# Deep Reinforcement Learning for Spacecraft Attitude Control During Atmospheric Re-Entry

**Alexander Fabisch, Melvin Laux, Mariela De Lucas Álvarez, Edoardo Caroselli, Julian Theis**

**Keywords:** Attitude Control, Spacecraft Re-Entry, Continuous Model-Free Reinforcement Learning, Task Scheduling, Out-of-Distribution Generalization

## Summary

Deep reinforcement learning has the potential to solve attitude control problems more adaptively, precisely, and robustly by handling nonlinear dynamics, uncertainties, and failure cases more effectively than traditional attitude control approaches. We explore reinforcement learning (RL) for attitude control in spacecraft re-entry. An industry-standard proportional–integral–derivative controller with gain scheduling serves as a strong baseline for model-free RL and hybrid controllers that combine these two approaches. We formalize the application in the RL framework to apply continuous, off-policy RL. State-of-the-art RL achieves comparable performance to traditional control approaches in this domain. However, its out-of-distribution generalization is not sufficient. Hence, we use dynamics randomization to introduce challenging task variations during training and enforce generalization in a predefined operational envelope. Finally, we assess the best obtained RL-based controllers with application-specific metrics to show superior performance in comparison to traditional controllers in the operational envelope, that is, hybrid controllers are able to track the angle of attack better and are more robust under variations of mass, inertia tensor, and flap actuator bandwidth.

## Contribution(s)

1. We apply state-of-the-art deep reinforcement learning algorithms for continuous control to the problem of attitude control during hypersonic re-entry of a spacecraft. This is a challenging tracking problem that requires gain scheduling for changing atmospheric conditions. We design a simple and effective reward function for attitude control and find that MR.Q excels without task-specific tuning.
   **Context:** We apply existing reinforcement learning algorithms (e.g., MR.Q (Fujimoto et al., 2025)). Prior work with older algorithms, less thorough evaluation, or more complicated reward functions in the application domain exists (e.g., Elkins et al., 2020a).
2. Although the best learned controller performs better than the industry-standard baseline controller under nominal conditions, we explore hybrid control architectures that combine the baseline controller with reinforcement learning and compare these against pure reinforcement learning to enhance out-of-distribution generalization.
   **Context:** Liu et al. (2022a) propose a similar hybrid controller that integrates reinforcement learning in a traditional controller. In comparison to this work, we use state-of-the-art algorithms and explicitly investigate generalization and robustness of learned controllers.
3. To improve robustness and generalization explicitly, we compare dynamics randomization and task scheduling approaches for reinforcement learning in this attitude control problem and determine the best training strategy. We find that the resulting policies considerably improve robustness in comparison to the baseline controller.
   **Context:** Uniform dynamics randomization has been used successfully for various applications in robotics (Antonova et al., 2017; Peng et al., 2018; Tan et al., 2018; Andrychowicz et al., 2020). Task scheduling approaches have been proposed in the context of deep reinforcement learning (Cho et al., 2024) and contextual policy search (Fabisch & Metzen, 2014).

# Deep Reinforcement Learning for Spacecraft Attitude Control During Atmospheric Re-Entry

**Alexander Fabisch**[1], **Melvin Laux**[1], **Mariela De Lucas Álvarez**[1], **Edoardo Caroselli**[2], **Julian Theis**[2]
{alexander.fabisch,melvin.laux,mariela.de_lucas_alvarez}@dfki.de,
{edoardo.caroselli,julian.theis}@airbus.com

[1]**Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI GmbH), Germany**
[2]**Airbus Defence and Space GmbH, Germany**

## Abstract

Deep reinforcement learning has the potential to solve attitude control problems more adaptively, precisely, and robustly by handling nonlinear dynamics, uncertainties, and failure cases more effectively than traditional attitude control approaches. We explore reinforcement learning (RL) for attitude control in spacecraft re-entry. An industry-standard proportional–integral–derivative controller with gain scheduling serves as a strong baseline for model-free RL and hybrid controllers that combine these two approaches. We formalize the application in the RL framework to apply continuous, off-policy RL. State-of-the-art RL achieves comparable performance to traditional control approaches in this domain. However, its out-of-distribution generalization is not sufficient. Hence, we use dynamics randomization to introduce challenging task variations during training and enforce generalization in a predefined operational envelope. Finally, we assess the best obtained RL-based controllers with application-specific metrics to show superior performance in comparison to traditional controllers in the operational envelope, that is, hybrid controllers are able to track the angle of attack better and are more robust under variations of mass, inertia tensor, and flap actuator bandwidth.

## 1 Introduction

Deep reinforcement learning (RL) for continuous control (e.g., Lillicrap et al., 2016) has the potential to handle nonlinear dynamics and failure cases more easily and to be more robust against and adaptable to unforeseen conditions than other attitude control approaches (see, e.g., Elkins et al., 2020b; Bernini et al., 2024; Liu et al., 2022a). However, the introduction of RL into a safety-critical attitude control system requires a fundamental shift in verification and validation (V&V) for space-grade software. V&V for guidance, navigation and control traditionally relies on deterministic and formally verifiable control laws, e.g., proportional–integral–derivative (PID) controllers. For these, the system's behavior is auditable through formal analysis and exhaustive simulation of well-defined corner cases. Core challenges

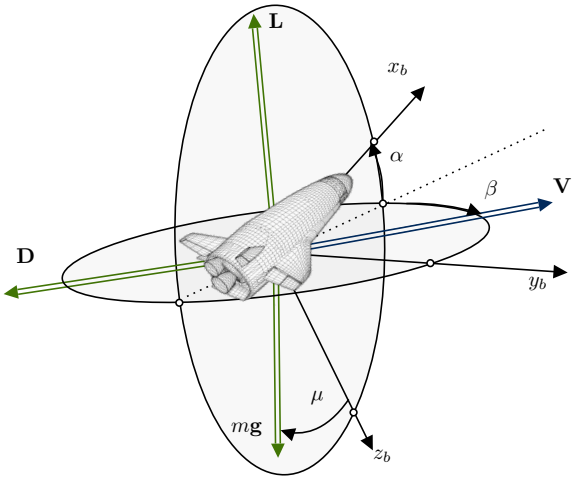



Figure 1: Sketch of the hypersonic re-entry vehicle.

with deep RL are the out-of-distribution generalization and opacity of the learned policy. It is a black box, making it infeasible to formally prove its stability and performance across the entire state space and parametric envelope. While explainable artificial intelligence advances transparency in the decision-making process of deep RL (Li et al., 2025; Goel et al., 2025; Luss et al., 2023), its integration into safety-critical control applications is not fully established. We propose the use of hybrid control designs (e.g., residual RL, Johannink et al., 2019) to mitigate black-box concerns.

We specifically discuss attitude control for a hypersonic re-entry vehicle, as illustrated in Figure 1. Such spacecraft return personnel and cargo from space and, hence, are a key technology for future human spaceflight, planetary exploration, and for establishing an extraterrestrial resource extraction economy, e.g., lunar mining. We attempt on the one hand to replace classical control engineering by RL, reducing the expert's effort, and on the other hand to combine PID control with RL-based control to form a hybrid controller that more effectively handles challenging changes to the environment dynamics and failure cases. We compare three different types of controllers. (1) Baseline controller: The previous solution is a PID controller with gain scheduling. (2) Only RL: RL without any additional controller, but with the model of the spacecraft to generate kinematically feasible reference values. (3) Hybrid controller: RL modifies the baseline controller. The hybrid controller is our primary candidate for a flight-ready system. This architecture is inherently safer because the PID controller acts as a verified baseline. The RL agent's authority can be explicitly bounded by a supervisor logic that monitors the system's state and the RL agent's output, disengaging the RL agent when the vehicle approaches the edge of its validated safe domain. We aim to provide statistical evidence that the controller is reliable and safe within a strictly defined operational envelope.

