Learning Reference Governor for Constrained Spacecraft Rendezvous and Proximity Maneuvering

Kosuke Ikeya*, Kaiwen Liu †, Anouck Girard‡ and Ilya Kolmanovsky§
University of Michigan, Ann Arbor, Michigan 48109

Spacecraft automated rendezvous, proximity maneuvering, and docking (ARPOD) plays a significant role in many space missions including on-orbit servicing, and active debris removal. Precise modeling and prediction of spacecraft dynamics can be challenging due to the uncertainties and perturbation forces in the spacecraft operating environment and due to the multi-layered structure of its nominal control system. Despite this complication, spacecraft maneuvers need to satisfy required constraints (thrust limits, line of sight cone constraints, relative velocity of approach constraints, etc.) to ensure safety and achieve ARPOD objectives. This paper considers an application of a Learning-based Reference Governor (LRG) to spacecraft ARPOD operations to enforce constraints without relying on a dynamic model of the spacecraft during the mission. Similar to the conventional Reference Governor (RG), the LRG is an add-on supervisor to a closed-loop control system, serving as a pre-filter on the command generated by the ARPOD planner. The LRG modifies, if it becomes necessary, the reference command to a constraint-admissible value to enforce specified constraints. The LRG is distinguished, however, by the ability to rely on learning instead of an explicit model of the system, and guarantees constraints satisfaction during and after the learning. In this paper, the LRG is applied to the control of combined translational and rotational motion of a chaser spacecraft, and three case studies with different sets of safety constraints and thruster assumptions are used to demonstrate the benefits of the LRG in ARPOD missions.

The material in this paper was partially presented as Paper 2022-2514 at AIAA SCITECH 2022 Forum, San Diego, CA & Virtual, January 3–7, 2022.

*Formerly M.Sc. Student, Department of Aerospace Engineering. Student Member AIAA.
†Ph.D. Student, Department of Aerospace Engineering.
‡Professor, Department of Aerospace Engineering. Associate Fellow, AIAA.
§Professor, Department of Aerospace Engineering. Associate Fellow, AIAA.

Nomenclature

D	=	output maximal response
$\mathcal{D}$	=	collected measurements dataset
d	=	distance to constraint boundaries
F	=	magnitude of thrust force acting on the chaser spacecraft
f, g	=	generic functions
I_{rw}	=	reaction wheel moment of inertia about the spin axis
I_{sp}	=	specific impulse of the chaser spacecraft thruster
I_{zz}	=	chaser spacecraft moment of inertia about $\hat{z}_{\text{B}}$-axis
K	=	linear–quadratic regulator gain
K_{p}	=	proportional gain
K_{d}	=	derivative gain
$k_{\max}$	=	number of adjustments at each command in the self-learning algorithm
L	=	Lipschitz constant
m_{c}	=	chaser spacecraft mass
n	=	mean motion on the target spacecraft's circular orbit.
$n_{\max}$	=	number of commands in the self-learning algorithm
P_{target}	=	diameter of the approach corridor at $\delta y = 0$
r_0	=	circular orbit radius of the target spacecraft.
r	=	reference command
T	=	time duration between each reference adjustment
t	=	time instant
u	=	input vector of the system
v	=	target position vector, input vector to the controller
Y	=	constraint admissible set
Y^C	=	complement of the constraint admissible set
y_c	=	output vector of the system
y_{cv}	=	steady state output of the system
$\hat{\boldsymbol{x}}_{\text{B}}, \hat{\boldsymbol{y}}_{\text{B}}, \hat{\boldsymbol{z}}_{\text{B}}$	=	chaser spacecraft body-fixed frame
$\hat{\boldsymbol{x}}_{\text{H}}, \hat{\boldsymbol{y}}_{\text{H}}, \hat{\boldsymbol{z}}_{\text{H}}$	=	Hill's reference frame
$\delta x, \delta y, \delta z$	=	relative coordinates of the spacecraft in the Hill frame
Γ	=	feedforward gain

ζ = state vector of the system

ζ_v = reference input of the system

θ = chaser spacecraft rotation angle

κ = scalar function for command adjustment

μ = the standard gravitational parameter of the Earth

ξ = line of sight cone angle

ρ = duty cycle of pulse-width modulation

ϕ = output response/trajectory

$\ddot{\psi}$ = angular acceleration of the chaser spacecraft reaction wheel

I. Introduction

SPACE missions are becoming increasingly more complex and autonomous, and they frequently include rendezvous, proximity operations, and docking maneuvers. For instance, in the International Space Station (ISS) program, rendezvous and docking on Low Earth Orbits (LEOs) are essential capabilities for resupply and orbit maintenance. Docking with non-cooperative targets is also a critical part of on-orbit servicing (OOS) and active debris removal (ADR) missions, see e.g., [1–4]. In 2021, Northrop Grumman's MEV-2 satellite successfully docked to another satellite, Intelsat 10-02, to extend its life, thereby demonstrating the potential for OOS [5]. The growing attention to OOS and ADR missions leads to the need for safe and autonomous rendezvous control schemes.

In general, automated rendezvous, proximity operations, and docking (ARPOD) missions result in challenging control problems because of constraints and uncertainties. Many control methods have been proposed to improve safety and robustness at every stage of rendezvous. Lopez and McInnes [6] developed an artificial potential field for obstacle avoidance by a chaser spacecraft. Lee and Pernicka [7] proposed an integrated guidance, navigation, and control (GNC) system for ARPOD missions and demonstrated its robustness against model uncertainties and disturbances satisfying Line of Sight (LoS) cone constraints through a simulation of ARPOD.

Model Predictive Control (MPC) based solutions have also been proposed for ARPOD. In particular, Weiss et al. [8] applied linear quadratic MPC to control the spacecraft during both rendezvous and docking phases while demonstrating capabilities of avoiding debris and exclusion zones. Gavilan et al. [9] successfully developed a chance-constrained MPC to satisfy constraints in a probabilistic sense, and demonstrated this approach for spacecraft rendezvous while handling LoS cone constraints and external disturbances. This approach was further extended by Sanchez et al. [10] to handle rendezvous on a Near Rectilinear Halo Orbit. In [11], tube-based Robust MPC is applied to ARPOD which handles set-bounded disturbances while satisfying mission constraints. Zaman et al. [12] considered applying MPC to ARPOD problems with various safety constraints.

In addition to these studies, adaptive and learning control methods for applications to ARPOD missions have been researched extensively as well. For instance, Wang et al. [13] considered an application of robust and adaptive backstepping control to a rendezvous mission. Dong et al. [14] exploited adaptive control and the method of potential functions for a safe target approach. Artificial potential functions have also been combined with backstepping [15] and sliding mode control [16] to achieve robust and safe proximity operations. Machine learning techniques have also been considered to facilitate ARPOD missions [17–23]. Yang et al. [17] used model-based reinforcement learning and neural networks to address ARPOD mission requirements and computational constraints. Riano-Rios et al. [18] developed a differential atmospheric drag-based control algorithm by designing a Lyapunov-based adaptive controller that compensates for the uncertain ballistic coefficient. Broida and Linares [19] applied Reinforcement Learning (RL), in particular Proximal Policy Optimization (PPO), to spacecraft rendezvous. Federici et al. [20] further investigated Behavioral Cloning as well as PPO, and compared trajectories to those of the MPC solution. The authors of [21] used an ε-constraint approach to meta-reinforcement learning to decrease the maximum thrust constraint violation in a multi-target mission. Hovell and Ulrich [22] combined deep reinforcement learning with conventional control theory. Through simulations and experiments, they demonstrated the viability of such an approach for spacecraft proximity operations with obstacle avoidance. In [23], collision avoidance between a chaser and a target was also achieved by leveraging deep reinforcement learning. Many of these articles showcased successful simulated docking maneuvers with obstacles or collision avoidance. However, handling state constraints while providing explicit constraint satisfaction guarantees has remained as both a theoretical and a practical challenge.

The present paper considers the application of a recently proposed Learning Reference Governor (LRG) [24–27] to performing ARPOD maneuvers. The LRG is an add-on scheme to a nominal control system and is used to enforce pointwise-in-time state and control constraints. Its viability has been previously demonstrated in applications to vehicle rollover avoidance [24–26] and misfire avoidance in spark-ignition engines [27]. In our application to spacecraft control, the LRG monitors and modifies the command generated by a higher-level ARPOD planning algorithm when it becomes necessary to avoid constraint violations. One distinct feature of the LRG, as compared to the conventional Reference Governor (RG) [28], is that its operation is data-informed and relies on learning. In particular, it does not require an explicit dynamic model of the system nor it limits the uncertainty to unknown parameters or disturbances only. The learning can be performed through experimentation with an actual spacecraft before the mission or through simulations on a high-fidelity, black-box model of the spacecraft (e.g., its digital twin). The safety-critical version of the LRG [25] guarantees constraint enforcement both during and after learning, and is adopted in this work. A notable feature of our safety-critical LRG approach is that learning can be terminated at any time and the resulting operation of the system is guaranteed to be safe (of course, conservative if the learning is interrupted too early).

A chaser spacecraft we consider in this paper has one thruster, and the direction of thrust force is controlled by orienting the spacecraft body using a reaction wheel. Hence, the translational and rotational motions of this spacecraft

are coupled. A translational controller provides commands to the rotational controller to track the desired relative position in the orbital plane. We note that even though physics-based models exist for the translational and rotational dynamics of the chaser spacecraft, a typical multi-layered controller architecture, especially with an on-off thrust control system, makes the resulting closed-loop system behavior nonlinear and quite complex. Furthermore, the closed-loop response becomes a function of the controller calibration. This motivates a learning-based, rather than a model-based, reference governor design approach pursued in this paper.

The rest of the paper is organized as follows. Sec. II describes the spacecraft model used for LRG training and simulations. This model represents translational and rotational spacecraft dynamics under a nominal controller, which operates a single thruster and a reaction wheel, to track the target's relative position in the orbital plane. Sec. III introduces the LRG algorithm, and Sec. IV demonstrates the application of the LRG to rendezvous on a LEO orbit subject to thrust magnitude, LoS cone, and relative velocity magnitude constraints. The effectiveness of the LRG is also demonstrated in a case study where thrust forces are approximately realized through pulse width modulation (PWM). Furthermore, the performance of the LRG is compared with the conventional RG. Finally, concluding remarks are made in Sec. V.

II. Problem Formulation

This section first introduces a model for spacecraft relative translational and rotational motion. Then a nominal controller is designed to allow the chaser spacecraft to track the commanded relative position. This nominal controller exploits a hierarchical architecture in which a Linear Quadratic Regulator (LQR) with a feedforward is employed to generate the required thrust force vector for the translational motion while the direction of the desired thrust force informs the desired orientation of the spacecraft tracked by a Proportional-plus-Derivative (PD) controller.

A. Spacecraft Relative Motion Model and Dynamics

The model of the relative motion of the chaser spacecraft with respect to the target spacecraft is based on [29] and [30], and accounts for both the translational and rotational motions of the chaser spacecraft, see Fig. 1. The $\hat{\boldsymbol{x}}_{\mathrm{H}}$ and $\hat{\boldsymbol{y}}_{\mathrm{H}}$ are the radial (R-bar) and in-track (V-bar) unit vectors of the Hill (H) frame (also known as the LVLH frame). The target spacecraft is assumed to be at the origin of the Hill frame, and the target spacecraft orbit is circular. The chaser spacecraft has a single thruster and a single reaction wheel. The spacecraft body-fixed frame is denoted by B and the thrust force is directed along the spacecraft body-fixed frame $\hat{\boldsymbol{y}}_{\mathrm{B}}$-axis, while the spin axis of the reaction wheel is aligned with the $-\hat{\boldsymbol{z}}_{\mathrm{B}}$-axis.

