frac{\lambda}{N+\kappa} \quad w_0^{(c)} = \frac{\lambda}{N+\lambda} + (1-\alpha^2) + \beta \quad w_i = \frac{1}{2(N+\lambda)} \tag{7}$$



a) Normal lighting condition **b) Harsh lighting condition**

Fig. 8 Overlapping boxes visualization under normal and harsh lighting conditions.

where $w_0^{(m)}$ is the weight for the first sigma point when calculating the weighted mean; $w_0^{(c)}$ is the weight for the first sigma point when calculating the weighted covariance; w_i is the weight for all other sigma points; N is the number of states; α, κ, and β are all tunable values; and $\lambda = \alpha^2(N + \kappa) - N$. The UKF prediction stage consists of calculating, propagating, and approximating a set of sigma points $\mathcal{X}$. Then, the UKF update stage transfers the state sigma points to the measurement space and updates the state estimates $\hat{x}$. For more details on the prediction stage and state estimate equations, the readers are referred to the work of Becker [53].

VI. Artificial Potential Function Guidance

The APF method uses the gradient of a potential field, composed of both attractive and repulsive potentials, to derive a suitable path to reach the goal while avoiding obstacles [54]. The desired path descends the gradient of the artificial potential field until the desired, final states are reached. Three potentials are used to describe the environment in this work. The attractive potential ϕ_a establishes a global minimum at the desired state $\boldsymbol{x}_f$ defined to be the state at which the chaser is docked to the target. The high-potential keep-out zone around the obstacle ϕ_r is represented by the attenuating Gaussian function, which repulses the chaser away from the obstacle. A target keep-out zone ϕ_b in the shape of a cardioid allows the chaser, when near the boundary but not yet at the docking cone, to follow the boundary while avoiding collisions with the target. The combination of repulsive and attractive potentials ensures that the chaser is guided smoothly around the boundary, preventing collisions while directing it toward the desired docking position. These three potentials, given by

$$\phi_a = \frac{k_a}{2}\boldsymbol{r}_{cf}^{\top}\boldsymbol{Q}_a\boldsymbol{r}_{cf} \quad \phi_r = \psi \exp\left\{-\frac{\boldsymbol{r}_{co}^{\top}\boldsymbol{N}\boldsymbol{r}_{co}}{\sigma}\right\}$$
$$\phi_b = k_r \exp\{-\boldsymbol{r}_{cb}^{\top}\boldsymbol{P}_b\boldsymbol{r}_{cb}\} \tag{8}$$

are summed to produce the total potential function $\phi_{\text{tot}} = \phi_a + \phi_r + \phi_b$. The attractive potential is defined by a positive scaling constant k_a set to unity, a symmetric, positive-definite attractive potential shaping matrix $\boldsymbol{Q}_a$, and the relative difference between the actual and desired chaser states, $\boldsymbol{r}_{cf} = \boldsymbol{x}_c - \boldsymbol{x}_f$. The obstacle repulsive potential is defined by scalars ψ and σ, which are used to shape the repulsive field about the shaping matrix $\boldsymbol{N}$. Specifically in this work, ψ is a binary variable used to activate the obstacle repulsive field, σ is set to 0.1, and $\boldsymbol{N}$ is set to the identity matrix $\boldsymbol{I}_3$. Note that $\boldsymbol{r}_{co} = \boldsymbol{x}_c - \hat{\boldsymbol{x}}_o$. The target repulsive potential is defined by a positive scaling constant k_r set to 2 and a positive-definite scaling matrix $\boldsymbol{P}_b$ set to s. The subscript b denotes the boundary of the target keep-out zone. The relative position vector can be expressed as $\boldsymbol{r}_{cb} = \boldsymbol{r}_m - d(\alpha)(\boldsymbol{r}_m/||\boldsymbol{r}_m||)$, where $\boldsymbol{r}_m = \boldsymbol{x}_m - \hat{\boldsymbol{x}}_t$ and $\boldsymbol{x}_m = [x_c \; y_c \; \theta_t]^{\top}$. The variable $\boldsymbol{r}_m$ is the state of the chaser relative to the target with the attitude component equal to zero such that the repulsion from the target is due to the relative position only. The subscript m denotes the modified field defined for the purposes of this work. The estimated target and obstacle states $\hat{x}_t$ and $\hat{\boldsymbol{x}}_o$, respectively, are obtained from the UKF, as shown in Fig. 1. The cardioid boundary constraint, as shown in Fig. 10, is aligned with the chaser body-fixed axis along the line of sight of its docking port ($\boldsymbol{B}_{cx}$) and centered at the intersection of $\boldsymbol{B}_{cx}$ with the target body-fixed axis parallel to its docking port ($\boldsymbol{B}_{tx}$), when the chaser is in its desired docking state $\boldsymbol{x}_f$. At $\boldsymbol{x}_f$, the offset of the chaser's COM along $\boldsymbol{B}_{tx}$ is $r_{\text{off}} = 0.165$ m. The value $d(\alpha)$, defined as the distance from the center of the cardioid to the intersection with its boundary along the vector to the chaser position, is defined in polar coordinates by [55]

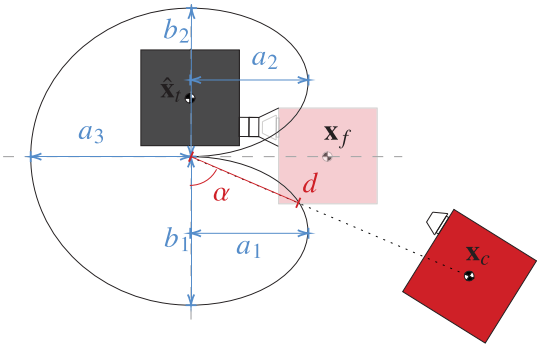


Fig. 10 Parameters for repulsive constraint of the target.

$$d(\alpha) = \begin{cases} \dfrac{2b_1a_1^2\cos(\alpha)}{a_1^2\cos^2(\alpha) + b_1^2\sin^2(\alpha)}, & \alpha \in \left[0, \dfrac{\pi}{2}\right) \\ \dfrac{-2b_2a_2^2\cos(\alpha)}{a_2^2\cos^2(\alpha) + b_2^2\sin^2(\alpha)} & \alpha \in \left[\dfrac{\pi}{2}, \pi\right) \\ \dfrac{2b_2a_3}{\sqrt{a_3^2\cos^2(\alpha) + 4b_2^2\sin^2(\alpha)}} & \alpha \in \left[\pi, \dfrac{3\pi}{2}\right) \\ \dfrac{2b_1a_3}{\sqrt{a_3^2\cos^2(\alpha) + 4b_2^2\sin^2(\alpha)}} & \alpha \in \left[\dfrac{3\pi}{2}, 2\pi\right) \end{cases} \tag{9}$$

where a_1, a_2, b_1, b_2 are defined in the target body-fixed frame as shown in Fig. 10. These constants were manually selected based on the dimensions of the target and chaser platforms to ensure that when the chaser's center of mass follows the boundary, it maintains a safe distance from the target. This work uses values of $a_1 = a_2 = 0.3$, $a_3 = 0.4$, and $b_1 = b_2 = 0.175$ m.

