Learning to Satisfy Constraints in Spacecraft Rendezvous and Proximity Maneuvering: A Learning Reference Governor Approach

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

This paper illustrates an approach to integrate learning into spacecraft automated rendezvous, proximity maneuvering, and docking (ARPOD) operations. Spacecraft rendezvous plays a significant role in many spacecraft missions including orbital transfers, ISS re-supply, on-orbit refueling and servicing, and debris removal. On one hand, 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 multi-layered structure of its nominal control system. On the other hand, spacecraft maneuvers need to satisfy required constraints (thrust limits, line of sight cone constraints, relative velocity of approach, etc.) to ensure safety and achieve ARPOD objectives. This paper considers an application of a learning-based reference governor (LRG) 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. As the RG, LRG modifies, if it becomes necessary, the command to a constraint-admissible reference 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. Simulations of spacecraft constrained relative motion maneuvers on a low Earth orbit are reported that demonstrate the effectiveness of the proposed approach.

I. Introduction

SPACE missions are becoming increasingly more complex and autonomous, and they invariably integrate rendezvous, proximity operations, and docking maneuvers. For instance, 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 on-orbit servicing. In the International Space Station (ISS) program, rendezvous and docking on Low Earth Orbits (LEOs) are essential capabilities for the resupply and orbit maintenance. Moreover, concomitant with the increasing number of space debris, many active debris removal missions have been proposed, which involve docking with non-cooperative targets, see e.g., [1, 2].

In general, automated rendezvous, proximity operations, and docking (ARPOD) missions result in challenging control problems because of constraints and uncertainties. A large number of control methods have been proposed to improve safety and robustness at every stage of rendezvous. For instance, Weiss et al. [3] applied linear quadratic Model Predictive Control (MPC) to generate spacecraft trajectories for both rendezvous and docking phases while demonstrating the capabilities to avoid debris and exclusion zones. MPC has also been applied by Zaman et al.[4] to ARPOD problems with safety constraints.

Adaptive and learning control methods for applications to ARPOD missions have also been studied. For instance, Dong et al. [5] exploited an adaptive control law and potential functions for a safe target approach. Reference [6] considered the application of robust and adaptive backstepping control to the rendezvous problems. Riano-Rios et al. [7] 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 [8] applied Reinforcement Learning (RL),

*M.Sc.Eng. 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. The fourth author would like to acknowledge support by the National Science Foundation award number CMMI-1904394.

in particular Proximal Policy Optimization (PPO), to spacecraft rendezvous. Federici et al. [9] further investigated Behavioral Cloning as well as PPO, and compared trajectories to those of the MPC solution. While these efforts simulated successful docking with collision avoidance, handling state constraints while providing explicit state constraint satisfaction guarantees remains both a theoretical and a practical challenge.

The present paper considers the use of a recently developed Learning Reference Governor (LRG) [10–12] for 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. In our proposed application, the LRG monitors and modifies the command generated by a higher-level ARPOD planning algorithm when it becomes necessary to enforce constraints. One distinct feature of LRG, as compared to the conventional RG [13], 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 LRG [11] guarantees constraints enforcement both during and after learning, and is adopted in this work.

The rest of the paper is organized as follows. The spacecraft model used for LRG training and simulations is described in Sec. II. This model represents translational and rotational spacecraft dynamics under a nominal controller that operates a single thruster and a reaction wheel to track the target relative position. Sec. III introduces the LRG algorithm, and Sec. IV demonstrates the application of LRG for rendezvous on a low Earth circular orbit (LEO) subject to constraints on thrust magnitude and on approaching the target within the Line of Sight (LoS) cone. 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 target 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-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 motivated by [14], see Figure 1. The $\hat{\boldsymbol{x}}_{\mathrm{H}}$, $\hat{\boldsymbol{y}}_{\mathrm{H}}$ are the radial (R-bar) and in-track (V-bar) spacecraft coordinates in the Hill's (H) frame. The target spacecraft is assumed to be at the origin of Hill's frame, and the target spacecraft orbit is circular. The spacecraft body fixed frame is denoted by B and the thrust force is directed along the spacecraft body fixed frame $\hat{\boldsymbol{x}}_{\mathrm{B}}$-axis, while the spin axis of the reaction wheel is aligned with the $-\hat{z}_{\mathrm{B}}$-axis.

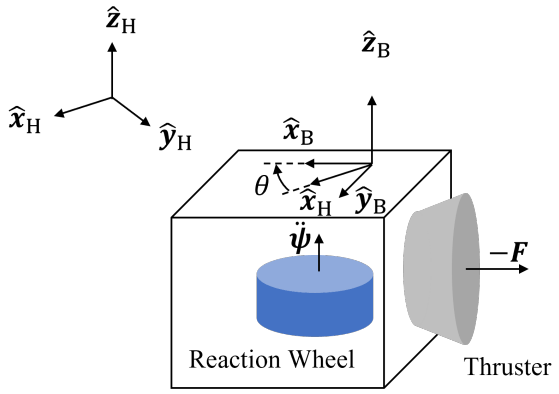


Fig. 1 Spacecraft schematics.

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

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

where the mean motion $n = \sqrt{\mu/r_0^3}$. First two equations represents the translational dynamics of the chaser spacecraft in the Hill's frame, whereas the third equation expresses the rotation of the chaser. $F\cos\theta$ and $F\sin\theta$ represent thrust force components in $\hat{\boldsymbol{x}}_\mathrm{H}$ direction and $\hat{\boldsymbol{y}}_\mathrm{H}$ directions, respectively. The rotation of the chaser is performed by controlling the reaction wheel, which defines the direction of thrust. Table 1 lists symbols and their definitions used in this paper. Note that this paper considers a simplified setting with the translational motion of the chaser restricted to x-y orbital plane.

Table 1 Definitions of symbols.

Symbol	Definition
δx, δy, δz	Components of the (relative) position vector of the spacecraft in Hill's frame.
θ	Angle that prescribed the direction of the thrust vector with respect to $\hat{\boldsymbol{x}}_\mathrm{B}$.
F	Magnitude of thrust force acting on the chaser spacecraft.
$\ddot{\psi}$	Angular acceleration of the reaction wheel.
$n = \sqrt{\mu/r_0^3}$	Mean motion on the target spacecraft's circular orbit.
m_c	Chaser spacecraft mass.
I_{zz}	Chaser spacecraft moment of inertia about $\hat{\boldsymbol{z}}_\mathrm{B}$-axis.
I_rw	Reaction wheel moment of inertia about the spin axis.
r_0	Circular orbit radius of the target spacecraft.
μ	Earth gravitational parameter.