In the above setting, the equations of motion are given by:

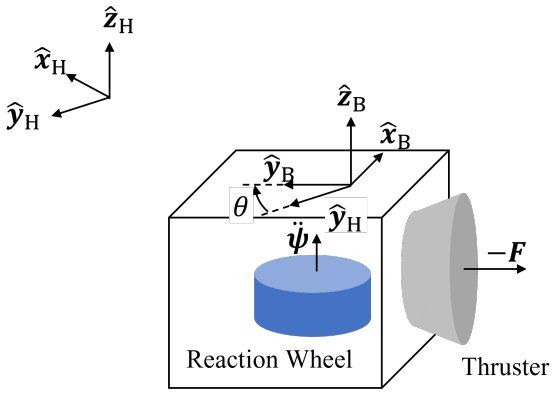


Fig. 1 Spacecraft schematics.

$$
\begin{aligned}
\delta\ddot{x} &= 3n^2\delta x + 2n\delta\dot{y} - \sin\theta\frac{F}{m_c},\\
\delta\ddot{y} &= -2n\delta\dot{x} + \cos\theta\frac{F}{m_c},\\
\ddot{\theta} &= -I_{\mathrm{rw}}\frac{\ddot{\psi}}{I_{zz}},
\end{aligned}
\tag{1}
$$

where $n = \sqrt{\mu/r_0^3}$ is the mean motion of the target spacecraft, δx and δy denote the components of the (relative) position vector of the chaser spacecraft expressed in the Hill frame, and θ represents the orientation of the spacecraft with respect to the Hill frame. The mass and the moment of inertia about $\hat{z}_B$-axis of the chaser spacecraft are denoted by m_c and I_{zz}, respectively, and the moment of inertia of the reaction wheel is denoted by I_{rw}. The magnitude of the thrust force and the angular acceleration of the reaction wheel are expressed as F and $\ddot{\psi}$, respectively. The first two equations represent the translational dynamics of the chaser spacecraft in the orbital plane relative to the target, whereas the third equation represents the rotational motion of the chaser. The terms $F\cos\theta$ and $F\sin\theta$ represent the thrust force components in $\hat{\boldsymbol{x}}_H$ and $\hat{\boldsymbol{y}}_H$ directions, respectively. The orientation of the chaser is modified by a reaction wheel, and defines the direction of the thrust. Note that the operation of a Learning Reference Governor (LRG) does not require a specific model; hence the LRG can also be applied to non-planar maneuvers. We focus on scenarios in which the motion of the chaser is confined to $\hat{\boldsymbol{x}}_H$-$\hat{\boldsymbol{y}}_H$ plane (yet accounting for the rotational dynamics of the chaser) to ease the exposition and follow [29] where such a scenario is proposed as a benchmark.

By introducing the state vector $\zeta = [\delta x, \delta y, \delta \dot{x}, \delta \dot{y}, \theta, \dot{\theta}]^\mathsf{T}$, (1) can be rewritten in the following form:

$$\dot{\zeta}(t) = A\zeta(t) + B(\theta)u(t),\ A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 3n^2 & 0 & 0 & 2n & 0 & 0 \\ 0 & 0 & -2n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},\ B(\theta) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\sin\theta & 0 \\ \cos\theta & 0 \\ 0 & 0 \\ 0 & -I_{\mathrm{rw}} \end{bmatrix}, \tag{2}$$

where $u = [F/m_\mathrm{c}, \ddot{\psi}/I_{zz}]^\mathsf{T}$ is the control vector.

As maneuvers are performed, the spacecraft mass and moment inertia change due to fuel consumption. The time rates of change of the chaser mass and its moment of inertia are modeled as follows:

$$\begin{aligned} \frac{dm_\mathrm{c}}{dt} &= -\frac{F}{I_\mathrm{sp} g_0}, \\ \frac{dI_{zz}}{dt} &= \alpha \frac{dm_\mathrm{c}}{dt}, \end{aligned} \tag{3}$$

where I_sp denotes the specific impulse of the thruster, whereas g_0 denotes the gravitational acceleration on the surface of the Earth. The coefficient α is computed as

$$\alpha = \frac{I_{zz}(0)}{m_\mathrm{c}(0)}, \tag{4}$$

where $I_{zz}(0)$ and $m_\mathrm{c}(0)$ are the initial values of the mass and the moment of inertia about $\hat{\boldsymbol{z}}_\mathrm{B}$-axis of the chaser spacecraft, respectively.

B. Nominal Controller

In this paper, a hierarchical control architecture is employed. The schematic of the nominal controller is shown in Fig. 2. The nominal controller is designed such that in the outer-loop, an LQR feedback law is utilized to generate thrust force and commanded orientation of the chaser to reach the desired position; in the inner-loop, a thrust force controller is used to provide the specified thrust force, and a PD controller is employed to control the reaction wheel to rotate the chaser to the target orientation. Note that in such a hierarchical control design, the inner-loop control (including the thruster and PD controller) should have a faster response than the outer-loop LQR.

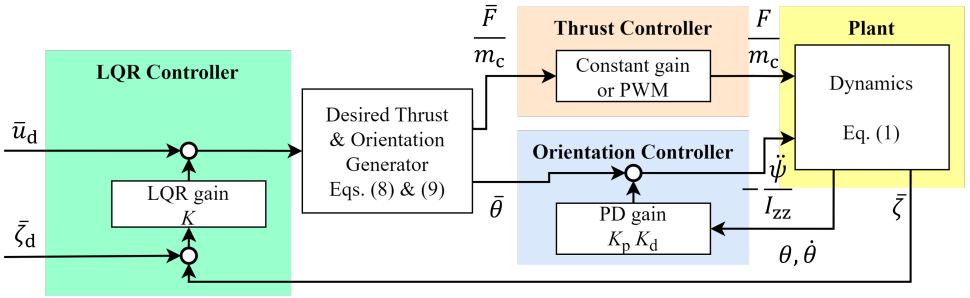


Fig. 2 Diagram of the nominal controller employed for the rendezvous problem.

For the outer-loop design, the LQR controller determines the desired thrust-induced relative acceleration vector as

$$\bar{u} = \begin{bmatrix} \bar{u}_x \\ \bar{u}_y \end{bmatrix} = \bar{u}_\mathrm{d} - K\left(\bar{\zeta} - \bar{\zeta}_\mathrm{d}\right), \quad \bar{\zeta}_\mathrm{d} = \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix}, \tag{5}$$

where $\bar{\zeta} = [\delta x, \delta y, \delta \dot{x}, \delta \dot{y}]^\mathsf{T}$ corresponds to the translational part of the state, $v = [v_x, v_y]^\mathsf{T}$ is the commanded coordinate in the Hill frame for the chaser spacecraft, and $\bar{u}_\mathrm{d}$ is the desired relative acceleration of the chaser in steady-state. The feedback gain K is computed by minimizing the cost function,

$$J = \int_0^\infty \left((\bar{\zeta}(t) - \bar{x}_\mathrm{d})^\mathsf{T} Q (\bar{\zeta}(t) - \bar{\zeta}_\mathrm{d}) + (\bar{u}(t) - \bar{u}_\mathrm{d})^\mathsf{T} R (\bar{u}(t) - \bar{u}_\mathrm{d}) \right) dt, \tag{6}$$

subject to $\dot{\bar{\zeta}} = \bar{A}\bar{\zeta} + \bar{B}\bar{u}$, $0 = \bar{A}\bar{\zeta}_\mathrm{d} + \bar{B}\bar{u}_\mathrm{d}$, where

$$\bar{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{7}$$

The matrices $Q \succeq 0$ and $R \succ 0$ aggregate weights for the state and control variables, respectively. Note that for the purely in-track commands (typical of V-bar approaches), $v_x = 0$ and $\bar{u}_\mathrm{d} = 0$. Then, the desired thrust force magnitude $\bar{F}$ and the desired orientation of the chaser $\bar{\theta}$ are calculated as follows:

$$\bar{F} = m_\mathrm{c} \|\bar{u}\|_2, \tag{8}$$

$$\bar{\theta} = \mathrm{atan2}\left(\bar{u}_y / \|\bar{u}\|_2, \bar{u}_x / \|\bar{u}\|_2\right). \tag{9}$$

The thrust force magnitude is commanded to the thruster controller. As typical thrusters are on-off devices, the thrust force magnitude can be approximately realized using a pulse width modulation (PWM) [31]. The duty cycle ρ of the PWM signal (fraction of time the thruster stays on) is calculated as

$$\rho = \min\left(1, \frac{\bar{F}}{F_{\max}}\right). \tag{10}$$

We note that values of ρ smaller than a threshold cannot be realized by thrusters due to the minimum impulse bit limitation. Similarly, the values of ρ close to 1 but not 1 cannot be realized as they require thrusters to cycle off and on again over a very short period. To exemplify the capability of the LRG to handle different closed-loop dynamics and constraints, in this paper, we first assume that the commanded thrust force magnitude can be accurately realized (Sec. IV.A). Then, the PWM effects are considered in the second case study (Sec. IV.B). In Sec. IV.C, to compare the performance of the LRG and the conventional Reference Governor (RG), the former assumption is made again since the conventional RG solutions that handle PWM implementation of thrust currently do not appear to exist.

For the inner-loop design, a PD controller is used to compute the reaction wheel rotational acceleration command as follows:

$$-\frac{\ddot{\psi}}{I_{zz}} = K_{\mathrm{p}}(\bar{\theta} - \theta) - K_{\mathrm{d}}\dot{\theta}, \tag{11}$$

where K_{p} and K_{d} denote the proportional and derivative gains, respectively.

The closed-loop stability and tracking performance of the overall closed-loop system can be verified using simulations. However, the nominal controller does not handle constraints. During the actual operation, the chaser spacecraft may be required to adhere to multiple constraints, such as thrust limits and staying within the Line of Sight (LoS) cone during the approach. In order to enforce these constraints, an RG [28] that modifies the commands v_x and v_y to constraint-admissible commands can be utilized. However, the design of such a conventional RG is impeded by the hierarchical and nonlinear characteristics of the nominal closed-loop system and can be even harder when PWM effects are considered. To tackle this challenge, we propose the use of the LRG that is able to enforce constraints through learning with minimal knowledge of the system.

III. Learning Reference Governor (LRG) Algorithm

The LRG is applied to the closed-loop spacecraft relative motion dynamics represented by the following equations of motion:

$$\begin{aligned} \dot{\zeta}(t) &= f(\zeta(t), v(t)), \\ y_c(t) &= g(\zeta(t), v(t)), \end{aligned} \tag{12}$$

where $\zeta(t)$ denotes the state of the spacecraft at time t, $v(t)$ is the vector of commanded δx and δy spacecraft coordinates, taking values in a compact and convex set V, and $y_c(t)$ is the output of the system at time t on which the constraints are imposed. Note that the LRG is applied to the closed-loop, pre-stabilized system introduced in Sec. II.B, consisting of the plant being controlled and a nominal controller (Fig. 3).

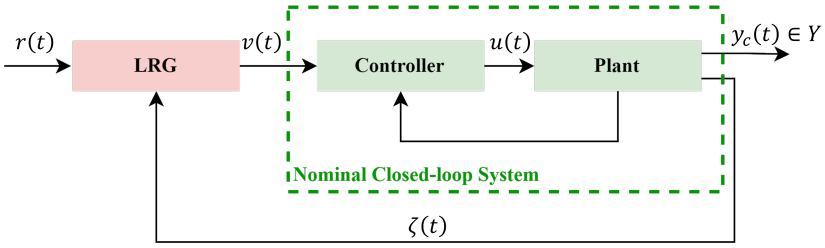


Fig. 3 Diagram of a nominal closed-loop system augmented with a Learning Reference Governor (LRG) to enforce constraints (modified from [32]). The command $r(t)$ is generated either by a higher-level planning algorithm or by a human operator.

The pointwise-in-time constraints are imposed on the output as

$$y_c(t) \in Y, \quad \forall t \in [0, \infty). \tag{13}$$

It is assumed that the closed-loop system (12) for any constant $v \in V$, i.e., the dynamics represented by

$$\dot{\zeta}(t) = f(\zeta(t), v), \tag{14}$$

have a unique state equilibrium, denoted by $\zeta_v = \zeta_v(v)$, which is (globally) asymptotically stable.

Let $\eta(\cdot, \zeta_0, v) : [0, \infty) \to \mathbb{R}^6$ denote the solution to (14) with the initial condition $\zeta(0) = \zeta_0$ and constant reference $v \in V$, and let $\phi(\cdot, \zeta_0, v) = g\left(\eta(\cdot, \zeta_0, v), v\right)$ denote the corresponding trajectory of output y_c. We define the function D as

$$D(v, \Delta v, \Delta\zeta) := \sup_{t \in [0,\infty)} \|\phi(t, \zeta_v(v) + \Delta\zeta, v + \Delta v) - y_{cv}(v)\|, \tag{15}$$

where $y_{cv}(v) = g\left(\zeta_v(v), v\right)$ denotes the steady-state output corresponding to the reference v. See Fig. 4 for a graphical expression of $D(v, \Delta v, \Delta\zeta)$ in a single output system. We assume that the function D is Lipschitz continuous with a Lipschitz constant L, i.e., for any $z_1, z_2 \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^6$,

$$|D(z_1) - D(z_2)| \leq L \|z_1 - z_2\|. \tag{16}$$

It is also assumed that given a trajectory over a time duration of length T, we can obtain an estimate of D, denoted as $\tilde{D}$,

such that $D \le \tilde{D} \le D + \varepsilon$, where $\varepsilon > 0$ is a sufficiently small, known constant.

We further assume that the state $\zeta(t)$, the output $y_c(t)$, and the distance from the steady-state output trajectory associated with the current reference, $y_{cv}(v)$, to the constraint boundary,

$$d(v) := \text{dist}(y_{cv}(v), Y^C) = \inf_{y_c \in Y^C} \left\| y_{cv}(v) - y_c \right\|, \tag{17}$$

are measured, where Y^C is the complement of the constraint admissible set Y. Fig. 4 illustrates the function $d(v)$ for a single output system.