The repulsive target potential function ϕ_b is a reformulation of existing functions [54,55] to make the cardioid boundary compatible with a closed-loop feedback control law. It was noted that previous keep-out zones around the target did not behave as intended when a closed-loop feedback control law was used as opposed to the open-loop feedback controller used in the referenced papers. In simulation, as the chaser approached the desired orientation before reaching the target, the constraint around the target started to diminish, losing its repulsive effect on the chaser. The enhanced target potential function ϕ_b proposed in this work ensures the angle orientation of the chaser has no effect on the repulsion, such that constraint around the target is always present. The chaser is still driven to the correct attitude by the attractive potential ϕ_a.

Figure 11 plots the contours of the total potential field around the target and obstacle on SPOT. Note that the potential fields depend not just on x and y, but also on θ. For these plots, it was set that $\theta_c = \theta_t$ for every (x, y) on the table. Since APF guidance involves descending the 3D gradient of ϕ_{tot}, it is invaluable to have an

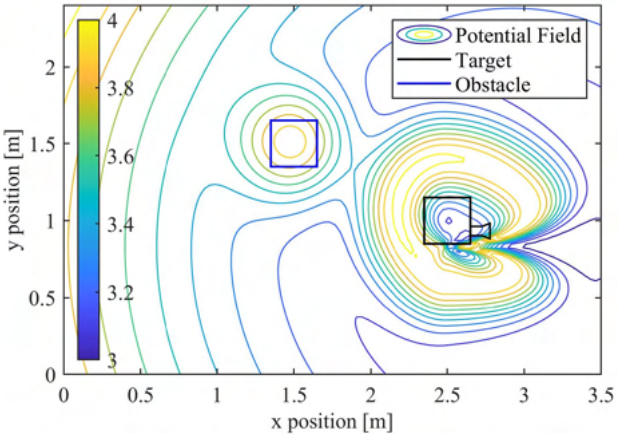


Fig. 11 Contour plot of total potential function.

analytical expression for the gradient of this field with respect to $\boldsymbol{x}_c$ at any point. This gradient is given by

$$\nabla_{\boldsymbol{x}_c}\phi_{\text{tot}} = k_a \boldsymbol{Q}_a \boldsymbol{r}_{cf} - \frac{2\psi}{\sigma}\exp\left\{-\frac{\boldsymbol{r}_{co}^{\top}\boldsymbol{N}\boldsymbol{r}_{co}}{\sigma}\right\}\boldsymbol{N}\boldsymbol{r}_{co} - 2k_r \exp[-\boldsymbol{r}_{cb}^{\top}\boldsymbol{P}_b\boldsymbol{r}_{cb}]\boldsymbol{P}_b\boldsymbol{r}_{cb} \tag{10}$$

A. Gradient Descent

The gradient descent method—appealing due to its relative computational efficiency—is used to generate the chaser docking trajectory $\tilde{\boldsymbol{x}}_c$. The differentiable function $\phi_{\text{tot}}(\boldsymbol{x}_c)$ is minimized by adjusting the position of the chaser in the opposite direction of the gradient of the potential with respect to the current position of the chaser, $\nabla_{\boldsymbol{x}_c}\phi_{\text{tot}}(\boldsymbol{x}_c)$, as defined in Eq. (10). Following the opposite direction of the gradient will direct the chaser to the low potential of the field—the desired state $\boldsymbol{x}_f$. Two gradient descent methods are used in this work. Note that for both algorithms, the parameters were selected initially as per best practices recommended in Ref. [56]. Then, they were fine-tuned empirically through simulations. These values were tested in different simulated scenarios to evaluate their performance. Once docking was successfully achieved in each scenario, the corresponding parameter values were chosen to be implemented in experiments.

1. Adam

`Adam` [56] is a gradient descent algorithm that updates the learning rate for each parameter. It stores an exponentially decaying average of the past gradients $\boldsymbol{m}_t$ (first moment) and an exponentially decaying average of the past squared gradients $\boldsymbol{v}_t$ (second moment) given by

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)[\nabla_{\boldsymbol{x}_c}\phi(\boldsymbol{x}_{c,t})] \tag{11}$$

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)[\nabla_{\boldsymbol{x}_c}\phi(\boldsymbol{x}_{c,t}) \odot \nabla_{\boldsymbol{x}_c}\phi(\boldsymbol{x}_{c,t})] \tag{12}$$

Since $\boldsymbol{m}_t$ and $\boldsymbol{v}_t$ are initialized as vectors of zeros, they exhibit a bias toward zero, particularly in the initial time steps and especially when the decay rates are small (i.e., when β_1 and β_2 are close to 1). These biases are corrected by computing the first and second moment estimates, which are given by

$$\hat{m}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t} \tag{13}$$

$$\hat{v}_t = \frac{\boldsymbol{v}_t}{1-\beta_2^t} \tag{14}$$

The chaser docking trajectory $\tilde{\boldsymbol{x}}_c$ at each time step is then calculated as

$$\tilde{\boldsymbol{x}}_c = \boldsymbol{x}_c - \frac{\eta}{\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon}\hat{\boldsymbol{m}}_t \tag{15}$$

where all operations are elementwise. The chosen values for β_1, β_2, and ϵ in this work are 0.99, 0.999, and 0.01, respectively. The learning rate η is a parameter that determines the speed with which the gradient is descended and is set to 0.0163. When compared to other gradient descent algorithms in simulations, `Adam` demonstrates a slower approach with minimum trajectory overshoot and superior consistency across experiments.