Our contributions are the following. (1) We apply state-of-the-art deep reinforcement learning algorithms for continuous control to the problem of attitude control during hypersonic re-entry of a spacecraft. This is a challenging tracking problem that requires gain scheduling for changing atmospheric conditions. We design a simple and effective reward function for attitude control and find that MR.Q excels without task-specific tuning. (2) Although the best learned controller performs better than the industry-standard baseline controller under nominal conditions, we explore hybrid control architectures that combine the baseline controller with reinforcement learning and compare these against pure reinforcement learning to enhance out-of-distribution generalization. (3) To improve robustness and generalization explicitly, we compare dynamics randomization and task scheduling approaches for reinforcement learning in this attitude control problem and determine the best training strategy. We find that the resulting policies considerably improve robustness in comparison to the baseline controller.

## 2 Background: Spacecraft Re-Entry and Reinforcement Learning

### 2.1 Spacecraft Re-Entry Problem

The present paper considers flight path control for a lifting-body vehicle with low lift-to-drag ratio during hypersonic re-entry (see Figure 1). We focus on the endo-atmospheric part of the flight trajectory and disregard the preceding deorbit maneuver. Specifically, we focus on the descent from the upper mesosphere down to the troposphere (altitude $93\,\mathrm{km}$ to $10\,\mathrm{km}$). The vehicle speed at the start is $7378\,\mathrm{m/s}$ and decreases to approximately $150\,\mathrm{m/s}$ at the end, when a parachute can be deployed for landing. Atmospheric flight creates aerodynamic forces and moments that act on the vehicle. The main objective is to control the aerodynamic forces such that they result in a desired translational motion of the re-entry vehicle. In addition, the magnitude of the forces must remain limited to not endanger structural integrity of the vehicle. These objectives are addressed by tracking a specific angle of attack and bank angle. The angle of attack $\alpha$ describes the pitch attitude of the vehicle with respect to its direction of flight, while the bank angle $\mu$ describes the roll attitude of the vehicle with respect to this direction. Furthermore, the nose of the vehicle needs to be aligned with its direction of flight, i.e., the sideslip angle $\beta$ must remain small. Figure 1 illustrates these angles. Attitude control must minimize the errors between the commanded and actual angles over the course of the trajectory. The vehicle in this study has two flaps at its tail that deflect to produce aerodynamic

moments about the body-fixed axes $x_b$ and $y_b$, and thrusters arranged in a configuration to exert a moment about the axis $z_b$. The actions of attitude controllers therefore manipulate deflection commands to the aerodynamic control surfaces ($\delta_{\text{e,cmd}}$ for symmetric and $\delta_{\text{a,cmd}}$ for antisymmetric deflection) and a thruster command $\boldsymbol{\tau} = [\tau_x, \tau_y, \tau_z]^T$. The observations encompass command values for angle of attack ($\alpha_{\text{cmd}}$) and bank angle ($\mu_{\text{cmd}}$) as well as estimates of the actual angles $\alpha, \beta, \mu$, and body-fixed angular rates from the navigation filter.

### 2.2 Simulation Model

In order to train and evaluate the controllers, we rely on a simulation that encompasses the vehicle dynamics, actuator and sensor lags, atmosphere, as well as guidance, navigation, and control algorithms that serve as a baseline for this study. The models are briefly described next with additional details given in Appendix D. The step size of the simulator is $1/140\,\text{s}$ and it accounts for multi-rate subsystems and computational delay. For example, control algorithms are executed at $14\,\text{Hz}$.

A six-degree-of-freedom rigid-body flight dynamics model for a rotating, spherical earth is the core of the simulation (see, e.g., Stevens & Lewis, 2003; Stengel, 2022). In addition to the notation from Figure 1, the aerodynamic side force $Y$ (perpendicular to aerodynamic lift $L$ and drag $D$), and the vehicle mass $m$ are needed in the evolution of the flight path, defined in terms of velocity $V$, flight path angle $\gamma$ (vertical direction relative to the local horizontal), and course angle $\chi$ (horizontal direction relative to local north) as

$$\begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{\chi} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -D - mg\sin\gamma \\ (L\cos\mu - Y\sin\mu - mg\cos\gamma)/V \\ (L\sin\mu + Y\cos\mu)/V \end{bmatrix}. \tag{1}$$

The change of the aerodynamic angles $\alpha$, $\beta$, and $\mu$ is

$$\begin{bmatrix} \dot{\mu} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} (\omega_x\cos\alpha + \omega_z\sin\alpha)/\cos\beta - \dot{\chi}\sin\gamma + \tan\beta(\dot{\gamma}\cos\mu + \dot{\chi}\cos\gamma\sin\mu) \\ \omega_y - \tan\beta(\omega_x\cos\alpha + \omega_z\sin\alpha) - (\dot{\gamma}\cos\mu + \dot{\chi}\cos\gamma\sin\mu)/(\cos\beta) \\ \omega_x\sin\alpha - \omega_z\cos\alpha + \dot{\chi}\cos\gamma\cos\mu - \dot{\gamma}\sin\mu \end{bmatrix},$$

where we denote angular rates about the body-fixed axes by $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$. The rotational dynamics are described by $\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}(\boldsymbol{M} - \boldsymbol{\omega} \times \boldsymbol{I}\,\boldsymbol{\omega})$, where $\boldsymbol{M} \in \mathbb{R}^3$ denotes the moments resulting from aerodynamics and thruster usage and $\boldsymbol{I}$ is the inertia tensor with respect to the center of mass. The aerodynamic forces and moments $D, L, Y, \boldsymbol{M}$ are complicated to model accurately and are, in general, nonlinear functions of the vehicle state and environmental conditions. We employ an industry-grade high-fidelity aerodynamics model that represents these dependencies in terms of tabulated coefficients that depend on the flight conditions. This is a standard approach in flight dynamics modeling (see, e.g., Schmidt, 2011; Stengel, 2022). For example, the lift is modeled as $L = \bar{q}SC_L(Ma, \alpha, \beta, \boldsymbol{\omega}, \boldsymbol{\delta})$. Here, $\bar{q} = \frac{1}{2}\rho V_a^2$ is the dynamic pressure, resulting from the velocity of the vehicle relative to air $V_a$ (i.e., $V$ accounted for wind) with the local air density $\rho$, and $S$ is the effective surface area of the vehicle. The coefficient $C_L$ depends on the Mach number $Ma$ (the ratio of flight speed and local speed of sound) and the flap deflections $\boldsymbol{\delta} = [\delta_\text{e}, \delta_\text{a}]^T$ in addition to the state variables $\alpha$, $\beta$, and $\boldsymbol{\omega}$. The simulation encompasses $Ma \in [0.5, 26.8]$ and $\bar{q} \in [50, 5825]\,\text{Pa}$, leading to considerable variations in the aerodynamic influence. The international standard atmosphere model is used to determine environmental conditions such as the density of air and speed of sound.