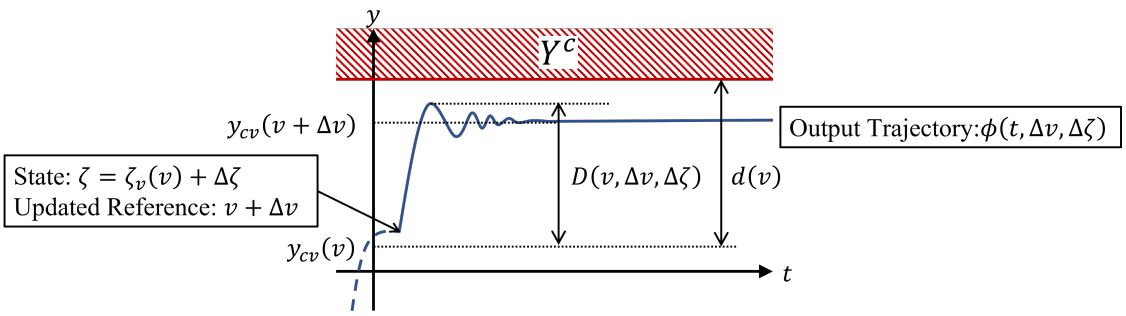


Fig. 4 Graphical explanation of functions $D(v, \Delta v, \Delta\zeta)$ and $d(v)$ for a single output system.

During the operation, the spacecraft is commanded to reach certain desired relative coordinates in the Hill frame. However, if directly using the designed nominal controller in Sec. II.B, constraint violations may occur. Therefore, instead of directly tracking the command, we use the LRG to modify the command to make it constraint-admissible. The LRG updates the reference at sample time instants $\{t_k\}_{k=0}^{\infty} \subset [0, \infty)$ based on the following reference update law,

$$v(t^+) = v(t^-) + \kappa(t)(r(t) - v(t^-)), \tag{18}$$

where $v(t^-)$ and $v(t^+)$ denote the reference input values before and after the update, respectively, $r(t)$ denotes the command (Fig. 3), and $\kappa(t)$ is a scalar computed according to

$$\begin{aligned} &\kappa(t) = \max \kappa \quad \text{subject to } 0 \le \kappa \le 1 \text{ and} \\ &D\left(v(t^-), \kappa(r(t) - v(t^-)), \zeta(t) - \zeta_v(v(t^-))\right) \le d(v(t^-)). \end{aligned} \tag{19}$$

Note that when $\kappa(t) = 1$, we have $v(t^+) = r(t)$, and the command is directly applied. To minimize the difference between $r(t)$ and $v(t)$, it is desirable to pick $\kappa(t)$ as large as possible within $[0, 1]$ range. According to (18) and (19), the reference governor maintains $v(t)$ as constant over each time interval $[t_k, t_{k+1}]$. Meanwhile, if the function D is known, the reference governor algorithm (19) can enforce the constraints by limiting the reference changes at each

sample time instants t_k [25]. However, for the spacecraft system, D is not known a priori. Therefore, we adopt the following Algorithms 1 and 2 from [25] to improve an estimate of D while ensuring safety during the learning process.

Algorithm 1 Safe learning algorithm

1: Initialize the spacecraft with a constraint-admissible steady-state initial condition, $\zeta(0) = \zeta_\nu\big(\nu(0^-)\big)$ for some $\nu(0^-) \in V$ with $y(0^-) \in Y$, and initialize the dataset $\mathcal{D} = \emptyset$;
2: **for** $n = 0 : n_{\max} - 1$ **do**
3: Generate $r_n \in V$, e.g., randomly according to a uniform distribution;
4: **for** $k = 0 : k_{\max} - 1$ **do**
5: At the sample time instant $t = (nk_{\max} + k)T$, compute $\kappa(t)$ with Algorithm 2;
6: Adjust the reference according to (18);
7: At the sample time instant $t' = t + T$, measure $\tilde{D}(t) = \tilde{D}\big(\nu(t^-), \Delta\nu(t), \Delta\zeta(t)\big)$, where $\Delta\nu(t) = \kappa(t)\big(r(t) - \nu(t^-)\big)$ and $\Delta\zeta(t) = \zeta(t) - \zeta_\nu\big(\nu(t^-)\big)$;
8: Add the measured data to the dataset, i.e., $\mathcal{D} = \mathcal{D} \cup \big(\nu(t^-), \Delta\nu(t), \Delta\zeta(t), \tilde{D}(t)\big)$.
9: **end for**
10: **end for**

At the beginning of the learning, we assume that the spacecraft is at a constraint-admissible steady-state and the measurement set is an empty set (Line 1). For each learning epoch, a command which prescribes the desired relative coordinates in the Hill frame is generated (Line 3). Then on Line 5, the LRG adjusts the reference towards the command according to (19), where $D(v, \Delta v, \Delta\zeta)$ is replaced by the current estimate $\bar{D}(v, \Delta v, \Delta\zeta)$ given by

$$\bar{D}(v, \Delta v, \Delta\zeta) = \min\left(L\left\|\begin{bmatrix}\Delta v\\ \Delta\zeta\end{bmatrix}\right\|, \min_{i\in\mathcal{D}}\left(\tilde{D}_i + L\left\|\begin{bmatrix}v\\ \Delta v\\ \Delta\zeta\end{bmatrix} - \begin{bmatrix}v_i\\ \Delta v_i\\ \Delta\zeta_i\end{bmatrix}\right\|\right)\right), \tag{20}$$

and the value of κ is obtained from Algorithm 2. After each reference adjustment, $\tilde{D}$ is measured for an elapsed time T and is added to the measurement set $\mathcal{D}$ (Lines 7-8).

Algorithm 2 Kappa($\zeta, r, v, d, \mathcal{D}$)

1: **for** $(v_i, \Delta v_i, \Delta\zeta_i, \tilde{D}_i) \in \mathcal{D}$ **do**

2: Compute κ_i as the solution to the optimization problem,

$$\max \ \kappa \quad \text{subject to } 0 \le \kappa \le 1 \text{ and} \tag{21}$$
$$\left\|\kappa(r - v) - \Delta v_i\right\| \le \frac{d - \tilde{D}_i}{L} - \left\| \begin{bmatrix} v \\ \zeta - \zeta_v(v) \end{bmatrix} - \begin{bmatrix} v_i \\ \Delta\zeta_i \end{bmatrix} \right\|,$$

if a solution exists, and $\kappa_i = 0$ otherwise;

3: **end for**

4:

$$\kappa' = \underset{[0,1]}{\text{sat}} \ \frac{d - L\left\|\zeta - \zeta_v(v)\right\|}{L\left\|r - v\right\|}, \tag{22}$$

5: **return** $\kappa = \max\left(\max_i \kappa_i, \kappa'\right)$.

Under reasonable assumptions (see [25] and [32]), the LRG algorithm can guarantee the following properties:

Proposition 1: The constraints (13) are guaranteed to be satisfied during the learning process.

Proposition 2: The constraints (13) are enforced during the operating phase when v is adjusted according to (19) and Algorithm 2, and this constraint enforcement guarantee does not depend on the length of the learning.

Proposition 3: If a steady-state constraint-admissible command is constantly applied, the actual reference v updated by (19) converges to the command in finite time.

The reader is referred to [25] and [32] for the rigorous statements, assumptions, and proofs of these propositions.

Remark 1: The LRG algorithms assume that the Lipschitz constant L is known. Typically, there is a range of values of $L \in [L_{\min}, \ +\infty)$ that could be chosen, where $L_{\min}$ is the minimum feasible Lipschitz constant. It is advantageous to select the value of L as close as possible to $L_{\min}$ as unnecessarily large (i.e., conservative) values of L may slow down learning and lead to a more conservative (slower) response. To estimate L, a high-fidelity simulation model could be used where the values of $D(v_i, \Delta v_i, \Delta\zeta_i)$, $i = 1, \cdots, N_D$, are computed over a randomized set of N_D points; then L can be chosen as an upper bound (plus a safety margin) on the difference quotients between all pairs of points in this data set. Such a high-fidelity model can also depend on uncertain parameters the values of which vary within prescribed ranges; then the final value of L can be set to overbound the Lipschitz constants estimated for different (sampled) parameter values.

IV. Case Studies

In this section, simulation case studies of applying the Learning Reference Governor (LRG) described in Sec. III to the spacecraft rendezvous are reported. In these case studies, the chaser spacecraft is commanded to reach a specified in-track position (as in a typical V-bar approach). The nominal controller described in Sec. II.B is utilized to control the chaser spacecraft towards the commanded position modified by the LRG (Fig. 5). In the first case study (Sec. IV.A), two constraints on the magnitude of the thrust and on the position of the chaser spacecraft to remain within the LoS cone of the target spacecraft for successful docking (Fig. 6) are jointly imposed to highlight characteristics of the LRG. In the second case study (Sec. IV.B), we assume that the thrust is realized using an on-off thruster with the PWM method; in this case, the constraint on the thrust is automatically satisfied due to saturation in (10) and hence only the LoS cone angle constraint is considered. Finally, the third case study (Sec. IV.C) considers constraints imposed on relative velocities in both $\hat{\boldsymbol{x}}_{\mathrm{H}}$ and $\hat{\boldsymbol{y}}_{\mathrm{H}}$ directions. Furthermore, the performance of the LRG is compared with that of the conventional Reference Governor (RG) in this case.

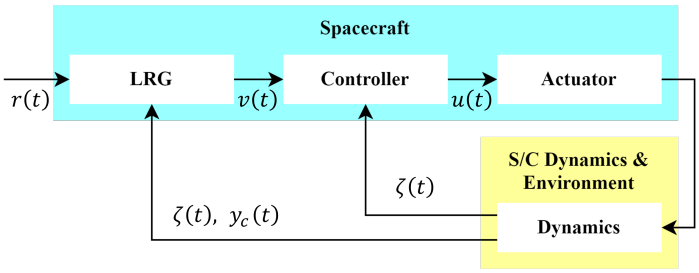


Fig. 5 Diagram of LRG application to spacecraft rendezvous case studies.

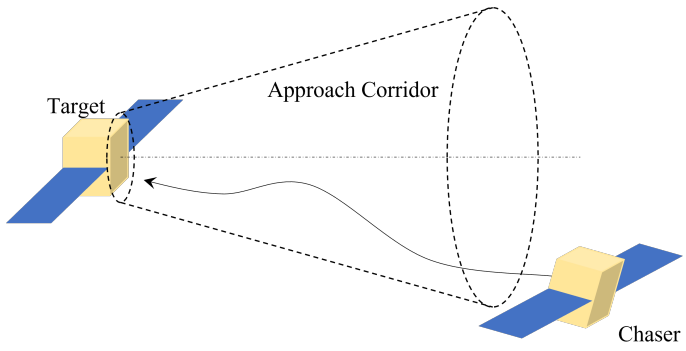


Fig. 6 LoS cone angle constraint defining the approach corridor for the rendezvous.

A. Case Study 1: Constraints on Thrust and Line of Sight Cone Angle

1. Constraints

This first case study demonstrates the effectiveness of the LRG in handling two constraints simultaneously: the maximum thrust limit constraint and the constraint on the position of the chaser spacecraft to remain within the maximum LoS cone of the target spacecraft. In this paper, the former constraint is called "thrust constraint" and the latter constraint is called "LoS constraint." We note that Algorithm 2 handles multiple constraints (i.e., y_c is allowed to be a vector) directly; however, another approach, handling constraints one by one, is chosen in this paper. This approach computes $\kappa_1,\ \kappa_2,\ \cdots,\ \kappa_{n_c}$ for each of n_c constraints based on Algorithm 2, respectively, and employs the minimum value as κ: i.e.,

$$\kappa = \min\left(\kappa_1, \kappa_2, \cdots, \kappa_{n_c}\right). \tag{23}$$

In this implementation, separate $\tilde{D}$ functions are maintained and learned for each of the constraints from the same reference trajectories (as they are generated by the same κ).

Thrust Constraint. The thrust constraint is expressed as $|F(t)| \leq F_{\max} = 1.5$ N, i.e., we seek to ensure that the nominal controller always generates the thrust force within the thruster capability range to avoid potential closed-loop stability loss and performance degradation due to saturation. The constraint is imposed on the output $y_c(t)$ defined as

$$y_c(t) = \frac{F(t)}{m_{\mathrm{c}}(t)}. \tag{24}$$

Note that the mass of the spacecraft, $m_{\mathrm{c}}(t)$ changes with time as fuel is consumed due to thrust. As the controller prescribes the acceleration, the change in mass translates into changes in the constraints, Y and Y^C, with time. Also, note that as we consider only in-track translational commands (corresponding to unforced relative motion equilibria), $v_x = 0$ and

$$\forall v,\ y_{cv}(v) = 0. \tag{25}$$

We then modify (21) as

$$\left\|\kappa(r - v) - \Delta v_i\right\| \leq \frac{F_{\max} - m_i \tilde{D}_i}{m_i L_a} - \left\|\Lambda \begin{bmatrix} v \\ \zeta - \zeta_v(v) \end{bmatrix} - \begin{bmatrix} v_i \\ \Delta\zeta_i \end{bmatrix}\right\|, \tag{26}$$

where L_a denotes the Lipschitz constant corresponding to F/m_c, and

$$m_i = m_c(t_i). \tag{27}$$

Here and subsequently, $\|\cdot\|$ stands for the 1-norm. Each component in $\|\cdot\|$ on the right-hand side of (26) is normalized, with the normalization factors aggregated into a diagonal matrix Λ as this promotes faster learning.