2. Adamax

A variant of `Adam` called `Adamax` outperforms `Adam` in some scenarios. `Adamax` differs from `Adam` in its utilization of the infinity norm $\boldsymbol{l}_\infty$ of the past gradient instead of applying the average decaying of the past squared gradients $\boldsymbol{v}_t$. The first moment is calculated using Eq. (13); however, Eq. (14) is replaced by the infinity norm-constrained $\boldsymbol{v}_{t,\infty}$, calculated as

$$\begin{aligned}\boldsymbol{v}_{t,\infty} &= \beta_2^{\infty}\boldsymbol{v}_{t-1,\infty} + (1-\beta_2^{\infty})|\nabla_{\boldsymbol{x}_c}\phi(\boldsymbol{x}_{c,t})|^{\infty} \\ &= \max(\beta_2 \boldsymbol{v}_{t-1,\infty}, |\nabla_{\boldsymbol{x}_c}\phi(\boldsymbol{x}_{c,t})|)\end{aligned} \tag{16}$$

where the second term of the max operation is the absolute value of the gradient vector. The chaser docking trajectory $\tilde{\boldsymbol{x}}_c$ at each time step is then calculated as

$$\tilde{\boldsymbol{x}}_c = \boldsymbol{x}_c - \frac{\eta}{\boldsymbol{v}_{t,\infty}}\hat{\boldsymbol{m}}_t \tag{17}$$

In this work, `Adamax` constants β_1 and β_2 are set to 0.5 and 0.999, respectively, with the learning rate η set to 0.014. The use of $\boldsymbol{v}_{t,\infty}$ inherently removes the need for bias correction, ensuring that the estimates are not biased toward zero, even during the early stages of time steps when the estimates are initialized. One notable advantage of `Adamax` is its efficient handling of sparse gradients, such as those near the desired state where the gradient may become very small, allowing for faster convergence speed. This can also be useful when the chaser is close to an obstacle where gradient values may become large. In these cases, `Adamax` provides more stable updates, reducing potential instability compared to `Adam`.

B. Modification for Computer Vision

To ensure consistent computer vision state data of the target is available and to minimize the system's reliance on the Kalman filter state propagation, the chaser must maintain its attitude such that its stereo camera is always facing the target. To accomplish this, the APF is further modified to neglect the angular component of the gradient and instead replace it with a separate controller that ensures the chaser is constantly looking at the target. This could introduce issues in the presence of distributed obstacles where the chaser cannot keep all objects of interest in view at one time. In this case, a multicamera or wide-angle vision system would be beneficial. Alternatively, the estimated object states could be propagated using a filter while they are out of frame. To remove the attitude from the APF, the attractive potential shaping matrix takes the form $\boldsymbol{Q}_a = \text{diag}(Q_{a_{11}}, Q_{a_{22}}, 0)$ where $Q_{a_{11}}, Q_{a_{22}}$ are tuning parameters. In this work, $\boldsymbol{Q}_a = \text{diag}(5, 5, 0)$. The reference angle for error tracking, i.e., the θ-component of $\tilde{x}_c$, is instead defined as

$$\tilde{\theta}_c = \text{atan2}(r_{cl,y}, r_{cl,x}) + \pi \tag{18}$$

where $r_{cl,x} = x_c - (\hat{x}_t + r_{\text{off}}\cos\hat{\theta}_t)$ and $r_{cl,y} = y_c - (\hat{y}_t + r_{\text{off}}\sin\hat{\theta}_t)$. An additional phase of π is added in Eq. (18) for alignment with the face on the chaser where the camera is mounted.

VII. Adaptive Trajectory Tracking Control

The single-axis adaptive control law is given by [57]

$$u(t) = K_1(t)e(t) + K_2(t)\dot{e}(t) \tag{19}$$

where $e(t)$ is a component of the output error signal vector $\boldsymbol{e}$ defined in Eq. (4) and $u(t)$ is the associated component of the control input vector $\boldsymbol{u}$ defined in Eq. (5). The adaptation mechanisms for both adaptive parameters $K_1(t)$ and $K_2(t)$ are given as

$$\dot{K}_1(t) = -\lambda_1 p e(t)^2 - (\lambda_1 + p\lambda_{12})e(t)\dot{e}(t) - \lambda_{12}\dot{e}(t)^2 \tag{20}$$

$$\dot{K}_2(t) = -\lambda_{12} p e(t)^2 - (\lambda_{12} + p\lambda_2)e(t)\dot{e}(t) - \lambda_2\dot{e}(t)^2 \tag{21}$$

where λ_1, λ_2, λ_{12}, and p are tunable constants. In this work, the constants used in the DACs for the translational motion were set as $\lambda_1 = \lambda_2 = \lambda_{12} = p = 0.4$ and for the rotational motion were set as $\lambda_1 = \lambda_2 = \lambda_{12} = p = 0.05$. The adaptive parameters for the translational motion DACs were initialized to $K_1(0) = 3.5$ and

$K_2(0) = 28$, and for the rotational motion, DACs were initialized to $K_1(0) = K_2(0) = 1$.

VIII. Experimental Results

This section presents the experimental results of the integrated GNC system incorporating computer vision techniques, APF guidance, and DAC as discussed in the previous sections. Two test cases were considered. The first case demonstrates rendezvous and docking with a spinning target, and the second demonstrates rendezvous and docking with a spinning target while avoiding an unknown obstacle. Test case 1 was conducted using a sun simulator that generates harsh lighting conditions, increasing glare and harsh contrast as seen by the stereo camera, which must be overcome by the computer vision technique, whereas test case 2 was performed in normal ambient lighting conditions, as illustrated in Fig. 12. The initial states of the platforms for each test case are listed in Table 1. Note that the target states are able to be estimated accurately, even for angular rates larger than 1.5 deg/s, as long as its motion is close to torque-free motion. The target platform's maximum rate of rotation is limited by the actuation capabilities of the chaser platform's thrusters. Indeed, if the rotation rates of the target are high, the control inputs required by the chaser would be too high to be physically delivered through the thruster hardware. Otherwise, the actual target rotation rate would not have any impact on the performance. Note that the actual angular rate of the obstacle does not have any effect on the results, as its APF potential is a symmetric Gaussian distribution.

The `Adam` gradient descent is implemented for test case 1, and `Adamax` is implemented for test case 2. A video showcasing successful docking in both test cases is available online at https://youtu.be/31x2ANIjW_I. During each experiment, the following sequence is carried out.

