doi: 10.82124/CEAS-GNC-2026-038 CEAS-GNC-2026-038

EuroGNC Navigation Guidance Control CEAS Madrid, Spain May 5th-7th 2026
uc3m Universidad Carlos III de Madrid AIAA

Safe Deep Reinforcement Learning for Spacecraft Reorientation with Pointing Keep-Out Constraint

Juntang Yang Postdoc, University of Würzburg, Würzburg, Germany. juntang.yang@uni-wuerzburg.de

Mohamed Khalil Ben-Larbi Professor, University of Würzburg, Würzburg, Germany. khalil.ben-larbi@uni-wuerzburg.de

ABSTRACT

This paper implements deep reinforcement learning (DRL) with a safety filter for spacecraft reorientation control with a single pointing keep-out zone. A new state space representation is designed which includes a compact representation of the attitude constraint zone. A reward function is formulated to achieve the control objective while enforcing the attitude constraint. The soft actor-critic (SAC) algorithm is adopted to handle continuous state and action space. A curriculum learning approach is implemented for agent training. To guarantee the compliance of the attitude constraint, a control barrier function (CBF)-based safety filter is implemented for agent deployment. Simulation results demonstrate the effectiveness of the proposed state space presentation and the designed reward function. Monte Carlo simulations underscore that reward shaping alone cannot guarantee the safety during reorientation maneuver. In contrast, with the CBF-based safety filter, the constraint can be guaranteed during maneuvers.

Keywords: Deep Reinforcement Learning, Safe Reinforcement Learning, Spacecraft Attitude Control, Pointing Constraint, Safety Filter, Control Barrier Function

Nomenclature

A_3	=	3-by-3 submatrix of M_F
h	=	Control barrier function for κ
$\mathcal{H}$	=	Inner constraint set
$\mathcal{H}^\Delta$	=	Subset of $\mathcal{H}$ with margin Δ
$\mathbb{H}$	=	Set of quaternions
I	=	Moment of inertia of spacecraft
I_3	=	3-by-3 identity matrix
$M_2^+, M_2^-, M_3^+, M_3^-$	=	Constants related to $\ddot{\kappa}$
M_F	=	Matrix for formulation of keep-out zone constraint
$\bar{n}_F$	=	Unit central direction vector of keep-out zone
$P_{\text{f-zone}}$	=	Keep-out zone penalty
p_h	=	Polynomial as upper bound evolution of h
p_κ	=	Polynomial as upper bound evolution of κ
$\boldsymbol{q}$	=	Quaternion

$\boldsymbol{q}^*$	=	Conjugate of quaternion
$\boldsymbol{q}_d$	=	Desired attitude in unit quaternion
$\boldsymbol{q}_e$	=	Relative attitude in unit quaternion
q_0	=	Scalar part of quaternion $\boldsymbol{q}$
q_{e0}	=	Scalar part of quaternion $\boldsymbol{q}_e$
$\bar{q}$	=	Vector part of quaternion $\boldsymbol{q}$
$\mathcal{Q}$	=	Safe set
$\mathcal{Q}^\delta$	=	Subset of $\mathcal{Q}$ with margin δ
r, r_1	=	Reward functions
$\bar{r}_F$	=	Unit boresight vector of instrument
$\mathbb{R}$	=	Set of real numbers
s	=	State observation
$\mathbb{S}^3$	=	Set of unit quaternions
t	=	time
t_k	=	Current time step
t_{k-1}	=	Previous time step
T	=	Time step of discretization
$\mathbb{U}$	=	Set of allowable control inputs
$\mathbb{U}_z$	=	Set of guaranteed safe control inputs
$\bar{x}$	=	Full state, $\bar{x} = (\boldsymbol{q}, \bar{\omega})$
$\mathcal{Z}$	=	Robust inner constraint set
α, β	=	Positive constants for keep-out zone penalty
δ	=	Margin (small and positive) for $\mathcal{Q}^\delta$
Δ	=	Margin (small and positive) for $\mathcal{H}^\Delta$
θ	=	Angle between boresight vector $\bar{r}_F$ and avoiding direction $\bar{n}_F$
θ_F	=	Half angle of keep-out zone
θ_{margin}	=	Safety margin angle, $\theta_{\text{margin}} = \theta - \theta_F$
κ	=	Keep-out zone constraint function
μ	=	Parameter for control barrier function h
$\bar{\tau}$	=	Control torque
$\bar{\tau}_{\max}$	=	Max available control torque
$\bar{\tau}_{\text{RL}}$	=	Control torque output by RL agent
$\bar{\tau}_{\text{safe}}$	=	Control torque output by safety filter
$\bar{\omega}$	=	Angular rate
$\bar{\omega}^\times$	=	Skew matrix defined based on $\bar{\omega}$
$\boldsymbol{\omega}$	=	Quaternion form of $\bar{\omega}$
ϕ	=	Attitude error angle, $\phi = 2\arccos(q_{e0})$
ψ	=	Certain component of $\ddot{\kappa}$ (2nd time derivative of κ) under no disturbances
$\Delta\bar{n}_F$	=	Unit relative avoiding direction, $\Delta\bar{n}_F = \frac{\bar{n}_F - \bar{r}_F}{\|\bar{n}_F - \bar{r}_F\|}$
Δt	=	Arbitrary positive time increment
$\Delta\bar{\tau}$	=	Change of control torque, $\Delta\bar{\tau}(t_k) = \bar{\tau}(t_k) - \bar{\tau}(t_{k-1})$
Superscript B	=	For variable expressed in body frame
Superscript I	=	For variable expressed in inertial frame
$\otimes$	=	Quaternion multiplication

1 Introduction

Spacecraft reorientation maneuvers are often required to comply with pointing constraints to prevent sensitive onboard instruments (e.g., optical cameras) from exposure to bright celestial objects like the Sun.

It is challenging to control spacecraft reorientation under such constraints. There are different approaches for spacecraft constrained reorientation, examples of which include attitude planning methods [1, 2], nonlinear model predictive control (MPC)-based methods [3], and artificial potential field (APF)-based controllers [4, 5]. Attitude planning and nonlinear MPC-based methods are computationally intensive and therefore challenging for real-time application. Despite their computational efficiency, APF-based controllers usually suffer from becoming trapped in local minima.

Deep reinforcement learning (DRL) offers a promising alternative by combining the computational efficiency of a trained policy with the ability to handle complex, nonlinear dynamics. This makes it well suitable for onboard constrained attitude control. Although initial training is resource-intensive, the resulting agent can generate optimal control commands in real time based on state observations. While DRL has been successfully applied to unconstrained attitude control [6–9], its extension to pointing-constrained scenarios remains limited. Recent studies have begun to address this gap. For instance, Jiang et al. [10] used a Deep Q-Network (DQN) for attitude maneuver planning under forbidden and boundary constraints with a discretized action space, which potentially limits control precision in continuous dynamics. Cai et al. [11] applied the Deep Deterministic Policy Gradient (DDPG) algorithm to a formation flying problem under multiple constraints; however, their state-space design lacked explicit information about the constraint zones, limiting its adaptability.