To obtain the value of L_a, 80 points corresponding to $(v, \Delta v, \Delta\zeta)$ values were chosen at random, and the corresponding values of D were computed through simulation experiments. From these, a bound on the difference quotients was determined. As a result, the Lipschitz constant for this constraint, L_a needs to satisfy $L_a \geq 8.55 \times 10^{-5}$. We chose a more conservative value of $L_a = 1.40 \times 10^{-4}$ for our implementation to demonstrate that a conservative estimate of the Lipschitz constant is sufficient for the algorithm albeit it may reduce the learning rate.

Line of Sight Constraint. In this case study, in addition to the thrust constraint discussed above, we also impose a constraint on the LoS cone angle ξ as $|\xi(t)| \leq \xi_{\max} = 2.5^\circ = 0.0436$ rad and consider the corresponding output of the system $y_c(t)$ defined as $y_c(t) = \xi$. Fig. 7 illustrates the definition of the LoS cone angle ξ. The diameter of the approach corridor at $\delta y = 0$, P_{target} is defined as 10 m, and thus, δy_{offset}, which determines the position of the LoS cone vertex to the docking side of the target spacecraft in orbital track, can be derived as

$$\delta y_{\text{offset}} = \frac{P_{\text{target}}}{2\tan\xi_{\max}} = 114.5\text{m}. \tag{28}$$

The output, $y_c(t)$, which is the LoS cone angle ξ, is expressed as

$$y_c(t) = \xi(t) = \arctan\frac{\delta x(t)}{\delta y(t) - \delta y_{\text{offset}}}, \tag{29}$$

where $\delta x(t)$ and $\delta y(t)$ denote, respectively, the $\hat{\boldsymbol{x}}_{\text{H}}$ and $\hat{\boldsymbol{y}}_{\text{H}}$ components of the position vector of the chaser relative to the target spacecraft.

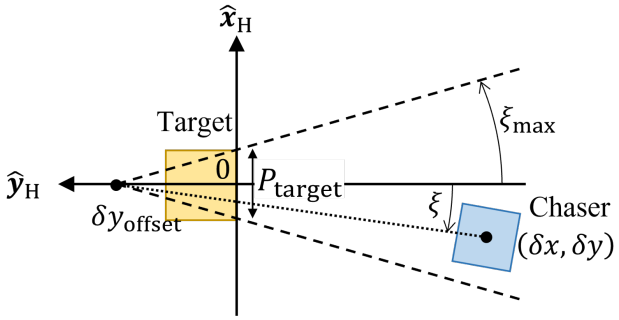


Fig. 7 Definition of LoS cone.

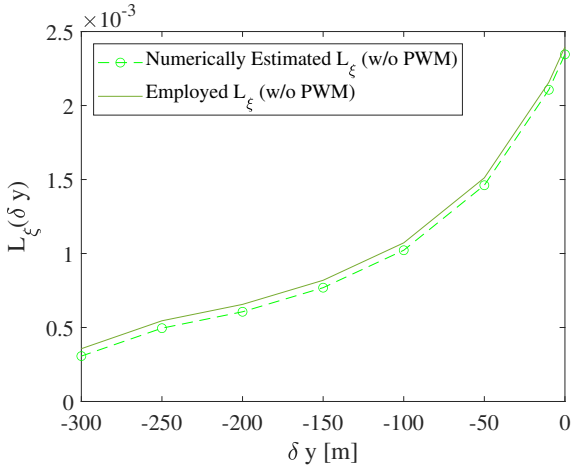


Fig. 8 Comparison of numerically estimated Lipschitz constant and employed Lipschitz constant $L_\xi(\delta y)$ in Case Study 1 for LoS constraint.

Note that the form of (29) implies that a small $|\delta y - \delta y_{\text{offset}}|$ (when the spacecraft approaches the target) makes ξ more sensitive to the change in δx. As a result, the Lipschitz constant L, defined in (16), needs to be sufficiently large to ensure that (16) is satisfied where $|\delta y - \delta y_{\text{offset}}|$ is small. However, a large L will slow down the learning process [32]. For the sake of faster learning, in this and the following case study, we make the Lipschitz constant corresponding to ξ, L_ξ, a function of δy, $L_\xi(\delta y)$. To accommodate the dependence of L_ξ on δy, the only modification we need is in Algorithm 2, with the estimate $\bar{D}(v, \Delta v, \Delta\zeta)$ in (20) replaced by

$$\bar{D}(v, \Delta v, \Delta\zeta) = \min\left(L_\xi(v, \Delta\zeta) \left\| \begin{bmatrix} \Delta v \\ \Delta\zeta \end{bmatrix} \right\|, \min_{i \in \mathcal{D}} \left(\tilde{D}_i + L_\xi(v, \Delta\zeta) \left\| \begin{bmatrix} v \\ \Delta v \\ \Delta\zeta \end{bmatrix} - \begin{bmatrix} v_i \\ \Delta v_i \\ \Delta\zeta_i \end{bmatrix} \right\| \right) \right), \tag{30}$$

where, with a slight abuse of notation, $L_\xi(v, \Delta\zeta) = L_\xi(\delta y)$ since δy is a component of ζ and $\zeta = \zeta_v(v) + \Delta\zeta$.

Fig. 8 compares the employed $L_\xi(\delta y)$ and numerically estimated Lipschitz constant corresponding to $\delta y \in S = [0, -10, -50, -100, -150, -200, -250, -300]$ m. The values of $L_\xi(\delta y)$ for $\delta y \notin S$ are computed by linear interpolation/extrapolation. The numerical estimates of $L_\xi(\delta y)$ were obtained consistently with Remark 1 and as follows: For each $\delta y' \in S$, 500 points corresponding to different $(v, \Delta v, \Delta\zeta)$ values and satisfying $\{(v, \Delta v, \Delta\zeta) \mid v + \Delta\zeta_2 = \delta y'\}$ were chosen, where v is the reference (i.e., the commanded δy) and $\Delta\zeta_2$ is the second element of $\Delta\zeta$. Then the difference quotients for D were numerically calculated, and $L_\xi(\delta y')$ was set to their maximum plus an offset (0.00005 in this case study), see Fig. 8.

Sec. IV.A.3 reports the simulation results with the proposed modification of representing L_ξ. While in a practical sense we have found that this modification improves the ability to deal with the LoS constraint, the development of the

supporting theoretical guarantees for this approach is left to future work.

2. Parameters

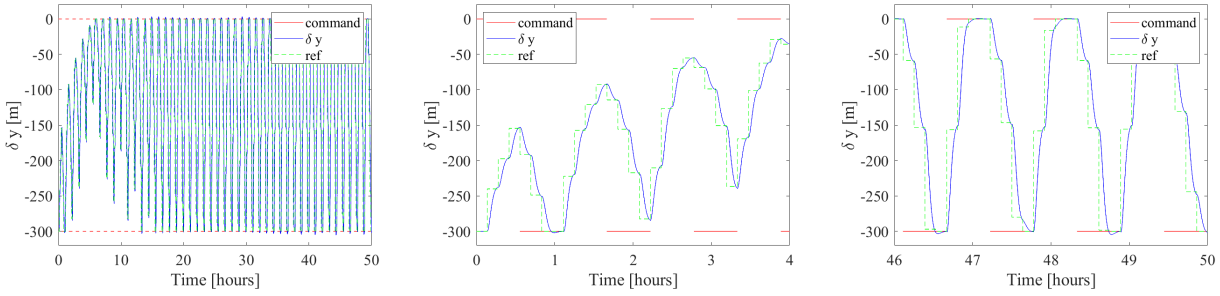
Table 1 lists the parameters used in the simulation. In Case Studies 1 and 2, the initial state of the chaser is $\zeta_0 = [0, -300\text{ m}, 0, 0, 0, 0]^\mathsf{T}$.

Table 1 Parameters Used in Cases Studies 1 and 2.

Parameter		Assumed value
Dynamics Parameter		
Semi-major axis	r_0	6,778 km
Specific impulse	I_{sp}	300 s
Chaser spacecraft initial mass	$m_{\text{c}}(0)$	100 kg
Chaser spacecraft initial moment of inertia about $\hat{z}_{\text{B}}$-axis	$I_{\text{zz}}(0)$	40 kg $\cdot$ m^2
Reaction wheel spin axis moment of inertia	I_{rw}	0.04 kg $\cdot$ m^2
LRG Parameter		
Number of commands	$n_{\max}$	0 - 90
Number of reference adjustments at each command	$k_{\max}$	4
Time duration between each reference adjustment	T	500 s
Lipschitz constant for thrust constraint	L_a	0.00025
Lipschitz constant for LoS constraint	$L_\xi(\delta y)$	See Figs. 8 and 15
Constraint		
Maximum thrust	$F_{\max}$	1.5 N
Maximum LoS cone angle	$\xi_{\max}$	2.5 degree = 0.0436 rad
Diameter of the approach corridor at $\delta y = 0$	P_{target}	10 m

3. Results

Safe Learning Results. Figs. 9 and 10 illustrate that the LRG adjusts the reference and is able to successfully enforce constraints on both thrust and LoS cone angle. Fig. 9 illustrates how the LRG modifies the reference commands. In particular, a comparison of Figs. 9b and 9c reveals that the modification becomes smaller as learning progresses.



(a) Command profile during learning. **(b) At the beginning of learning.** **(c) At the end of learning.**

Fig. 9 Time histories of δy in Case Study 1 with thrust and LoS constraints.

Figs. 10a and 10b show that the chaser violates neither of the constraints during the learning process. Fig. 10a shows the time history of the thrust force F during learning corresponding to the commands in Fig. 9. The LRG operates the spacecraft conservatively at first due to the initially conservative bounds on the response as informed by the initial assumptions. As learning progresses, the LRG is able to operate the spacecraft more and more aggressively without violating constraints. Fig. 10b also shows that the LRG operates the spacecraft conservatively at the beginning of the learning and becomes more aggressive without violating the LoS constraint as learning progresses. Indeed, as can be seen from Fig. 10c, the trajectory of the chaser stays within the approach corridor as the LRG enforces the LoS constraint. Fig. 10c also indicates that the LoS constraint is dominant when the chaser is close to the target, while the thrust constraint limits the operation of the chaser when the chaser is far from the target. In particular, there are regions visible in Fig. 10c (top right and bottom right) that the chaser could not reach and explore due to the thrust limit.

The progress of the learning can also be visualized by plotting the tracking error, which is the average of $\|r(t) - v(t)\|$ over a moving time window of the most recent 8000 seconds (Fig. 10d). At the beginning of the learning, the tracking error is large, and then it gradually decreases as the learning proceeds. Learning is accomplished in about 35 hours as shown in Fig. 10d.

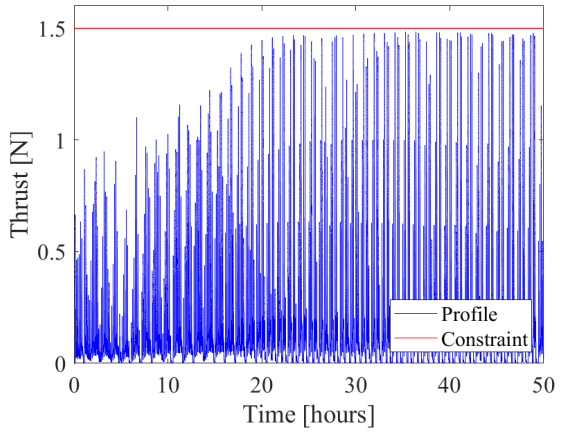
Performance of LRG after Learning. After the learning is terminated, commands are applied to the chaser spacecraft, and the responses of the chaser spacecraft without the LRG and with the LRG after different learning epochs are shown in Fig. 11. As shown in Fig. 11a, in all cases the chaser is able to follow the command. Note that $n = 0, 18, 36, 90$ correspond to $0, 10, 20, 50$ hours of learning, respectively. Fig. 11a also indicates that the chaser arrives at the target position faster after longer learning. Without the LRG, the command is tracked rapidly, but this leads to a significant constraint violation as seen in Figs. 11b and 11c. With the LRG, the spacecraft is able to arrive at the target position without violating either of the constraints. Fig. 11d visualizes the different trajectories of the chaser corresponding to each learning epoch.

