AAS 21-419

# CHALLENGE PROBLEM: ASSURED SATELLITE PROXIMITY OPERATIONS*

**Christopher D. Petersen† Sean Phillips ‡ Kerianne L. Hobbs § and Kendra Lang¶**

Assuring safety of a spacecraft during autonomous rendezvous, proximity operations, and docking is a non-trivial, multi-constraint challenge. This paper presents a challenge problem relating to in-plane autonomous rendezvous and docking of a basic 6U CubeSat model with thrust in one axial direction, a reaction wheel for attitude control, and a gimbaled sensor. To assure the safety of the satellite throughout the entire operation, a variety of constraints are introduced with notional values. Moreover, we list several metrics to compare the developed algorithms. Lastly, the benefits and disadvantages for several different solution approaches to the challenge problem are discussed.

## INTRODUCTION

The space community is in the middle of a complexity paradigm shift. Many are looking to large numbers of heterogeneous small satellites to accomplish missions never before attempted. Commercial interests are looking to constellations of satellites for space-based services like on-demand imagery and global internet service. There are growing enterprises, new infrastructures and technologies being developed for complicated missions such as satellite servicing and active debris removal. All of this involves a level of complexity never before seen in mission design and operation. However spacecraft operations today are developed by human teams days in advance [1], and thus would be difficult to support the missions stated above. To close this gap, development in autonomous techniques are necessary to enable human-on-the-loop operations that are more supportive to evolving complex needs. This paper in particular discusses the complexity involved in developing autonomy for safe and efficient proximity operations, offering up the challenge to develop techniques for solving the mission criteria imposed.

While autonomy could enable faster and larger numbers of spacecraft operations, there is valid concern for the safety of the missions. Rendezvous and docking as an autonomous operation is particularly relevant to the satellite servicing and debris removal applications required to manage increased space traffic. Applications are under continuous development, but mission safety is especially critical and challenging. Multiple spacecraft collisions have occurred during rendezvous

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PUBLIC AFFAIRS RELEASE APPROVAL AFMC-2020-0303.

†Research Aerospace Engineer, Space Vehicles Directorate, Kirtland AFB, New Mexico.

‡Research Mechanical Engineer, Space Vehicles Directorate, Kirtland AFB, New Mexico.

§Research Aerospace Engineer, Autonomy Capability Team (ACT3), Air Force Research Laboratory 2210 8th Street, Wright-Patterson AFB, Ohio, 43433

¶Technical Director, Space and Autonomy, 6100 Uptown Blvd NE, Albuquerque, New Mexico, 87110.

[2, 3, 4, 5]. In the 1990s, two incidents occurred during manned docking with the MIR space station. In 1994, the Soyuz TM-17 ferry spacecraft caused minor damage when it docked with the MIR space station, resulting in only minor damage [2], and in 1997, the Progress M-34 spacecraft damaged MIR solar panels, radiators, and punctured the hull in a collision. During an autonomous docking experiment in 2005 the DART spacecraft collided with the MUBLCOM spacecraft, sending MUBLCOM into a higher orbit with no significant damage [5]. Metrics are therefore needed for such operations to compare solutions and to devise specifications that define success.

The challenge problem presented here seeks to meet the above-mentioned need by providing a general benchmark for spacecraft autonomous rendezvous, proximity operations, and docking (ARPOD) for an under-actuated spacecraft system. The problem objective and safety constraints introduced reflect the complexity necessary for safe and efficient rendezvous and docking. A list of multiple (but not all inclusive) techniques are discussed to address the objective, demonstrating how placing a layer of autonomy is required for this operation. In addition, specifications and suggested metrics are included for evaluating solution quality and reasonableness.

The paper is organized as follows. First, previously posed ARPOD challenge problems are discussed and compared with this challenge problem. In the next section, the reference frame, dynamics, and logic notation used to describe the problem are introduced. Next, a problem formulation is presented including the problem statement and constraints before introducing several performance metrics. The last section outlines different approaches to solving the satellite docking planning and control problem from classical control to learning techniques.

## RELATED WORK

There has been a fair amount of work on the problem of autonomous spacecraft rendezvous and docking in the literature. A benchmark problem for hybrid control and estimation during ARPOD [6] was described in 2016. In this problem, a "chaser" spacecraft transports "target" passive components. The paper enumerates the phases of the ARPOD mission, including the sensors and dynamics available at each phase. The following on-board sensors are considered: radar, GPS, laser range finders ($<$ 1 km), stereo cameras ($<$100 m). Additional orbit and rendezvous benchmarks based on the Clohessy-Wiltshire Hill (CWH) equations have been proposed in [7, 8]. In [9], an obstacle avoidance algorithm was developed based on a sliding-mode controller to ensure that the chaser spacecraft does not enter into the keep-out zone. Similarly, finite-time convergence has been shown in [10]. A Gaussian mixture model is leveraged to reconstruct a satellite shape and coupled with an artificial potential function for collision avoidance in proximity operations [11]. In [12], a robust adaptive control method is shown to have the asymptotic convergence to the origin.

There are several papers examining the topic of rendezvous and docking with non-cooperative spacecraft; e.g., a tumbling or malfunctioning satellite body in need of servicing. In [13], the authors show the asymptotic stability of a sliding-mode controller coupled with artificial potential functions to dock with a non-cooperative spacecraft while adhering to keep-out zones of virtual obstacles. In [14], the authors consider multiple cases of tumbling satellites including tumbling with respect to the axis of the docking port, two axes are not orthogonal and, lastly, for the case when two axes are in line. In [15], the authors leverage model predictive control for autonomous rendezvous employing constraints in the control input, collision avoidance, velocity and a dock-enabled conditions.

This paper builds on [6], by examining a hypothetical spacecraft with coupled translational and rotational dynamics via actuator design, enumerating several safety constraints, describing the pros and challenges of different controls approaches, and defining metrics for success of an ARPOD

algorithm. There are some preliminary results that address components of this problem, however, to the best of our knowledge, there hasn't been a full closed-loop solution to this problem with the dynamics and constraints. In [16], the authors leverage a geometric control method to achieve the desired docking configuration without constraints. In [17], a model predictive control (MPC) method is used for successful docking with both control input and spacecraft velocity constraints, but does not consider keep-out zones constraints.

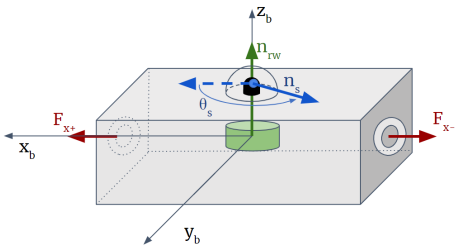


**Figure 1**: Hypothetical 6U Cubesat with thrusters (red) aligned with positive and negative $x$-axes, a sensor (blue) that is able to gimbal around the $z$-axis to any direction within the $xy$-plane, and a reaction wheel (green) for single axis attitude control aligned with the $z$-axis

.

## PRELIMINARIES

### Model

The model used in this challenge problem is a 6U CubeSat measuring 10 cm x 20 cm x 30 cm, as pictured in Fig. 1. In this model, thrust is only possible from two thrusters assumed to be perfectly aligned with the spacecraft body x-axis $x_b$ on either side of the spacecraft. A gimbaled sensor with boresight $n_s$, only able to rotate around the body z-axis, is equipped to the satellite, with its boresight $\theta_s$ angle measured from the x-axis. The attitude of the spacecraft about its z-axis $n_{rw}$ is controlled by a reaction wheel. Specific values for the vehicle can be found in Table 1. Note that while numeric values are given, techniques developed can be utilized for other akin satellite configurations.