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

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

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

As maneuvers are performed, the spacecraft mass and moment inertia change due to fuel being consumed. 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 at the Earth surface. The coefficient α is computed as

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

B. Nominal Controller

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

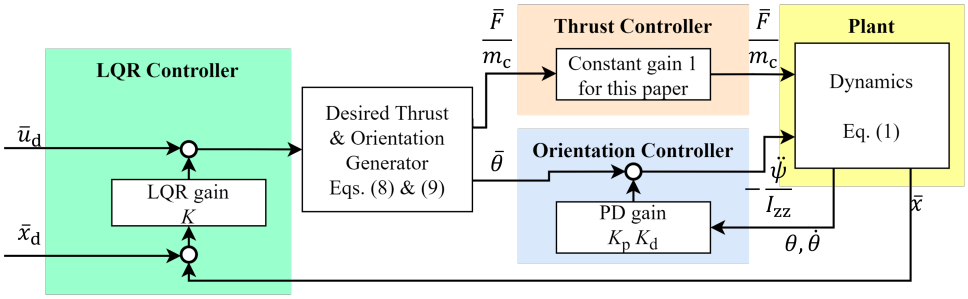


Fig. 2 Diagram of the nominal controller employed for the rendezvous problem.

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

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

where $\bar{x} = [\delta x, \delta y, \delta \dot{x}, \delta \dot{y}]^\mathsf{T}$ corresponds to the translational part of the state, $v = [v_x, v_y]^\mathsf{T}$ is the vector of target δx and δy relative coordinates for the chaser spacecraft, $\bar{u}_\mathrm{d}$ is the target acceleration of the chaser, and K is a feedback gain computed by minimizing the cost functional,

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

subject to $\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}\bar{u}$, $0 = \bar{A}\bar{x}_\mathrm{d} + \bar{B}\bar{u}_\mathrm{d}$, 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 $Q \geq 0$ and $R > 0$ aggregate weights for the state and control variables, respectively. Note that for in-track commands, $v_x = 0$ 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:

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

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

The thrust force magnitude is commanded to the thruster. As typical thrusters are on-off devices, the thrust force magnitude can be realized using a pulse width modulation (PWM). In this paper, it is assumed that the thrust force magnitude can be accurately realized, noting, however, that the effects of PWM can be handled by LRG, further highlighting the advantages of the LRG approach.

For the inner-loop design, a PD controller computes the reaction wheel acceleration command so that

$$-\frac{\ddot{\psi}}{I_{zz}} = K_\mathrm{p}(\bar{\theta} - \theta) - K_\mathrm{d}\dot{\theta}, \tag{10}$$

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 design of the nominal controller does not consider system constraints. During the actual operation, the system may be subjected to multiple constraints, such as thrust limits and LoS cone angle considered in this paper. In order to enforce these constraints during runtime, a reference governor [13] that modifies the commands v_x and v_y to constraint-admissible references to enforce point-wise in time constraints can be utilized. However, the design of the conventional reference governor is impeded by the hierarchical and nonlinear characteristics of the nominal closed-loop system and can be even harder when PWM effects are considered. As a result, we propose the use of a Learning Reference Governor (LRG) that is able to enforce constraints through learning with minimal knowledge of the system.

III. Learning Reference Governor (LRG) Algorithm

To apply LRG, we consider the closed-loop spacecraft relative motion dynamics as represented by the following equations of motion:

$$\begin{aligned} \dot{x}(t) &= f(x(t), v(t)), \\ y(t) &= g(x(t), v(t)) \end{aligned} \tag{11}$$

where $x(t) = [\delta x, \delta y, \delta \dot{x}, \delta \dot{y}, \theta, \dot{\theta}]^\mathrm{T}$ 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 $y(t)$ is the output of the system at time t on which the constraints are imposed. Note that LRG is applied to the closed-loop, pre-stabilized system introduced in Subsection II.B, consisting of the plant being controlled and a nominal controller (Fig. 3).

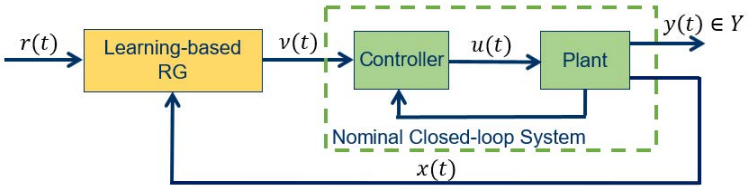


Fig. 3 Diagram of a nominal closed-loop system augmented with a Learning Reference Governor (LRG) to enforce constraints. $r(t)$ is the command generated either by a human operator or a higher-level planning algorithm.

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

$$y(t) \in Y, \quad \forall t \in [0, \infty). \tag{12}$$

For any constant $v \in V$, the closed-loop system,

$$\dot{x}(t) = f(x(t), v), \tag{13}$$

has a unique equilibrium, denoted by $x_v = x_v(v)$, which is asymptotically stable.

Let $\eta(\cdot, x_0, v) : [0, \infty) \to \mathbb{R}^6$ denote the solution to (13) with the initial condition $x(0) = x_0$ and constant reference $v \in V$, and $\phi(\cdot, x_0, v)$ denote the corresponding trajectory of output y. We define the function D as

$$D(v, \Delta v, \Delta x) := \sup_{t \in [0,\infty)} \|\phi(t, x_v(v) + \Delta x, v + \Delta v) - y_v(v)\|, \tag{14}$$

where $y_v(v) = g\big(x_v(v), v\big)$ denotes the steady-state output corresponding to the reference v. We assume the function D is Lipschitz continuous with a Lipschitz constant L, i.e., for any $z_1, z_2 \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^6$,

$$|D(z_1) - D(z_2)| \leq L \|z_1 - z_2\|. \tag{15}$$

We further assume that the state $x(t)$, the output $y(t)$, and the distance from the steady-state output trajectory associated with the current reference, $y_v(v)$, to the constraint boundary,

$$d(v) := \mathrm{dist}(y_v(v), Y^C) = \inf_{y \in Y^C} \big\|y_v(v) - y\big\|, \tag{16}$$

are measured, where Y^C is the complement of the safe set Y.

We also assume that given a trajectory over a time duration of length T, we can obtain an estimate of D, denoted as $\tilde{D}$, such that $D \leq \tilde{D} \leq D + \varepsilon$, where $\varepsilon > 0$ is a sufficiently small, known constant.

During the operation, the spacecraft is commanded to reach certain target relative coordinates in Hill's frame. However, if directly using the designed nominal controller in Sec. II.B, constraint violations may occur. As a result, 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,

$$v(t^+) = v(t^-) + \kappa(t)(r(t) - v(t^-)), \tag{17}$$

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

$$\kappa(t) = \max \kappa \in [0, 1], \quad \text{subject to} \tag{18}$$

$$D\left(v(t^-), \kappa(r(t) - v(t^-)), x(t) - x_v(v(t^-))\right) \leq d(v(t^-)).$$

Note that when $\kappa(t) = 1$, we have $v(t^+) = r(t)$, and the command is directly applied. Meanwhile, if the function D is known, the reference governor algorithm (18) can enforce the constraints by limiting the reference changes at each sample time instants t_k [11]. However, for the spacecraft system, D is not known a prior. Therefore, following from [11], we have the following Algorithms 1 to 2 to improve an estimate of D while ensuring safety during the learning process. More detailed descriptions of the algorithms, required assumptions, and theoretical analysis of the underlying properties can be found in [11].