4. Discussion

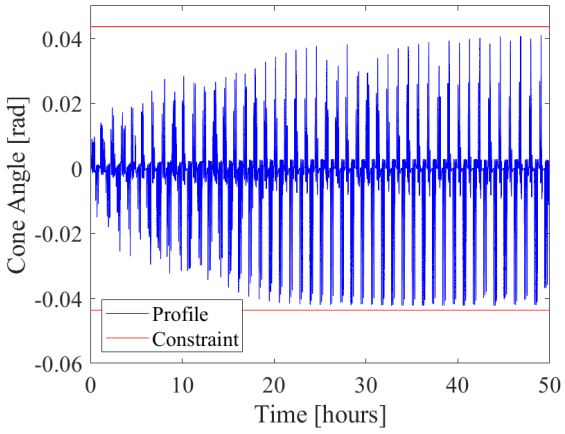
As shown in Fig. 11, the chaser behavior between learning epochs of $n = 36$ and $n = 90$ is very similar. Thus, 20 hours of learning is practically enough to achieve fast maneuvers while enforcing the thrust and LoS constraints during rendezvous.

It is also important to note that while the LRG successfully protected the chaser from violating constraints, the learning process consumes a certain amount of time and propellant. Fig. 12 illustrates the change of mass during the learning for this case study. Thus fast learning is desirable for the sake of saving propellant. The proposed modification of L_ξ wherein it is made a function of δy speeds up the learning and is beneficial in terms of saving the propellant.

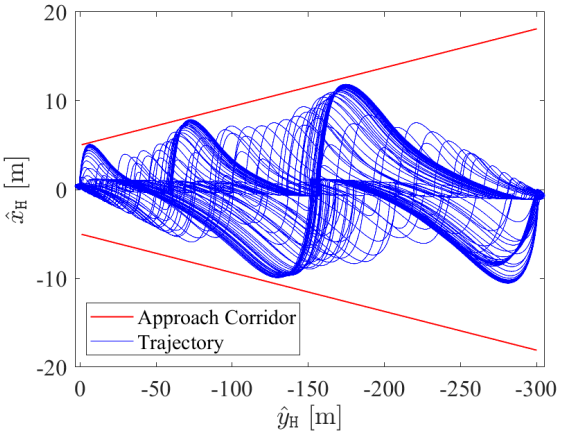
Finally, Fig. 13 compares the time history of the fraction of the chaser mass to the initial mass, $m_c(t)/m_c(0)$,



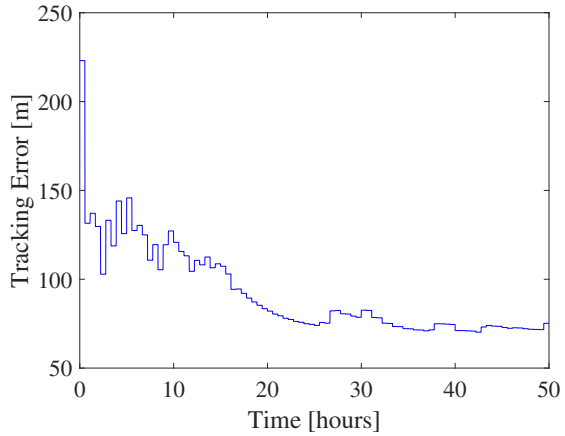
(a) Thrust profile during learning.



(b) LoS cone angle profile during learning.



(c) Chaser trajectories during learning.



(d) Tracking error profile during learning.

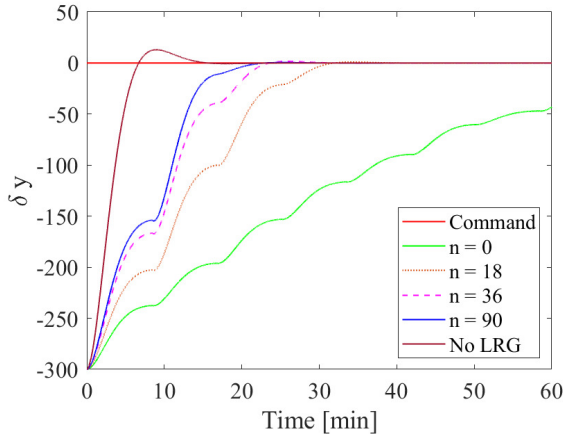
Fig. 10 Learning process in Case Study 1 with thrust and LoS constraints.

corresponding to the maneuvers shown in Fig. 11. Larger $m_c(t)/m_c(0)$ implies smaller propellant consumption. The chaser prior to learning ($n = 0$) consumes the least amount of propellant, and this is due to the fact that less thrust is generated to achieve the smaller reference adjustments Δv. When Δv becomes larger, as the controller becomes more aggressive, the chaser generates a larger thrust to track the reference command and consumes more propellant.

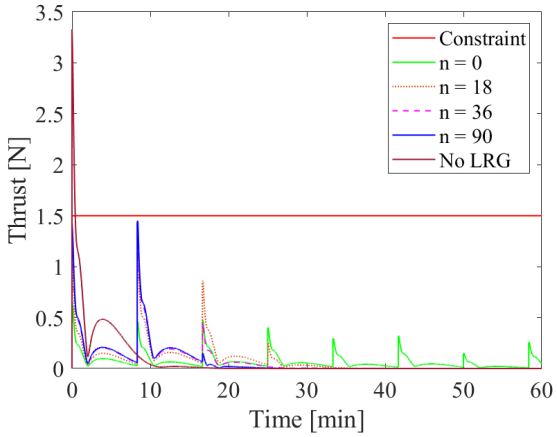
B. Case Study 2: Constraint on Line of Sight Cone Angle with Pulse-Width Modulation Thrust Controller

1. Overview and Pulse-Width Modulation Thrust Controller

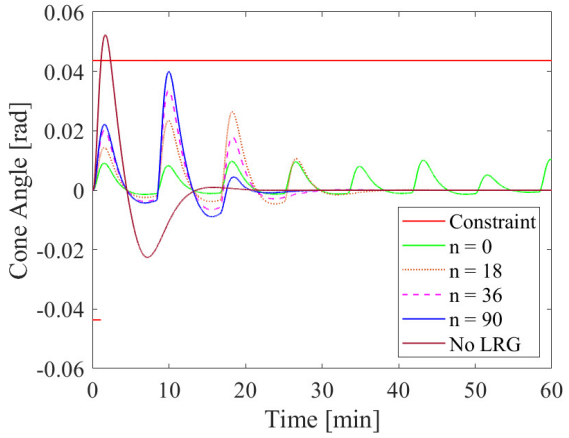
In this case study, the chaser spacecraft is assumed to have an on-off thruster, and the PWM method discussed in Sec. II.B is used to realize the commanded thrust force. With this PWM approach, the thrust magnitude constraint is automatically satisfied (i.e., we no longer treat the thrust limit as a constraint but consider thrust saturation as a nonlinearity). Consequently, we focus only on handling the LoS constraint. The parameters used in this case study are the same as Case Study 1 except for $L_\xi(\delta y)$.



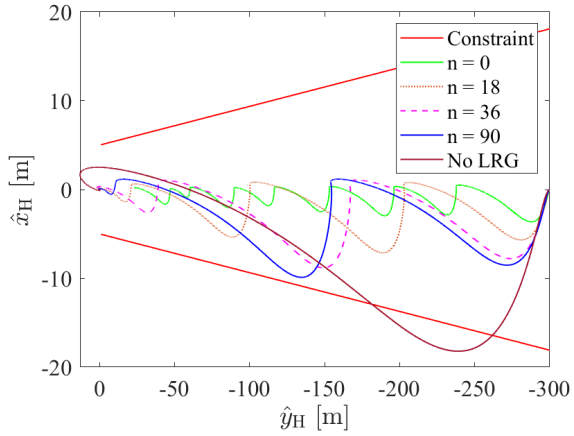
(a) Time history of δy in response to a constant command corresponding to rendezvous at the origin.



(b) Thrust profile during mission.



(c) LoS cone angle profile during mission.



(d) Chaser trajectories.

Fig. 11 Comparison of rendezvous trajectories with the LRG after different learning epochs and without the LRG in Case Study 1.

The minimum period the thruster can stay on or off is assumed to be 0.1 sec. Therefore, values of ρ computed from (10) close enough to 0 are set to 0 and values close enough to 1 are set to 1. Fig. 14 compares trajectories and thrust profiles of the controller with and without the PWM effect corresponding to a step command in δy from -300 m to 0 m.

Following the same procedure in Sec. IV.A.1, $L_\xi(\delta y)$ was informed from simulation results and an offset of 0.0002 has been added. Fig. 15 shows the result of the numerical estimation and the actual L_ξ employed in this case.

2. Results

Similar to Case Study 1 described in Sec. IV.A.3, the LRG adjusts the reference, and the magnitude of the reference adjustment becomes larger as learning progresses as shown in Fig. 16, which demonstrates that the LRG operates the spacecraft more aggressively. Fig. 17a shows that the LRG operates the spacecraft conservatively at the beginning of the learning and becomes more aggressive without violating LoS constraint as learning progresses. As we can observe from

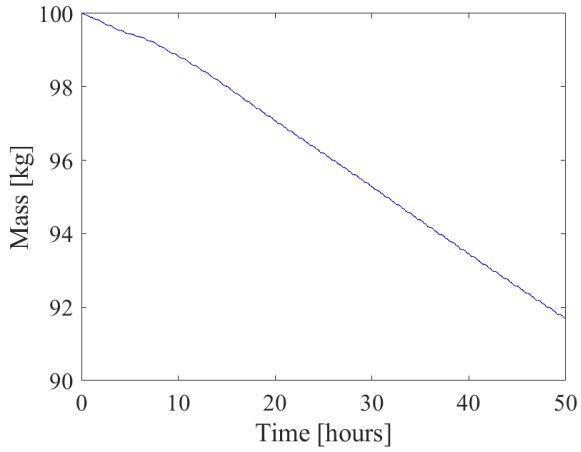


Fig. 12 Mass change during learning in Case Study 1.

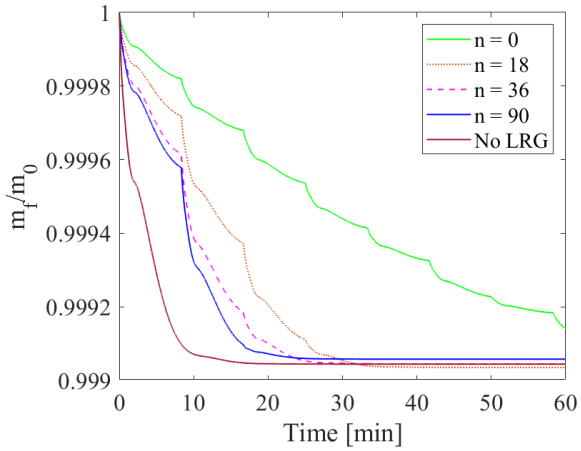


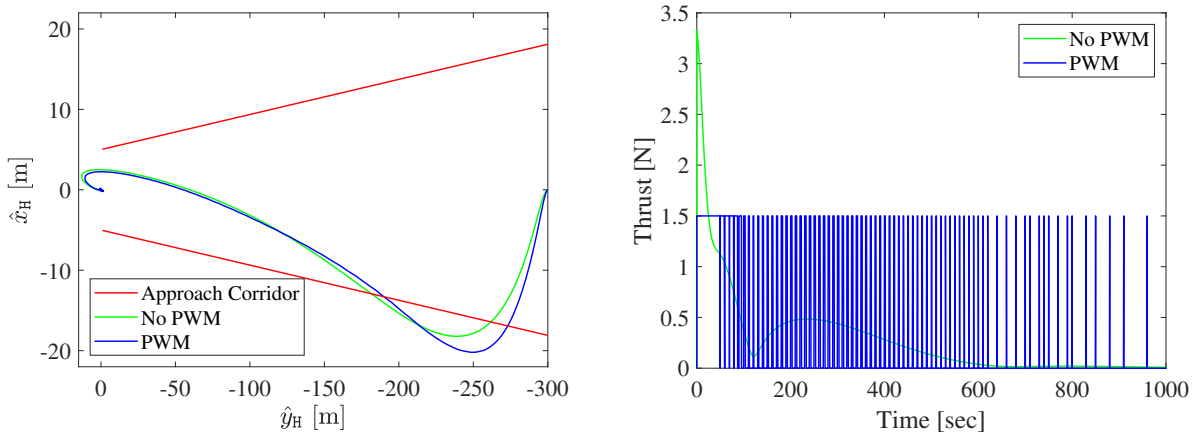
Fig. 13 Time history of the fraction of mass to initial mass during maneuvers after learning.

Fig. 17b, the trajectory of the chaser stays within the approach corridor as the LRG enforces the LoS constraint. Fig.17c indicates that learning takes longer than in Case Study 1.

Fig. 18 compares the responses of the chaser spacecraft without the LRG and with the LRG after different learning epochs. As shown in Fig. 18a, the response is slower with shorter learning. Without the LRG, the LoS constraint is violated as seen from Figs. 18b and 18c. With the LRG, the chaser spacecraft stays in the approach corridor at all times.