**Table 1**: Mission Parameters

| Variable | Description | Value |
|---|---|---|
| $D$ | reaction wheel spin axis mass moment of inertia | $4.1\times10^{-5}$kg-m$^2$ |
| $I_{xx}$ | spacecraft mass moment of inertial in x-axis | 0.022kg-m$^2$ |
| $I_{yy}$ | spacecraft mass moment of inertial in y-axis | 0.044kg-m$^2$ |
| $I_{zz}$ | spacecraft mass moment of inertial in z-axis | 0.056kg-m$^2$ |
| $m_j$ | spacecraft mass | 12 kg |
| $\mu$ | Earth's gravitational parameter | $(3.986004418\pm0.000000008)\times10^{14}\,\frac{\text{m}^3}{\text{s}^2}$) |
| $n$ | mean motion | 0.001027 rad/s |

### Equations of Motion

Consider two spacecraft (one of which is the CubeSat described above) orbiting about Earth, with the overarching goal to rendezvous and dock both satellites together. One spacecraft, called

the target will be the subject of docking. The other, called the chaser, will approach the target and dock with it (this is the CubeSat discussed above). To begin, the following assumptions are made for the equations of motion.

**Assumption 1** *Both satellites are rigid bodies.*

**Assumption 2** *The mass of Earth is significantly greater than the mass of the satellites.*

**Assumption 3** *The mass loss of the spacecraft is significantly smaller than the total mass of the spacecraft.*

Assumption 1 applies to most modern spacecraft as fuel slosh and moving mass are typically not a significant part of the spacecraft vehicle dynamics. Assumption 2 consolidates the magnitude of gravity into a universal gravity parameter $\mu$. Assumption 3 results in mass remaining constant and is reasonable as propellant usage over short time intervals is not tremendous. Under these assumptions, both the target and chaser satellites revolve around the Earth, governed by the following equations of motion.

$$\ddot{\vec{R}}_j = -\frac{\mu}{|\vec{R}_j|^3}\vec{R}_j + \frac{1}{m_j}(\vec{U}_j + \vec{W}_j) \tag{1}$$

where $j \in \{t, c\}$ are the subscripts denoting the target and chaser, respectively, $\vec{R}_j \in \mathbb{R}^3$ is the position vector of the spacecraft in the inertial frame, $\ddot{\vec{R}}_j$ is the acceleration of position with respect to the inertial frame, $m_j$ is the mass, $\vec{U}_j \in \mathbb{R}^3$ are the actuation forces, and $\vec{W}_j \in \mathbb{R}^3$ are forces from other sources or higher fidelity gravity modeling. If the terms $\vec{U}_j$ and $\vec{W}_j$ are neglected, the equations of motion in (1) can yield stable, elliptical orbits about the center of gravity of the frame, which due to Assumption 2, is approximately the center of the Earth. The following additional assumptions are made to linearize the spacecraft dynamics.

**Assumption 4** *The target spacecraft is in a circular orbit with radius* $|\vec{R}_t| = r_0$.

**Assumption 5** *The distance from the target spacecraft to the chaser spacecraft is significantly less than that of the distance from the target spacecraft to the center of the Earth.*

Assumption 4 is reasonable as a good number of satellites are near-circular. With regards to Assumption 5, rendezvous typically happens on the order of tens of kilometers, while the distance to any satellite from the Earth is on the order of thousands of kilometers at least.

With the target spacecraft in an equilibrium orbit due to Assumption 4, a non-inertial frame is attached to it specifically for the purposes of rendezvous. The following axes were chosen to define the non-inertial reference frame used in this paper:

- $\hat{e}_R$ for "radial direction" that points outwards from Earth's center $\left(\hat{e}_R = \frac{\vec{R}_t}{||\vec{R}_t||}\right)$,

- $\hat{e}_N$ for "normal direction" that is aligned with the angular momentum vector of an orbit, which is constant and always points orthogonal to the orbital plane $\left(\hat{e}_N = \frac{\vec{R}_t \times \dot{\vec{R}}_t}{||\vec{R}_t \times \dot{\vec{R}}_t||}\right)$.
- $\hat{e}_T$ for "tangential direction" which completes an orthogonal coordinate system with $\hat{e}_R$ and $\hat{e}_N$ ($\hat{e}_T = \hat{e}_N \times \hat{e}_R$). Note that for a circular orbit, $\hat{e}_T$ and the inertial orbital velocity $\dot{\vec{R}}$ are aligned.

The above frame is called the Hill's Frame and was developed in the 1870s to describe the orbit of one body about each other [18]. Other similar frames include the local-vertical local-horizontal (LVLH) or radial-tangential normal (RTN) frame, which differ only slightly in terms of axis definition. See Figure 2.

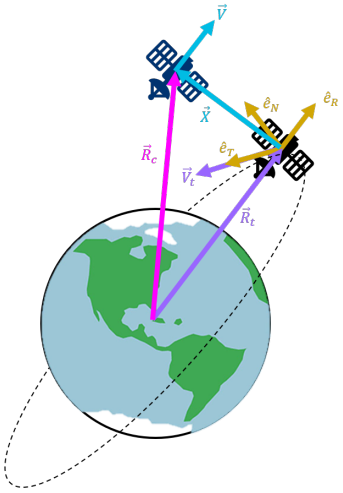


**Figure 2**: Hill's reference frame.

Define the relative position vector of the chaser with respect to the chief as $\vec{X}$, given by the following relation

$$C\vec{R}_c = \begin{bmatrix} r_0 \\ 0 \\ 0 \end{bmatrix} + \vec{X}$$

where the rotation matrix $C$ transforms a vector in the inertial frame to the Hill's frame. Utilizing Assumptions 4 and 5 the relative equations of motion abide by the following dynamics:

$$\begin{aligned} \dot{\vec{X}} &= \vec{V} \\ \dot{\vec{V}} &= A_x\vec{X} + A_v\vec{V} + B(\vec{F} + C(\vec{W}_c - \vec{W}_t)) \end{aligned} \tag{2}$$

where $\vec{V}$ is the velocity in the Hill's frame, $\vec{F} = C\vec{U}_c$ is the actuation force in the Hill's frame and $A_x$, $A_v$ and $B$ are given by

$$A_x = \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix}, \qquad A_v = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} \frac{1}{m_c} & 0 & 0 \\ 0 & \frac{1}{m_c} & 0 \\ 0 & 0 & \frac{1}{m_c} \end{bmatrix} \tag{3}$$

where $n = \sqrt{\frac{\mu}{r_0^3}}$ is the mean motion on the target spacecraft's circular orbit (this is also the circular angular velocity magnitude of the chief spacecraft in the inertial frame). Note that the linear equations of motion can be decoupled into the orbital in-plane motion (which lives in the $\hat{e}_R - \hat{e}_T$ plane) and out-of-plane motion (which lives in the $\hat{e}_N$ direction). While under more complex modeling, these equations of motion become re-coupled, it is outside the scope of this problem. In this paper only in-plane motion is considered. If $\vec{X} = [x,\ y,\ z]$, representing the $\hat{e}_R,\ \hat{e}_T$, and $\hat{e}_N$ respectively, then the in-plane motion is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m_c & 0 \\ 0 & 1/m_c \end{bmatrix} (\vec{F}_{xy} + (\vec{w}_{c,xy} - \vec{w}_{t,xy})) \tag{4}$$

where $\vec{F}_{xy}$, $\vec{w}_{c,xy}$, $\vec{w}_{t,xy}$ are the chaser control forces, the chaser other forces, and target other forces in the $\hat{e}_R$-$\hat{e}_T$ plane, respectively.