Trajectory generation is a complex, mission-specific endeavor that requires precalculation and predictor-corrector algorithms (Lu, 2008; Vernis et al., 2011; Spreng et al., 2011). We consider a generic atmospheric re-entry mission with a simplified flight path guidance loop. One objective for the controller is to keep the sideslip angle small, $\beta \approx 0$, which implies $Y \approx 0$. For this case, Equation (1) simplifies and provides insight into the principal mechanics behind guidance for re-entry. The gravitational force ($mg$) accelerates the vehicle during descent ($\gamma < 0$), while the aerodynamic drag $D$ decelerates the vehicle. The change in $\gamma$ depends on the term $L\cos\mu$, which represents the upward component of the aerodynamic lift $L$ that opposes the gravitational force ($mg\cos\gamma$).

Hence, the vertical direction of flight can be controlled by changing the magnitude of the bank angle $\mu$. However, the equation for $\chi$ reveals that this will also change the horizontal direction due to the sideward lift component $L \sin \mu$. We employ a simple guidance strategy: (1) The angle of attack determines the magnitude of lift and drag. The command value $\alpha_{\text{cmd}}$ issued by the guidance is a setpoint that ensures structural integrity of the vehicle, while resulting in sufficient deceleration along the trajectory. (2) The magnitude of the bank angle command $\mu_{\text{cmd}}$ is calculated by a feedback loop to track a predefined flight path angle $\gamma$, i.e., to offset the gravity component $mg \cos \gamma$ in Equation (1). (3) Bank angle reversal is triggered, i.e., the sign of the command value $\mu_{\text{cmd}}$ is switched, whenever a specified heading angle deviation from the reference is exceeded. This strategy leads to sufficiently rich variations in the reference commands for the present study.

### 2.3 Baseline Controller

We focus on the atmospheric control mode, which is more complex compared to exo-atmospheric attitude control. To provide a challenging baseline for benchmarking, a state-of-the-art control algorithm is included in the simulation. This baseline consist of separate control laws for the longitudinal motion ($\alpha \to \alpha_{\text{cmd}}$ tracking) and for the lateral-directional motion ($\mu \to \mu_{\text{cmd}}$ tracking and $\beta \to 0$ regulation). Longitudinal control uses a gain-scheduled proportional-integral-derivative (PID) law in combination with an inverse model $f$ of the flight dynamics in feedforward:

$$\begin{aligned} \delta_{\text{e}}(t) \quad = \quad & f(Ma(t), \bar{q}(t), \alpha_{\text{cmd}}(t)) + k_p(Ma(t), \bar{q}(t)) \cdot (\alpha_{\text{cmd}}(t) - \alpha(t)) \\ & + k_i(Ma(t), \bar{q}(t)) \cdot \int_0^t \alpha_{\text{cmd}}(s) - \alpha(s) ds - k_d(Ma(t), \bar{q}(t)) \cdot \dot{\alpha}(t) \end{aligned}$$

This is a deterministic, time-varying control policy with continuous changes of the controller gains based on the current estimates of the Mach number $Ma(t)$ and dynamic pressure $\bar{q}(t)$. Similarly, lateral-directional control laws are gain-scheduled PID for bank angle tracking and proportional-derivative (PD) control for sideslip angle regulation. The lateral directional control laws involve crossfeeds and hence they are multivariable. The controller aims at yielding as little as possible error between the commanded and the estimated values of the angle of attack, bank angle, and sideslip angle. The gains for the control laws were selected based on the well-established pole-placement technique that seeks to match a closed-loop response with desired characteristics specified by a low-order transfer function reference model (see, e.g., Enns et al., 1994; Stevens & Lewis, 2003). Designs were performed for 21 linear snapshot models obtained along the considered nominal flight trajectory. The control laws are implemented in terms of state feedback gain schedules with linear interpolation in between design grid points. The scheduling variables are calculated based on estimated Mach number and dynamic pressure (similar to Ganet-Schoeller et al., 2008).

### 2.4 Reinforcement Learning for Continuous Control and Generalization of Policies

Deep Deterministic Policy Gradients (DDPG, Lillicrap et al., 2016) enabled using neural networks for continuous model-free RL. Since then, many off-policy RL algorithms for continuous control were influenced by Soft Actor Critic (SAC, Haarnoja et al., 2018) and DDPG's extension TD3 (Fujimoto et al., 2018). TD3 adds clipped double Q-learning, target policy smoothing, and delayed policy updates to DDPG. SAC is similar to TD3, but uses a stochastic policy in the maximum entropy RL framework. TD3 was extended to TD7 (Fujimoto et al., 2023) and MR.Q (Fujimoto et al., 2025). TD7 adds a state-action encoder, loss-adjusted prioritized experience replay (LAP, Fujimoto et al., 2020), and checkpointing. Similarly, MR.Q uses model-based representation learning and LAP. SAC was extended to DroQ (Hiraoka et al., 2022), CrossQ (Bhatt et al., 2024), BRO (Nauman et al., 2024), and SimbaV2 (Lee et al., 2025). Across various benchmarks for continuous control, model-free RL (Nauman et al., 2024; Fujimoto et al., 2025; Lee et al., 2025) currently seems to outperform model-based RL, i.e., DreamerV3 (Hafner et al., 2025) and TD-MPC2 (Hansen et al., 2024).

To obtain robust policies and enable sim-to-real transfer, dynamics randomization has been used in robotics (Antonova et al., 2017; Peng et al., 2018; Tan et al., 2018; Andrychowicz et al., 2020).

In its simplest form, dynamics randomization varies parameters of the dynamics in the simulation randomly during training without informing the RL agent. The agent has to learn a policy that is robust under these unknown variations while only relying on observations.

We have a smoothly varying parametrization of the dynamic properties over which we want to generalize, e.g., actuation characteristics. Hence, we adopt the formalism of Contextual Markov Decision Processes (CMDPs, Hallak et al., 2015), in which the variation of tasks is described as a tuple $(\mathcal{C}, \mathcal{S}, \mathcal{A}, \mathcal{M}(c))$ , with context space $\mathcal{C}$, state space $\mathcal{S}$, action space $\mathcal{A}$, and a function $\mathcal{M}(c) = (\mathcal{S}, \mathcal{A}, P_c(s'|s,a), R_c(r|s,a,s'), \mu_c(s_0))$ mapping contexts $c \in \mathcal{C}$ to individual Markov Decision Processes (MDPs) with context-dependent state transition probabilities $P_c$, reward function $R_c$, and initial state distribution $\mu_c$. In our case, the agent only has access to observations that depend on context and previous states $o_t = \phi(c, s_t, s_{t-1}, \ldots)$, and it is supposed to select actions $a \in \mathcal{A}$. Our goal is to find a policy $\pi$ that maximizes the objective function $\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} J(\mathcal{M}(c), \pi)$, where $J(\mathcal{M}, \pi)$ is the expected return of policy $\pi$ in task $\mathcal{M}$. CMDPs allow us to define distinct sets of contexts for training and testing to evaluate zero-shot generalization (Kirk et al., 2023).