A primary challenge in deploying RL for safety-critical systems is the lack of safety guarantees. To address this, the field of safe RL has developed various approaches to ensure operational safety [12]. One prominent strategy involves augmenting a performance-driven RL policy with a separate safety filter (SF) [13]. At runtime, the safety filter checks the actions proposed by the agent. If an action is deemed safe, it is executed; if not, the filter overrides it with a safe alternative. While safe RL has seen application in domains like autonomous driving [14], robotics [15], and spacecraft tasking [16, 17], its potential for ensuring safety in constrained spacecraft attitude control remains largely unexplored.

Motivated by these research gaps, this paper implements DRL for spacecraft reorientation control with a single pointing keep-out constraint in the framework of safe RL. The soft actor-critic (SAC) algorithm [18] is adopted to handle continuous state and action spaces. The core of our approach includes a state space designed with an explicit and compact representation of the attitude constraint zone, and a tailored reward function to encourage constraint-compliant behavior. To improve training efficiency, a curriculum learning strategy [19, 20] is employed, which progressively increases task difficulty. Crucially, to ensure operational safety, a control barrier function (CBF)-based safety filter [21] is incorporated during agent deployment to guarantee that all maneuvers avoid the keep-out zone.

The remainder of this paper is organized as follows: Section 2 details the methodology, Section 3 presents and discusses the simulation results, and Section 4 concludes this paper.

2 Preliminaries

2.1 Rotational kinematics and dynamics

In this paper the set of quaternions is denoted by $\mathbb{H}$ and the set of unit quaternions is denoted by $\mathbb{S}^3$. The rotational kinematic and dynamic equations of a rigid spacecraft are described as [22]

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega} \tag{1}$$

$$I\dot{\bar{\omega}} = -\bar{\omega}^{\times} I\bar{\omega} + \bar{\tau} \tag{2}$$

where $\boldsymbol{q} = \left[q_0, \bar{q}^T\right]^T \in \mathbb{S}^3$ is the attitude of spacecraft in unit quaternion with $q_0 \in \mathbb{R}$ and $\bar{q} = [q_1, q_2, q_3]^T \in \mathbb{R}^3$ as the scalar and vector parts, respectively, $\boldsymbol{\omega} = \left[0, \bar{\omega}^T\right]^T \in \mathbb{H}$ is the vector quaternion form of the angular velocity $\bar{\omega} = [\omega_1, \omega_2, \omega_3]^T \in \mathbb{R}^3$, $I \in \mathbb{R}^{3\times3}$ is the moment of inertia of the spacecraft, $\bar{\tau} = [\tau_1, \tau_2, \tau_3]^T \in \mathbb{R}^3$ is the control torque. Note that both the angular velocity $\bar{\omega}$ and the control torque $\bar{\tau}$ are expressed in the body frame. The skew matrix $\bar{\omega}^\times \in \mathbb{R}^{3\times3}$ is defined as

$$\bar{\omega}^\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{3}$$

$\otimes$ denotes quaternion multiplication. Given two quaternions $\boldsymbol{a} = \left[a_0,\ \bar{a}^T\right]^T \in \mathbb{H}$ and $\boldsymbol{b} = \left[b_0,\ \bar{b}^T\right]^T \in \mathbb{H}$, the quaternion multiplication between $\boldsymbol{a}$ and $\boldsymbol{b}$ is defined as

$$\boldsymbol{a} \otimes \boldsymbol{b} = \begin{bmatrix} a_0 b_0 - \bar{a} \cdot \bar{b} \\ a_0 \bar{b} + b_0 \bar{a} + \bar{a} \times \bar{b} \end{bmatrix}$$

The conjugate of quaternion $\boldsymbol{a}$ is defined as

$$\boldsymbol{a}^* = \left[a_0,\ -\bar{a}^T\right]^T$$

2.2 Pointing keep-out zone

A pointing keep-out zone (or forbidden zone) defines an inertial direction that must be avoided by sensitive onboard payloads like telescopes. As shown in Fig. 1, the constraint is geometrically represented by a cone, parameterized by its central direction vector $\bar{n}_F$ and half-angle θ_F, which the telescope's boresight vector $\bar{r}_F$ is forbidden to enter. Note that here both $\bar{n}_F$ and $\bar{r}_F$ are unit vectors.

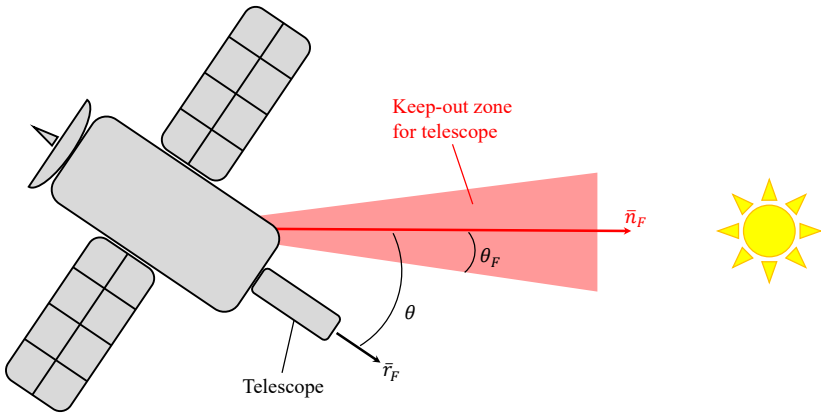


Fig. 1 Spacecraft sketch with keep-out zone for telescope.

The mathematical formulation of the requirement of the forbidden zone is written as

$$\bar{r}_F \cdot \bar{n}_F - \cos\theta_F < 0 \tag{4}$$

which requires the angle between $\bar{r}_F$ and $\bar{n}_F$ (illustrated as θ in Fig. 1) to be larger than θ_F.

The formulation in Eq. (4) can be rewritten as [23]

$$\boldsymbol{q}^T M_F \boldsymbol{q} < 0 \tag{5}$$

with

$$M_F = \begin{bmatrix} \bar{r}_F^B \cdot \bar{n}_F^I - \cos\theta_F & \left(\bar{r}_F^B \times \bar{n}_F^I\right)^T \\ \bar{r}_F^B \times \bar{n}_F^I & A_3 \end{bmatrix} \tag{6}$$

where $A_3 = \bar{r}_F^B \left(\bar{n}_F^I\right)^T + \bar{n}_F^I \left(\bar{r}_F^B\right)^T - \left(\bar{r}_F^B \cdot \bar{n}_F^I + \cos\theta_F\right) I_3$ with $I_3 \in \mathbb{R}^{3\times3}$ being an identity matrix. The superscripts B and I indicate variables expressed in the body frame and the inertial frame, respectively.

2.3 Reinforcement learning

Reinforcement learning (RL) is a machine learning paradigm concerned with how an agent ought to take actions in an environment so as to maximize a notion of cumulative reward [24, 25]. In this learning process, the agent interacts with its environment iteratively. Under the assumption of the Markov property, RL problems are conventionally framed as Markov decision process (MDP). An MDP is formally defined by the tuple (S, A, R, P, γ), where S and A are the state and action spaces, respectively. The function $R : S \times A \to \mathbb{R}$ specifies the reward function, P defines the state transition function, and $\gamma \in [0, 1)$ is the discount factor. The agent's learning cycle consists of selecting an action $a \in A$ based on the observation of the current state $s \in S$ which results in a transition to a successor state s' and the receipt of a reward $R(s, a)$. The objective in RL is to identify a policy that maximizes the expected cumulative discounted return over the decision horizon.
The RL framework for spacecraft reorientation control subject to a single pointing keep-out constraint is formalized as follows.