The rotational equations of motion are dictated by the conversation of angular momentum. With regards to the above planar case, the rotational degree of freedom is about only the $\hat{e}_N$ axis. As the inertia of the spacecraft about that axis is $I_{zz}$, the inertia of a reaction wheel about that spin axis then $D$, and $\theta_3$ is the rotation of the spacecraft about $\hat{e}_N$, measured from the $\hat{e}_T$ direction to the $x_b$ axis, the rotational equations of motion are

$$I_{zz}\ddot{\theta}_3 = -D\dot{\psi} \tag{5}$$

where $\dot{\psi}$ is commanded acceleration given to the reaction wheel to produce a required torque.

With the rotational degree defined, the translational equations in (4) are modified to incorporate the single axis thruster discussed above. This can be done by defining the thruster magnitude $F_x$. Then

$$\vec{F}_{xy} = \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix} F_x. \tag{6}$$

Then, (4) can be rewritten as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m_c & 0 \\ 0 & 1/m_c \end{bmatrix} \left( \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix} F_x + (\vec{w}_{c,xy} - \vec{w}_{t,xy}) \right). \tag{7}$$

**Temporal Logic and Set Notation**

Constraints in this challenge problem are expressed in *past time linear temporal logic* (ptLTL), which uses temporal operators to describe the past states of an execution trace relative to the current point of reference [19]. A list of ptLTL symbols is listed in precedence order in Table 2, and the semantics are described below.

**Definition 6 (Semantics of ptLTL [20, Definition 2])** *A linear structure $\pi$ over a finite set of propositions $\mathcal{A}$ is a function $\pi : \mathbb{N} \to 2^{\mathcal{A}}$ [20]. Let $\pi$ be a linear structure over $\mathcal{A}$, let $\varphi$ and $\psi$ be ptLTL formulas, and let $i, j, k \in \mathbb{N}$. $\varphi$ is true in $\pi$, written $\pi \models \varphi$ if and only if (iff) $(\pi, 0) \models \varphi$. Then $\varphi$*

**Table 2**: Summary of Past Time Linear Temporal Logic (ptLTL) Symbols.

| Symbol | Description | Translation | Alternative Symbols |
|---|---|---|---|
| $\square$ | historically/always | "always" | $[*]$ |
| $\bigcirc^{-1}$ | previous step | "previously" | |
| $\neg$ | negation | "not" | ! |
| $\cup$ | until | "until" | U |
| $\wedge$ | conjunction | "and" | |
| $\vee$ | disjunction | "or" | |
| $\Rightarrow$ | implication | "implies" | $->$ |
| $\Leftrightarrow$ | logical equivalence | "is equivalent to" | $<->$ |

*holds in $\pi$ at time $i$, written $(\pi, i) \models \varphi$, is inductively defined as follows:*

$$
\begin{array}{lcl}
(\pi,i) \models p & \iff & p \in \pi(i) \\
(\pi,i) \models \square\varphi & \iff & \forall j \leq i, (\pi,j) \models \varphi \\
(\pi,i) \models \bigcirc^{-1}\varphi & \iff & i > 0 \textit{ and } (\pi, i-1) \models \varphi \\
(\pi,i) \models \neg\varphi & \iff & (\pi,i) \not\models \varphi \\
(\pi,i) \models \varphi \cup \psi & \iff & \exists j \leq i, ((\pi,j) \models \psi \textit{ and } \forall k : j < k \leq i, (\pi,k) \models \varphi \\
(\pi,i) \models \varphi \wedge \psi & \iff & (\pi,i) \models \varphi \textit{ and } (\pi,i) \models \psi \\
(\pi,i) \models \varphi \vee \psi & \iff & (\pi,i) \models \varphi \textit{ or } (\pi,i) \models \psi
\end{array}
$$

In addition to ptLTL, some requirements are expressed in set notation where $x \in \mathcal{A}$ indicates $x$ is an element of $\mathcal{A}$, $x \in [x_{ll}, x_{ul}]$ indicates $x_{ll} \leq x \leq x_{ul}$, and $x \in (x_{ll}, x_{ul})$ indicates $x_{ll} < x < x_{ul}$.

## FORMULATION

### Problem Statement

Building on the previous benchmark problem in [6], this paper considers an ARPOD problem, wherein the chaser satellite must ensure safety of itself and the target satellite at all times during rendezvous and docking. Given the above preliminaries above, the problem can be summarized as follows.

*Assume a chaser satellite with initial state and a target satellite with the Hill's frame fixed to the target. Assume the chaser satellite is initialized at $r(0) > 10\ km$ and achieves the following:*

*P1) Asymptotically dock with target satellite at $(x, y, \dot{x}, \dot{y}, \theta_3, \dot{\theta}_3) = (0, 0, 0, 0, 0, 0)$.*

*P2) Ensure that the camera is never pointed at the sun.*

*P3) Maintain safety of both the chaser satellite and the target satellite throughout the duration of the solution.*

In this problem, the notion of safety is encoded in numerous imposed constraints explored in the next section.

## Constraints

The solution to the challenge problem presented in this document should adhere to the constraints described in this section, and any verification approach should attempt to verify that all or a subset of these constraints are met. A summary of the safety constraint formalization is presented in Table 3.

**Table 3**: Summary of formalized safety constraints in ptLTL.

| Requirement | Description | Value |
|---|---|---|
| $\varphi_{F_{x_{max}}}$ | $\square F_x \in [F_{x_{min}}, F_{x_{max}}]$ | $F_{x_{min}}$ = -1 N and $F_{x_{max}}$ = 2 N |
| $\varphi_{\dot{\boldsymbol{X}}_{lim}}$ | $\square(\lvert\dot{x}\rvert \leq v_{x_{max}}) \wedge (\lvert\dot{y}\rvert \leq v_{y_{max}})$ | $v_{x_{max}} = v_{y_{max}} = v_{max} = 10$ m/s |
| $\varphi_{rv}$ | $\square\lvert\lvert\vec{v}_H\rvert\rvert \leq v_{H_{max}}$ | $v_{H_{max}} = \frac{(f_s)(\lvert\lvert\vec{r}_H\rvert\rvert)}{T_c(F_{max}, m_j, \lvert\lvert\vec{r}_H\rvert\rvert)}$ |
| $\varphi_{cnmt}$ | $\square\lVert\vec{v}_{\mathrm{H}}\rVert \leq \nu_0 + \nu_1\lVert\vec{r}_{\mathrm{H}}\rVert$ | $\nu_0 = 0.2$ m/s, $\nu_1 = 2n\ s^{-1}$ |
| $\varphi_{\ddot{\theta}_{max}}$ | $\square\Big(\ddot{\theta}_3 \in [\ddot{\theta}_{3_{min}}, \ddot{\theta}_{3_{max}}]\Big)$ | $\ddot{\theta}_{3_{min}} = -1$ deg/s$^2$, $\ddot{\theta}_{3_{max}} = 1$ deg/s$^2$ |
| $\varphi_{\dot{\theta}_{max}}$ | $\square\Big(\dot{\theta}_3 \in [\dot{\theta}_{3_{min}}, \dot{\theta}_{3_{max}}]$ | $\dot{\theta}_{3_{min}} = -2$ deg/s, $\dot{\theta}_{3_{max}} = 2$ deg/s |
| $\varphi_{\psi_{max}}$ | $\square\lvert\psi\rvert \leq \psi_{max}$ | $\psi_{max} = 576.0$ rad/s |
| $\varphi_{\dot{\psi}_{max}}$ | $\square\lvert\dot{\psi}\rvert \leq \dot{\psi}_{max}$ | $\dot{\psi}_{max} = 181.3$ rad/s$^2$ |
| $\varphi_{\theta_{EZ}}$ | $\square\theta_{EZ} > \frac{\alpha_{EZ}}{2} + \beta_{EZ}$ | $\beta = 15°$ and $\alpha = 8°$ |
| $\varphi_{\theta_s}$ | $\square\theta_R < \frac{\alpha_s}{2}$ | $\alpha_s = 25°$ |

*Thrust Limits*

- *Bounded thrust force* ($\square F_x \in [F_{x_{min}}, F_{x_{max}}]$): The spacecraft should obey reasonable thrust limitations. There are a variety of ways to represent this such as a bounded control input [21, 22, 23, 24], bounded thrust force [25, 26], or bounded change in velocity (i.e., $\Delta V$) [27, 28, 29]. In this challenge problem, a net limit is considered, resulting in a net maximum and minimum thrust level.