Going one step further, we want to select task variations intelligently during training to improve sample efficiency and final performance. Task scheduling approaches have been proposed for multi-task RL (Sharma et al., 2018; Cho et al., 2024) and contextual RL (Fabisch & Metzen, 2014). Fabisch & Metzen (2014) frame the task selection problem as a non-stationary multi-armed bandit problem and solve it via discounted upper-confidence bound (D-UCB, Kocsis & Szepesvári, 2006; Garivier & Moulines, 2011). In this framework, they compare task selection strategies based on the episodic return: easiest tasks first (best reward), hardest task first, and focus on tasks with largest improvement in returns (e.g., monotonic progress). We call this active multi-task training (AMT). Scheduled multi-task training (SMT, Cho et al., 2024) focuses on solving hard tasks first.

## 3 Learning Attitude Control for Hypersonic Spacecraft Re-Entry

**Action space (see also Appendix A)** The vehicle uses its flaps to control pitch and roll motion, and thrusters for yaw motion. We compare several architectures for attitude control. The simplest learned controller (Only RL) will directly output yaw thrust and changes of the flap angles. The controller described in Section 2.3 serves as a baseline for evaluation, but also as a building block in two hybrid controllers. In additive hybrid control, the output of the RL policy network is added to the output of the baseline controller (also known as residual RL, Johannink et al., 2019). A more advanced hybrid control approach uses the policy for gain scheduling in the baseline controller, i.e., we compute twelve continuous gain factors $f_{\text{gain}}$ to scale gains in the baseline control laws. We represent these multipliers as values of decibel in accordance with classical control notion, i.e., we transform each policy output by $f_{\text{gain}}(x) = 10^{x/20}$ with $x \in [-6, 6]$, such that a policy output of $x = 0$ results in a factor of 1. The maximum is $f_{\text{gain}}(6) \approx 2$, and the minimum is $f_{\text{gain}}(-6) \approx 0.5$.

**Observation space (see also Appendix B)** The observable state of the vehicle includes the aerodynamic angles $(\alpha, \beta, \mu)$, and angular velocities $\boldsymbol{\omega}$ (see Section 2). In addition, we add the commanded aerodynamic angles $(\alpha_{\text{cmd},t}, \beta_{\text{cmd},t}, \mu_{\text{cmd},t})$, the previously commanded flap deflections $(\delta_{e,\text{cmd},t-1}, \delta_{a,\text{cmd},t-1})$, and, for hybrid control, also the current baseline control command to the observation space. To mimic the structure of a conventional PID error-feedback controller, we add for each aerodynamic angle the current error, its derivative, and its integral to the observation space. This improves the ability of a feedforward policy network in this partially observable MDP. The dynamics of the environment are influenced by air density and speed of the vehicle, which determine the dynamic pressure. We measure the altitude, velocity, and dynamic pressure and provide it as observations to the controller. Observations are normalized to the range $[-1, 1]$.

**Reward function (see also Appendix C)** Our main objective is to minimize the attitude cost

$$c(t) = \left(\frac{\alpha_{\text{cmd}}(t) - \alpha(t)}{\alpha_{\text{range}}}\right)^2 + \left(\frac{\beta_{\text{cmd}}(t) - \beta(t)}{\beta_{\text{range}}}\right)^2 + \left(\frac{\mu_{\text{cmd}}(t) - \mu(t)}{\mu_{\text{range}}}\right)^2,$$

which we define based on commands and estimates of the aerodynamic angles (in radian). The individual ranges allow us to define a desired range for each error. This approach is known as *Bryson's rule* in classical linear quadratic optimal control. We focus the attitude cost on the angle of attack $\alpha$ with $\alpha_{\text{range}} = 2°, \beta_{\text{range}} = 10°, \mu_{\text{range}} = 10°$.

However, taking the negative attitude cost as a reward has the drawback that the reward is always negative, which makes early termination of an episode by running into failure states appealing. We solve this problem by defining a strictly positive reward

$$\text{attitude reward}\,(t) = \max(0.001, \exp(-w_c \cdot c(t))) \in [0.001, 1]$$

with weight $w_c = 2$. Since the attitude reward is at least 0.001, it discourages the agent from terminating early and shifts focus to the main objective.

We combine the attitude reward with a control cost to form the reward function in each step $t'$ as $r_{t'} = \text{attitude reward}\,(t) - \text{control cost}\,(t)$, where control cost is the sum of thruster and flap costs

$$\text{control cost}\,(t) = w_{\tau_z}\left(\frac{\tau_z(t)}{\tau_{z,\max}}\right)^2 + w_{\delta_e}\left(\frac{\Delta\delta_{e,\text{cmd}}(t)}{\Delta\delta_{e,\max}}\right)^2 + w_{\delta_a}\left(\frac{\Delta\delta_{a,\text{cmd}}(t)}{\Delta\delta_{a,\max}}\right)^2,$$

with weights $w_{\tau_z} = 1$ and $w_{\delta_e} = w_{\delta_a} = 0.05$. The difference between successive flap commands $\Delta\delta_{\circ,\text{cmd}}(t) = \delta_{\circ,\text{cmd}}(t) - \delta_{\circ,\text{cmd}}(t-1)$ has a maximum of $\Delta\delta_{\circ,\max} = \Delta t \cdot \dot{\delta}_{\circ,\max} = \frac{1}{14} \cdot 15° \approx 1.07°$. The yaw torque $\tau_z(t)$ generated by the thrusters is limited by $\tau_{z,\max} = 300\,\text{N}$.

**Context space (see also Appendix D)** For dynamics randomization, we parametrize the simulation with the mass of the vehicle $m_0 \in [1312, 1968]$ kg, fractions of the principal moments of inertia $\boldsymbol{f} \in [0.9, 1.1]^3$, misalignment of the principal axis by means of rotation $\boldsymbol{\omega} \in \{\omega = \theta\hat{\omega} | \hat{\omega} \in S^2, \theta \in [-10°, 10°]\} \subset \mathbb{R}^3$, and flap actuator bandwidth $\omega_0 \in [12, 30]$ rad/s. Hence, the parameter space is $\mathcal{C} \subset \mathbb{R}^8$. Nominal conditions correspond to $c_{\text{nominal}} = (1640, 1, 1, 1, 0, 0, 0, 30)^T$.

## 4 Experiments

In our experiments, we want to answer the following research questions:

1. Can we apply deep RL algorithms to spacecraft attitude control? (Section 4.1.1)
2. What is the best achievable performance with pure RL? (Section 4.1.1)
3. Which control architecture works best? (Section 4.1.2)
4. How well do controllers generalize to out-of-distribution conditions? (Section 4.1.3)
5. Can we enforce generalization with domain randomization or task scheduling? (Section 4.2.1)
6. Which is the most robust controller? (Section 4.2.2)

**Evaluation of RL algorithms:** Following Agarwal et al. (2021), we analyze the performance of RL algorithms with sample-efficiency curves using the interquartile mean (IQM) and bootstrapped 95% confidence intervals of the cumulative episode reward, i.e., undiscounted return. For direct comparison of policies, we compute the probability of improvement.

**Application-specific performance metrics:** Key metrics for attitude control are the absolute errors of the angle of attack $|e_\alpha(t)| = |\alpha_{\text{cmd}}(t) - \alpha(t)|$, sideslip angle $|e_\beta(t)| = |\beta_{\text{cmd}}(t) - \beta(t)|$, and bank angle $|e_\mu(t)| = |\mu_{\text{cmd}}(t) - \mu(t)|$. We prioritize the angle of attack to maintain structural and thermal integrity of the vehicle and create sufficient drag to decelerate. We want to avoid large errors, while small errors over a longer period of time are allowed. Hence, we analyze the error distribution and report percentiles. As secondary metrics, we report the control costs in all three action components. For multiple test contexts, we furthermore report the success rate, which is the percentage of contexts, in which the controller was able to follow the trajectory completely. Episodes terminate when any angle is outside of its safe domain, which is defined based on the aerodynamic angles: $\alpha \in [0°, 60°]$, $\beta \in [-20°, 20°]$, and $\mu \in [-90°, 90°]$.