State space: The state observation, $s(t_k)$, is defined as

$$s(t_k) = \left[\boldsymbol{q}_e(t_k), \bar{\omega}(t_k), \bar{r}_F^B(t_k), \theta_{\text{margin}}(t_k), \theta(t_k), \Delta\bar{n}_F^B(t_k), q_{e0}(t_{k-1})\right] \tag{7}$$

where t_k and t_{k-1} indicate the current and previous time step, respectively. The state observation $s(t_k)$ contains the following information: the attitude error $\boldsymbol{q}_e(t_k)$ (calculated as $\boldsymbol{q}_e(t_k) = \boldsymbol{q}_d^* \otimes \boldsymbol{q}(t_k)$ with $\boldsymbol{q}_d^*$ being the conjugate of the desired attitude $\boldsymbol{q}_d$), the angular rate $\bar{\omega}(t_k)$, the payload boresight unit vector in the body frame $\bar{r}_F^B(t_k)$, the angle $\theta(t_k)$ between the boresight vector and the avoid vector as illustrated in Fig. 1, the safety margin $\theta_{\text{margin}}(t_k) = \theta(t_k) - \theta_F$, the relative direction vector $\Delta\bar{n}_F^B(t_k)$ (expressed in the body frame), and the previous quaternion scalar component $q_{e0}(t_{k-1})$ providing temporal information. Note that $\theta_{\text{margin}}(t_k) > 0$ must be satisfied to ensure the boresight vector stays outside the keep-out zone.

The relative direction vector $\Delta\bar{n}_F^B(t_k)$ informs the agent of the correct avoidance maneuver direction and is defined as

$$\Delta\bar{n}_F^B(t_k) = \frac{\bar{n}_F^B(t_k) - \bar{r}_F^B(t_k)}{\|\bar{n}_F^B(t_k) - \bar{r}_F^B(t_k)\|} \tag{8}$$

where $\bar{n}_F^B(t_k)$ is the unit vector $\bar{n}_F$ (see Fig. 1) expressed in the body frame.

Action space: the action space is defined by the body-frame control torque $\bar{\tau} \in \mathbb{R}^3$, which is normalized to $[-1, 1]$ during training.

Reward function: The reward function $r(t_k)$ at the current time step is formulated based on the design by Elkins et al. [6] as follows:

$$r(t_k) = \begin{cases} r_1(t_k) + 9, & \phi(t_k) \leq 0.25^\circ \\ r_1(t_k), & \text{otherwise} \end{cases} \tag{9}$$

$$r_1(t_k) = \begin{cases} e^{\frac{-\phi(t_k)}{0.14*2\pi}} - 0.05\frac{\|\bar{\tau}(t_k)\|}{\|\bar{\tau}_{\max}\|} - 0.005\|\Delta\bar{\tau}(t_k)\| - \text{P}_{\text{f-zone}}(t_k), & q_{e0}(t_k) > q_{e0}(t_{k-1}) \\ e^{\frac{-\phi(t_k)}{0.14*2\pi}} - 0.05\frac{\|\bar{\tau}(t_k)\|}{\|\bar{\tau}_{\max}\|} - 0.005\|\Delta\bar{\tau}(t_k)\| - \text{P}_{\text{f-zone}}(t_k) - 1, & \text{otherwise} \end{cases} \tag{10}$$

where $\phi(t_k) = 2\arccos(q_{e0}(t_k))$ is the attitude error angle, $\bar{\tau}_{\max}$ is the max available control torque, and $\Delta\bar{\tau}(t_k) = \bar{\tau}(t_k) - \bar{\tau}(t_{k-1})$ is the change of the torque. The keep-out zone penalty is defined as:

$$\mathrm{P}_{\text{f-zone}}(t_k) = \begin{cases} \beta, & \theta_{\text{margin}}(t_k) \le 0 \\ \beta e^{-\alpha\theta_{\text{margin}}(t_k)}, & \text{otherwise} \end{cases} \tag{11}$$

where α and β are positive constants, and $\theta_{\text{margin}}(t_k)$ is in radians.

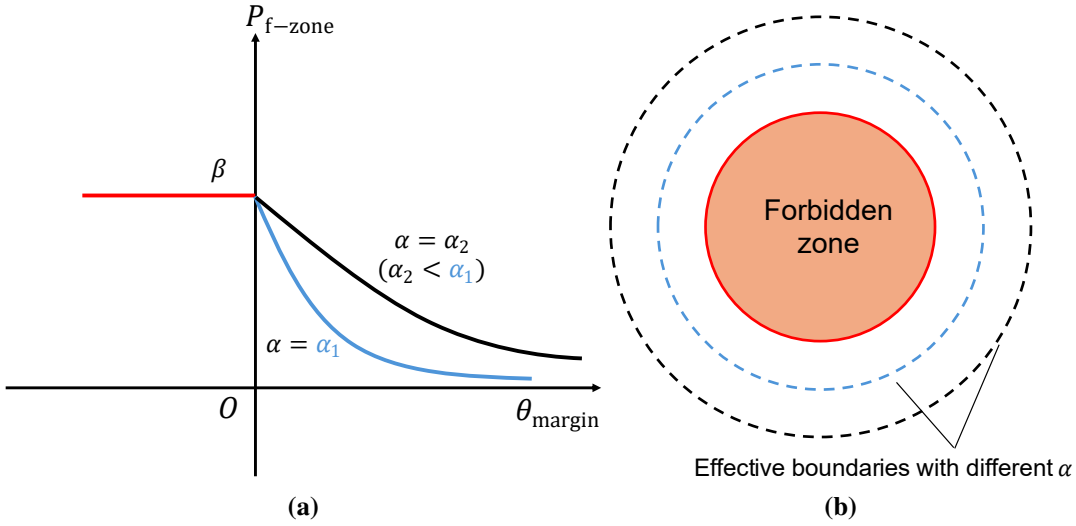


Fig. 2 Illustration of keep-out zone penalty $\mathrm{P}_{\text{f-zone}}$

The reward function in Eq. (9) is composed of two primary elements: a base reward $r_1(t_k)$ and an extra reward of 9 when achieving the desired pointing accuracy of 0.25°. The base reward $r_1(t_k)$ in Eq. (10) consists of weighted terms promoting attitude convergence, minimizing control effort and its change, and penalizing violations of the keep-out zone constraint via the penalty function $\mathrm{P}_{\text{f-zone}}(t_k)$. Figure 2 visualizes the keep-out zone penalty function with two different α values (see Fig. 2a) and illustrates how the parameter α affects the effective boundary (indicated as the dashed lines in Fig. 2b), outside which the keep-out zone penalty is below a defined small threshold value and can be ignored. Increasing α narrows this boundary, concentrating the penalty closer to the zone.

State transition: The state transition is governed by the rotational kinematics and dynamics, as given by Eqs. (1) and (2).