Phase 1 is the target acquisition phase. In this phase, the chaser holds its position and rotates with a constant angular rate of 0.025 rad/s using an LQR scheme until the target is detected.

Phase 2 corresponds to the visual observation phase and is initiated once the target is detected. Then, the chaser maintains its position and line of sight to the target (via an LQR scheme) to collect additional visual information about the target and obstacle states, which allows the Kalman filter's error covariance to properly converge before initiating the rendezvous and docking maneuver. Indeed, Kalman filters are known to exhibit a large variance between state estimates when initiated with the first measured data point [53].

Finally, at time 25 s, Phase 3, which corresponds to the rendezvous and docking maneuver based on the integrated GNC approach (see Fig. 1), is initiated. Using the developed computer vision algorithms, the relative pose of the target and obstacle (if applicable) is determined online and converted to the inertial frame using knowledge of the chaser state via PhaseSpace. The UKF estimates the inertial pose of the target and obstacle and feeds this information —along with the known chaser state—to the APF guidance law, which calculates the desired docking trajectory in real-time. A DAC is used to actively track the desired trajectory.

A. Test Case 1: Rendezvous and Docking with a Spinning Target

Five distinct trials were successfully demonstrated for test case 1. Figure 13 shows the chaser and target platforms pictured in their initial (transparent) and final (solid) states, along with the different trajectories that represent different trials. Despite variance in the paths across the five trials, docking was achieved in each one. The 3σ error in the inertial states (Fig. 13) and adaptive controller tracking (Fig. 14) grows as the experiment progresses through time, especially in terms of attitude, which is expected due to the variability of the real-time APF-generated path during each trial. With only a single trial requiring longer than 130 s to dock, the 3σ error collapses to that single trial thereafter. Performance is quantified in terms of root-mean square error (RSME), total control effort, and total path length. The RMSE of the adaptive controller over the course of a sample trial was determined to be 8.4 ± 0.1, 9.2 ± 0.1 mm, and 2.59 ± 0.08 deg in the x, y, and θ states, respectively. The control input is integrated numerically over the experiment time to obtain the total impulse for the maneuver. The distance between the (x, y) coordinates of each adjacent time step is summed to yield the total path length. For trials 1–5, the total impulse was 24.76, 15.69, 17.44, 17.66, and 4.27 N·s, respectively, and the total path length was 4.55, 3.05, 3.14, 3.00, and 1.28 m, respectively. The average total impulse is 15.96 N·s, and the average path length is 3.00 m. Both averages are lowered due to trial 5, where the chaser docked significantly faster than the others.

The real-time computer vision algorithm executed onboard the NVIDIA Jetson Orin was restricted by the camera frame rate of 15 fps. Although data transfer from the NVIDIA Jetson Orin to the NVIDIA Jetson Xavier—which runs the remaining GNC algorithms—was sampled every 0.013 s, the target state estimates

a) Test case 1 **b) Test case 2**

Fig. 12 Lighting conditions for both test cases.

Table 1 Platform initial states for experiment test cases

		Chaser	Target	Obstacle
Initial states	**Test case 1**	(2.7, 0.4, −180, 0, 0,0)	(1.3, 1.25, 270, 0, 0, −1.5)	— —
$(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})$[a]	**Test case 2**	(2.7, 0.4, −180, 0, 0,0)	(1.3, 1.25, 90, 0, 0, −1.5)	(1.75, 0.5, 188, 0, 0, 1.5)

[a]Translational states are given in units of m and m/s. Rotational states are given in units of deg and deg/s.

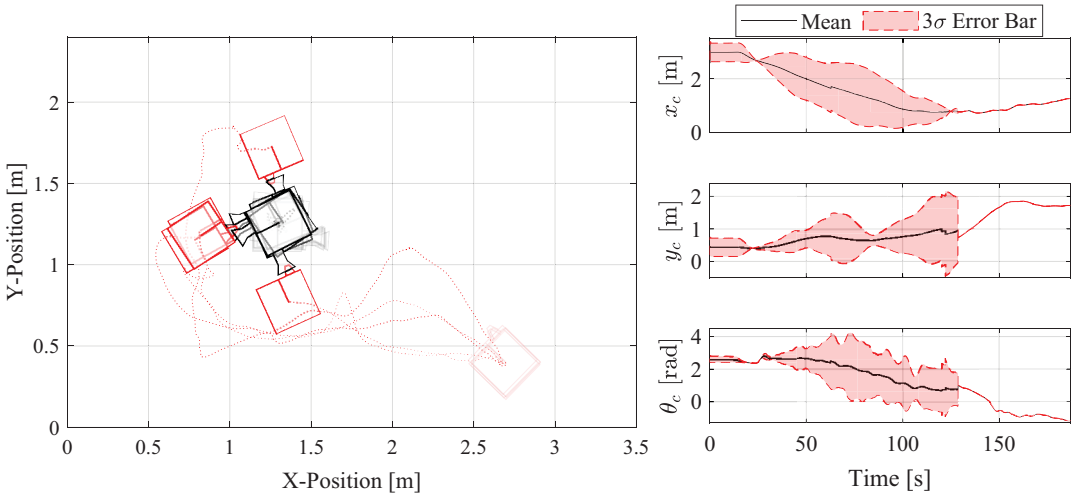


Fig. 13 Experimental results for test case 1.

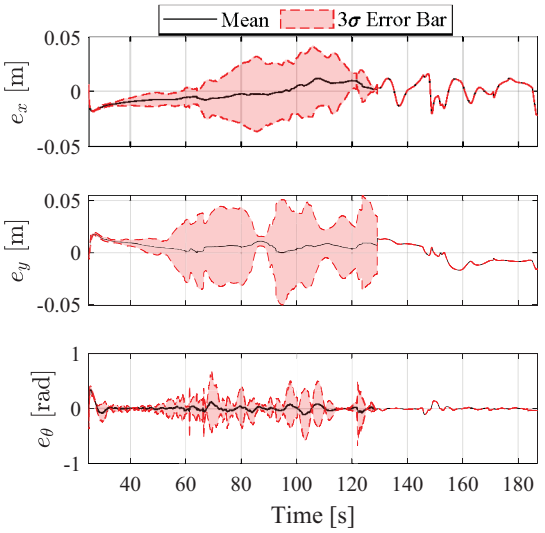


Fig. 14 Trajectory tracking error for test case 1.