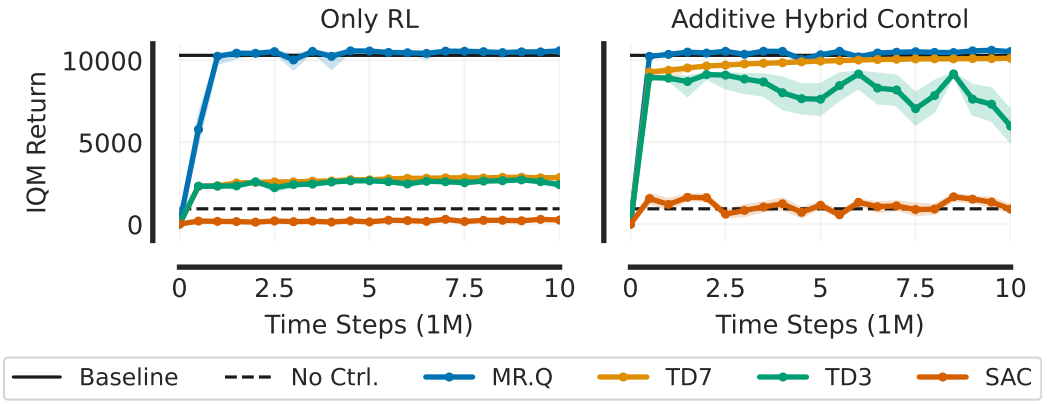



Figure 2: MR.Q surpasses the baseline controller. Learning curves with interquartile means (IQM) and 95% bootstrap confidence intervals for 10 training seeds (Agarwal et al., 2021).

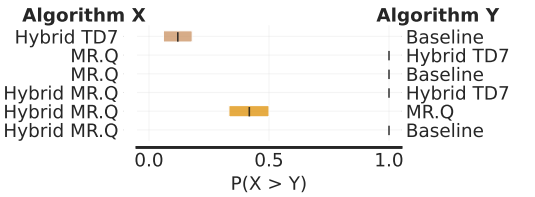
### 4.1 Training Under Nominal Conditions

We start episodes with constant initial attitude, initial mass, inertia, and flap actuator bandwidth, but randomize parameters of the trajectory to prevent overfitting the trajectory (see Appendix D for details). We analyze which control architecture and RL algorithm performs best and what is the maximum achievable performance. Since we do not vary dynamics during training, we can test out-of-distribution generalization of the learned controllers.

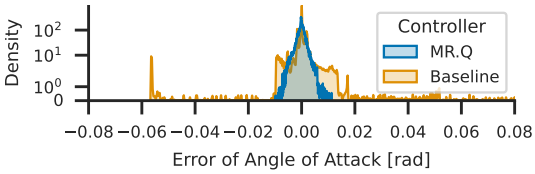
#### 4.1.1 RL Algorithm Selection

We initially selected SAC and TD3 to explore the feasibility RL-based control. We tried CrossQ as an extension of SAC as well as TD7 and MR.Q as extensions of TD3. In this study, we focus on TD3-based algorithms as they showed consistent behavior in preliminary experiments without task-specific tuning. Hence, we evaluate only SAC, TD3, TD7, MR.Q. Hyperparameters are listed in Appendix E.

With the learning curves in Figure 2, we evaluate the performance and sample efficiency of RL algorithms and control architectures during training. We evaluate combinations of learning algorithm and control architecture with 10 random seeds for training and 10 evaluation episodes without exploration noise. For comparison, we plot the performance of no control and the baseline controller averaged over 10 evaluation episodes. Individual learning curves per training run are in Appendix M.




(a) Probability of improvement (Agarwal et al., 2021).




(b) Error distribution (kernel density estimation) for angle of attack. Density is plotted on a logarithmic scale to highlight the tails. The peak of the baseline controller around 0 is more pronounced on a linear scale.

Figure 3: Comparison under nominal conditions.

MR.Q excels without task-specific tuning. The main reasons for this are the multi-step return, which reduces bias in the target value for the critic loss, and the model-based representation (detailed analysis in Appendix G). Most algorithms perform better in the hybrid control architecture than their non-hybrid counterpart, with the exception of MR.Q. For the non-obvious cases, we compare the best obtained policies per run with the probability of improvement and a 95% confidence interval in Figure 3a. We attribute that MR.Q outperforms the baseline controller to the fact that MR.Q learns to continuously adapt to environment conditions, which is more efficient than the gain scheduling of the baseline. In particular, the error distribution for our primary control objective, the angle of attack, is narrower for the learned controller than for the baseline (see Fig-

ure 3b, see also Appendix J for obtained trajectories and control commands). However, the baseline has a higher concentration of errors close to 0. Hence, MR.Q works without task-specific tuning and is able to achieve a higher return than the baseline controller by tracking the angle of attack more accurately.

#### 4.1.2 Comparison of Control Architectures

With MR.Q, we compare control architectures (see Figure 4). Our simplest RL-based controller maps from observations to flap changes and thruster commands. We compare it to additive and a gain-scheduling hybrid control. In both hybrid architectures, the baseline controller's command is in the observation space.

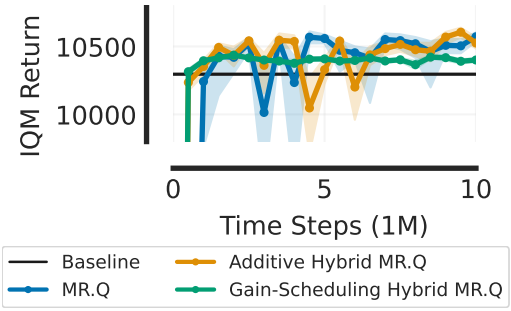


Figure 4: Comparisons of control architectures.

For maximum performance pure RL or additive hybrid control seem to be the best options under nominal evaluation conditions. Although the best possible performance of the gain-scheduling hybrid controller is lower than for the two other architectures, the bounded influence of the policy makes learning considerably more stable across runs and over time, and the performance is still better than the baseline.