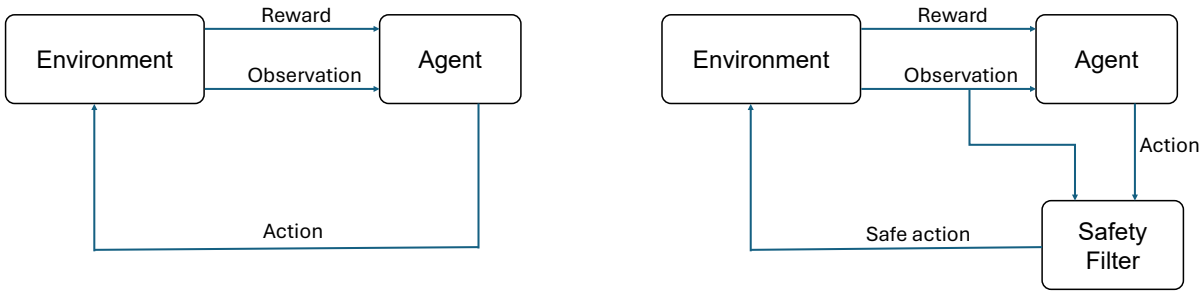
2.4 RL with safety filter

Figure 3 compares a standard RL framework with a safety-augmented variant. The key difference is that the latter interposes a safety filter between the agent and the environment (Fig. 3b) to modify actions for safety, unlike the direct application in the standard framework (Fig. 3a).

In this paper, we implement a control barrier function (CBF)-based safety filter developed by Breeden and Panagou [21]. The basic idea of the safety filter is as follows: at each time step t_k, the safety filter finds a control torque $\bar{\tau}_{\text{safe}}(t_k)$ closest to the agent action, $\bar{\tau}_{RL}(t_k)$, while satisfying the control limit and safety constraints by solving the following optimization problem

$$\bar{\tau}_{\text{safe}}(t_k) = \arg\min_{\bar{\tau}(t_k)\in\mathbb{U}\cap\mathbb{U}_z} \|\bar{\tau}(t_k) - \bar{\tau}_{RL}(t_k)\|^2 \tag{12}$$

where $\mathbb{U}$ is the set of allowable control inputs, and $\mathbb{U}_z$ is the set of control inputs guaranteeing the compliance of the keep-out zone constraint.



(a) Standard RL **(b) RL with safety filter**

Fig. 3 Standard RL and RL with safety filter

The control set $\mathbb{U}$ is derived from the operational limits of the actuators. The calculation of $\mathbb{U}_z$ is briefly explained in the remainder of this section, with slight adaptation of the notation used in [21]. Note that, for the sake of brevity, the dependency of $\boldsymbol{q}$ and $\bar{\omega}$ on the time t is omitted in the following explanation.

The safe set $\mathcal{Q}(t)$ is defined as

$$\mathcal{Q}(t) \triangleq \{(\boldsymbol{q}, \bar{\omega}) | \kappa(t, \boldsymbol{q}) < 0\} \tag{13}$$

with

$$\kappa(t, \boldsymbol{q}) \triangleq \boldsymbol{q}^T M_F \boldsymbol{q} \tag{14}$$

The states in $\mathcal{Q}$ ensures the satisfaction of the keep-out zone constraint at the current time instant.

Breeden and Panagou [21] utilize the CBF theory to ensure state trajectories always inside the safe set and they choose the CBF for κ as

$$h(t, \boldsymbol{q}) = \kappa(t, \boldsymbol{q}) + \frac{\dot{\kappa}(t, \boldsymbol{q}) |\dot{\kappa}(t, \boldsymbol{q})|}{2\mu} \tag{15}$$

with the parameter $\mu \in (0, 0.0025]$. Note that the CBF with a larger μ is less conservative.

The inner constraint set $\mathcal{H}(t)$ for the keep-out zone is defined as

$$\mathcal{H}(t) \triangleq \{(\boldsymbol{q}, \bar{\omega}) | h(t, \boldsymbol{q}) \leq 0\} \tag{16}$$

The state trajectories inside $\mathcal{H}$ cannot leave $\mathcal{H}$, i.e., $\mathcal{H}$ is a controlled-invariant set [21].

In order to account for input constraints, disturbances, and controller sampling, the robust inner constraint set is defined as [21]

$$\mathcal{Z}(t) = \mathcal{Q}^{\delta}(t) \cap \mathcal{H}^{\Delta}(t) \tag{17}$$

where $\mathcal{Q}^{\delta}(t)$ and $\mathcal{H}^{\Delta}(t)$ are the subset of $\mathcal{Q}(t)$ with margin δ and the subset of $\mathcal{H}(t)$ with margin Δ, respectively, defined as

$$\mathcal{Q}^{\delta}(t) \triangleq \{(\boldsymbol{q}, \bar{\omega}) | \kappa(t, \boldsymbol{q}) \leq -\delta\} \tag{18}$$

$$\mathcal{H}^{\Delta}(t) \triangleq \{(\boldsymbol{q}, \bar{\omega}) | h(t, \boldsymbol{q}) \leq -\Delta\} \tag{19}$$

with δ and Δ as small positive parameters.

For control based on sampled data, the robust inner constraint set $\mathcal{Z}(t)$ with suitable δ and Δ can ensure the safety between time steps [21].

The requirements in Eqs. (18) and (19) are further transformed as the requirements on the upper bound evolution of κ (denoted as p_κ) and the upper bound evolution of h (denoted as p_h), which is expressed as

$$p_\kappa(t, \bar{x}, \bar{\tau}, \Delta t) \leq -\delta \tag{20}$$

$$p_h(t, \bar{x}, \bar{\tau}, \Delta t) \leq -\Delta \tag{21}$$

with p_κ and p_h as polynomials in Δt (an arbitrary positive time increment) defined as follows

$$p_\kappa(t, \bar{x}, \bar{\tau}, \Delta t) \triangleq \kappa(t, \bar{x}) + \dot{\kappa}(t, \bar{x})\Delta t + \frac{1}{2}\psi(t, \bar{x}, \bar{\tau})\,(\Delta t)^2 + \frac{1}{2}M_2^+\,(\Delta t)^2 + \frac{1}{6}M_3^+\,(\Delta t)^3 \tag{22}$$

$$p_h(t, \bar{x}, \bar{\tau}, \Delta t) \triangleq p_\kappa(t, \bar{x}, \bar{\tau}, \Delta t) + \frac{1}{2\mu}\mathrm{ssq}\left(\dot{\kappa}(t, \bar{x})\Delta t + \psi(t, \bar{x}, \bar{\tau})\Delta t + M_2^+\Delta t + \frac{1}{2}M_3^+\,(\Delta t)^2\right) \tag{23}$$

where $\bar{x} = (\boldsymbol{q}, \bar{\omega})$, $\psi(t, \bar{x}, \bar{\tau})$ is the certain component of $\ddot{\kappa}$ under no disturbances, the constant M_2^+ represents the upper bound on the uncertainty in $\ddot{\kappa}$ because of unknown disturbances, the constant M_3^+ describes the uncertainty in the evolution of $\psi(t, \bar{x}, \bar{\tau})$ between time steps due to both the zero-order-hold (ZOH) sampling and disturbances. $\mathrm{ssq}(\lambda) \triangleq \lambda|\lambda|$ for brevity. Note that the dependency of $\bar{x}$ and $\bar{\tau}$ on the time is omitted for the sake of brevity.