Algorithm 1 Safe learning algorithm

1: Initialize the spacecraft with a constraint-admissible steady-state initial condition, $x(0) = x_v\big(v(0^-)\big)$ for some $v(0^-) \in V$ 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

$$\kappa(t) = \texttt{Kappa}\big(x(t), r_n, v(t^-), d\big(v(t^-)\big), \mathcal{D}\big); \tag{19}$$

6: Adjust the reference according to

$$v(t^+) = v(t^-) + \kappa(t)\big(r(t) - v(t^-)\big); \tag{20}$$

7: At the sample time instant $t' = t + T$, measure $\tilde{D}(t) = \tilde{D}\big(v(t^-), \Delta v(t), \Delta x(t)\big)$, where $\Delta v(t) = \kappa(t)\big(r(t) - v(t^-)\big)$ and $\Delta x(t) = x(t) - x_v\big(v(t^-)\big)$;
8: $\mathcal{D} = \mathcal{D} \cup \big(v(t^-), \Delta v(t), \Delta x(t), \tilde{D}(t)\big)$.
9: **end for**
10: **end for**

At the beginning of the learning, we initialize the spacecraft system with a constraint-admissible steady state and the measurement set as an empty set (Line 1). For each learning epoch, a command (target relative coordinates in Hill's frame) is generated (Line 3). Then on Line 5, the LRG will adjust the reference towards the command according to (18), where $D(v, \Delta v, \Delta x)$ is replaced by the current estimate $\bar{D}(v, \Delta v, \Delta x)$ given by

$$\bar{D}(v, \Delta v, \Delta x) = \min\left(L\left\|\begin{bmatrix}\Delta v \\ \Delta x\end{bmatrix}\right\|, \min_{i \in \mathcal{D}}\left(\tilde{D}_i + L\left\|\begin{bmatrix} v \\ \Delta v \\ \Delta x\end{bmatrix} - \begin{bmatrix} v_i \\ \Delta v_i \\ \Delta x_i\end{bmatrix}\right\|\right)\right), \tag{21}$$

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$(x, r, v, d, \mathcal{D})$

1: **for** $(v_i, \Delta v_i, \Delta x_i, \tilde{D}_i) \in \mathcal{D}$ **do**
2: Compute κ_i as the solution to the optimization problem,

$$\max \ \kappa \in [0,1], \quad \text{subject to} \tag{22}$$
$$\left\|\kappa(r-v) - \Delta v_i\right\| \leq \frac{d-\tilde{D}_i}{L} - \left\|\begin{bmatrix} v \\ x - x_v(v) \end{bmatrix} - \begin{bmatrix} v_i \\ \Delta x_i \end{bmatrix}\right\|,$$

if a solution exists, and $\kappa_i = 0$ otherwise;
3: **end for**
4:

$$\kappa' = \operatorname*{sat}_{[0,1]} \frac{d - L\|x - x_v(v)\|}{L\|r - v\|}, \tag{23}$$

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

The LRG algorithm can guarantee the following properties:

Proposition 1: The constraints (12) are guaranteed to be satisfied during the learning process.

Proposition 2: The constraints (12) are enforced during the operating phase where v is adjusted according to (18) 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 (18) converges to the command in finite time.

For more detailed assumptions, proofs of the propositions and other properties of the algorithm, please refer to [15].

IV. Case Studies

In this section, case studies of implementing and applying LRG to the spacecraft rendezvous problem are considered. The operation of the spacecraft is subject to constraints which prescribe the thrust limit and target approach within LoS cone. The nominal controller designed in Sec. II.B does not explicitly consider these constraints, so the LRG described in Sec. III is exploited to guard the spacecraft from constraint violations during the rendezvous and improve maneuver agility over time.

A. Case Studies Overview

The simulated case studies involve a chaser spacecraft and a target spacecraft in LEO. The chaser spacecraft needs to achieve the rendezvous with the target spacecraft. In the case study, the chaser spacecraft is commanded to reach a target position (in terms of relative coordinates) to accomplish its mission. The nominal controller described in Sec. II.B is utilized to control the chaser spacecraft towards the target position. During the rendezvous of the chaser spacecraft, two main constraints are considered including the propulsion or thrust limit and the LoS angle with the target spacecraft for successful docking. In this regard, the LRG is exploited to enforce these constraints through experimentation and learning while guarantee safety (constraint satisfaction) during the learning. After the learning is accomplished, the LRG is able to guard the chaser spacecraft from violating constraints and operate the spacecraft aggressively. In the case studies, constraints on maximum propulsion or thrust and LoS angle are considered separately at first, which demonstrates the ability of LRG to enforce these constraints through learning with minimal system information. Then, these constraints are combined together, and the LRG is able to operate the spacecraft safely and non-conservatively during the rendezvous mission.

Table 2 lists parameters used for the simulation. Accordign to Figure 4, the LRG is augmented to the nominal closed-loop system (including the controller and the plant) to handle additionally imposed constraints on the system. The LRG starts with a conservative design, performs experimentation with various step changes of the reference command, collects response data, and gradually learns to operate the system aggressively. After the learning is finished, the trained LRG is applied to the spacecraft to enforce constraints during rendezvous missions.

In what follows, we first demonstrate the ability of LRG to enforce the constraint on the maximum thrust magnitude (Sec. IV.B). Then in Sec. IV.C, we illustrates LRG's ability to handle the LoS cone angle constraints (Fig. 5). In order to improve the learning speed of the LRG, instead of setting L in (15) as a constant, we propose to construct L as a function of the state (more specifically, δy). Finally, the ability of LRG to handle both of these constraints simultaneously is

Table 2 Case Study Parameters

Parameter		Assumed value
Dynamics Parameter		
Semi-major axis	r_0	6,778 km
Specific impulse	I_{sp}	300 s
Chaser spacecraft initial mass	m_c	100 kg
Chaser spacecraft initial moment of inertial in $\hat{z}_B$-axis	I_{zz}	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 - 72
Number of reference adjustment at each command	k_{max}	4
Time duration between each reference adjustment	T	500 s
Lipschitz constant for thrust constrained case	L_F	0.00014
Lipschitz constant for cone angle constrained case	L_ξ	0.04
Constraint		
Maximum thrust	F_{max}	1.5 N
Maximum LoS cone angle	ξ_{max}	2.5 degree = 0.0436 rad
Diameter of the approach corridor at $\delta y = 0$	D_{target}	10 m

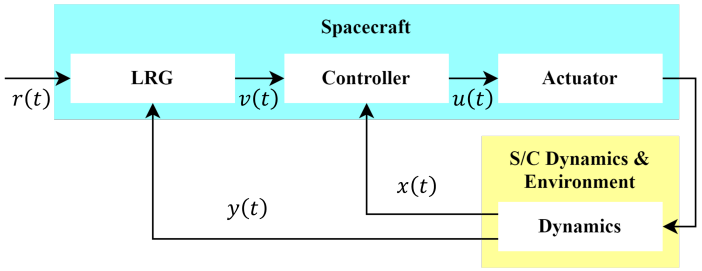


Fig. 4 Diagram of a case study with a learning reference governor.