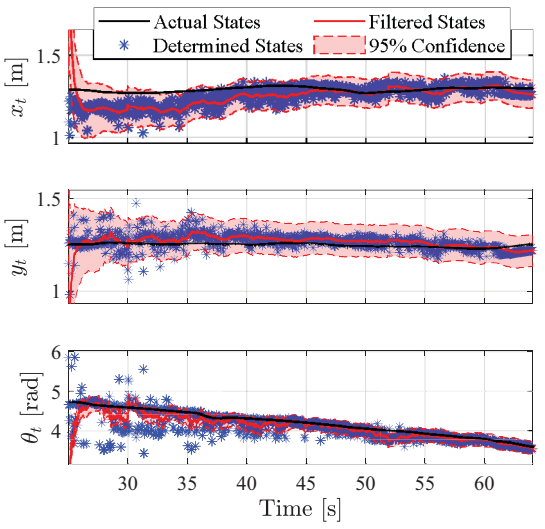


Fig. 15 Computer vision and UKF data for target pose determination for test case 1.

were only updated every 4–5 samples (about 15–19 Hz) once a new stereo image was captured. The target platform states, as determined from the real-time computer vision algorithm for a sample trial, are plotted by the blue markers in Fig. 15.

The filtered states of the target platform for a sample trial are also plotted in Fig. 15, which are compared to the actual states of the target as determined using the PhaseSpace motion capture system. The root-mean-squared error (RMSE) of the filtered target state estimates with a 95% confidence interval obtained using a bootstrapping method was determined for this specific trial to be 6.1 ± 0.1, 2.78 ± 0.04 cm, and 6.3 ± 0.1 deg in the x, y, and θ states, respectively.

B. Test Case 2: Rendezvous and Docking with a Spinning Target and Stationary Obstacle

A successful trial was demonstrated for test case 2 that further considers an unknown obstacle. In this experiment, the sun simulator was not used since the obstacle detection software—unlike the ArUco-based target pose determination software—was not capable of consistently overcoming harsh lighting conditions. Figure 16 plots the experimental results of the successful docking experiment. Similar to test case 1, the RMSE of the adaptive controller (Fig. 17) was determined to be 9.9 ± 0.1, 10.0 ± 0.1 mm, and 6.7 ± 0.2 deg in the x, y, and θ states, respectively.

Executing the sequentially combined target pose detection software and obstacle detection software on the NVIDIA Jetson Orin, the real-time implementation of the computer vision algorithm was limited by the computational expense of the obstacle detection code. On average, the obstacle detection code required 2 s to execute, and the target pose determination code required 0.14 s to execute. To achieve suitable sampling performance, the obstacle detection code was executed once for every 10 executions of the target pose determination code. The blue markers in Fig. 18 show this clear distinction in sampling rates. The response of the UKF is also shown in Fig. 18. In particular, the UKF exhibits a significant delay in responding to new measurements at around 80 s of the θ-plot in Fig. 18a; however, this algorithm still was more sensitive to new measurements than other tested filters. Figure 18b further showcases the outlier rejection and propagation capabilities of the UKF. Indeed, in the absence of measurements, the UKF propagates the position of the obstacle, assuming a torque-free dynamics model. As such, the chaser is always aware of the obstacle position, even when the obstacle is out of frame.

The RMSE in the filtered target states was determined to be 8.30 ± 0.07, 4.19 ± 0.03 cm, and 17.4 ± 0.1 deg in the x, y, and θ states, respectively. These RMSE values are all larger than those found during test case 1, which is likely due to the lower sampling frequency used during test case 2. The RMSE in

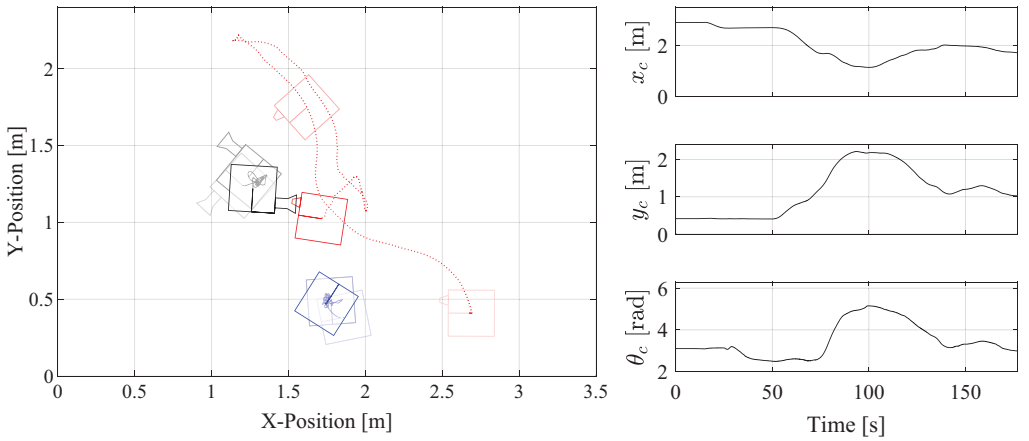


Fig. 16 Experimental results for test case 2.