Based on p_κ and p_h, $\mathbb{U}_z$ in Eq. (12) is defined as

$$\mathbb{U}_z = \left\{\bar{\tau} \in \mathbb{R}^3 | p_\kappa(t_k, \bar{x}, \bar{\tau}, T) \leq -\delta \text{ and } p_h(t_k, \bar{x}, \bar{\tau}, T) \leq -\Delta\right\} \tag{24}$$

where t_k is the sample time and T is the time step of discretization. Note that both $p_\kappa(t_k, \bar{x}, \bar{\tau}, T) \leq -\delta$ and $p_h(t_k, \bar{x}, \bar{\tau}, T) \leq -\Delta$ in Eq. (24) can be encoded in a quadratic program (QP). As for the calculation of $\mathbb{U}_z$, readers are referred to [21] for more details.

2.5 Agent training

This work employs the SAC algorithm [18] for agent training due to its suitability for continuous action spaces and strong exploration capabilities. As an off-policy actor-critic method, SAC offers high sample efficiency. SAC maximizes an entropy-augmented objective, which balances task performance with exploration by encouraging stochastic behavior. The implementation is based on the Stable-Baselines3 library [26].

For agent training, the simulation runs with a time step $T = 0.1$ s over episodes of 100 s duration. For attitude regulation scenarios, the initial angular rate is set as zero with small random perturbations. The spacecraft's moment of inertia is fixed at

$$I = \begin{bmatrix} 60 & 5 & 1 \\ 5 & 50 & 2 \\ 1 & 2 & 70 \end{bmatrix} \mathrm{kg} \cdot \mathrm{m}^2$$

The boresight vector is $[1, 0, 0]^T$. The torque limit for each axis is 2 Nm. The keep-out zone penalty parameters are $\beta = 10$, $\alpha = 66$.

Agent training followed a two-phase curriculum learning strategy. In Phase 1, the agent learned a baseline attitude control policy without keep-out zones. The initial attitude error was randomized, with the maximum initial deviation angle progressively increased from 25° to 180°. The resulting policy and experience replay buffer from this phase were saved for initialization in further training. In Phase 2, the pre-trained agent was fine-tuned with the keep-out zone penalty activated. For each episode, a single keep-out zone was generated by placing its center vector, $\bar{n}_F$, along the shortest rotation path between the initial attitude (with randomized error between 80° and 180°) and the target. The cone half angle, θ_F,

was randomized between 15° and 30°. This placement ensures the zone intersects the agent's likely path since the policy from Phase 1 generates short-path rotations. The maximum initial deviation angle was again gradually increased from 80° to 180°.

Agent training was conducted on a desktop PC (Intel i7-14700 KF, 32 GB RAM, NVIDIA RTX 4070 GPU). The default SAC configuration from Stable-Baselines3 was used, which features an MlpPolicy with ReLU activations and a two-layer fully connected network architecture with 256 units per layer. The hyperparameters used for training are provided in Table 2.

Table 2 SAC hyperparameters for agent training

Parameter	Value
Batch size	256
Buffer size	10^6
Discount factor (γ)	0.99
Entropy coefficient	Auto
Learning rate (without F-zone)	0.0001
Learning rate (with F-zone)	0.0001
Soft update coefficient	0.005

3 Numerical Results

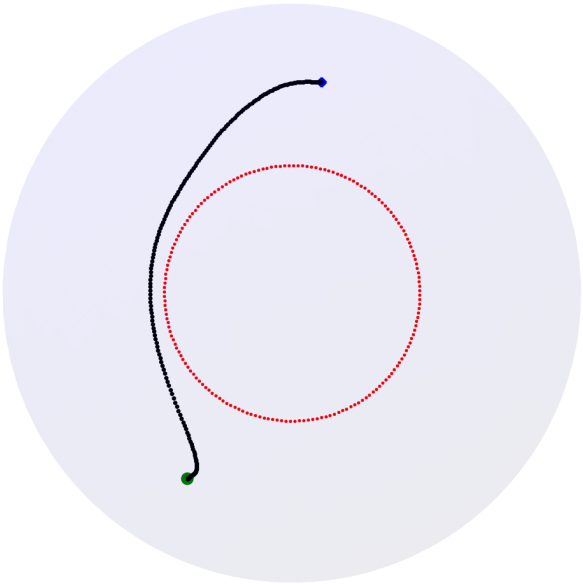
A reorientation scenario with an initial deviation angle of 100° and one F-zone was tested with an agent trained in Phase 2. The parameters are detailed in Table 3. An example result under the agent is

Table 3 Parameters for example simulation

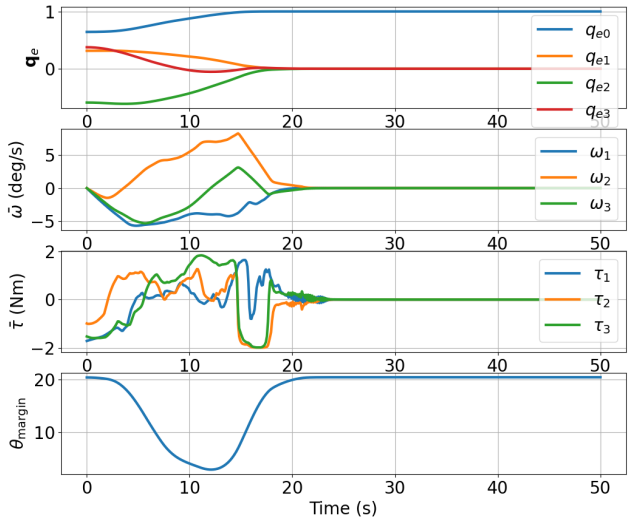
Parameter	Value
Avoid vector ($\bar{n}_F^I$)	$[0.703, 0.263, 0.661]^T$
Boresight vector ($\bar{r}_F^B$)	$[1, 0, 0]^T$
Half angle (θ_F)	25 deg
Initial relative attitude ($\boldsymbol{q}_e$)	$[0.6428, 0.3138, -0.5892, 0.3757]^T$
Initial angular rate ($\bar{\omega}$)	$[-5.7, -1.1, -9.9]^T * 10^{-4}$ deg/s
Target attitude	$[1, 0, 0, 0]^T$

presented in Fig. 4. Figure 4a shows the boresight vector's trace on the unit sphere. The trace (black) starts at the initial pointing (blue cross) and ends at the target (green point), successfully avoiding the keep-out zone bounded by the red circle. The corresponding time histories in Fig. 4b confirm that the spacecraft achieves the desired attitude as $\boldsymbol{q}_e$ converges to the identity quaternion, while maintaining a positive margin angle θ_{margin} throughout the maneuver. These results validate that the design of the state space and the reward function is effective for RL-based attitude control with a single pointing keep-out zone.