3. Discussion

In contrast to Case Study 1 (Fig. 11), Fig. 18 illustrates that 20 hours of learning (n = 36) may not be enough for this case with the PWM-based thrust realization. This longer learning requirement is caused by larger values of the Lipschitz constant L associated with the closed-loop system under PWM effects (see Fig. 19).



(a) Trajectories. **(b) Thrust profile (first 1000 seconds).**

Fig. 14 Comparison of trajectories and thrust profiles of a chaser spacecraft with and without a PWM thrust controller.

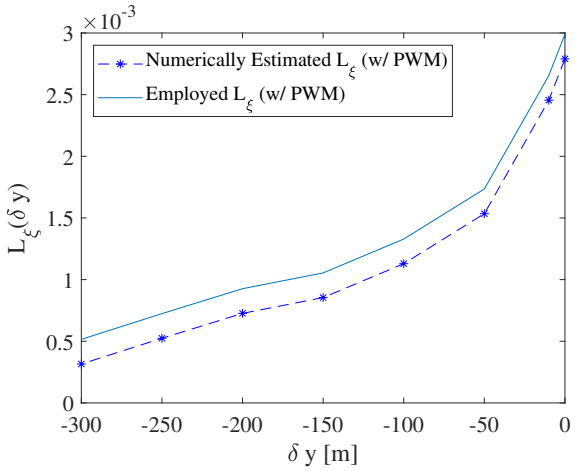
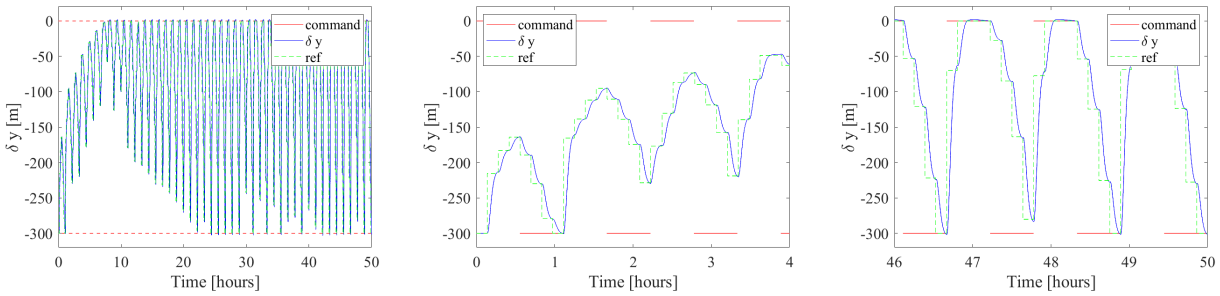
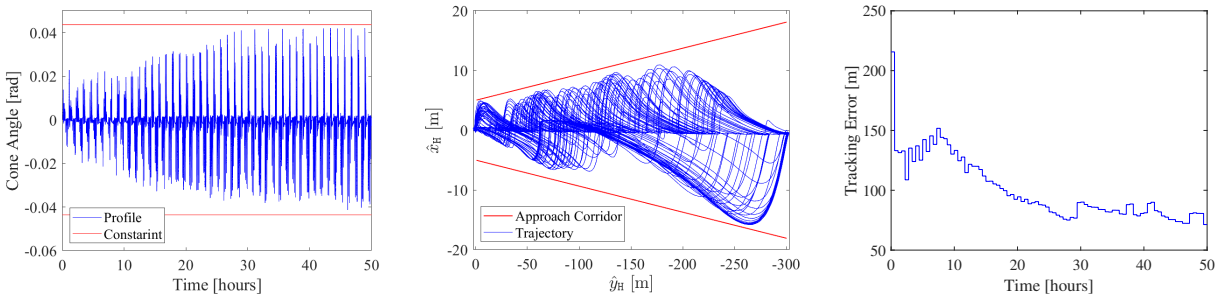


Fig. 15 Comparison of numerically estimated Lipschitz constant and employed Lipschitz constant $L_\xi(\delta y)$ in Case Study 2 with LoS constraint and PWM thrust realization.



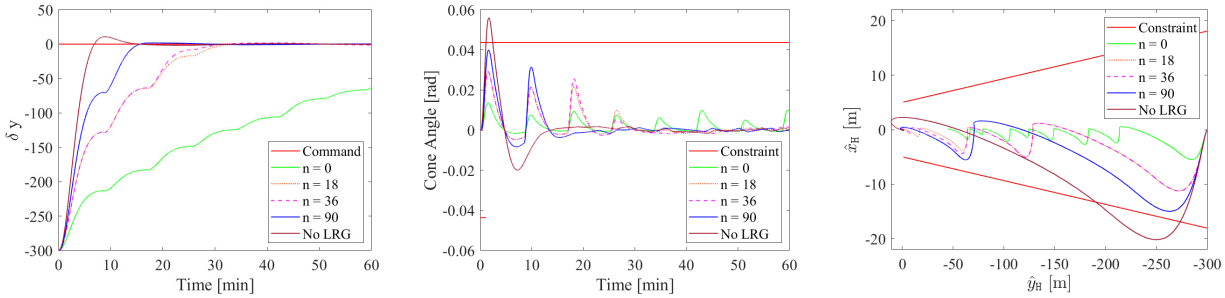
(a) Command profile during learning. **(b) At the beginning of learning.** **(c) At the end of learning.**

Fig. 16 Time histories of δy in Case Study 2 with LoS constraint and PWM thrust realization.



(a) LoS cone angle profile. **(b) Chaser trajectories during learning.** **(c) Tracking error profile during learning.**

Fig. 17 Learning process in Case Study 2 with LoS constraint and PWM thrust realization.



(a) Response to a step δy command. **(b) LoS cone angle response.** **(c) Chaser trajectories.**

Fig. 18 Comparison of rendezvous trajectories with the LRG after different learning epochs and without the LRG in Case Study 2.

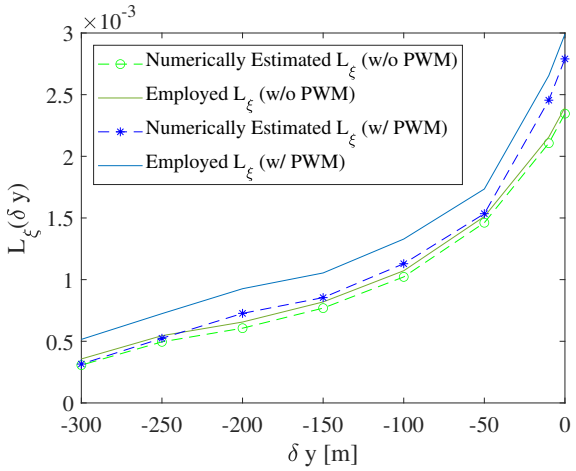


Fig. 19 Comparison of Lipschitz constants for LoS constraint in Case Studies 1 and 2.

C. Case Study 3: Constraints on Relative Velocities in $\hat{x}_H$ and $\hat{y}_H$ Directions

1. Constraints and Parameters

Finally, this case study involves constraints on relative velocities in both $\hat{x}_H$ and $\hat{y}_H$ directions. These constraints can be expressed as $|\delta\dot{x}| \leq \delta\dot{x}_{\max} = 0.006$ m/s, and $|\delta\dot{y}| \leq \delta\dot{y}_{\max} = 0.015$ m/s. The Lipschitz constants corresponding to

these constraints are calculated following the same procedure as for L_a in Sec. IV.A.1.

Table 2 lists the parameters used for the simulation. In this case study, the initial state of the chaser is $\zeta_0 = [0, -100\text{ m}, 0, 0, 0, 0]^\mathsf{T}$. Note that all dynamics parameters listed in Table 1 are also used in this case study, and hence, they are omitted from Table 2.

Table 2 Parameters Used in Case 3.

Parameter		Assumed value
LRG Parameter		
Number of commands	$n_{\max}$	0 - 150
Number of reference adjustments at each command	$k_{\max}$	4
Time duration between each reference adjustment	T	300 s
Lipschitz constant for relative velocity constraint	$L_{\delta\dot{x}}$	0.0038
	$L_{\delta\dot{y}}$	0.0101
Constraint		
Maximum relative velocity	$\delta\dot{x}_{\max}$	0.06 m/s
	$\delta\dot{y}_{\max}$	0.15 m/s

2. Conventional Reference Governor

We also implemented the conventional Reference Governor (RG) to compare its performance with that of the LRG in this relative-velocity-constrained case study. We consider the same closed-loop system in Fig. 2, i.e., the closed-loop system is nonlinear. Note that the implementation of coupled dynamics of translational and rotational motions significantly complicates the development of a conventional RG solution. Hence, a simplifying assumption is made that the desired thrust and orientation can be exactly realized during the implementation of the conventional RG. This assumption is reasonable because the thrust and orientation controller responds faster than the outer-loop LQR controller as discussed in Sec. II.B. Under this assumption, the closed-loop system in Fig. 2 is reduced to a linear system.

Following [28], we apply the conventional RG to this linear system, which relies on computing a finitely determined approximation $\tilde{O}_\infty$ to the maximal constraint admissible set [33] defined as

$$O_\infty := \left\{(v, \zeta) : \phi(t, \zeta, v) \in Y, \forall t \in [0, \infty)\right\} \tag{31}$$

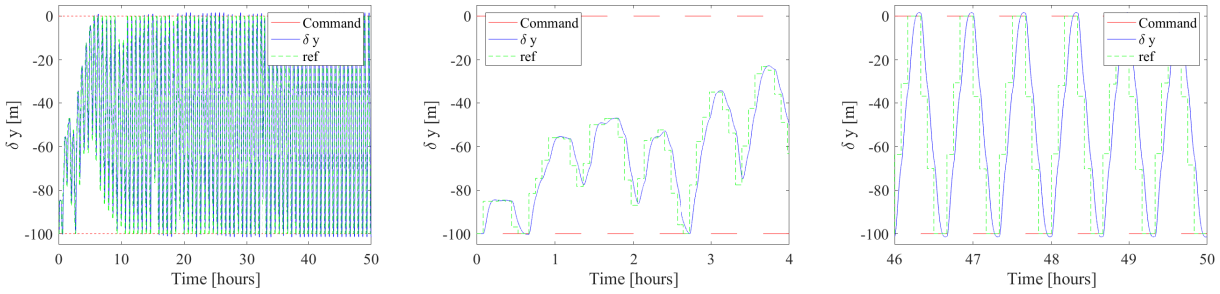
Based on the computed $\tilde{O}_\infty$, the operation of the conventional RG is similar to (18) and (19), wherein the reference

command is updated according to (18) at every time step with $\kappa(t)$ computed by

$$\kappa(t) = \max \kappa \quad \text{subject to } 0 \leq \kappa \leq 1 \text{ and} \tag{32}$$
$$\Big(\zeta(t), v(t^-) + \kappa\big(r(t) - v(t^-)\big)\Big) \in \tilde{O}_\infty.$$

3. Results

Safe Learning and Performance after Learning. Figs. 20 and 21 indicate that the LRG adjusts the reference and is able to successfully enforce constraints on relative velocities in both directions. Similar to the other two cases, the LRG operates the chaser spacecraft conservatively at first, and the operation becomes more aggressive without violating the constraints as learning proceeds. Learning is accomplished in about 40 hours as shown in Fig. 21d.



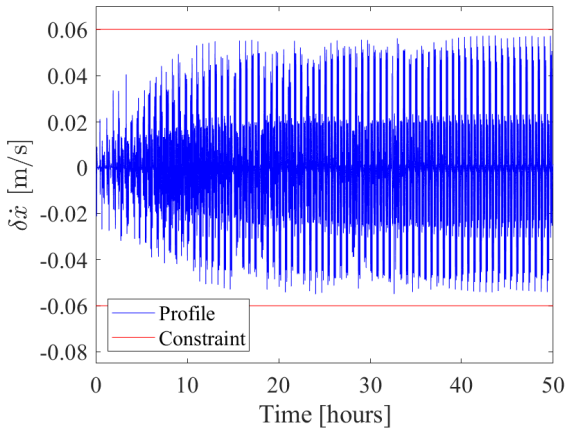
(a) Command profile during learning. **(b) At the beginning of learning.** **(c) At the end of learning.**

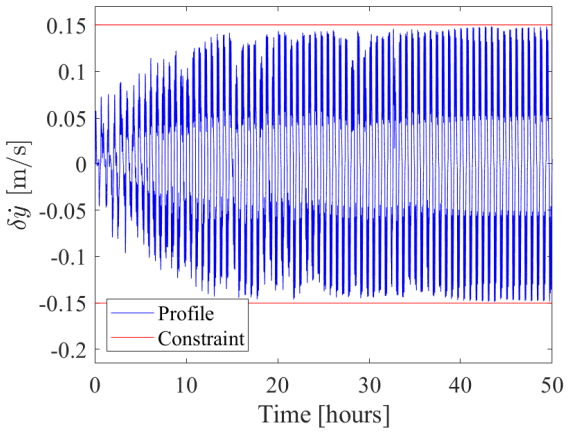
Fig. 20 Time histories of δy in Case Study 3 with constraints imposed on relative velocities.