- *Asymmetric bounded thrust* (let $F_{x_{min}}$ = -1 N and $F_{x_{max}}$ = 2 N): In some cases, thrust capabilities may not be equal in all directions, or acceleration may be limited in some directions but not in others when there are deployed solar panel, booms, antennas, etc. In this case, specific thrust limits may be assigned for each direction. Due to size and weight constraints, it is assumed in this problem that one thruster is twice as powerful as the other.

*Velocity Limits*

- *Recoverable relative velocity limit*: Spacecraft maneuvers shall maintain recoverable relative motion. While the linearized dynamics of the system are completely controllable, it is possible for the relative velocity to be so high that it exceeds capabilities of the actuators to arrest motion within a limited time frame. One way to deal with this is to place limits on the maximum velocity:

$$\varphi_{v_{max}} = \square(\lvert\dot{x}\rvert \leq v_{x_{max}}) \wedge (\lvert\dot{y}\rvert \leq v_{y_{max}}). \tag{8}$$

  Let $v_{max}$ be the maximum velocity that the spacecraft can travel in the x or y direction and stop within one minute given the available thrust.

$$v_{x_{max}} = v_{y_{max}} = v_{max} = \frac{F_{x_{max}}}{m_j}(t_{stop}) = \frac{2 \text{ kg m/s}^2}{12 \text{ kg}}(60s) = 10 \text{ m/s} \tag{9}$$

- *Bounded relative velocity limit*: The chaser spacecraft should not be traveling exceedingly fast relative to it's distance from the target. In cases where spacecraft formation flying or a controlled collision (docking) is intended, the magnitude of the acceptable relative velocity moving may be based on the distance between the spacecraft, where acceptable relative velocity decreases as the spacecraft distance decreases. This concept is notionally borrowed from the idea of using a temporal rather than distance requirement to avoid collisions where the time-to-collision $T_c$ is the separation distance $||\vec{r}_H||$ over closure velocity $||\vec{v}_H||$ [30]:

$$T_c = \frac{||\vec{r}_H||}{||\vec{v}_H||}. \tag{10}$$

  Alternatively, $T_c$ may be a function of the available thrust, spacecraft mass, and distance from the target, where

$$T_c(F_{max}, m_j, ||\vec{r}_H||) = \sqrt{\Big(\frac{2m_j}{F_{max}}(||\vec{r}_H||)\Big)} \tag{11}$$

  The relative velocity of the target spacecraft should be less than the limit defined by a safety factor $f_s$ the following equation:

$$\varphi_{rv} = \square||\vec{v}_H|| \leq v_{H_{max}}, v_{H_{max}} = \frac{(f_s)(||\vec{r}_H||)}{T_c(F_{max}, m_j, ||\vec{r}_H||)} \tag{12}$$

  A second variation of the velocity constraint may be based on a "Plan B" to maneuver to a parking orbit, such as a an elliptical closed natural motion trajectory (CNMT), around the target. Plan B might be selected if the original docking plan is determined to be unsafe. Using this concept as inspiration, the system is defined to be safe if it obeys the following safety constraint,

$$\varphi_{cnmt} = \square\|\vec{v}_{\mathrm{H}}\| \leq \nu_0 + \nu_1\|\vec{r}_{\mathrm{H}}\| \tag{13}$$

  where, $\nu_0,\ \nu_1 \in \mathbb{R}_{\geq 0}$, and

$$\|\vec{r}_{\mathrm{H}}\| = (x^2 + y^2)^{1/2}, \quad \|\vec{v}_{\mathrm{H}}\| = (\dot{x}^2 + \dot{y}^2)^{1/2}. \tag{14}$$

  Like Eq. 12, the constraint in Eq. (13) is a distance-dependent speed limit, where $\nu_0$ is the maximum allowable docking speed and $\nu_1$ is the rate at which deputy must slow down as it approaches the chief. These speeds are based on the property that the maximum speed along an CNMT at the distance $\|\boldsymbol{r}_{\mathrm{H}}\|$ is $\|\boldsymbol{v}_{\mathrm{H}}\| = 2n\|\boldsymbol{r}_{\mathrm{H}}\|$ m/s (which occurs when $y = 0, x = ||\boldsymbol{r}_H||$). Example values may include values $\nu_0 = 0.2$ m/s, and $\nu_1 = 2n \text{ s}^{-1}$.

*Angular Velocity and Acceleration Constraints*

- *Maximum angular acceleration*: Excessive rotational acceleration and may cause damage to the spacecraft structure, payload (which may be a sensitive instrument), or one of many possible appendages and deployables like solar panels, antennas, booms, and tethers. This occurred in April 2016 on the Japanese Hitomi X-ray observatory, when a combination of a

design flaw in reaction wheel rotation direction and bad settings for rocket firings made the spacecraft spin out of control, shedding portions of its solar panels or deployable telescope as a result [31]. In this problem, the reaction wheel array only leads to rotation about the z-axis so that the spacecraft can align thrust with any direction in the x-y plane. Assuming the max acceleration is $\pm 1$ deg/s$^2$, then the constraint becomes:

$$\varphi_{\ddot{\theta}_{max}} = \square\Big(\ddot{\theta}_3 \in [\ddot{\theta}_{3_{min}}, \ddot{\theta}_{3_{max}}]\Big) \tag{15}$$

- *Maximum angular velocity*: Spacecraft rotational velocity limits are in place for the same reason as the translational velocity limits - to enable the spacecraft to react or recover from commands in a reasonable time frame. Spacecraft rotation is typically limited to relatively low velocities. For this problem it is assumed that the rotation rate is limited to 2 degrees per second (0.034907 rad/s) and the constraint is as follows:

$$\varphi_{\dot{\theta}_{max}} = \square\Big(\dot{\theta}_3 \in [\dot{\theta}_{3_{min}}, \dot{\theta}_{3_{max}}]\Big). \tag{16}$$

*Reaction Wheel Actuation Constraints*

- *Maximum reaction wheel velocity*: While the maximum velocity of the reaction wheel faces physical limitations, there may be a lower velocity that designers aim to stay below to improve the longevity by reducing overall wear [32]. Loss of two reaction wheels occurred in the Kepler Space Telescope mission. After the first wheel was lost, the wheel vendor recommended keeping the wheel speeds below 300 revolutions per minute to preserve functionality of the remaining wheels for as long as possible [33]. For this problem, assure that the maximum velocity is limited to $\psi_{max} = 576.0$ rad/s [34], and the constraint is as follows:

$$\varphi_{\psi_{max}} = \square|\psi| \leq \psi_{max} \tag{17}$$

- *Maximum reaction wheel acceleration*: Repeated and extended operation of reaction wheels at the extreme ends of the positive and negative acceleration can also lead to excessive wear and premature failure of reaction wheels [32]. For this problem, assume that the maximum acceleration is limited to $\dot{\psi}_{max} = 181.3$ rad/s$^2$ [34].