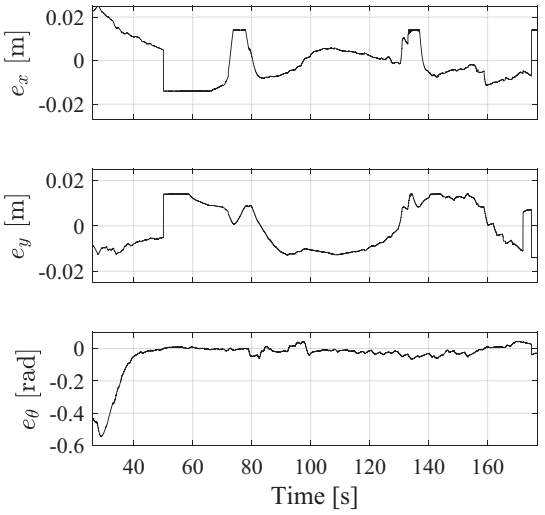


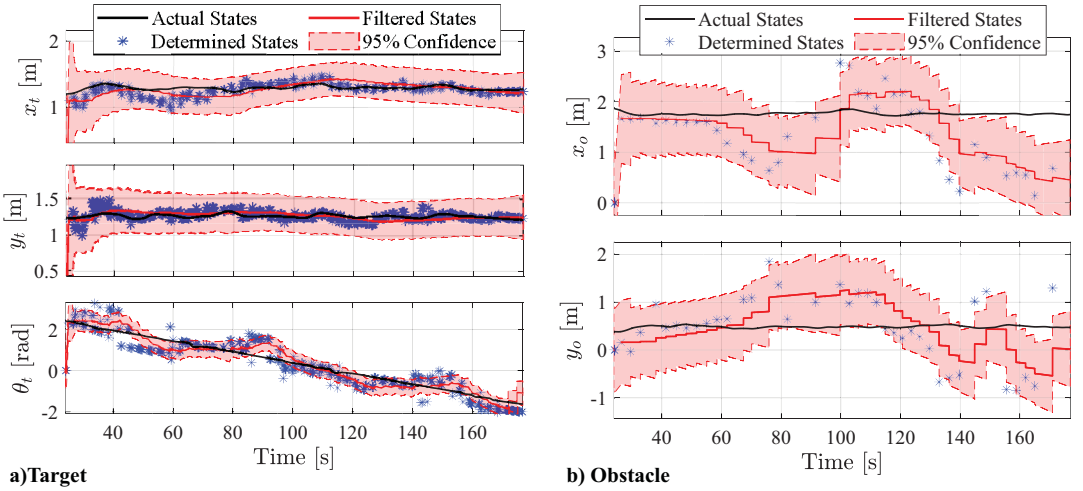
Fig. 17 Trajectory tracking error for test case 2.

the filtered obstacle states was determined to be 63.1 ± 0.5 and 51.4 ± 0.3 cm in the x and y states, respectively. The data obtained from the obstacle detection has quite a large distribution, and with a large enough obstacle keep-out zone defined in the APF guidance law, such large uncertainties in the knowledge of the obstacle state can be overcome. This, however, comes at a cost of overly conservative trajectories that may prohibit certain maneuvers. In this experiment, the large distribution comes from the fact that objects other than the obstacle were likely detected. Indeed, since the chaser was programmed to point at the estimated target location, the obstacle was not (fully) in frame throughout the experiment. The plots in Fig. 18b show that the determined states agree with the true obstacle states until about 60 s. Detections thereafter when the obstacle was out of the frame should have been rejected—an area for future work.

IX. Conclusions

Autonomous rendezvous and docking in a frictionless planar environment was experimentally achieved through the integration of computer vision, guidance, and control techniques. Cooperative target relative pose determination is accomplished through ArUco marker detection. Simultaneously, obstacle pose estimation is achieved using bounding box manipulation and stereo depth mapping. This data is fed through an unscented Kalman filter (UKF) to filter noise before being used to generate an artificial potential field (APF). Docking was achieved in two cases: in harsh lighting conditions with no obstacle and in ambient lighting conditions with an uncooperative obstacle. The `Adam` and `Adamax` gradient descent algorithms, for each case, respectively, were integrated with a direct adaptive controller (DAC) to ensure that the chaser was able to quickly and smoothly approach and dock with the rotating target.

Although this work demonstrated the proposed architecture in a planar environment, application of these methods in upcoming spacecraft proximity operations will be done in a six-degree-of-freedom (6-DoF) environment. The OpenCV functions for



a)Target **b) Obstacle**

Fig. 18 Computer vision and UKF data for target pose determination and obstacle detection for test case 2.

ArUco-based pose estimation already provide 6-DoF pose parameters; however, further modifications will need to be made for the remainder of the control architecture. For example, the UKF implementation would require modifications to the dynamic model to handle full 6-DoF motion. Ideally, it would operate on an error-state formulation using modified Rodrigues parameters. The APF guidance law would require new, higher-dimensional potential functions, including a 6D function for the attractive potential and 3D functions for the target and the obstacle. The DAC was derived based on the planar system dynamics and would need to be reworked to account for the full 6-DoF spacecraft dynamics, including attitude coupling effects.

Acknowledgments

Special thanks to Sydney Moses, Adrian Comisso, Laurenne Tynski, Hayden Arms, Ryan McPherson, Adriana Steckel, and Finnian Tuck for their technical contributions to this work. This research was financially supported in part by the Natural Sciences and Engineering Research Council of Canada under the Discovery Grant program in addition to the Ontario Graduate Scholarship program, Carleton University's Kostiuk Engineering Funding Collective, the Future Funder crowdfunding platform, and Carleton University's Department of Mechanical and Aerospace Engineering.

References

[1] Kessler, D. J., and Cour-Palais, B. G., "Collision Frequency of Artificial Satellites: The Creation of a Debris Belt," *Journal of Geophysical Research*, Vol. 83, No. A6, 1978, pp. 2637–2646. https://doi.org/10.1029/JA083iA06p02637

[2] Eberle, C., and Sebesvari, Z., "Technical Report: Space Debris," *Interconnected Disaster Risks 2023: Risk Tipping Points*, United Nations University - Institute for Environment and Human Security (UNU-EHS), 2023. https://doi.org/10.53324/YIKU7602

[3] ESA Space Debris Office, "ESA's Annual Space Environment Report," GEN-DB-LOG-00288-OPS-SD, TR, European Space Agency, 2023.

[4] Kessler, D. J., Johnson, N. L., Liou, J., and Matney, M., "The Kessler Syndrome: Implications to Future Space Operations," *33rd Annual AAS Guidance and Control Conference*, Vol. 137, Advances in the Astronautical Sciences, Breckenridge, CO, 2010, pp. 1–15.

[5] Kasai, T., Oda, M., and Suzuki, T., "Results of the ETS-7 Mission-Rendezvous Docking and Space Robotics Experiments," *5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (ISAIRAS)*, Vol. 440, Noordwijk, Netherlands, 1999, p. 299. https://adsabs.harvard.edu/full/1999ESASP.440.299K.

[6] Friend, R. B., "Orbital Express Program Summary and Mission Overview," *SPIE Defense and Security Symposium, Sensors and Systems for Space Applications II*, Vol. 6958, SPIE, Orlando, FL, 2008, pp. 11–21. https://doi.org/10.1117/12.783792

[7] Biesbroek, R., Aziz, S., Wolahan, A., Cipolla, S.-f., Richard-Noca, M., and Piguet, L., "The ClearSpace-1 Mission: ESA and ClearSpace Team Up to Remove Debris," *8th European Conference on Space Debris*, ESA Space Debris Office, Darmstadt, Germany, 2021, pp. 1–3, https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/320/SDC8-paper320.pdf.