#### 4.1.3 Out-Of-Distribution Generalization of Policies Trained Under Nominal Conditions

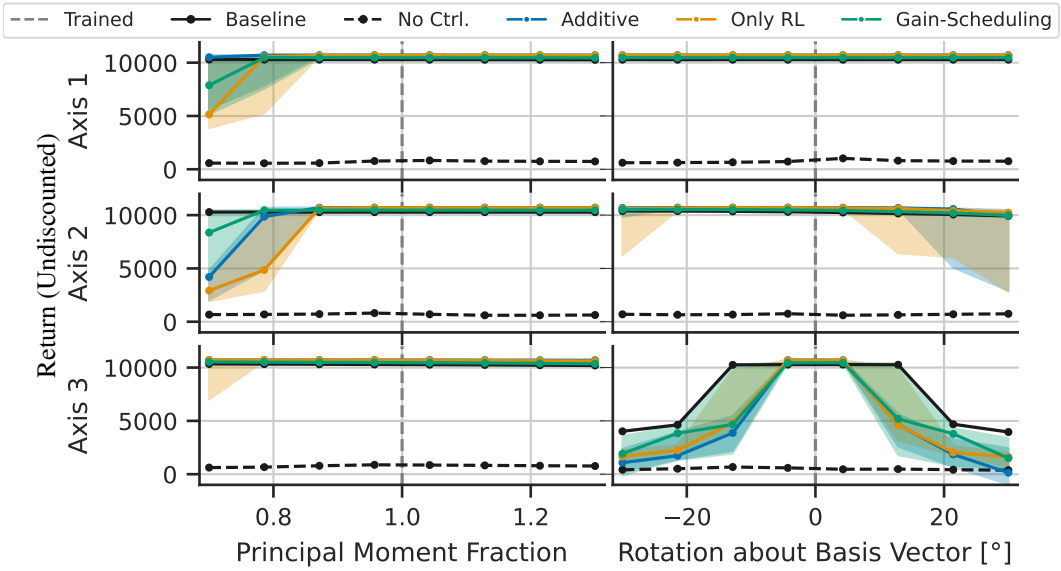
To analyze generalization of the standard RL training paradigm with random exploration in action space, we test robustness with respect to variations that were not present during training. We compare the undiscounted return of an episode for the best obtained policies per run (see Figure 5).

**Inertia tensor (Figure 5a):** The inertia tensor $\boldsymbol{I}$ has six degrees of freedom. To systematically vary it, we multiply the principal moments of inertia $I_{xx}, I_{yy}, I_{zz}$ by factors from $[0.7, 1.3]$ and apply rotations about the basis vectors with angles from $[-30°, 30°]$. Rotation of $\boldsymbol{I}$ about the z-axis has the most impact and the baseline controller is more robust than the learned controllers to it.

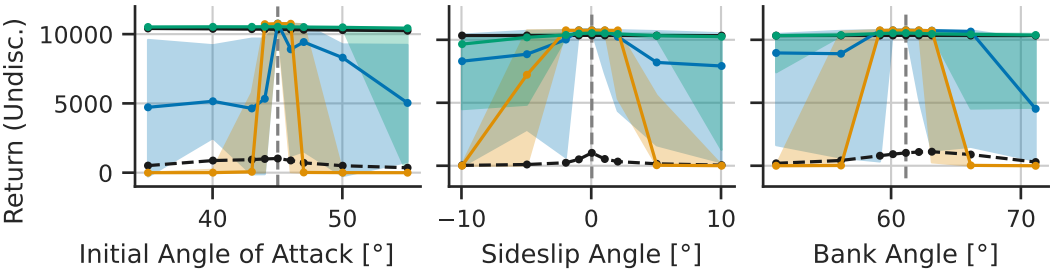
**Initial attitude (Figure 5b):** We vary the initial values of the aerodynamic angles. Although policies trained on constant initial attitude outperform the baseline under nominal conditions, they do not generalize well. Even for a small envelope around nominal conditions, we cannot guarantee stability without explicit testing, which demonstrates that action noise is not enough to learn policies that are robust against deviations from nominal conditions. However, the additive hybrid controller's performance deteriorates less drastically in most out-of-distribution cases than pure RL and the gain-scheduling hybrid controller is almost as reliable as the baseline.

**Flap actuator bandwidth (Figure 5c):** Similarly, we test generalization to variations of the flap actuator bandwidth. Lower values make the control problem harder and simulate failures of the actuation. Nominal conditions correspond to $30\,\mathrm{rad/s}$. Hence, all of the tested values are out of distribution. The baseline controller's performance drastically drops for actuator bandwidths below $14\,\mathrm{rad/s}$. Interestingly, completely learned controllers are consistently more robust to lower values. We suspect that random exploration in action space during training makes the resulting policy robust to this scenario although the exact stable range cannot be guaranteed. Both hybrid control architectures show more variation, but their median performance is similar to the baseline.

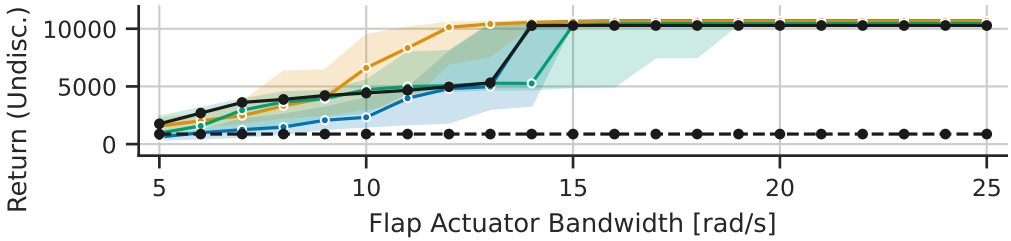
The bounded influence of the gain-scheduling hybrid controller makes its behavior more similar to the baseline with respect to variations of the initial conditions. However, the stability of learned controllers cannot be guaranteed in the same range as for the baseline controllers when we do not explicitly enforce the operational envelope during training. Although MR.Q is more robust to actuation failures than the baseline controller, it is less stable with respect to variations of the inertia tensor and considerably less stable under variations of the initial attitude than the baseline controller.




(a) Performance evaluated under variations of $\boldsymbol{I}$. On the left, we scale the principal moments of inertia along three axes (from top to bottom: x-, y-, z-axis). On the right, we rotate $\boldsymbol{I}$ about the three basis vectors.




(b) Performance with respect to changes of the initial attitude without training for it. Performance is measured at the initial attitude during training $+\epsilon \in [-10, -5, -2, -1, 0, 1, 2, 5, 10]$ degree per component.




(c) Performance evaluated under variations of flap actuator bandwidth.

Figure 5: Generalization of policies trained under nominal conditions. The solid lines show median performance over training runs and shaded areas the $[5, 95]$-percentile interval.

### 4.2 Training Under Dynamics Randomization

We randomize not just the trajectories, but also the initial attitude to ensure that the RL-based controllers generalize to these variations. Furthermore, we systematically vary the following dynamics parameters: initial mass, inertia tensor, and flap actuator bandwidth (see Appendix D for details). Each combination of dynamics parameters defines a context. We compare task scheduling approaches to select contexts during training and evaluate generalization of the obtained controllers.

To test generalization and robustness of controllers, we uniformly sample a test set $\mathcal{C}_{\text{test}}$ of 100 contexts and measure the obtained return in each test context. We select the best controllers per training procedure for this analysis based on the minimum return in a smaller test set of 30 contexts from regularly saved checkpoints per run (checkpoint interval: 200k time steps).

### 4.2.1 Comparison of Task Scheduling Methods for Dynamics Randomization

We sample $|\mathcal{C}_{\text{train}}| = 50$ context vectors that define individual tasks. We compare SMT, which prefers hard tasks, and active task selection with the best-reward (AMT-B, prefers easy tasks) and monotonic progress (AMT-M, prefers tasks with high expected learning progress) strategies. Hyperparameters are listed in Appendix F. We use continuous uniform random sampling of contexts (dynamics randomization, DR) and round robin (RR) as baselines for task scheduling.

Task scheduling methods do not perform better than dynamics randomization or round robin. Since the results are inconclusive, we report them in Appendix L. Among task selection strategies, focusing on hard tasks in the beginning (SMT) is the worst strategy. Focusing on the easiest tasks first (AMT-B) and focusing on tasks with the maximum expected learning progress (AMT-M) work equally well and are comparable to the baselines DR and RR with no clear advantage.