$$\varphi_{\dot{\psi}_{max}} = \square|\dot{\psi}| \leq \dot{\psi}_{max} \tag{18}$$

*Angular Orientation Constraints*

- *Spacecraft attitude exclusion zone geometries*. Depending on the spacecraft mission or payload, there may be attitudes where it is unsafe to point a spacecraft. The most common example is the solar exclusion angle; where sensitive instruments must not be pointed too close to the sun to avoid damage. This is the motivation behind exclusion zone guidance methods for spacecraft such as that in [35]. The spacecraft adheres to attitude exclusion zone geometries when the angle between the sensor boresight and the direction of exclusion $\hat{r}_{EZ}$, denoted $\theta_s$ is less than half the sensor field of view $\alpha_{EZ}$ plus a safety buffer angle $\beta_{EZ}$, as depicted in Fig.3. For this problem assume that $\beta = 15^\circ$ [36], and $\alpha = 8^\circ$ [37]. Then the safety requirement becomes:

$$\varphi_{\theta_s} = \square\theta_s > \frac{\alpha_{EZ}}{2} + \beta_{EZ} \tag{19}$$

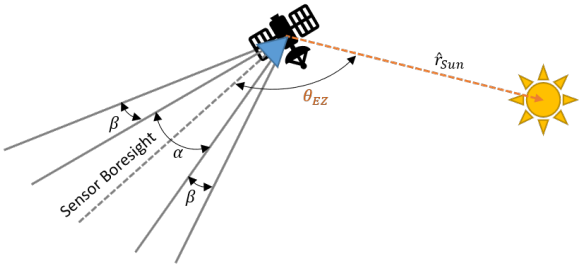


**Figure 3**: Notional depiction of attitude keep out (or exclusion) zone geometry.

- *Line of sight constraints*: During some ARPOD missions, it may be important to maintain the target with the chaser's line of sight [21, 22]. In this case the angle between the sensor boresight and the target, a.k.a fixation angle $\theta_R$, should be less than half the sensor field of view $\alpha_s$. For this problem, assume that $\alpha_s = 25^\circ$.

$$\varphi_{\theta_s} = \square\theta_R < \frac{\alpha_s}{2} \tag{20}$$

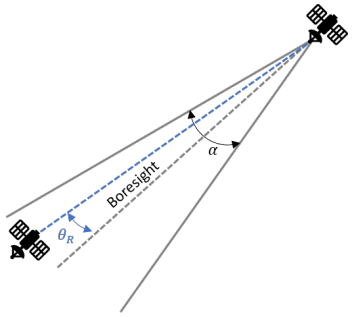


**Figure 4**: Notional depiction of sensor attitude keep in zone geometry.

## METRICS

Metrics are used to evaluate, compare, and contrast various approaches and are generally required to define a formal specification for almost every engineering discipline. Specification also requires a target value with specific units, i.e., track a step input with 10% error. However, there is no standard metric for every application, instead the engineering teams developing the products must adhere to standardized specifications (i.e., IEEE 802 Standards for networking protocols). For the case where no such standard or safety requirements exist (for instance in many Autonomy Platforms) , then there are several high level metrics that can be used to compare the performance of the autonomous maneuvers.

In classical control design, metrics tend to come in only several basic categories: time and response. More specifically, these metrics include performance properties like *stability*, *rise time*,

*settling time*, *tracking errors*, and *robustness*. These metrics may be analyzed through frequency domain analysis which looked at the Bode Plot or Root Locus of the system to ensure that the *poles* or *modes* of the feedback control system are appropriately designed. While classical control systems are still a wildly popular industry standard, control design techniques have continued to advance and apply to potentially more complex systems with nonlinear time-varying dynamics with multiple operational modes. For such complex systems, classical control techniques may no longer apply directly as linearization of complex systems become more difficult or impossible. Consider the case of a closed-loop system with two controllers and a switching signal that dictates which controller is active, for this case a frequency based approach can be used to argue the stability of the intramode dynamics, however, other arguments need to me made to discuss the intermode stability such as invariance of the states with a mode, switching frequencies or guarantees.

For the autonomous docking problem introduced in this paper, there are numerous types of metrics that can be considered. In the remainder of this section, we will outline several metric approaches.

1. *Robustness*:
   The block diagram in Fig.6 shows an interconnection between many components of a closed loop control algorithm, each arrow being the interface between one component to another. Errors may occur at each interface in terms of signal error, or even external disturbances onto the plant, sensors or actuators that may be difficult to model. Also, each block is a mathematical representation of a physical process and, as such, perfect representations are not possible due to manufacturing errors, misalignment and modeling errors.

   There are several ways to discuss robustness through a systems theoretical approach. Robustness can be investigated through an input-to-state stability (ISS)[38] analysis which is typically referred to as the notion that if the nominal system (the system without disturbances or errors) has the origin globally asymptotically stable and if the solutions are bounded by a function of the size of the input for all times. Consider a closed-loop system described by a time-invariant nonlinear differential equation

   $$\dot{x} = f(x, u) \tag{21}$$

   where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is a Lebesgue measurable external input and the map $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous.* A system is called input-to-state stable if there exists functions $\gamma \in \mathcal{K}_\infty$ and $\beta \in \mathcal{KL}$ such that

   $$|x(t)| \leq \beta(|x_0|, t) + \gamma(||u||_\infty), \qquad x_0 \in \mathbb{R}^n$$

   where $x(t)$ is the solution to (21), all admissible input $u$, times $t \geq 0$ and $||\cdot||_\infty$ is the infinity norm. The function $\gamma$ is known as the gain of the system which characterizes the amount of influence the input has on the solutions to the system.

2. *Time-based Metrics*
   In classical controls, time-based metrics determine how the trajectory of the system dampens and comes to a steady state. There are several types of time-based metrics that can be used as

*We define $\mathcal{K}_\infty$ as the set of unbounded, positive, increasing and continuous functions $\gamma$ with $\gamma(0) = 0$ and denote such a function as $\gamma \in \mathcal{K}_\infty$. We also define $\beta \in \mathcal{KL}$ if $\beta(r, s) \in \mathcal{K}_\infty$ for each $s$ and is continuous and strictly decreasing to zero for all $r > 0$.

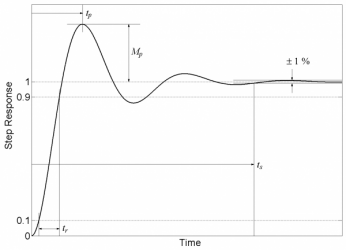


**Figure 5**: Time-based metrics for a unit step response to a system determine how a solution evolves over time.

shown in Figure 5 for a step response input into a system. Such metrics come in the form of settling time, rise time, and peak time. The settling time is the time it takes for the solution to reach $\pm 1\%$ of the steady state value. The peak time is the maximum overshoot time, and the rise time is the time from 10% to 90% of the commanded value. For linear continuous-time systems, we can use the pole placements of the dominant poles to dictate the convergent behavior. For more complex systems, different types of stability notions can identify the rate of convergence and behaviors of the solutions to such systems. For example, consider the following definition.