The performance after different learning epochs is compared in Fig. 22. As can be seen from Fig. 22, in all cases, the chaser is able to follow the command without violating the relative velocity constraints. Figs. 22b and 22c also show that the nominal controller cannot enforce the relative velocity constraints without protection by the LRG. Note that $n = 0, 30, 60, 150$ correspond to $0, 10, 20, 50$ hours of learning, respectively. A comparison between the chaser behavior for $n = 60$ and $n = 150$ suggests 20 hours of learning is practically enough in this case.

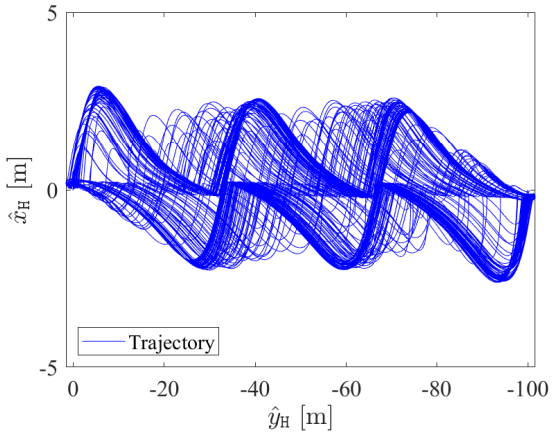
Comparison with Conventional RG. Finally, we compare the performances of the LRG and the conventional RG in Fig. 23. Fig. 23a shows the reference commands generated by both the LRG and the conventional RG, as well as the δy time histories. This result indicates that the conventional RG operates more aggressively than the LRG; however, the conventional RG developed with the approach followed above fails to strictly enforce the constraints (see Figs. 23b and 23c) even though the constraint violation is not very large.

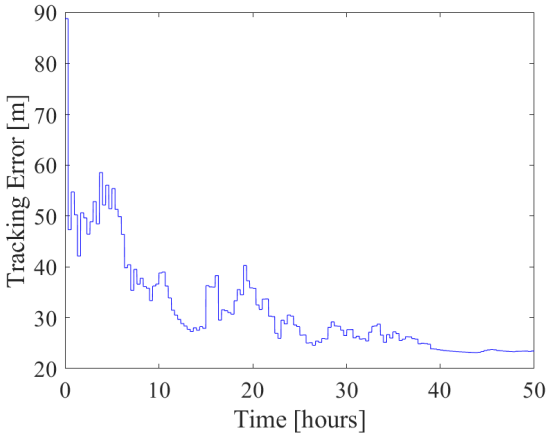
Furthermore, this comparison highlights the fact that the response with the LRG, after sufficient learning, is not significantly slower as compared to RG. In other words, the LRG is able to enforce the constraints with similar





(a) Profile of relative velocity in $\hat{x}_{\mathrm{H}}$ direction during learning. (b) Profile of relative velocity in $\hat{y}_{\mathrm{H}}$ direction during learning.





(c) Chaser trajectories during learning. **(d) Tracking error profile during learning.**

Fig. 21 Learning process in Case Study 3 with constraints imposed on the relative velocities.

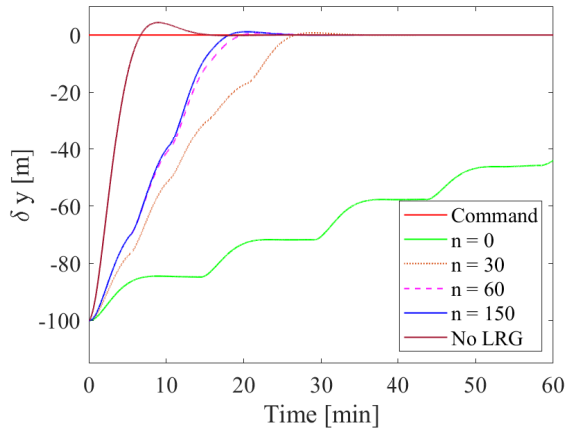
performance as the conventional RG but without relying on a dynamic model of the system (e.g., A and B matrices).

4. Discussion

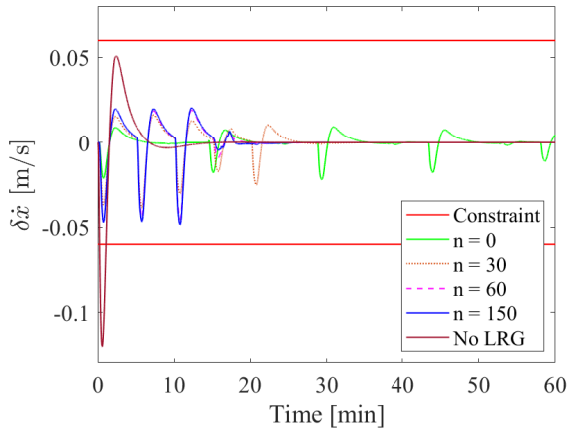
The performance of the LRG shown in Fig. 23 can be potentially improved by tuning the LRG parameters listed in Table 2. The analysis of sensitivity to each parameter is left as a subject for future work.

The constraint violations by the conventional-RG-protected chaser observed in Figs. 23b and 23c are attributed to the model simplifications we employed in Sec. IV.C.2. Although these violations are relatively small in this case study, larger violations can be expected in the case of more significant model uncertainty.

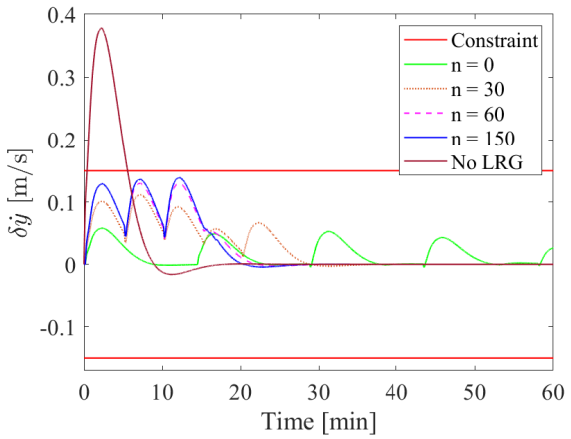
Along these lines, we consider another hypothetical scenario to highlight the capability of the LRG compared to the conventional RG. In particular, a change or hazardous damage to spacecraft may occur during spacecraft operation. As an example, a JAXA's X-ray space telescope, Hitomi, partially broke apart due to a human error involving an incorrect update of control parameters from the ground station. Suppose that the conventional RG is implemented using a model



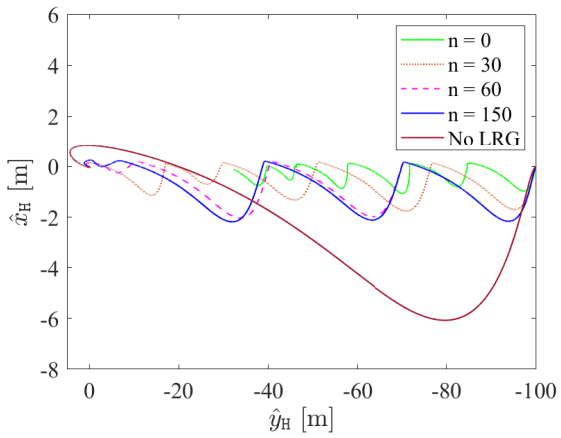
(a) Response to a step δy command.



(b) Profile of relative velocity in $\hat{x}_{\mathrm{H}}$ direction during mission.



(c) Profile of relative velocity in $\hat{y}_{\mathrm{H}}$ direction during mission.



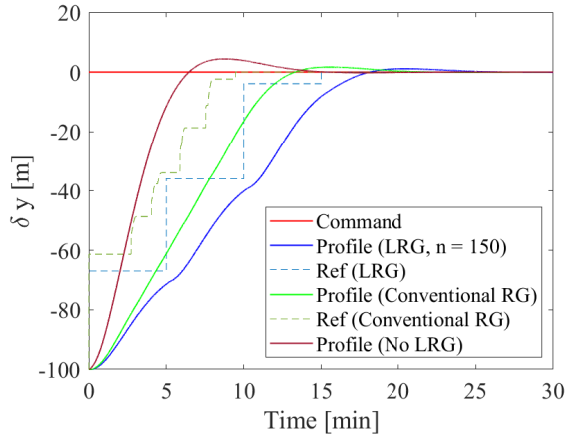
(d) Chaser trajectories.

Fig. 22 Comparison of rendezvous trajectories with the LRG after different learning epochs and without the LRG in Case Study 3.

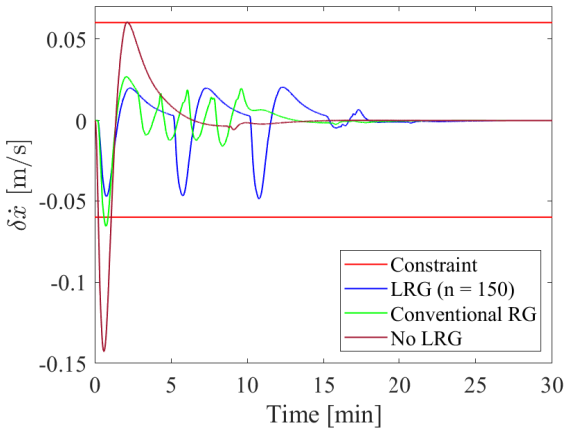
with

$$\bar{B}_{\mathrm{f}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.7 & 0 \\ 0 & 0.7 \end{bmatrix} \tag{33}$$

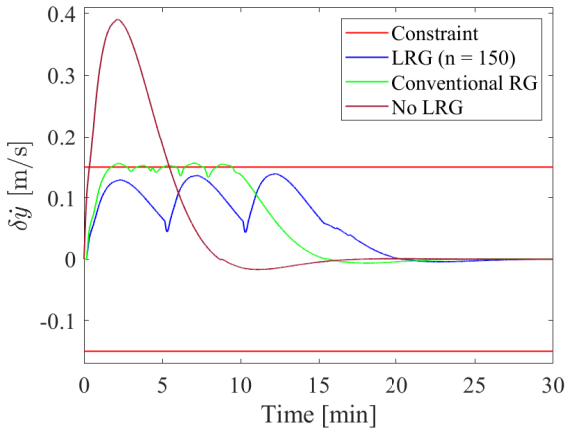
instead of $\bar{B}$ in (7), where $\bar{B}_{\mathrm{f}}$ corresponds to operation in a fault mode and suppose such $\tilde{O}_{\infty}$ is by mistake communicated to the spacecraft operating in the normal mode. Fig. 24 compares the profiles of relative velocities of the chaser with the LRG, the traditional RG, and the traditional RG with this model error. As can be seen from these plots, the constraint violations became larger due to the larger model error. The significance of the LRG is that we can reduce the risk of hazardous accidents by employing learning without the need for developing, identifying, and simplifying detailed



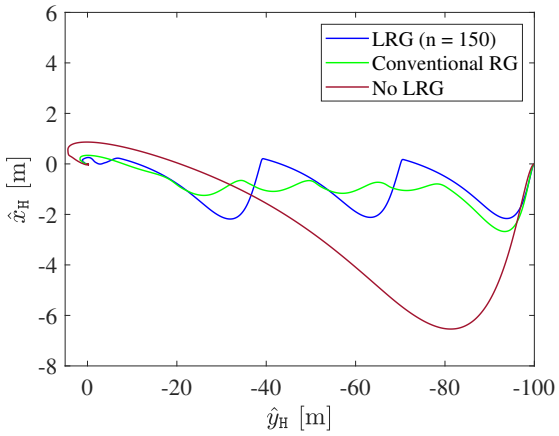
(a) Response to a step δy command.



(b) Profile of relative velocity in $\hat{x}_{\mathrm{H}}$ direction during mission



(c) Profile of relative velocity in $\hat{y}_{\mathrm{H}}$ direction during mission



(d) Chaser trajectories

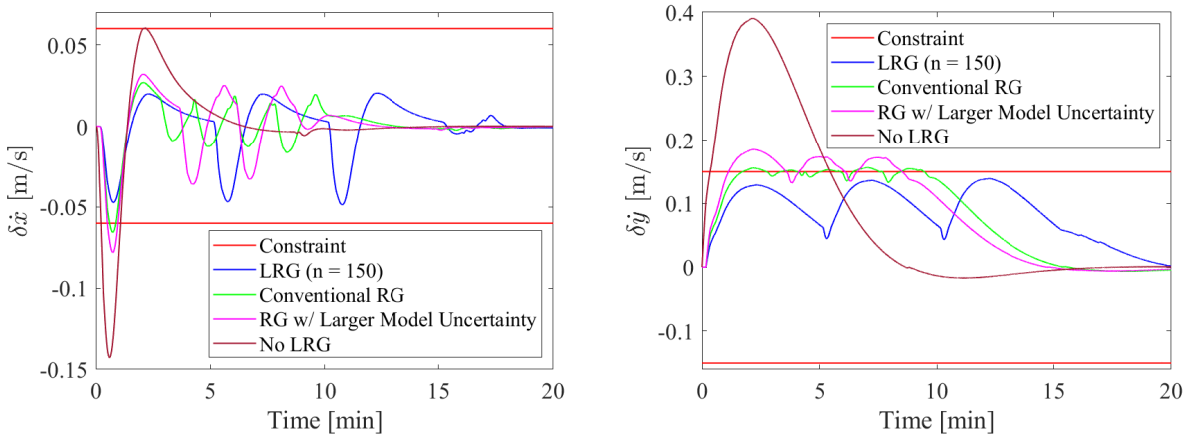
Fig. 23 Comparison of rendezvous trajectories with the LRG after a learning epoch and traditional RG in Case Study 3.

models of system dynamics that can be difficult in certain situations.