A Monte Carlo analysis comprising 10,000 simulations was conducted to statistically evaluate the performance of the best rewarded agent from Phase 2. Each scenario featured a random initial deviation angle between 80° and 180° and an initial angular rate uniformly distributed in the range $[-0.001, 0.001]$ °/s. The simulation uses the same time step, duration, and moment of inertia as in the training phase. Each test was evaluated based on the cumulative reward, settling time (defined as the



(a) Trace of boresight vector on unit sphere



(b) Time history of relative attitude, angular velocity, control torque, and θ_{margin}

Fig. 4 Example result under agent trained in Phase 2 (with one F-zone)

Table 4 Results of Monte Carlo simulation

Evaluation metrics	Results		
	Standard RL	RL with SF (μ = 0.0025)	RL with SF (μ = 0.0001)
Mean Reward	7281.91 ± 688.85	7232.96 ± 642.41	6372.58 ± 1327.31
Mean settling time* (sec)	27.81 ± 5.24	28.47 ± 5.30	37.21 ± 12.36
Mean control effort* (N^2m^2s)	76.02 ± 25.76	73.31 ± 24.21	68.65 ± 18.00
Mean control accuracy* (deg)	0.08 ± 0.04	0.08 ± 0.04	0.08 ± 0.04
Rate of non-settled	0.32%	0.22%	0.70%
Rate of violation	2.66%	0%	0%

*Non-settled samples are ignored.

time at which the attitude error enters and subsequently remains within the desired accuracy of 0.25 deg), control accuracy, and the control effort, defined as

$$E\left(t_{end}\right) = \int_0^{t_{end}} \|\bar{\tau}\|^2 dt$$

The results of Monte Carlo simulation using the framework of standard RL are summarized in Fig. 5a showing that the agent successfully reached the target orientation while complying with the constraint zone in approximately 97% of cases (blue samples). The failures (about 3%) were due to constraint violations (purple samples, 2.66%) and failure to converge within the simulation duration (orange samples, 0.32%). Note that the latter failures were assigned a settling time of 200 s for visualization. The detailed results of evaluation metrics are listed in Table 4 (2nd column). The insight from the Monte Carlo analysis is that incorporating the keep-out zone penalty into the reward function alone does not guarantee full constraint compliance, as evidenced by the 2.66% violation rate. This underscores the need for additional safety assurances when using the standard RL framework.

The same agent was used for a Monte Carlo simulation with the CBF-based safety filter in [21] in the loop. The parameters of the safety filter are presented in Table 5. Small M_2^+, M_2^-, M_3^+, and M_3^- are used since no disturbances are considered. For the least conservative CBF of the form in Eq. (15), the largest allowable parameter $\mu = 0.0025$ is used [21]. The calculation of δ and Δ is based on the setting of M_2^+, M_2^-, M_3^+, M_3^-, μ and the time step T (See Theorem 1 in [21] for more details).

Table 5 Parameters of CBF-based safety filter

Parameter	Value
T	0.1 s
M_2^+	1.64×10^{-5}
M_2^-	-1.64×10^{-5}
M_3^+	6.2×10^{-4}
M_3^-	-6.2×10^{-4}
μ	0.0025
δ	3.18×10^{-6}
Δ	3.18×10^{-6}

The Monte Carlo simulation results using the safe-RL framework are summarized in Fig. 5b. In contrast to the case using the standard RL framework, the violation of the keep-out zone is totally avoided with the safety filter in the loop. However, there are still non-settled cases accounting for 0.22%. These non-settled cases fall into two categories: simple biases and limit cycles around certain pointing other than the target pointing. Figures 6 and 7 show examples for both categories with detailed plots. Individual check of the non-settled cases show that they are caused by the agent itself instead of the safety filter, which implies that the agent from Phase 2 needs further training.

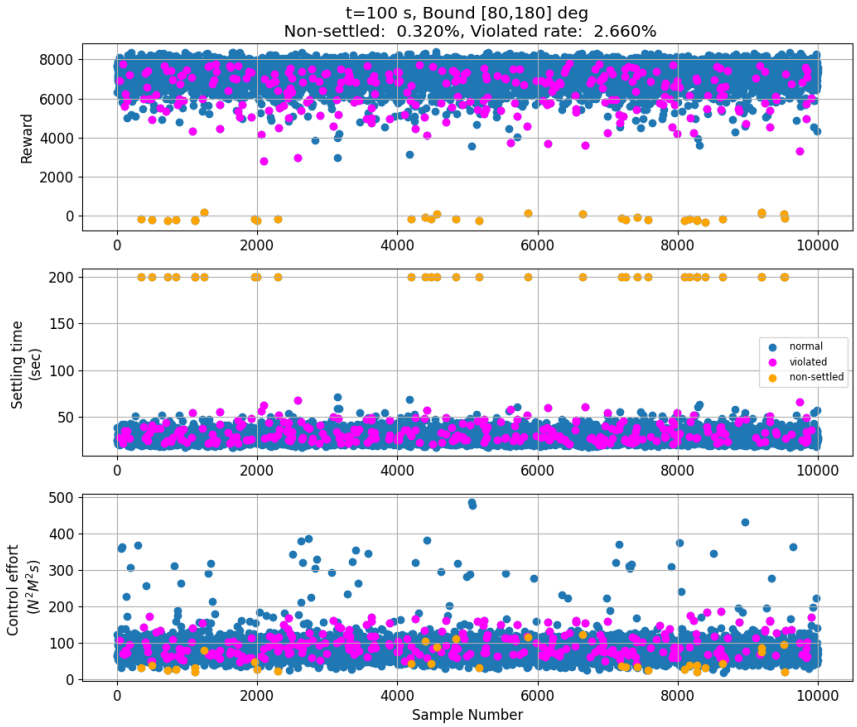
Table 4 (3rd column) shows the detailed results of evaluation metrics. By comparing the 2nd and the 3rd columns in Table 4, it is observed that the mean reward only reduces slightly from 7281.91 to 7232.96 and the mean settling time increases slightly from 27.81 s to 28.47 s. The mean control effort reduces (from 76.02 N^2m^2s to 73.31 N^2m^2s) and the mean control accuracy remains the same as 0.08 deg.

Figure 8 presents simulation examples illustrating how the safety filter changes the reorientation maneuver to ensure the keep-out zone avoidance. The black trace in Fig. 8a is from a simulation without the safety filter and the blue from a simulation with the safety filter ($\mu = 0.0025$). Figure 8b compares the time history of related data. It is observed that the safety filter with $\mu = 0.0025$ starts to change the maneuver only when the boresight vector approaches the keep-out zone to a certain angular distance.

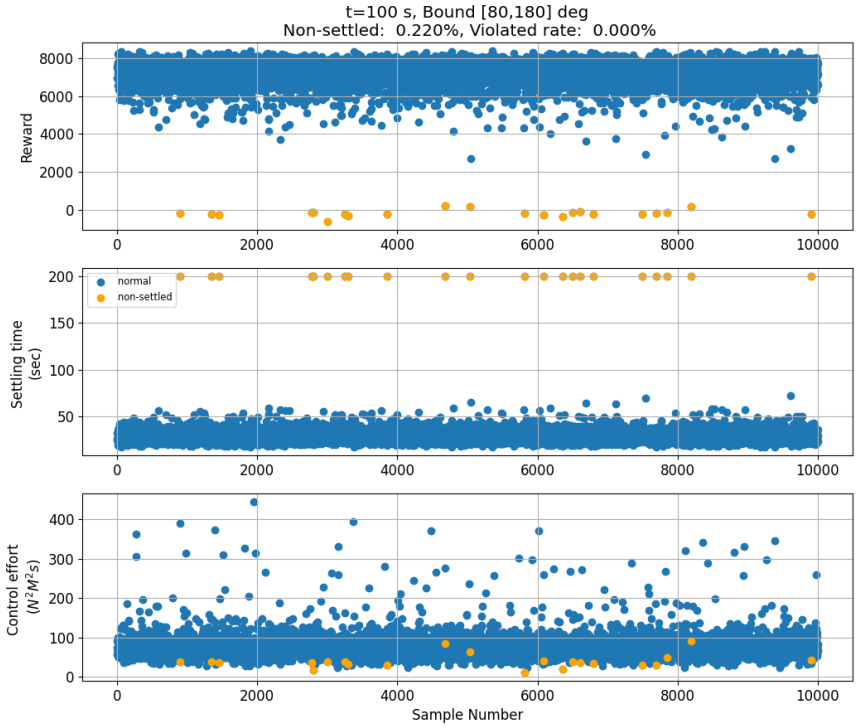
Simulations with different parameter settings for the CBF-based safety filter were performed. It turns out that the parameter μ has the most significant influence on the performance of the safety filter. Using a small μ that is not set suitably can lead to worse performance than that in cases without using a safety filter. The 4th column of Table 4 presents the result when using a safety filter with $\mu = 0.0001$, which is worse than that without a safety filter in terms of the mean reward, the mean settling time, and the rate of non-settled samples, even though the violation of the keep-out zone is avoided.