**Definition 7 (Exponential Stability)** *Given a solution $x$ to* (21)*, there exist positive scalars $\alpha$ and $\kappa$ such that every solution satisfies*

$$|x(t)| \leq \alpha \exp(-\kappa t)|x(0)|$$

*for all $t \geq 0$ and initial conditions $x(0) \in \mathbb{R}^n$.*

Note that from the above definition the rate of convergence is exponential with a rate $\kappa$. There are other stability notions such as *uniform asymptotic stability*, *finite-time stability*, *fixed-time stability*, and more.

3. *Computational Resources*
Many algorithms can be computationally inefficient and require large amounts computational power and memory to perform. This inefficiency can be a large hurdle for autonomous satellite systems since (as of writing this paper) the state of the art processor is the BAE RAD750 which has a Max CPU clock rate of 110 MHz to 200 MHz with 32 KM instruction and data cache. A common misconception is that you can increase algorithm performance by getting a faster computer, while this will decrease the run time of the algorithm, it does not provide a valid way to compare one algorithm with another. This necessitates a need for quantitative and qualitative metrics to compare algorithms implemented on a hardware system. These can be decomposed into two main properties: *computational time* and memory.

   One method to evaluate computational complexity is to determine the number of operations to handle $n$ items. This provides a meaningful comparison for large values of $n$. In the Big-O notation is a way of measuring the order of magnitude of a mathematical expression, $O(n)$ means the order of $n$. For example, an algorithm that operates at $O(n \log n)$ occurs on the order of $n$ times natural log of $n$. Another quantification of complexity/time is that of Floating Point OPerations (FLOPS), which quantify the number of operations performed in

an algorithm. If it is known how much time one operation takes, processor run-time can be extrapolated. For the second measure, the amount of memory that an algorithm requires is a critical. While matrix algebra is thought of as efficient, matrix storage can expend memory at an alarming rate. Thus, the size of static and allocated memory is on par of importance with that of run-time.

## SOLUTION

This section describes the potential solutions to the benchmark in this paper, including guidance on the perceived challenges and advantages to each approach. As presented, the satellite docking problem has both *objectives* and *constraints*. Objectives are to be optimized in some fashion, while constraints must be enforced. Such a structure clearly lends itself well to an optimization framework, although one with nonlinear functions, possible stochastic/disturbance elements, and a time component.

Any solution to the satellite docking problem as posed should be in the form of a planning and/or control algorithm that can be implemented by actuators (in this case reaction wheels and thrusters) onboard the satellite. The planner generates high-level commands, which are passed to the low-level controller to implement, which in turn generates actuation commands. As an example, the planner outputs a series of waypoints to be followed, the low-level controller outputs the required torques and thrust needed to achieve those waypoints, and the actuators convert low-level torque and thrust commands to power commands, gimbal rates, and so on. A complete system block diagram, including the plant (which generates the dynamics of the spacecraft) and sensors (which measure the state of the spacecraft) in a feedback architecture is given in Fig. 6.

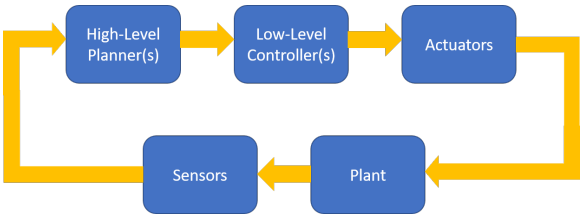


**Figure 6**: Block diagram of a complete satellite system that includes high-level planners and low-level controllers.

The reason to make the distinction between high- and low-level control is that in many cases the two systems are separate, and the low-level control is built into the satellite. For example, a satellite will often be built with a third-party attitude determination and control system (ADCS) that has its own proprietary tracking controllers. These are typically some variation of a proportional-integral-derivative (PID) controller. The solution can therefore either separate the planning and control into disjoint algorithms or combine them, depending on the approach and assumptions made. Reinforcement learning algorithms, for instance, often merge the two.

Most importantly, a solution is sought that both optimizes the objective given and *guarantees* that the constraints are satisfied. Consideration should be given to the following:

1. How well does the solution optimize the objectives?
2. How can the constraints be verified (proven to be satisfied)? Does the solution lend itself well to formal verification?

3. Can the solution be implemented on a satellite? Does it fit within an existing computational architecture used on satellites? What type of advances are required for it to be implemented on a satellite?

**Possible Approaches and Unique Challenges**

Some potential techniques for solving the above problem are described here, along with a summary of known advantages and disadvantages to each. This is not an exhaustive list, and other approaches are possible. The purpose of this section is to provide a starting point for solutions to the benchmark. An eventual comparison of derived solutions using various techniques would be of interest.

*Reinforcement Learning* Reinforcement learning uses some combination of simulation, experience, and expert guidance to iteratively improve upon control policies that achieve a desired outcome. A reward function captures the objectives that the agent (the one doing the learning) wants to optimize. There are numerous algorithms available to then learn a policy that seeks to optimize the reward function using control actions available to the agent. The policy is a function that maps the current environment state (whatever is observable by the agent) to an action. A preliminary use of reinforcement learning in application to satellite rendezvous is discussed in [39].

Using a reinforcement learning approach has the following advantages:

- Handles complex objectives,
- May combine planning and control,
- Accepts a wide variety of sensory inputs without pre-processing,
- Captures complex policies.

Additionally, once the training is set up properly, domain knowledge and hand-tuned policies are not necessary, meaning that the learning algorithm does the heavy lifting. Finally, there are available software tools (e.g., in Python and MATLAB) that implement reinforcement learning algorithms without the need to write one from scratch.

Challenges to implementing a reinforcement learning approach include:

- Enforcing constraints universally,
- Measuring robustness to changing initial conditions and sensor inputs,
- Tuning the reward to achieve desired performance,
- Including continuous or complex action spaces.

*Path Planners* Path planners, or motion planners, create a path for the vehicle to follow. They may include specifications for not only the vehicle's position but also any additional degrees of freedom the vehicle has, so that the path characterizes the desired full state of the vehicle at given points in time. Path planners typically operate by prescribing a set of waypoints for the vehicle to follow, although in some cases the path is defined using a smoothed curve such as a series of splines. A popular class of path planner uses randomized algorithms, such as Rapidly Exploring

Random Trees (RRTs) or Probabilistic Road Maps (PRMs). These algorithms sample points from the configuration space of the vehicle and find an optimal path that does not intersect any obstacles. Other planners use optimization, such as genetic algorithms, to directly find a set of waypoints that optimize an objective while satisfying constraints. Still other approaches use potential fields to drive the vehicle to a goal while avoiding obstacles. The found path is then communicated to the low-level planner, which must track that path. An example of using a path planning algorithm for satellite trajectory optimization is found in [40].

Using a path planning approach has the following advantages:

- Leverages existing algorithms and optimization solvers,
- Avoids designing a low-level controller,
- May develop computationally efficient ways to satisfy constraints.

Challenges to implementing a path planning solution include:

- Avoiding inter-sample constraint violations, due to discrete time commands given to a continuous time system,
- Computing a complex solution onboard a satellite,
- Handling moving obstacles,
- Verification of the planner, in particular randomized planners.

*Model Predictive Control* Spacecraft trajectory planning is often formulated as an *open loop* problem in which the trajectory and high-level controls are computed once at the beginning of the maneuver and this path is followed until maneuver termination (such as the path planners above). In some cases, the dynamics are simplified and assumed to be linear, and the problem is formulated as a linear or mixed-integer linear program, for which efficient solvers exist. To expand upon this concept, model predictive control (MPC) utilizes the same optimization techniques in a receding horizon fashion to simulate a closed loop response. At discrete points in time, the optimization problem is solved and the trajectory adjusted based on the current state of the vehicle, which is likely different than predicted in the purely open loop solution. An example of using MPC for spacecraft rendezvous is given in [41].