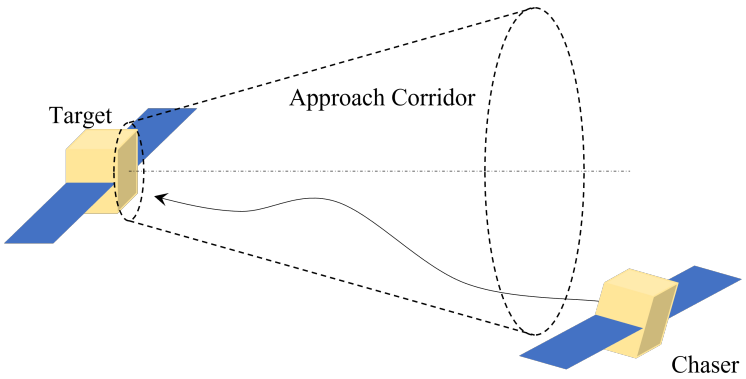
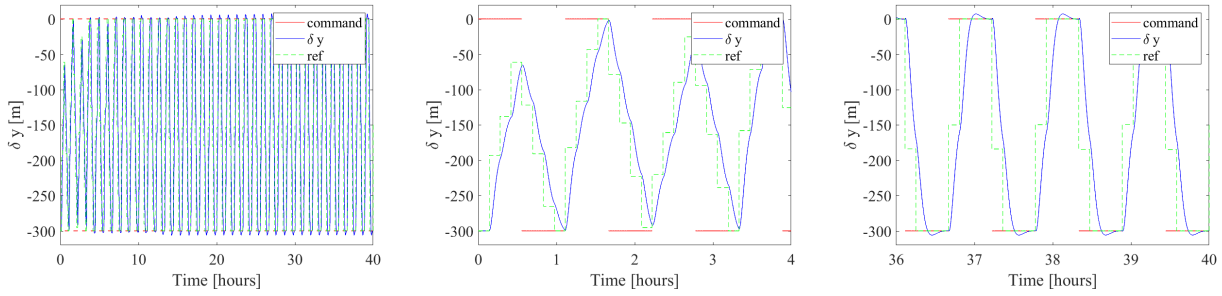


Fig. 5 LoS cone angle constraint defining the approach corridor for the rendezvous.

illustrated in Sec. IV.D. In each case, the initial state of the chaser is $x_0 = [0, -300\text{ m}, 0, 0, 0, 0]^\mathsf{T}$.



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

Fig. 6 The LRG adjusts references in Case 1 (constraint is imposed on thrust).

B. Case 1: Constraints on Thrust

1. Constraint and Modification of Algorithm

First, we consider the maximum thrust limit and the constraint $|F| \leq F_{\max} = 1.5$ N. Note that in the case studies, we assume that the thrust force determined by the nominal controller can be realized without saturation limit, and we want to limit the actual delivered thrust force to be within $F_{\max}$. The output of the system $y(t)$ is defined as

$$y(t) = \frac{F(t)}{m_c(t)}. \tag{24}$$

Since the mass of the spacecraft changes due to thrust, even though $F_{\max}$ is constant, Y^C changes during the learning. To deal with this, we restrict our commands v to only the in-track translation (i.e., $v_x = 0$). This implies

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

We then modify Eq.22 as

$$\left\|\kappa(r-v) - \Delta v_i\right\| \leq \frac{F_{\max} - M_i \tilde{D}_i}{M_i L} - \left\|\begin{bmatrix} v \\ x - x_v(v) \end{bmatrix} - \begin{bmatrix} v_i \\ \Delta x_i \end{bmatrix}\right\|, \tag{26}$$

where M_i is

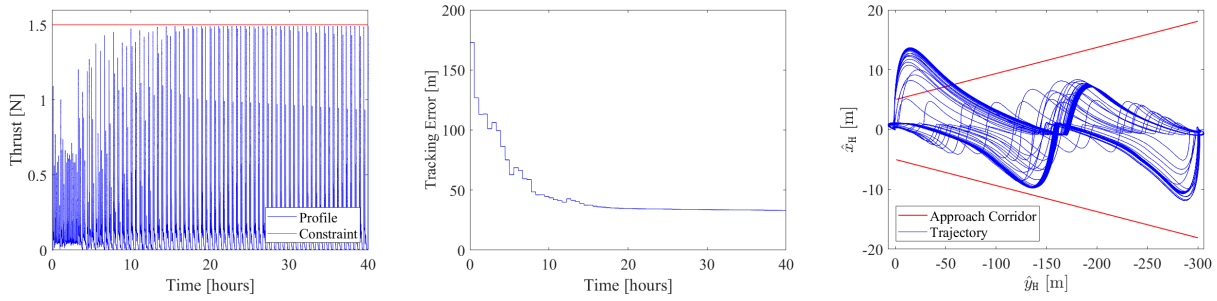
$$M_i := \max_{t \in [t_i, t_i+T]} m_c(t). \tag{27}$$

All case studies in this paper employ the 1-norm for every $\|\cdot\|$ involved in the algorithm. To obtain the value of the Lipschitz constant L, arbitrary 80 points in $(v, \Delta v, \Delta x)$ are sampled and corresponding derivative of D are numerically calculated. As a result, the Lipschitz constant L for this case needs to satisfy $L \geq 8.55 \times 10^{-5}$, and therefore we $L = 1.40 \times 10^{-4}$ to demonstrate that a conservative estimate of L is sufficient for the algorithm. Note that, however, conservative estimate of L can lead to slow learning rate.

2. Results

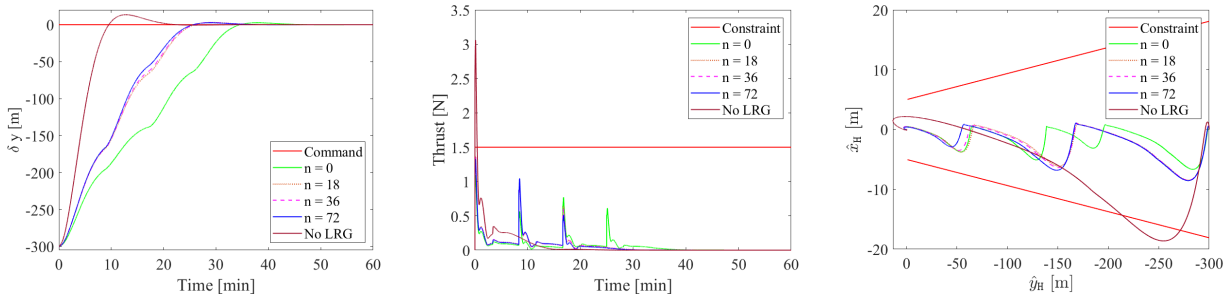
Fig. 6 illustrates the LRG modifies the references and outputs the commands. Especially, comparison of Figs. 6b and 6c indicates the modification becomes smaller as learning progresses.