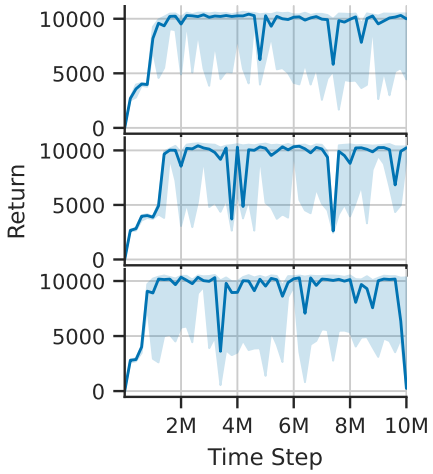



Figure 6: Learning curves of three training runs with DR and pure RL with MR.Q. Lines indicate median performance over 30 test contexts and shaded areas show the full range of returns.

Training with task scheduling is difficult for two reasons: (1) training on the full range of flap actuator bandwidths creates some problems that are considerably harder to solve, and (2) as we use a replay buffer for each task, infrequently selected tasks might contain transitions of low performing policies that are still selected relatively often, since we first sample the task uniformly and then the samples within the task for the batch that is required in the network updates. We mitigate the first issue by sampling flap actuator bandwidths from $[25, 30]$ rad/s, but still evaluate on the full range. The second problem causes instabilities after about 4M steps, which we avoid by stopping after 5M steps. A more detailed analysis is required to understand the problem and find solutions.

### 4.2.2 Generalization and Robustness of Best Controllers

Since task scheduling approaches do not provide any benefit in comparison to continuous uniform dynamics randomization (DR), we analyze more closely the performance of the best obtained controllers with DR. We compare DR for pure RL and additive hybrid control in Table 1 with domain-specific evaluation metrics. We focus on the error distribution of the aerodynamic angles and success rate. See Appendix L for a detailed analysis including the control costs. Exemplary training curves are displayed in Figure 6).

Table 1: Evaluation of best controllers with domain-specific metrics. Cell background color indicates best values or values better than the baseline. Appendix K contains more details on error distributions and Appendix L contains all results with control costs.

| Training | Seed | Success | Median | 90-Perc. | 95-Perc. | 98-Perc. | Median | 90-Perc. | 95-Perc. | 98-Perc. | Median | 90-Perc. | 95-Perc. | 98-Perc. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\lvert e_\alpha \rvert$ [deg] | | | | $\lvert e_\beta \rvert$ [deg] | | | | $\lvert e_\mu \rvert$ [deg] | | | |
| Baseline | - | 79% | 0.05 | 0.46 | 0.67 | 2.96 | 0.02 | 0.20 | 0.36 | 0.57 | 0.06 | 0.72 | 1.43 | 2.35 |
| Add. Hyb. | 0 | 93% | 0.05 | 0.23 | 0.34 | 0.59 | 0.07 | 0.25 | 0.41 | 0.65 | 0.17 | 0.42 | 0.59 | 0.87 |
| Add. Hyb. | 1 | 93% | 0.07 | 0.37 | 0.56 | 0.83 | 0.07 | 0.30 | 0.53 | 0.85 | 0.15 | 0.60 | 0.84 | 1.11 |
| Add. Hyb. | 2 | 94% | 0.07 | 0.33 | 0.47 | 0.71 | 0.11 | 0.39 | 0.60 | 0.84 | 0.17 | 0.60 | 0.84 | 1.19 |
| Only RL | 0 | 100% | 0.12 | 0.47 | 0.63 | 0.86 | 0.09 | 0.45 | 0.88 | 1.87 | 0.23 | 1.14 | 1.63 | 2.19 |
| Only RL | 1 | 99% | 0.10 | 0.45 | 0.62 | 0.84 | 0.10 | 0.53 | 1.06 | 2.03 | 0.34 | 1.31 | 1.82 | 2.57 |
| Only RL | 2 | 98% | 0.10 | 0.39 | 0.55 | 0.80 | 0.09 | 0.49 | 0.95 | 1.76 | 0.25 | 0.69 | 0.98 | 1.49 |

In comparison to the baseline controller, which fails particularly often at low flap actuator bandwidths and large rotations of the inertia tensor, both additive hybrid control and pure RL improve the success rate considerably. The additive hybrid controller's success rate seems to be limited by the baseline, although it obtains lower error 9x-percentiles in each aerodynamic angle. For the angle of attack, both architectures reduce the 90, 95, and 98 error percentiles in comparison to the baseline. The additive hybrid controller is even better at tracking the bank angle than the baseline in these percentiles. Improving these metrics comes at the cost of more expensive control, i.e., thruster usage and flap angle changes (see Appendix L).

## 5 Related Work

RL has been applied to attitude control in various ways for vehicles inside (Zhen et al., 2020; Bernini et al., 2021; Liu et al., 2022a; Rosa et al., 2023; Bernini et al., 2024; Bøhn et al., 2024; City University of London, 2024) and outside of an atmosphere (Su et al., 2019; Vedant et al., 2019; Elkins et al., 2020b;a; 2021; Mahfouz et al., 2022; Liu et al., 2022b; Xiao et al., 2023; Barrenechea et al., 2023). Q-learning (Liu et al., 2022b), PPO (Vedant et al., 2019; Elkins et al., 2020b; Zhen et al., 2020; Mahfouz et al., 2022), DDPG (Su et al., 2019; Rosa et al., 2023), TD3 (Elkins et al., 2020a; 2021; Liu et al., 2022a), and SAC (Bernini et al., 2024; Bøhn et al., 2024) were used in these applications. Most studies replace traditional control approaches by RL. However, Barrenechea et al. (2023) propose a residual RL architecture for online adaptation and Liu et al. (2022a) embed a policy for anti-disturbance control in a hybrid controller for hypersonic re-entry. Furthermore, Liu et al. (2022a) extend TD3 to EVE-TD3, which shares similarities with MR.Q: (1) a multi-step return (over a horizon of 30 steps) to improve the estimation of the target value for the critic, and (2) the policy network has a bottleneck (6–18 nodes) to force the policy to extract high-level features.

According to Elkins et al. (2020a), robustness of attitude control can be defined with respect to (1) tumble: nonzero initial angular velocity, (2) single, impulsive disturbance torque, (3) constant disturbance torque, and (4) different inertia tensors $\boldsymbol{I}$. RL can be robust against variations of $\boldsymbol{I}$ by adapting to it online (e.g., in takeover maneuvers, Liu et al., 2022b) or by handling variation of it as noise (Vedant et al., 2019). Robustness can also be defined in terms of handling disturbances caused by the applications of unknown external torques on the spacecraft (Xiao et al., 2023; Elkins et al., 2020a) or actuator faults (Xiao et al., 2023). Sources of disturbances might be (1) fault caused by bias (e.g., less available torque), (2) fault caused by failure to respond to control signals (e.g., motor fault), or (3) disturbance of environment: uncertainties in air resistance, gravity, solar pressure, etc. (Su et al., 2019). Bernini et al. (2024) measure robustness with respect to wind gusts in quadcopter control, which is similar to a single, impulsive disturbance force. Similarly, Rosa et al. (2023) verified a deep RL algorithm for guidance and control of a reusable launch vehicle in the landing phase and tested under various wind conditions.