Using an MPC approach has the following advantages:

- Exploits efficient optimization solvers to simulate closed loop control,
- Is robust to unanticipated behaviors, as trajectory is repeatedly adjusted,
- Handles changing constraints.

Challenges to implementing an MPC solution include:

- Simplifying the dynamics and constraints to exploit efficient optimization solvers (while preserving optimality),
- Avoiding inter-sample constraint violations,
- Verifying constraint satisfaction.

*Classical Control* Classical control methods include proportional-integral-derivative (PID) controllers, linear quadratic regulators (LQR) and Lyapunov techniques. These are all methods with a long heritage and well-established properties and methods of derivation. PID control is perhaps the most common control technique used to drive a system to its steady state or to track a desired path. It is optimal under only very specific assumptions but tends to work well in practice and can be tuned to achieve specific performance gains (under the assumption that the dynamics are linear). LQR assumes the dynamics are linear and the performance objective is a quadratic function of the state and observations. Additive Gaussian noise can also be included in the model. Under these and other assumptions on controllability and observability, the optimal controller is guaranteed to exist and has a closed form solution. Lyapunov techniques require finding a Lyapunov function over the system dynamics assuming the dynamics include a control input. The control input is then defined as the value that minimizes the Lyapunov function for a given system state. Lyapunov functions can be very difficult to find, however, except in certain cases, such as when the dynamics are linear.

Many spacecraft use some form of classical controller, but rely on operator-provided paths that the classical controller then tracks. For example, [42] uses an LQR controller in combination with an artificial potential field path planner.

Using a classical control approach has the following advantages:

- May be fast to implement with low computational overhead,
- May be easier to verify,
- Uses well-established techniques that have known properties.

Challenges to implementing a classical control solution include:

- Enforcing constraints or optimizing non-standard objectives,
- Satisfying assumptions on the dynamics (e.g., linearity) and still verifying constraints,
- Measuring robustness to changing model assumptions,
- Hand-tuning to achieve desired performance.

*Hybrid Control Systems* Many realistic systems comprise continuous-time dynamics coupled with a discrete-time control implementation or state-based switching logic. These types of systems are called hybrid systems. There are several approaches to hybrid system modeling and control that may be considered. The first is a switching system model with time-sequenced state updates, namely, impulsive updates occur at isolated time instances independent of the state. Hybrid automata theory is a popular approach for modeling and analyszing these systems, which leverages much of the work done for finite state machines. In the case of hybrid dynamics, however, complex state dynamics are fixed to each node in the state machine. More recently, a hybrid inclusion system approach has gained popularity. A hybrid inclusion consists of a differential inclusion that continuously evolves the state on a set called the flow set. Moreover, a difference inclusion dictates the impulsive jumps in the state when the state reaches a set of points called the jumps set. From this hybrid system model, controllers may be derived with, e.g., stability properties using Lyapunov theory specifically for hybrid systems. Controllers may specify switches between modes in combination with other controllers used within specific modes.
Using a hybrid control system approach has the following advantages:

- Models more realistic complex systems,
- Can certify robustness of asymptotic properties if certain regularity conditions are met,
- Provides a rich set of tools and approaches for analyzing asymptotic properties of solutions.

Challenges to implementing a hybrid control system solution include:

- Simulating hybrid systems with only limited tools available,
- Verifying hybrid controllers
- Understanding the complex mathematical architecture,
- Lack of a uniform modeling framework,
- Designing hybrid performance metrics.

*Runtime Assurance* Runtime assurance is distinct from the above approaches in that it combines multiple controllers, and is therefore a form of *hybrid control*. Runtime assurance incorporates three elements: 1) A high-performance or otherwise unverified controller, 2) A simple, trusted/verified controller, and 3) a monitoring scheme that determines when to switch between the two controllers. The high-performance controller can, for example, be derived using reinforcement learning, with a trusted PID controller as backup. The monitoring system observes the vehicle's behavior and when an unsafe configuration is encountered or predicted, the monitor switches to the PID controller. A primitive form of runtime assurance is currently implemented on satellites where the backup controller simply switches the satellite into "safe" mode when something goes wrong. More complex runtime assurance schemes have not yet been implemented to the best of our knowledge. An example of runtime assurance software applied to drones is provided in [43].

Using a runtime assurance approach has the following advantages

- Reduces the burden on verifying the nominal controller,
- Allows for higher performance controllers most of the time.

Challenges to implementing a runtime assurance solution include:

- Implementing an extra architectural/software layer,
- Determining switching conditions that prevent constraint violations before they occur can be challenging,
- Designing a second, verified and trusted simple controller,
- Ensuring that switching does not occur frequently and there is no chatter between the two controllers.

## CONCLUSION

In this paper, an autonomous spacecraft rendezvous and docking problem is for an underactuated planar "chaser" spacecraft modeled using the CWH equations with a thruster in only the positive and negative $x$-body frame directions and a reaction wheel which allows for rotation in the spacecraft. The objective of this problem is for the chaser spacecraft to converge to the origin of the CWH frame with zero angle and angular velocity while ensuring that safety requirements for the spacecraft and sensor are being adhered to. Several metrics for ensuring satisfaction are considered. Lastly, several potential methods for developing algorithms are presented in conjunction with the advantages and disadvantages of each.

## REFERENCES


[1] K. G. Symonds, T. Flohrer, N. Mardle, D. Fornarelli, X. Marc, and T. Ormston, "Operational Reality of Collision Avoidance Manoeuvres," *SpaceOps 2014 Conference*, 2014, p. 1746.

[2] B. Evans, "Dawn of a New Era," *Partnership in Space*, pp. 453–482, Springer, 2014.

[3] D. Holland, "6.4.1 A Case Study of the Near-Catastrophic Mir-Progress 234 Collision with Emphasis on the Human Factors/Systems-Level Issues Surrounding this Mishap," *INCOSE International Symposium*, Vol. 12, Wiley Online Library, 2002, pp. 820–827.

[4] J. Oberg, "Shuttle-Mir's Lessons for the International Space Station," *IEEE Spectrum*, Vol. 35, No. 6, 1998, pp. 28–37.

[5] L. D. Thomas, "Selected Systems Engineering Process Deificiencies and Their Consequences," tech. rep., National Aeronautics and Space Administration George C. Marshall Space Flight Center, 2007.

[6] C. Jewison and R. S. Erwin, "A Spacecraft Benchmark Problem for Hybrid Control and Estimation," *Proceedings of the IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, NV, IEEE, 12 2016, pp. 3300–3305.

[7] B. HomChaudhuri, M. Oishi, M. Shubert, M. Baldwin, and R. S. Erwin, "Computing Reach-avoid Sets for Space Vehicle Docking Under Continuous Thrust," *Decision and Control (CDC), 2016 IEEE 55th Conference on*, IEEE, 2016, pp. 3312–3318.

[8] N. Chan and S. Mitra, "Verifying Safety of an Autonomous Spacecraft Rendezvous Mission," *ARCH17. 4th International Workshop on Applied Verification of Continuous and Hybrid Systems* (G. Frehse and M. Althoff, eds.), Vol. 48 of *EPiC Series in Computing*, EasyChair, 2017, 10.29007/thb4.

[9] H. Dong, Q. Hu, and M. R. Akella, "Safety Control for Spacecraft Autonomous Rendezvous and Docking Under Motion Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 7, 2017, pp. 1680–1692, 10.2514/1.G002322.