4 Conclusions

This paper proposed a safe DRL solution for spacecraft reorientation subject to a single pointing keep-out zone. The approach is characterized by a novel state space representation, a constraint-aware reward function, and training with the SAC algorithm and curriculum learning. To ensure operational safety, a safety filter was integrated into the action loop, implementing a safe reinforcement learning

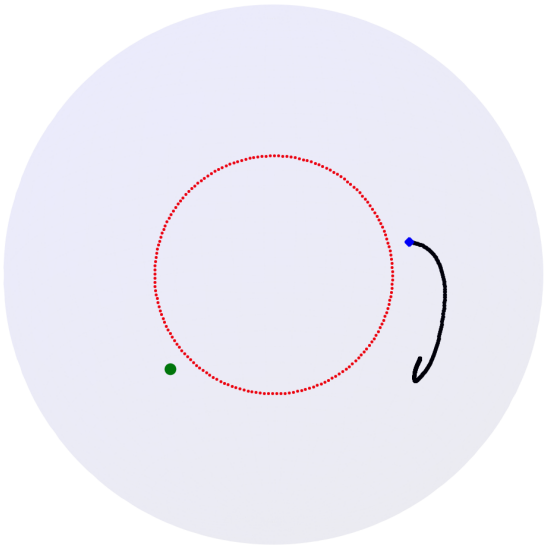


(a) Without using safety filter (namely, in standard RL framework)

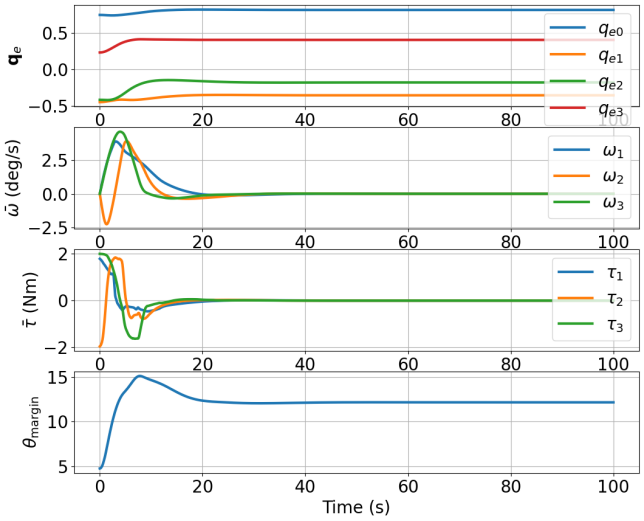


(b) Using safety filter (namely, in safe-RL framework)

Fig. 5 Monte Carlo simulation results (metrics vs sample number) under the best rewarded agent trained in Phase 2 (with one F-zone). The non-settled cases are assigned a settling time of 200 s for visualization.

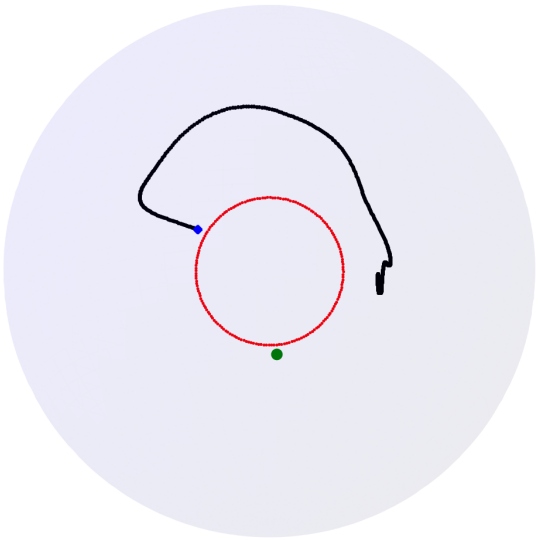


(a) Trace of boresight vector on unit sphere

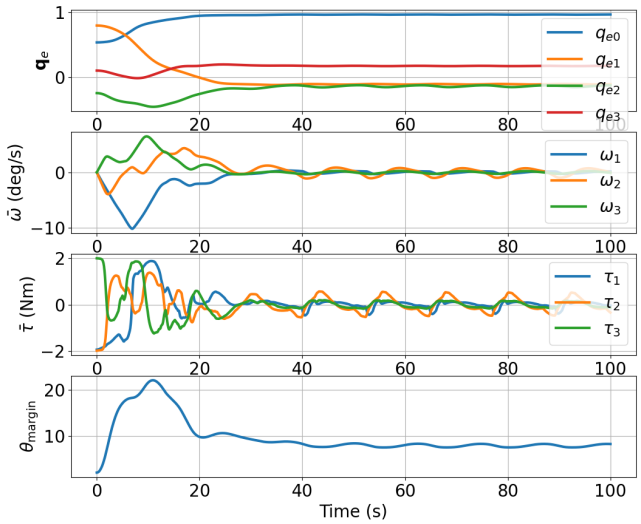


(b) Time history of relative attitude, angular velocity, control torque, and θ_{margin}

Fig. 6 Example of non-settled cases (bias)