Fig. 7a shows the response of F during learning corresponding to the commands in Fig. 6. The LRG operates the spacecraft conservatively at first due to the initial conservative bounds on the response as informed by the initial assumptions. As learning progresses, the LRG is able to operate the system more and more aggressively without violating constraints. The progress of the learning can be also visualized by plotting the tracking error, which is the average of $\|r - v\|$ over a time window of the most recent 8000 seconds (Fig. 7b). At the beginning of learning, the tracking error is high, and then it gradually decreases as the learning proceeds. Fig. 7c illustrates the chaser trajectory during the learning process in Hill's reference frame. Note that in this case, the chaser travels out of the specified approaching corridor as constraints on the maximum LoS cone angle are not imposed.



(a) Thrust profile during learning. **(b) Tracking error profile during learning.** **(c) Chaser trajectories during learning.**

Fig. 7 The learning process of the LRG applied to Case 1 (only thrust constraint is active).



(a) Response to a step δy command **(b) Thrust profile after learning** **(c) Chaser trajectories**

Fig. 8 Comparison of rendezvous mission with the LRG after different learning epochs and without the LRG of Case 1.

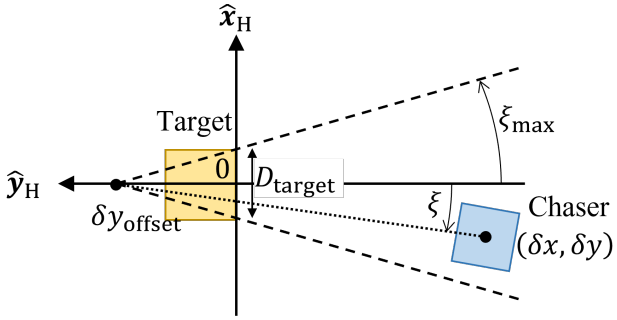


Fig. 9 Definition of LoS cone angle.

After learning is stopped, commands are applied to the chaser spacecraft, and the responses of the chaser spacecraft without LRG and with LRG after different learning epochs are shown in Fig. 8. As shown in Fig. 8a, in all cases the chaser is able to follow the command. Note that $n = 0, 18, 36, 72$ correspond to $0, 10, 20, 40$ hours of learning, respectively. Fig. 8a also indicates that the chaser arrives at the target position faster after longer learning. Without LRG, the system is able to track the command rapidly, but this leads to a significant constraint violations as seen from Fig. 8b. With LRG's protection, the spacecraft is able to arrive at the target position without violating constraints. Fig. 8c shows the trajectories of the chaser corresponding to each case.

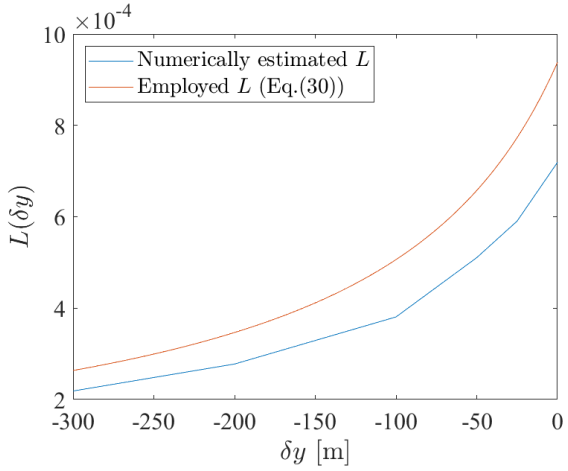


Fig. 10 Comparison of numerically estimated Lipschitz constant L and employed Lipschitz constant $L(\delta y)$ in (30) in Case 2 (LoS cone angle constraint).

C. Case 2: Constraints on Line of Sight Cone Angle

1. Constraint

In this case, we impose a constraint on the LoS cone angle ξ as $|\xi| \leq \xi_{\max} = 2.5^\circ = 0.0436$ rad and the output of the system $y(t)$ is defined as $y(t) = \xi$. Fig. 9 illustrates the definition of the LoS cone angle ξ. The diameter of the approach corridor at $\delta y = 0$, D_{target} is defined as 10 m, and thus, δy_{offset} can be derived as follows:

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

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

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

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

Note that the form of Eq. (29) implies that a small $|\delta y - \delta y_{\text{offset}}|$ (when the spacecraft approaches the corridor) makes ξ more sensitive to the change of δx. As a result, the Lipschitz constant L defined in Eq. (15) needs to be sufficiently large to make sure Eq. (15) is satisfied in the corridor area (where $|\delta y - \delta y_{\text{offset}}|$ is small). However, a large L will slow down the learning process [15]. For the sake of faster learning, in this case (and the next one), we treat the L as a function of δy, $L(\delta y)$. In this paper, $L(\delta y)$ is considered as follows:

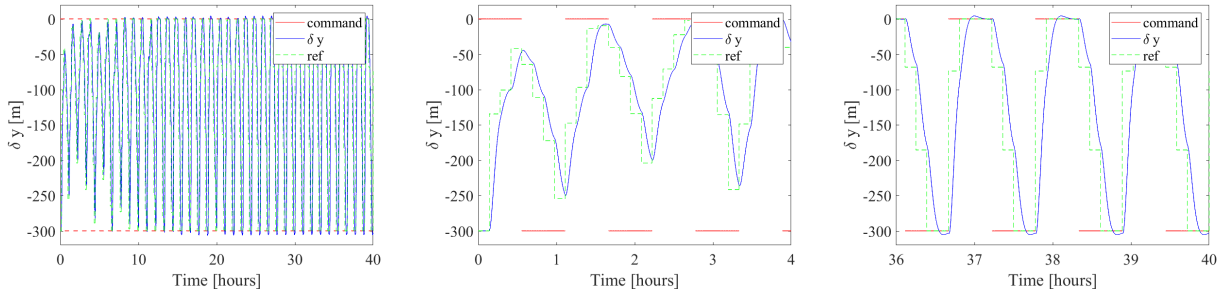
$$L(\delta y) = \frac{L_\xi}{\beta \, |\delta y - \delta y_{\text{offset}}| + 1}, \tag{30}$$

where the coefficient β is set as 2.75, and when $\delta y = \delta y_{\text{offset}}$, the Lipschitz constant L_ξ is set as 0.04. Note that this change of $L(\delta y)$ can be incorporated into the LRG algorithms directly. The only modification we need is in Algorithm 2, where state information x is available, and we first calculate $L(\delta y)$ by (30) in Algorithm 2 and replace L in (22) and (23) by $L(\delta y)$.