[8] El-Shawa, S., Cattani, B. M., Innocenti, L., and Delaval, J., "From Cradle to Grave: ESA Clean Space's Approach to Space Sustainability," *72nd International Astronautical Congress*, International Astronautical Federation, Dubai, United Arab Emirates, 2021, pp. 1–12.

[9] Sullivan, K., and Ben-Itzhak, S., "Securing the Space Domain with Active Orbital Debris Removal: Lessons from Europe and Japan Towards a US Strategy," *Astropolitics*, Vol. 21, No. 2-3, 2023, pp. 127–149. https://doi.org/10.1080/14777622.2023.2274922

[10] 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

[11] 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., Porto, Portugal, March 2012, https://elib.dlr.de/76201/.

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

[13] 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*, 2024, pp. 1–25. https://doi.org/10.1007/s42064-024-0198-4

[14] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J., "Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion," *Pattern Recognition*, Vol. 47, No. 6, 2014, pp. 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005

[15] MacLean, S. G., and Pinkney, H. F. L., "Machine Vision in Space," *Canadian Aeronautics and Space Journal*, Vol. 29, No. 1, 1993, pp. 63–77.

[16] Howard, R., Heaton, A., Pinson, R., and Carrington, C., "Orbital Express Advanced Video Guidance Sensor," *IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2008, pp. 1–10, https://doi.org/10.1109/IROS.1992.587290

[17] Hannah, S. J., "ULTOR Passive Pose and Position Engine For Spacecraft Relative Navigation," *SPIE Defense and Security Symposium*, SPIE, Orlando, FL, April 2008. https://doi.org/10.1117/12.777193

[18] Bodin, P., Larsson, R., Nilsson, F., Chasset, C., Noteborn, R., and Nylund, M., "PRISMA: An In-Orbit Test Bed for Guidance, Navigation, and Control Experiments," *Journal of Spacecraft and Rockets*, Vol. 46, No. 3, 2009, pp. 615–623. https://doi.org/10.2514/1.40161

[19] Vela, C., Fasano, G., and Opromolla, R., "Pose Determination of Passively Cooperative Spacecraft in Close Proximity Using a Monocular Camera and AruCo Markers," *Acta Astronautica*, Vol. 201, 2022, pp. 22–38. https://doi.org/10.1016/j.actaastro.2022.08.024

[20] Muralidharan, V., Makhdoomi, M. R., Baraba, K. R., Amaya-Mejia, L. M., Howell, K., Martineza, C., and Olivares-Mendeza, M. A., "Hardware-in-the-Loop Proximity Operations in Cislunar Space," *International Astronautical Congress*, International Astronautical Federation, Paris, France, Sept. 2022.

[21] Tweddle, B., and Saenz-Otero, A., "Relative Computer Vision-Based Navigation for Small Inspection Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 5, 2015, pp. 969–977. https://doi.org/10.2514/1.G000687

[22] Valverde, A., Filipe, N., Kontitsis, M., and Tsiotras, P., "Experimental Validation of an Inertia-Free Controller and a Multiplicative EKF for Pose Tracking and Estimation Based on Dual Quaternions," *38th AAS Guidance and Control Conference*, AAS, Brekenridge, CO, Feb. 2015.

[23] Rizzolo, G., "Unscented Kalman Filters for Vision-Based Relative Dynamics Estimation Between Cooperating Spacecraft," Ph.D. Thesis, Univ. of Padova, Padova, 2019.

[24] Jewison, C. M., "Guidance and Control for Multi-sStage Rendezvous and Docking Operations in the Presence of Uncertainty," Ph.D. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2017.

[25] Nolet, S., "Development of a Guidance, Navigation and Control Architecture and Validation Process Enabling Autonomous Docking to a Tumbling Satellite," Ph.D. Thesis, Massachusetts Inst. of Technol., Cambridge, MA, 2007.

[26] Kim, Y. H., and Spencer, D. B., "Optimal Spacecraft Rendezvous Using Genetic Algorithms," *Journal of Spacecraft and Rockets*, Vol. 39, No. 6, 2002, pp. 859–865. https://doi.org/10.2514/2.3908

[27] 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

[28] McCamish, S. B., Romano, M., Nolet, S., Edwards, C. M., and Miller, D. W., "Flight Testing of Multiple-Spacecraft Control on SPHERES During Close-Proximity Operations," *Journal of Spacecraft and Rockets*, Vol. 46, No. 6, 2009, pp. 1202–1213. https://doi.org/10.2514/1.43563

[29] Bevilacqua, R., Lehmann, T., and Romano, M., "Development and Experimentation of LQR/APF Guidance and Control for Autonomous Proximity Maneuvers of Multiple Spacecraft," *Acta Astronautica*,