[10] G.-Q. Wu, S.-M. Song, and J.-G. Sun, "Finite-time Antisaturation Control for Spacecraft Rendezvous and Docking with Safe Constraint," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 6, 2019, pp. 2170–2184, 10.1177/0954410018774678.

[11] R. Chen, Y. Bai, Y. Zhao, Z. Chen, and T. Sheng, "Safe Proximity Operation to Rotating Non-Cooperative Spacecraft with Complex Shape Using Gaussian Mixture Model-Based Fixed-Time Control," *Applied Sciences*, Vol. 10, 08 2020, p. 5986, 10.3390/app10175986.

[12] L. Sun and W. Huo, "Robust Adaptive Control of Spacecraft Proximity Maneuvers Under Dynamic Coupling and Uncertainty," *Advances in Space Research*, Vol. 56, No. 10, 2015, pp. 2206 – 2217, https://doi.org/10.1016/j.asr.2015.08.029.

[13] X. Li, Z. Zhu, and S. Song, "Non-cooperative Autonomous Rendezvous and Docking using Artificial Potentials and Sliding Mode Control," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 4, 2019, pp. 1171–1184, 10.1177/0954410017748988.

[14] J. Ge, J. Zhao, and J. Yuan, "A Novel Guidance Strategy for Autonomously Approaching a Tumbling Target," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 232, No. 5, 2018, pp. 861–871, 10.1177/0954410017699006.

[15] Q. Li, J. Yuan, B. Zhang, and C. Gao, "Model Predictive Control for Autonomous Rendezvous and Docking with a Tumbling Target," *Aerospace Science and Technology*, Vol. 69, 2017, pp. 700 – 711, https://doi.org/10.1016/j.ast.2017.07.022.

[16] A. Soderlund, S. Phillips, A. Zaman, and C. Petersen, "Autonomous Satellite Rendezvous and Proximity Operations via Geometric Control Methods," *2021 AIAA SciTech Forum*, 2021.

[17] A. Zaman, A. Soderlund, S. Phillips, and C. Petersen, "Autonomous Satellite Rendezvous and Proximity Operations via Model Predictive Control Methods," *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021.

[18] G. W. Hill, "Researches in the Lunar Theory," *American Journal of Mathematics*, Vol. 1, No. 1, 1878, pp. 5–26.

[19] Formal Systems Laboratory of the Department of Computer Science at the University of Illinois at Urbana-Champaign, "Past Time Linear Temporal Logic," `http://fsl.cs.illinois.edu/index.php/Past\_Time\_Linear\_Temporal\_Logic`, 2008. Accessed 15 April 2019.

[20] A. Cimatti, M. Roveri, and D. Sheridan, "Bounded Verification of Past LTL," *International Conference on Formal Methods in Computer-Aided Design*, Springer, 2004, pp. 245–259.

[21] S. Di Cairano, H. Park, and I. Kolmanovsky, "Model Predictive Control Approach for Guidance of Spacecraft Rendezvous and Proximity Maneuvering," *International Journal of Robust and Nonlinear Control*, Vol. 22, No. 12, 2012, pp. 1398–1427.

[22] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 4, 2015, pp. 1638–1647.

[23] K. Lee and C. Park, "Spacecraft Collision Avoidance with Constrained Control via Discrete-Time Generating Functions," *2018 26th Mediterranean Conference on Control and Automation (MED)*, IEEE, 2018, pp. 1–5.

[24] E. Denenberg and P. Gurfil, "Debris Avoidance Maneuvers for Spacecraft in a Cluster," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 6, 2017, pp. 1428–1440.

[25] A. Weiss, C. Petersen, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Safe Positively Invariant Sets for Spacecraft Obstacle Avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2014, pp. 720–732.

[26] A. Weiss, M. Baldwin, C. Petersen, R. S. Erwin, and I. Kolmanovsky, "Spacecraft Constrained Maneuver Planning for Moving Debris Avoidance using Positively Invariant Constraint Admissible Sets," *2013 American Control Conference*, IEEE, 2013, pp. 4802–4807.

[27] J. B. Mueller, P. R. Griesemer, and S. J. Thomas, "Avoidance Maneuver Planning Incorporating Station-keeping Constraints and Automatic Relaxation," *Journal of Aerospace Information Systems*, Vol. 10, No. 6, 2013, pp. 306–322.

[28] S. Alfano, "Collision Avoidance Maneuver Planning Tool," *15th AAS/AIAA Astrodynamics Specialist Conference*, 2005, pp. 7–11.

[29] C. Bombardelli and J. Hernando-Ayuso, "Optimal Impulsive Collision Avoidance in Low Earth Orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 2, 2015, pp. 217–225.

[30] F. Barfield, "Autonomous Collision Avoidance: the Technical Requirements," *National Aerospace and Electronics Conference, 2000. NAECON 2000. Proceedings of the IEEE 2000*, IEEE, 2000, pp. 808–813.

[31] S. Clark, "Attitude Control Failures Led to Break-up of Japanese Astronomy Satellite," 4 2016.

[32] K. H. Gross, M. A. Clark, J. A. Hoffman, E. D. Swenson, and A. W. Fifarek, "Run-time Assurance and Formal Methods Analysis Applied to Nonlinear System Control," *Journal of Aerospace Information Systems*, Vol. 14, No. 4, 2017, pp. 232–246.

[33] K. A. Larson, K. M. McCalmont, C. A. Peterson, and S. E. Ross, "Kepler Mission Operations Response to Wheel Anomalies," *SpaceOps 2014 Conference*, 2014, p. 1882.

[34] K. H. Gross, "Evaluation of Verification Approaches Applied to Nonlinear System Control," Master's thesis, Air Force Institute of Technology, 3 2016.

[35] J. D. Koenig, "Exclusion Zone Guidance Method for Spacecraft," 9 2010. US Patent 7,795,566.

[36] B. Aerospace, "CT-2020 Fact Sheet," tech. rep., Ball Aerospace, 2018.

[37] B. Aerospace, "High Accuracy Star Tracker," tech. rep., Ball Aerospace, 2018.

[38] E. D. Sontag, "Smooth stabilization implies coprime factorization," *IEEE Transactions on Automatic Control*, Vol. 34, No. 4, 1989, pp. 435–443, 10.1109/9.28018.

[39] J. Broida and R. Linares, "Spacecraft Rendezvous Guidance in Cluttered Environments Via Reinforcement Learning," *29th AAS/AIAA Space Flight Mechanics Meeting*, 2019, pp. 1 – 15.

[40] I. Garcia and J. P. How, "Trajectory Optimization for Satellite Reconfiguration Maneuvers with Position and Attitude Constraints," *American Control Conference*, 2005.

[41] A. Weiss, I. Kolmanovsky, M. Baldwin, and R. S. Erwin, "Model Predictive Control of Three Dimensional Spacecraft Relative Motion," *American Control Conference*, 2012, pp. 173 – 178.

[42] R. Bevilacqua, T. Lehmann, and M. Romano, "Development and Experimentation of LQR/APF Guidance and Control for Autonomous Proximity Maneuvers of Multiple Spacecraft," *Acta Astronautica*, Vol. 68, 2011, pp. 1260–1275.

[43] A. Desai, S. Ghosh, S. Seshia, N. Shankar, and A. Tiwari, "SOTER: A Runtime Assurance Framework for Programming Safe Robotics Systems," *49th Annual IEEE/FIP International Conference on Dependable Systems and Networks*, 2019, pp. 138–150.