As a result, the estimate $\bar{D}(v, \Delta v, \Delta x)$ in (21) is replaced by,

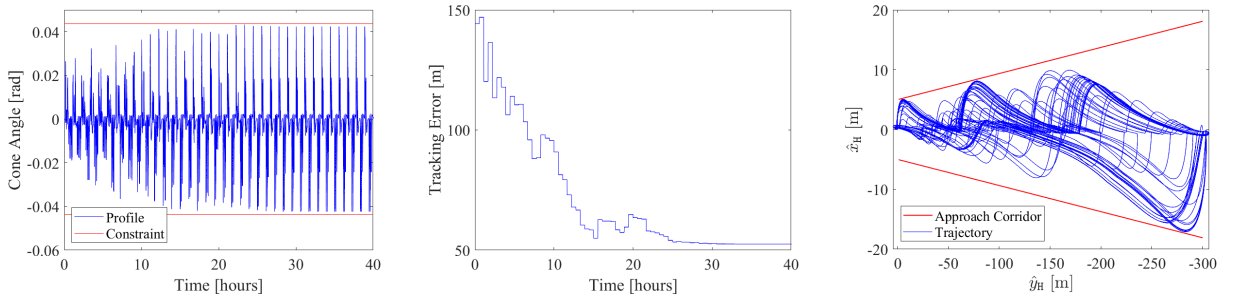
$$\bar{D}(v, \Delta v, \Delta x) = \min \left(L(v, \Delta x) \left\| \begin{bmatrix} \Delta v \\ \Delta x \end{bmatrix} \right\|, \min_{i \in \mathcal{D}} \left(\tilde{D}_i + L(v, \Delta x) \left\| \begin{bmatrix} v \\ \Delta v \\ \Delta x \end{bmatrix} - \begin{bmatrix} v_i \\ \Delta v_i \\ \Delta x_i \end{bmatrix} \right\| \right) \right), \tag{31}$$

where $L(v, \Delta x) = L(\delta y)$ because $x = x_v(v) + \Delta x$, which means the value of δy can be obtained using the information of v and Δx.



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

Fig. 11 The LRG adjusts references in Case 2 (constraint is imposed on LoS cone angle).



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

Fig. 12 The learning process of the LRG applied to Case 2 (only LoS cone angle constraint is active).

Fig. 10 compares the Lipschitz constant calculated from (30) and numerically estimated $L(\delta y)$ corresponding to $\delta y \in S = [0, -25, -50, -100, -200, -300]$ m. The numerical estimations of $L(\delta y)$ are achieved as follows: For each $\delta y' \in S$, arbitrary 80 points $(v, \Delta v, \Delta x)$ satisfying $\{(v, \Delta v, \Delta x) \mid v + \Delta x_2 = \delta y'\}$ are sampled, where v is the reference (i.e., the target δy) and Δx_2 is the second element of Δx. The constraint is imposed on sampling such that all sampled points satisfy $\delta y = v + \Delta x_2 = \delta y'$. Then, for each sampled $(v, \Delta v, \Delta x)$, the corresponding derivative of D are numerically calculated, and the maximum among all numerical derivatives of D is taken as $L(\delta y')$. From Fig. 10, we can observe that $L(\delta y)$ from (30) with chosen values of β and L_ξ is indeed a reasonable and conservative estimate of L over $\delta y \in [-300\text{ m}, 0\text{ m}]$.

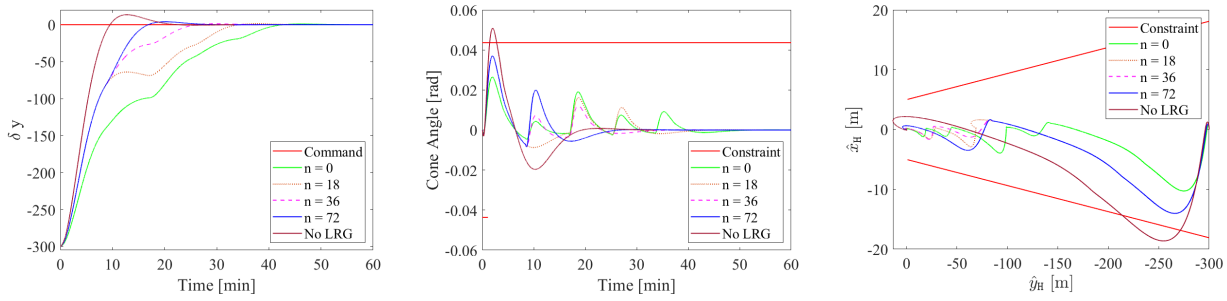
Note that the proposed modification of computing the Lipschitz constant relaxes the assumption of (15) and enables the LRG to address a broader class of problems. However, further theoretical analysis and simulation studies need to be performed to validate and evaluate the performance of the proposed relaxation of the assumption of (15). In what follows, we will show the simulation studies with the proposed modification of computing the Lipschitz constant.

2. Results

Similar to the thrust constrained case described in Subsection IV.B, the LRG adjusts the reference, and the magnitude of the reference adjustment becomes larger as learning processes as shown in Fig. 11, which demonstrates that the LRG operates the spacecraft more aggressively.

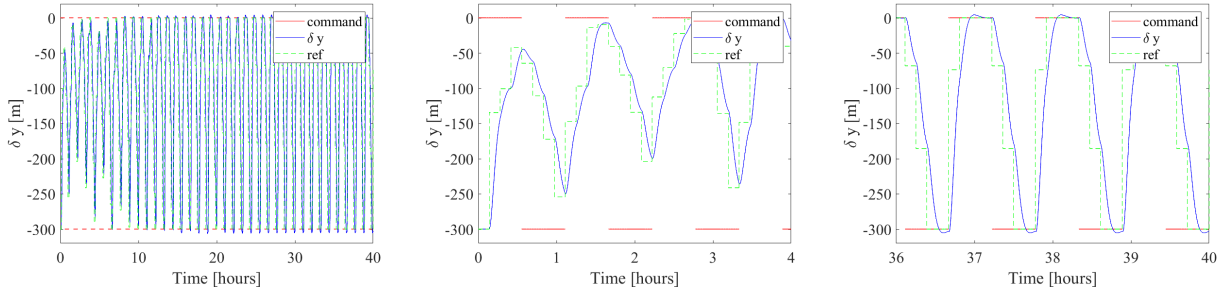
Fig. 12a shows that the LRG operates the spacecraft conservatively at the beginning of the learning and becomes more aggressive without violating LoS cone angle constraint as learning progresses. Learning is accomplished in about 35 hours as shown in Fig.12b. As we can observe from Fig. 12c, the trajectory of the chaser stays within the approach corridor as the LRG enforces LoS cone angle constraint.