Vol. 68, Nos. 7-8, 2011, pp. 1260–1275.
https://doi.org/10.1016/j.actaastro.2010.08.012
[30] Zappulla, R., Park, H., Virgili-Llop, J., and Romano, M., "Real-Time Autonomous Spacecraft Proximity Maneuvers and Docking Using an Adaptive Artificial Potential Field Approach," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 6, 2018, pp. 2598–2605.
https://doi.org/10.1109/TCST.2018.2866963
[31] Mahendrakar, T., Holmberg, S., Ekblad, A., Conti, E., White, R. T., Wilde, M., and Silver, I., "Autonomous Rendezvous with Non-Cooperative Target Objects with Swarm Chasers and Observers," *33rd AAS/AIAA Space Flight Mechanics Meeting*, AAS Paper 23-423, Jan. 2023.
https://doi.org/10.48550/arXiv.2301.09059
[32] Sachdev, J., "Spacecraft Formation Flying Using L1 Adaptive Control Techniques," Master's Thesis, Carleton Univ., Ottawa, ON, 2017.
[33] Hough, J., and Ulrich, S., "Immersion and Invariance Adaptive Control for Proximity Operations Under Uncertainties and Modeling Errors," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2020-1599, Jan. 2020.
https://doi.org/10.2514/6.2020-1599
[34] Edgardt, B., *Stability of Adaptive Controllers*, Springer, Berlin, 1979, pp. 1–8.
[35] Ulrich, S., Saenz-Otero, A., and Barkana, I., "Passivity-Based Adaptive Control of Robotic Spacecraft for Proximity Operations Under Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 6, 2016, pp. 1444–1453.
https://doi.org/10.2514/1.G001491
[36] Predmyrskyy, A., and Ulrich, S., "Design Heuristics and Optimization for Simple Adaptive Controllers with Applications to Spacecraft Proximity Operations," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 59, No. 3, 2023, pp. 2296–2306.
https://doi.org/10.1109/TAES.2022.3211825
[37] Ulrich, S., "Nonlinear Passivity-Based Adaptive Control for Spacecraft Formation Flying," *American Control Conference*, Inst. of Electrical and Electronics Engineers, New York, July 2016, pp. 7432–7437.
https://doi.org/10.1109/ACC.2016.7526846
[38] Chihabi, Y., and Ulrich, S., "Hybrid Guardian Map-Based Adaptive Control of Spacecraft Formation Flying on Highly Elliptical Orbits in the Restricted Three-Body Problem," *Acta Astronautica*, Vol. 180, 2021, pp. 370–385.
https://doi.org/10.1016/j.actaastro.2020.12.048
[39] Hong, J., and Bernstein, D. S., "Experimental Application of Direct Adaptive Control Laws for Adaptive Stabilization and Command Following," *38th IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, New York, Dec. 1999, pp. 779–784.
https://doi.org/10.1109/CDC.1999.832884
[40] Hong, J., and Bernstein, D. S., "Adaptive Stabilization of Non-Linear Oscillators Using Direct Adaptive Control," *International Journal of Control*, Vol. 74, No. 5, 2001, pp. 432–444.
https://doi.org/10.1080/713806130
[41] Zappulla, R., Park, H., Virgili-Llop, J., and Romano, M., "Real-Time Autonomous Spacecraft Proximity Maneuvers and Docking Using an Adaptive Artificial Potential Field Approach," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 6, 2019, pp. 2598–2605.
https://doi.org/10.1109/TCST.2018.2866963
[42] 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
[43] Virgili-Llop, J., Zagaris, C., Zappulla, R., Bradstreet, A., and Romano, M., "A Convex-Programming-Based Guidance Algorithm to Capture a Tumbling Object on Orbit Using a Spacecraft Equipped with a Robotic Manipulator," *International Journal of Robotics Research*, Vol. 38, No. 1, 2019, pp. 40–72.
https://doi.org/10.1177/0278364918804660
[44] Eun, Y., Kim, G.-N., Hyun, J., and Park, S.-Y., "Experimental Results on the Vision-Based Navigation System for Spacecraft Operation in Proximity Using Ground Test Facility," *2018 Space Flight Mechanics Meeting*, AIAA Paper 2018-2222, Jan. 2018.
https://doi.org/10.2514/6.2018-2222
[45] Opromolla, R., Vela, C., Nocerino, A., and Lombardi, C., "Monocular-Based Pose Estimation Based on Fiducial Markers for Space Robotic Capture Operations in GEO," *Remote Sensing*, Vol. 14, No. 18, 2022, p. 4483.
https://doi.org/10.3390/rs14184483
[46] Zhang, G., Vela, P., Tsiotras, P., and Cho, D.-M., "Efficient Closed-Loop Detection and Pose Estimation for Vision-Only Relative Localization in Space with A Cooperative Target," *AIAA SPACE 2014 Conference and Exposition*, AIAA Paper 2014-4262, Aug. 2014.
https://doi.org/10.2514/6.2014-4262
[47] Fourie, D., Tweddle, B. E., Ulrich, S., and Saenz-Otero, A., "Flight Results of Vision-Based Navigation for Autonomous Spacecraft Inspection of Unknown Objects," *Journal of Spacecraft and Rockets*, Vol. 51, No. 6, 2014, pp. 2016–2026.
https://doi.org/10.2514/1.A32813
[48] Kalaitzakis, M., Cain, B., Carroll, S., Ambrosi, A., Whitehead, C., and Vitzilaios, N., "Fiducial Markers for Pose Estimation: Overview, Applications and Experimental Comparison of the ARTag, AprilTag, ArUco and STag Markers," *Journal of Intelligent and Robotic Systems*, Vol. 101, No. 4, 2021, pp. 1–26.
https://doi.org/10.1007/s10846-020-01307-9
[49] Joshua, O., Ibiyemi, S., and Adu, B., "A Comprehensive Review on Various Types of Noise in Image Processing," *International Journal of Scientific and Engineering Research*, Vol. 10, No. 11, 2019, pp. 388–393.
[50] Rodríguez-Rodríguez, J. A., López-Rubio, E., Ángel-Ruiz, J. A., and Molina-Cabello, M. A., "The Impact of Noise and Brightness on Object Detection Methods," *Sensors*, Vol. 24, No. 3, 2024, p. 821.
https://doi.org/10.3390/s24030821
[51] Hassan, H. A., Tolstrup, W., Suriana, J. P., and Kiziloklu, I. D., "A Comparative Analysis Between the Additive and the Multiplicative Extended Kalman Filter for Satellite Attitude Determination," arXiv preprint arXiv:2307.06300, 2023.
https://doi.org/10.48550/arXiv.2307.06300
[52] Hajiyev, C., and Ata, M., "Unscented Kalman Filter Based Two-Stage Estimation of Spacecraft Position and Velocity with Single Station Antenna Tracking Data," *7th International Conference on Recent Advances in Space Technologies (RAST)*, IEEE Publ., Piscataway, NJ, June 2015, pp. 777–782.
https://doi.org/10.1109/RAST.2015.7208445
[53] Becker, A., "Non-Linear Kalman Filters," *Kalman Filter from the Ground Up*. 2nd ed., kalmanfilter.net, Israel, 2023, pp. 287–327.
[54] Zappulla, R., II, Park, H., Virgili-Llop, J., and Romano, M., "Real-Time Autonomous Spacecraft Proximity Maneuvers and Docking Using an Adaptive Artificial Potential Field Approach," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 6, 2019, pp. 2598–2605.
https://doi.org/10.1109/TCST.2018.2866963
[55] Zappulla, R., II, "Experimental Evaluation Methodology for Spacecraft Proximity Maneuvres in a Dynamic Environment," Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, 2017.
[56] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations*, ICLR, Appleton, WI, May 2015.
https://doi.org/10.48550/arXiv.1412.6980
[57] Rao, V. G., and Bernstein, D. S., "Naive Control of the Double Integrator," *IEEE Control Systems Magazine*, Vol. 21, No. 5, 2001, pp. 91–97.
https://doi.org/10.1109/37.954521

D. Selva
Associate Editor