V. Concluding Remarks

This paper developed and demonstrated in simulations a Learning Reference Governor (LRG) for the spacecraft Automated Rendezvous, Proximity Operations, and Docking (ARPOD) missions. The LRG learns to perform safe relative motion maneuvers through experimentation and without violating the imposed constraints such as on the maximum thrust, LoS cone angle, and relative velocity. As the case studies show, with the proposed safety-critical LRG approach, the learning can be interrupted at any time and the LRG can be deployed to ensure that the actual mission can be accomplished without constraint violations; however, with more learning, more agile maneuvers could be performed.

The development of more systematic procedures to experimentally estimate the Lipschitz constant L of the peak in the spacecraft response, as required by our approach is left to future work. Handling other constraints such as avoiding



(a) Profile of relative velocity in $\hat{x}_{\mathrm{H}}$ direction during mission **(b) Profile of relative velocity in $\hat{y}_{\mathrm{H}}$ direction during mission**

Fig. 24 Comparison of rendezvous trajectories with the LRG and the conventional RG developed on a model with larger uncertainty.

thrusting into the target is another direction to extend this research. The LRG could benefit other applications that include multiple instances of maneuvers being performed by similar spacecraft (e.g., in formation flying missions). Additionally, extensions of our approach to more complex spacecraft configurations and constraints, such as the multibody underactuated spacecraft considered in [34], obstacle avoidance as in [35], rendezvous on elliptic and Halo orbits, docking to a rotating chief spacecraft, and formation flight between two or more deputy spacecraft with respect to one chief spacecraft could be tractable; such extensions will be pursued in future research.

Acknowledgment

This work was supported in part by the National Science Foundation award number CMMI-1904394.

References

[1] Luu, M. A., and Hastings, D. E., "On-Orbit Servicing System Architectures for Proliferated Low-Earth-Orbit Constellations," *Journal of Spacecraft and Rockets*, Vol. 59, No. 6, 2022, pp. 1946–1965. https://doi.org/10.2514/1.A35393, URL https://doi.org/10.2514/1.A35393.

[2] Kelly, P. W., Bevilacqua, R., Mazal, L., and Erwin, R. S., "TugSat: Removing Space Debris from Geostationary Orbits Using Solar Sails," *Journal of Spacecraft and Rockets*, Vol. 55, No. 2, 2018, pp. 437–450. https://doi.org/10.2514/1.A33872, URL https://doi.org/10.2514/1.A33872.

[3] Bang, J., and Ahn, J., "Multitarget Rendezvous for Active Debris Removal Using Multiple Spacecraft," *Journal of Spacecraft and Rockets*, Vol. 56, No. 4, 2019, pp. 1237–1247. https://doi.org/10.2514/1.A34344, URL https://doi.org/10.2514/1.A34344.

[4] Shen, H.-X., Zhang, T.-J., Casalino, L., and Pastrone, D., "Optimization of Active Debris Removal Missions with Multiple

Targets," *Journal of Spacecraft and Rockets*, Vol. 55, No. 1, 2018, pp. 181–189. https://doi.org/10.2514/1.A33883, URL https://doi.org/10.2514/1.A33883.

[5] Northrop Grumman, "Northrop Grumman and Intelsat Make History with Docking of Second Mission Extension Vehicle to Extend Life of Satellite," https://news.northropgrumman.com/news/releases/northrop-grumman-and-intelsat-make-history-with-docking-of-second-mission-extension-vehicle-to-extend-life-of-satellite, Apr 2021. Accessed: 2022-11-09.

[6] Lopez, I., and McInnes, C. R., "Autonomous rendezvous using artificial potential function guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 2, 1995, pp. 237–241. https://doi.org/10.2514/3.21375, URL https://doi.org/10.2514/3.21375.

[7] Lee, D.-R., and Pernicka, H., "Integrated System for Autonomous Proximity Operations and Docking," *International Journal of Aeronautical and Space Sciences*, Vol. 12, No. 1, 2011, pp. 43–56. https://doi.org/http://dx.doi.org/10.5139/IJASS.2011.12.1.43, URL http://koreascience.or.kr/article/JAKO201115037886127.page.

[8] Weiss, A., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "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. https://doi.org/10.1109/TCST.2014.2379639.

[9] Gavilan, F., Vazquez, R., and Camacho, E. F., "Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation," *Control Engineering Practice*, Vol. 20, No. 2, 2012, pp. 111–122. https://doi.org/https://doi.org/10.1016/j.conengprac.2011.09.006, URL https://www.sciencedirect.com/science/article/pii/S0967066111001985.

[10] Sanchez, J. C., Gavilan, F., and Vazquez, R., "Chance-constrained Model Predictive Control for Near Rectilinear Halo Orbit spacecraft rendezvous," *Aerospace Science and Technology*, Vol. 100, 2020, p. 105827. https://doi.org/https://doi.org/10.1016/j.ast.2020.105827, URL https://www.sciencedirect.com/science/article/pii/S127096381932348X.

[11] Mammarella, M., Capello, E., Park, H., Guglieri, G., and Romano, M., "Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance," *Aerospace Science and Technology*, Vol. 77, 2018, pp. 585–594. https://doi.org/https://doi.org/10.1016/j.ast.2018.04.009, URL https://www.sciencedirect.com/science/article/pii/S1270963817321223.

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

[13] Wang, Y., Li, K., Yang, K., and Ji, H., "Adaptive backstepping control for spacecraft rendezvous on elliptical orbits based on transformed variables model," *International Journal of Control, Automation and Systems*, Vol. 16, No. 1, 2018, pp. 189–196. https://doi.org/10.1007/s12555-016-0696-0, URL https://doi.org/10.1007/s12555-016-0696-0.

[14] Dong, H., Hu, Q., and Akella, M. R., "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. https://doi.org/10.2514/1.G002322, URL https://doi.org/10.2514/1.G002322.

[15] Li, Q., Yuan, J., Zhang, B., and Wang, H., "Artificial potential field based robust adaptive control for spacecraft rendezvous and docking under motion constraint," *ISA Transactions*, Vol. 95, 2019, pp. 173–184. https://doi.org/https://doi.org/10.1016/j.isatra.2019.05.018, URL https://www.sciencedirect.com/science/article/pii/S001905781930240X.

[16] Li, Q., Sun, C., Song, S., Gou, Q., and Niu, Z., "Robust adaptive control for spacecraft final proximity maneuvers with safety constraint and input quantization," *ISA Transactions*, Vol. 111, 2021, pp. 35–46. https://doi.org/https://doi.org/10.1016/j.isatra.2020.10.064, URL https://www.sciencedirect.com/science/article/pii/S0019057820304675.

[17] Yang, Z., Xing, L., Gu, Z., Xiao, Y., Zhou, Y., Huang, Z., and Xue, L., "Model-Based Reinforcement Learning and Neural-Network-Based Policy Compression for Spacecraft Rendezvous on Resource-Constrained Embedded Systems," *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 1, 2023, pp. 1107–1116. https://doi.org/10.1109/TII.2022.3192085.

[18] Riano-Rios, C., Bevilacqua, R., and Dixon, W. E., "Adaptive control for differential drag-based rendezvous maneuvers with an unknown target," *Acta Astronautica*, Vol. 181, 2021, pp. 733–740. https://doi.org/https://doi.org/10.1016/j.actaastro.2020.03.011, URL https://www.sciencedirect.com/science/article/pii/S0094576520301399.

[19] Broida, J., and Linares, R., "Spacecraft Rendezvous Guidance in Cluttered Environments via Reinforcement Learning," *29th AAS/AIAA Space Flight Mechanics Meeting*, 2019.

[20] Federici, L., Benedikter, B., and Zavoli, A., "Machine Learning Techniques for Autonomous Spacecraft Guidance during Proximity Operations," *AIAA Scitech 2021 Forum*, 2021. https://doi.org/10.2514/6.2021-0668, URL https://arc.aiaa.org/doi/abs/10.2514/6.2021-0668.

[21] Federici, L., Scorsoglio, A., Zavoli, A., and Furfaro, R., "Meta-reinforcement learning for adaptive spacecraft guidance during finite-thrust rendezvous missions," *Acta Astronautica*, Vol. 201, 2022, pp. 129–141. https://doi.org/https://doi.org/10.1016/j.actaastro.2022.08.047, URL https://www.sciencedirect.com/science/article/pii/S009457652200460X.

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

[23] Qu, Q., Liu, K., Wang, W., and Lü, J., "Spacecraft Proximity Maneuvering and Rendezvous With Collision Avoidance Based on Reinforcement Learning," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 58, No. 6, 2022, pp. 5823–5834. https://doi.org/10.1109/TAES.2022.3180271.

[24] Liu, K., Li, N., Rizzo, D., Garone, E., Kolmanovsky, I., and Girard, A., "Model-free Learning to Avoid Constraint Violations: An Explicit Reference Governor Approach," *2019 American Control Conference (ACC)*, 2019, pp. 934–940. https://doi.org/10.23919/ACC.2019.8814772.

[25] Liu, K., Li, N., Kolmanovsky, I., Rizzo, D., and Girard, A., "Model-free Learning for Safety-critical Control Systems: A Reference Governor Approach," *2020 American Control Conference (ACC)*, 2020, pp. 943–949. https://doi.org/10.23919/ACC45564.2020.9147255.

[26] Liu, K., Li, N., Kolmanovsky, I., Rizzo, D., and Girard, A., "Tanker truck rollover avoidance using learning reference governor," *SAE International Journal of Advances and Current Practices in Mobility*, Vol. 3, No. 2021-01-0256, 2021, pp. 1385–1394. https://doi.org/https://doi.org/10.4271/2021-01-0256.

[27] Maldonado, B. P., Li, N., Kolmanovsky, I., and Stefanopoulou, A. G., "Learning reference governor for cycle-to-cycle combustion control with misfire avoidance in spark-ignition engines at high exhaust gas recirculation–diluted conditions," *International Journal of Engine Research*, Vol. 21, No. 10, 2020, pp. 1819–1834. https://doi.org/10.1177/1468087420929109, URL https://doi.org/10.1177/1468087420929109.

[28] Garone, E., Di Cairano, S., and Kolmanovsky, I., "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, Vol. 75, 2017, pp. 306–328. https://doi.org/https://doi.org/10.1016/j.automatica.2016.08.013, URL https://www.sciencedirect.com/science/article/pii/S0005109816303715.

[29] Petersen, C. D., Hobbs, K., Lang, K., and Phillips, S., "Challenge Problem: Assured Satellite Proximity Operations," *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021.

[30] Jia-Richards, O., and Lozano, P. C., "Analytical Maneuver Library for Remote Inspection with an Underactuated Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 4, 2022, pp. 611–622. https://doi.org/10.2514/1.G005766, URL https://doi.org/10.2514/1.G005766.

[31] Wie, B., *Space Vehicle Dynamics and Control*, 2nd ed., American Institute of Aeronautics and Astronautics, 2008.

[32] Liu, K., Li, N., Kolmanovsky, I., Rizzo, D., and Girard, A., "Safe Learning Reference Governor: Theory and Application to Fuel Truck Rollover Avoidance," *Journal of Autonomous Vehicles and Systems*, Vol. 1, No. 4, 2022. https://doi.org/10.1115/1.4053244, URL https://doi.org/10.1115/1.4053244, 041003.

[33] Gilbert, E. G., and Tan, K. T., "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Transactions on Automatic control*, Vol. 36, No. 9, 1991, pp. 1008–1020.

[34] Rui, C., Kolmanovsky, I. V., and McClamroch, N. H., "Nonlinear attitude and shape control of spacecraft with articulated appendages and reaction wheels," *IEEE Transactions on Automatic Control*, Vol. 45, No. 8, 2000, pp. 1455–1469. https://doi.org/10.1109/9.871754.

[35] Weiss, A., Petersen, C., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Safe Positively Invariant Sets for Spacecraft Obstacle Avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2015, pp. 720–732. https://doi.org/10.2514/1.G000115, URL https://doi.org/10.2514/1.G000115.