Fig. 13 compares the responses of the chaser spacecraft without LRG and with LRG after different learning epochs. As shown in Fig. 13a, in all cases the chaser is able to follow the command (meaning the actual state reaches the commanded state). Without the protection of the LRG, the chaser arrives at the target position rapidly, but the LoS cone angle constraints are violated as seen from Figs. 13b and 13c. With the LRG's protection, at different learning phases,



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

Fig. 13 Comparison of rendezvous mission with the LRG after different learning epochs and without the LRG of Case 2.



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

Fig. 14 The LRG adjusts references in Case 3 (constraints are imposed on both thrust and LoS cone angle).

the spacecraft remains within the approach corridor at all times.

D. Case 3: Constraints on Thrust and Line of Sight Cone Angle

1. Constraint

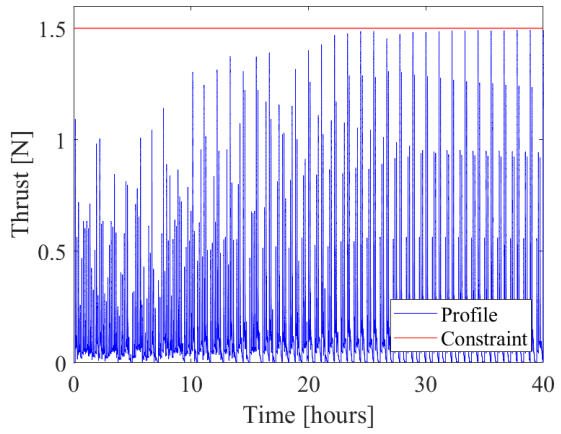
Finally, this case demonstrates the effectiveness of the LRG by imposing constraints on both thrust and LoS cone angle simultaneously. Imposing multiple constraints simultaneously is possible by calculating κ_1, $\kappa_2, \cdots$ for different constraints based on Algorithm 2 respectively, and employing the minimum value as κ: i.e.,

$$\kappa = \min(\kappa_1, \kappa_2, \cdots) \tag{32}$$

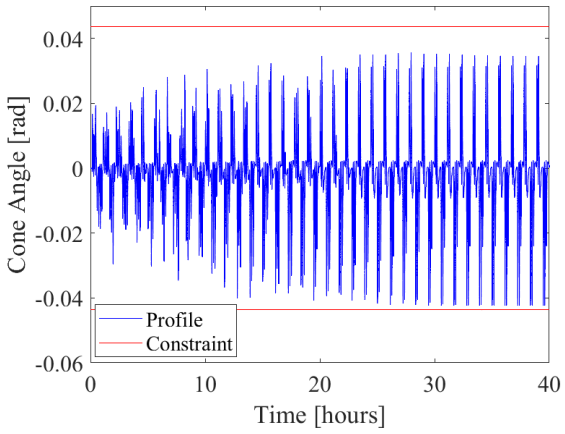
2. Results

Figs. 14 and 15 illustrate that the LRG adjusts the reference and is able to successfully enforce constraints on thrust and LoS cone angle. Especially, Figs. 15a and 15b show that the chaser violates neither of the constraints during the learning process. Learning is accomplished in about 35 hours as shown in Fig.15c. The trajectory of the chaser is shown in Fig. 15d, and this indicates that the LoS cone angle 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.

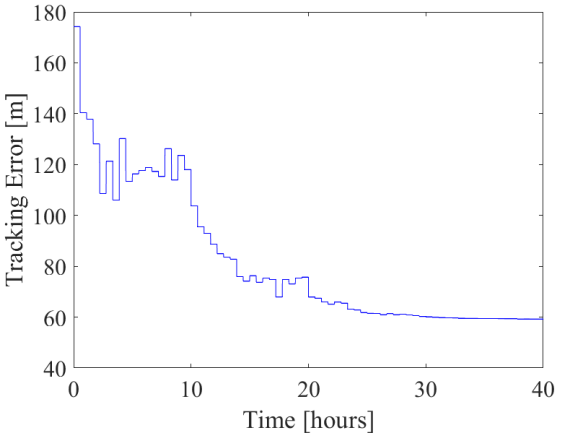
Fig. 16 compares the responses of the chaser without LRG and with LRG after different learning epochs. Fig. 13a shows in all cases the chaser is able to follow the command and reach the target state. Figs. 16b and 16c show that the chaser without the protection of the LRG violates both thrust and ling of sight cone angle constraints. In contrast, after learning for different number of epochs, the LRG is able to guard the chaser from violating either of the constraints. Fig. 16d visualizes the different trajectories of the chaser corresponding to each case.



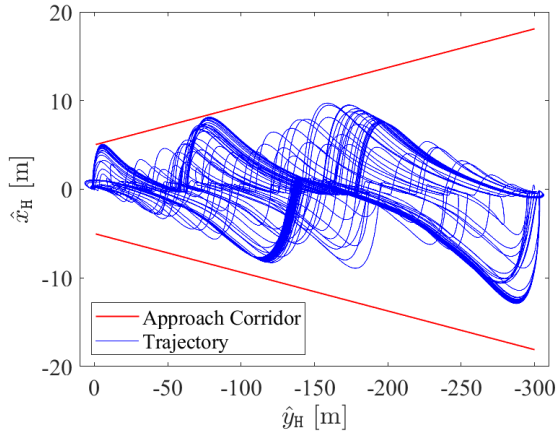
(a) Thrust profile during learning.



(b) LoS cone angle profile during learning.



(c) Tracking error profile during learning.



(d) Chaser trajectories during learning.

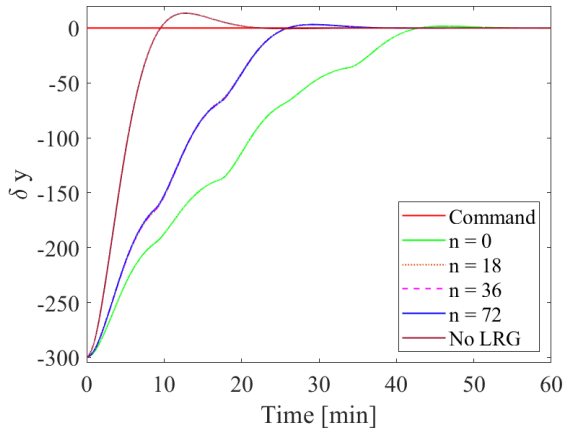
Fig. 15 The learning process of the LRG applied to Case 3 (thrust and LoS cone angle constraints are both active).