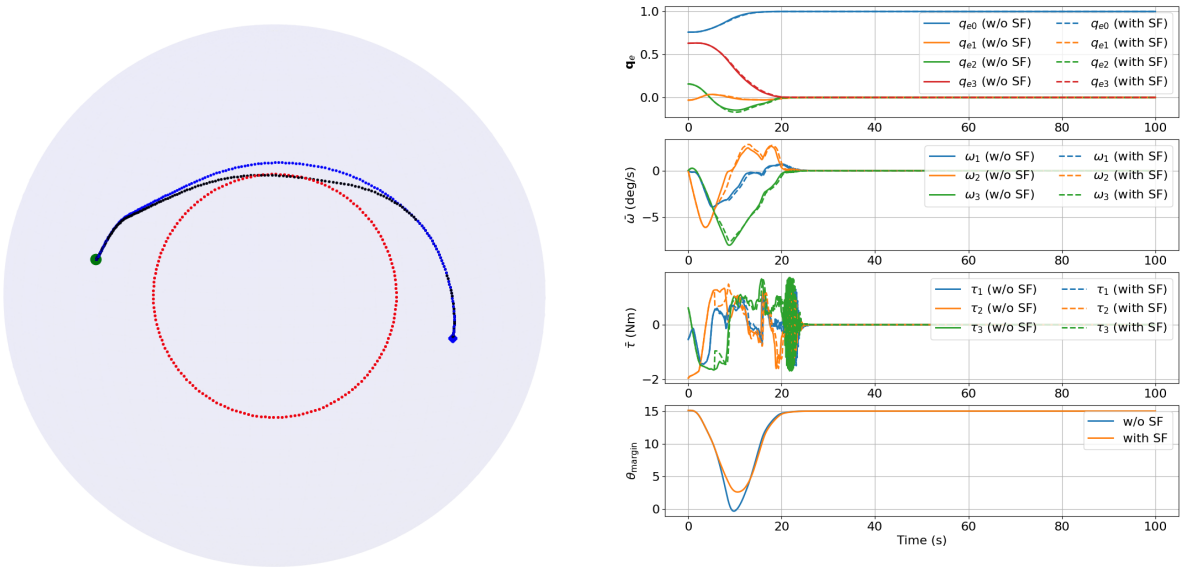


(a) Trace of boresight vector on unit sphere



(b) Time history of relative attitude, angular velocity, control torque, and θ_{margin}

Fig. 7 Example of non-settled cases (limit cycle)



(a) Traces of boresight vector on unit sphere (black: without SF; blue: with SF, $\mu = 0.0025$)

(b) Time history of relative attitude, angular velocity, control torque, and θ_{margin}

Fig. 8 Example results under agent with and without safety filter in the loop

strategy. The effectiveness of the state space design and the curriculum learning was validated through simulations. Monte Carlo simulations underscore that reward shaping alone cannot reliably prevent constraint violations. In contrast, the framework incorporating a safety filter successfully guarantee the constraint compliance. However, it is also observed that the agent from Phase 2 training may result in non-settled results under certain initial conditions. Further training of the agent is required to address this issue. The current work considers only one keep-out zone for a boresight vector. Future work will extend the current algorithm to account for multiple constraint zones by using long short-term memory (LSTM) networks to determine a single equivalent keep-out zone based on the configuration of the multiple keep-out zones and the current status of the spacecraft. Future work will also involve training and evaluating the agent using a high-fidelity simulation environment (e.g., Basilisk).

Appendix

Acknowledgments

This work was supported by JMU Seed Grant from University of Würzburg.

Declaration of Use of Artificial Intelligence

Deep reinforcement learning was used for spacecraft attitude control with a single pointing keep-out zone.

References

[1] E Feron, M Dahleh, E Frazzoli, and R Kornfeld. A randomized attitude slew planning algorithm for autonomous spacecraft. In *AIAA guidance, navigation, and control conference and exhibit*, page 4155, 2001.

[2] Henri C Kjellberg and E Glenn Lightsey. Discretized constrained attitude pathfinding and control for satellites. *Journal of Guidance, Control, and Dynamics*, 36(5):1301–1309, 2013.

[3] Rohit Gupta, Uroš V Kalabić, Stefano Di Cairano, Anthony M Bloch, and Ilya V Kolmanovsky. Constrained spacecraft attitude control on so (3) using fast nonlinear model predictive control. In *2015 American Control Conference (ACC)*, pages 2980–2986. IEEE, 2015.

[4] Unsik Lee and Mehran Mesbahi. Feedback control for spacecraft reorientation under attitude constraints via convex potentials. *IEEE Transactions on Aerospace and Electronic Systems*, 50(4):2578–2592, 2014.

[5] Juntang Yang, Yisheng Duan, Mohamed Khalil Ben-Larbi, and Enrico Stoll. Potential field-based sliding surface design and its application in spacecraft constrained reorientation. *Journal of Guidance, Control, and Dynamics*, 44(2):399–409, 2021.

[6] Jacob G Elkins, Rohan Sood, and Clemens Rumpf. Bridging reinforcement learning and online learning for spacecraft attitude control. *Journal of Aerospace Information Systems*, 19(1):62–69, 2022.

[7] Duozhi Gao, Haibo Zhang, Chuanjiang Li, and Xinzhou Gao. Satellite attitude control with deep reinforcement learning. In *2020 Chinese Automation Congress (CAC)*, pages 4095–4101. IEEE, 2020.

[8] K. Djebko, F. Puppe, S. Montenegro, T. Baumann, and M. Faisal. Learning attitude control. In *14th IAA Symposium on Small Satellites for Earth System Observation*, 2023.

[9] Snyoll Oghim, Junwoo Park, Hyochoong Bang, and Henzeh Leeghim. Deep reinforcement learning-based attitude control for spacecraft using control moment gyros. *Advances in Space Research*, 75(1):1129–1144, 2025.

[10] Shulei Jiang, Fanyu Zhao, Yuejie Chen, and Zhonghe Jin. Spacecraft attitude maneuver planning based on deep reinforcement learning under complex constraints. In *2023 9th International Conference on Control Science and Systems Engineering (ICCSSE)*, pages 61–67. IEEE, 2023.

[11] Yingkai Cai, Kay-Soon Low, and Zhaokui Wang. Reinforcement learning-based satellite formation attitude control under multi-constraint. *Advances in Space Research*, 74(11):5819–5836, 2024.

[12] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[13] Kim Peter Wabersich and Melanie N Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.

[14] David Isele, Alireza Nakhaei, and Kikuo Fujimura. Safe reinforcement learning on autonomous vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6. IEEE, 2018.

[15] Javier García and Diogo Shafie. Teaching a humanoid robot to walk faster through safe reinforcement learning. *Engineering Applications of Artificial Intelligence*, 88:103360, 2020.

[16] Islam Nazmy, Andrew Harris, Morteza Lahijanian, and Hanspeter Schaub. Shielded deep reinforcement learning for multi-sensor spacecraft imaging. In *2022 American Control Conference (ACC)*, pages 1808–1813. IEEE, 2022.

[17] Robert Reed, Hanspeter Schaub, and Morteza Lahijanian. Shielded deep reinforcement learning for complex spacecraft tasking. In *2024 American Control Conference (ACC)*, pages 2331–2337. IEEE, 2024.

[18] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[19] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

[20] Kashish Gupta, Debasmita Mukherjee, and Homayoun Najjaran. Extending the capabilities of reinforcement learning through curriculum: A review of methods and applications. *SN Computer Science*, 3(1):28, 2022.

[21] Joseph Breeden and Dimitra Panagou. Autonomous spacecraft attitude reorientation using robust sampled-data control barrier functions. *Journal of Guidance, Control, and Dynamics*, 46(10):1874–1891, 2023.

[22] F Landis Markley and John L Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*, chapter 2, 3, 7. Springer, New York, 2014.

[23] Unsik Lee and Mehran Mesbahi. Feedback control for spacecraft reorientation under attitude constraints via convex potentials. *IEEE Transactions on Aerospace and Electronic Systems*, 50(4):2578–2592, 2014.

[24] Richard S Sutton, Andrew G Barto, et al. Reinforcement learning: An introduction 2nd ed. *MIT press Cambridge*, 1(2):25, 2018.

[25] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE signal processing magazine*, 34(6):26–38, 2017.

[26] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of machine learning research*, 22(268):1–8, 2021.