E. Discussions

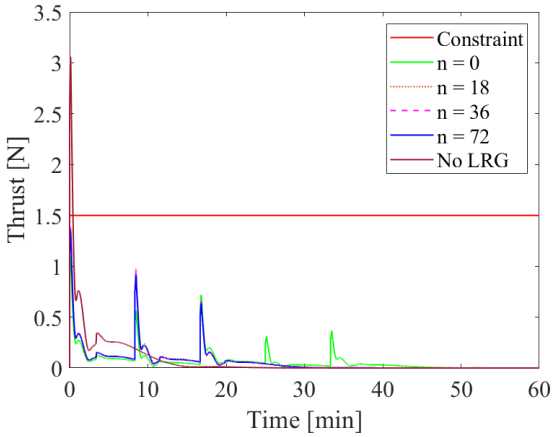
As shown by Figs. 8, 13, and 16, there is not a large difference in terms of the chaser's behavior between cases of $n = 18$ and $n = 72$. Thus, 10 hours' learning is practically enough for the case of enforcing thrust and LoS cone angle constraints during rendezvous mission considered in this paper.

While the LRG successfully protected the chaser from violating constraints, the learning process consumes a certain amount of time and propellant. Fig. 17 illustrates the change of mass during the learning for Case 3 (Sec. IV.D). Fast learning is desirable for the sake of saving propellant, where we have considered constructing the Lipschitz constant L as a function of δy so that we can employ a smaller value of L when appropriate to speedup the learning process.

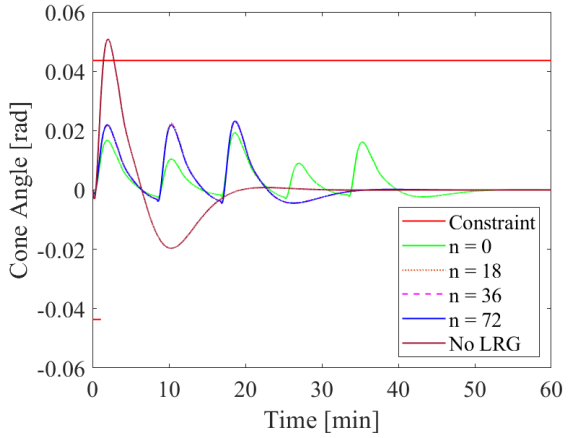
Finally, Fig. 18 compares the profile of the fraction of the chaser mass to the initial mass, $m_{\mathrm{c}}(t)/m_{\mathrm{c}}(0)$, corresponding to the maneuvers shown in Fig. 16. Larger $m_{\mathrm{c}}(t)/m_{\mathrm{c}}(0)$ implies smaller propellant consumption to achieve this rendezvous. The chaser without learning ($n = 0$) consumes the least amount of propellant, and this is due to the fact that less thrust is generated to achieve smaller reference adjustment Δv. When Δv becomes larger as the controller becomes more aggressive, the chaser generates larger thrust to track the reference and hence consumes more propellant. This result implies applying the LRG is effective in terms of fuel efficiency while not sacrificing the performance, and therefore, the LRG would benefit applications that include multiple times of maneuvers between spacecraft (e.g. formation flying missions).



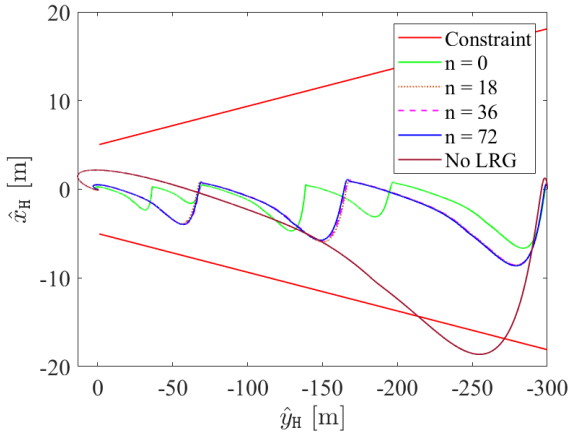
(a) Tracking results of a step δy command (rendezvous at origin)



(b) Thrust profile during mission



(c) LoS cone angle profile during mission



(d) Chaser trajectories

Fig. 16 Comparison of rendezvous mission with the LRG after different learning epochs and without the LRG of Case 3.

V. Concluding Remarks

This paper considered an application of the Learning Reference Governor (LRG) to the spacecraft autonomous rendezvous, proximity operations and docking (ARPOD) missions. The LRG learns to perform safe relative motion maneuvers through experimentation. With the proposed safety critical LRG approach, the learning can be interrupted and deployed at any time and the actual mission can be achieved without constraint violations; however, after more learning, more agile maneuvers could be performed.

The development of systematic procedures to estimate the Lipschitz constant L as required by our approach, and to handle other constraints such as relative spacecraft translational velocity to ensure soft docking, avoiding thrusting into the target, etc. is left for future work. Additionally, we will consider extensions of the approach to more complex spacecraft configurations, such as the multibody underactuated spacecraft considered in [16], obstacle avoidance as in [17], 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.

References

[1] 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

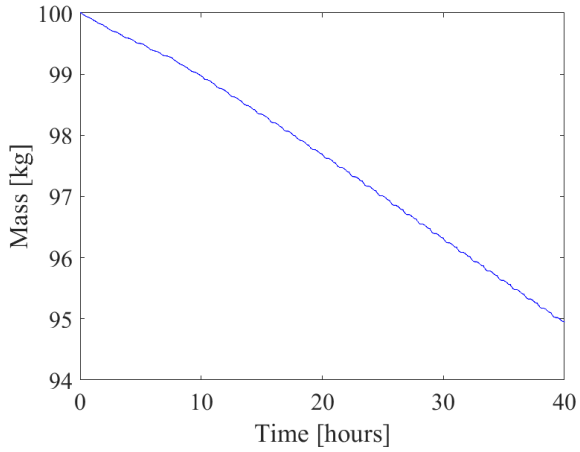


Fig. 17 Mass change during learning.

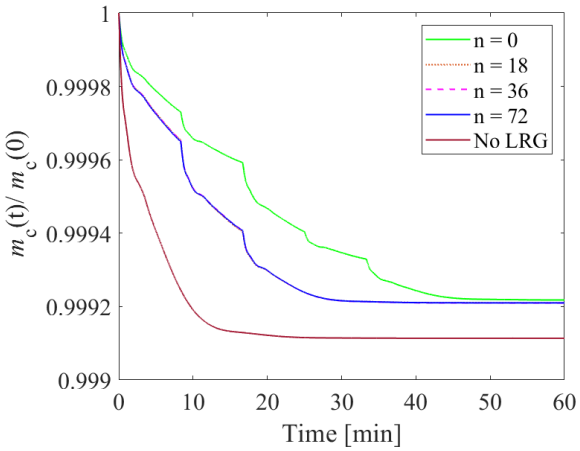


Fig. 18 Profile of fraction of mass to initial mass during maneuvers after learning.

https://doi.org/10.2514/1.A33872.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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)*, IEEE, 2019, pp. 934–940.

[11] 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)*, IEEE, 2020, pp. 943–949.

[12] 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.

[13] 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.

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

[15] 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*, 2021.

[16] 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.

[17] 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.