## 6 Conclusion

The MR.Q algorithm excels without task-specific tuning in the challenging attitude control problem of a hypersonic re-entry vehicle (see Section 4.1.1). Specifically, its model-based representation and multi-step returns are pivotal. Pure MR.Q can surpass the performance of the baseline controller and obtain a similar performance as hybrid controllers under nominal conditions. However, training under nominal conditions does not yield sufficient generalization, although hybrid control architectures often work better under out-of-distribution settings than pure RL (see Section 4.1.3). Hence, we enforce generalization with dynamics randomization and find that continuous uniform sampling of dynamic parameters is sufficient, i.e., task scheduling approaches do not perform better (see Section 4.2.1). Surprisingly, MR.Q is able to generalize over each condition that we varied during training. With dynamics randomization, obtained controllers are considerably more robust than the baseline with respect to uncertainties in dynamics parameters (see Section 4.2.2).

The additive hybrid controller is the most promising controller. It generalizes better than pure MR.Q in the out-of-distribution experiments with training under nominal conditions (see Section 4.1.3). It improves the robustness in comparison to the baseline by reducing the probability of larger errors of the angle of attack at the cost of using the thrusters and changing the flap angles more (see Section 4.2.2). The lower bound of the additive hybrid controller's performance is the baseline controller's performance if we deactivate the learned component in out-of-distribution settings.

To ensure stability of hybrid controllers, out-of-distribution conditions must be detected to disable the learned component and fall back to the formally verified baseline. This is straightforward to implement through outlier detection with data from the replay buffer. To enable real-time execution of policies in deployment, we plan to run them on Field Programmable Gate Arrays (FPGAs).

Being able to adapt to unforeseen conditions online would be a major advantage of reinforcement learning for attitude control (see, e.g., Barrenechea et al., 2023). The gain-scheduling controller, which exhibits less variance across training steps and runs (see Section 4.1.2), is our preferred candidate for online adaptation in a real scenario. However, the analysis is out of scope for this study. More experiments are needed to evaluate stability and performance of the online adaptation process.

**Acknowledgments**

This work was funded by the European Aerospace Agency under the GSTP programme, activity GT1I-602SA "Artificial Intelligence techniques for spacecraft attitude control and estimation" (project acronym: AI4AOCS), contract number 4000145154/24/NL/MGu, lead by Airbus Defence and Space GmbH. We thank Octavio Arriaga for feedback on our implementation of MR.Q, Shubham Vyas for feedback on the manuscript, and Arthur de Freitas Precht for technical support for the interface to the simulation software. This work was partially supported by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG).

## References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *International Journal of Robotics Research*, 39(1):3–20, 2020. DOI: 10.1177/0278364919887447.

Rika Antonova, Silvia Cruciani, Christian Smith, and Danica Kragic. Reinforcement learning for pivoting task. *CoRR*, abs/1703.00472, 2017. URL http://arxiv.org/abs/1703.00472.

Maria Carrillo Barrenechea, Pierre Lachevre, Ilke Karsli Terjan, Mark Watt, and Carlos Hervas Garcia. Enabling AI-in-the-loop AOCS algorithms on in-flight hardware: from conception to in-orbit demonstration in ESA OPS-SAT. In *ESA GNC Conference Papers Repository*, 2023. DOI: 10.5270/esa-gnc-icatt-2023-131.

Nicola Bernini, Mikhail Bessa, Remi Delmas, Arthur Gold, Eric Goubault, Romain Pennec, Sylvie Putot, and Francois Sillion. A few lessons learned in reinforcement learning for quadcopter attitude control. In *International Conference on Hybrid Systems: Computation and Control*, 2021. DOI: 10.1145/3447928.3456707.

Nicola Bernini, Mikhail Bessa, Rémi Delmas, Arthur Gold, Eric Goubault, Romain Pennec, Sylvie Putot, and François Sillion. Reinforcement learning with formal performance metrics for quad-

copter attitude control under non-nominal contexts. *Engineering Applications of Artificial Intelligence*, 127:107090, 2024. ISSN 0952–1976. DOI: 10.1016/j.engappai.2023.107090.

Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. Crossq: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PczQtTsTIX.

Eivind Bøhn, Erlend M. Coates, Dirk Reinhardt, and Tor Arne Johansen. Data-efficient deep reinforcement learning for attitude control of fixed-wing UAVs: Field experiments. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):3168–3180, 2024. ISSN 2162-2388. DOI: 10.1109/TNNLS.2023.3263430.

Myungsik Cho, Jongeui Park, Suyoung Lee, and Youngchul Sung. Hard tasks first: Multi-task reinforcement learning through task scheduling. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8556–8577. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/cho24d.html.

City University of London. Explainable secure deep learning software for spacecraft gnc systems. Technical report, European Space Agency, 2024. URL https://nebula.esa.int/content/explainable-secure-deep-learning-software-spacecraft-gnc-systems.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.07289.

Jacob G. Elkins, Rohan Sood, and Clemens Rumpf. Adaptive continuous control of spacecraft attitude using deep reinforcement learning. 2020a. URL https://www.researchgate.net/publication/343834157_Adaptive_Continuous_Control_of_Spacecraft_Attitude_Using_Deep_Reinforcement_Learning.

Jacob G. Elkins, Rohan Sood, and Clemens Rumpf. Autonomous spacecraft attitude control using deep reinforcement learning. In *International Astronautical Congress*, Virtual, 2020b. URL https://ntrs.nasa.gov/citations/20205008891.

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, 2021. DOI: 10.2514/1.I010958. URL https://arc.aiaa.org/doi/10.2514/1.I010958.

Dale Enns, Dan Bugajski, Russ Hendrick, and Gunter Stein. Dynamic inversion: an evolving methodology for flight control design. *International Journal of Control*, 59(1):71–91, 1994.

Alexander Fabisch and Jan Hendrik Metzen. Active contextual policy search. *Journal of Machine Learning Research*, 15(97):3371–3399, 2014. URL http://jmlr.org/papers/v15/fabisch14a.html.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Scott Fujimoto, David Meger, and Doina Precup. An equivalence between loss functions and non-uniform sampling in experience replay. In *Advances in Neural Information Processing Systems*, volume 33, pp. 14219–14230. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a3bf6e4db673b6449c2f7d13ee6ec9c0-Paper.pdf.

Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang (Shane) Gu, Doina Precup, and David Meger. For sale: State-action representation learning for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 61573–61624. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c20ac0df6c213db6d3a930fe9c7296c8-Paper-Conference.pdf.

Scott Fujimoto, Pierluca D'Oro, Amy Zhang, Yuandong Tian, and Michael Rabbat. Towards general-purpose model-free reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=R1hIXdST22.

Martine Ganet-Schoeller, Herve Charbonnel, and Bruno Cavrois. ARD-like gain scheduled control for re-entry vehicle. In *AIAA Guidance, Navigation and Control Conference*, 2008. DOI: 10.2514/6.2008-6831.

Aurelélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pp. 174–188, 2011.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 249–256. PMLR, 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 315–323. PMLR, 2011. URL https://proceedings.mlr.press/v15/glorot11a.html.

Diksha Goel, Kristen Moore, Jeff Wang, Minjune Kim, and Thanh Thi Nguyen. Unveiling the black box: A multi-layer framework for explaining reinforcement learning-based cyber agents. *CoRR*, abs/2505.11708, 2025. URL http://arxiv.org/abs/2505.11708.

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*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. DOI: 10.1038/s41586-025-08744-2.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov decision processes. *CoRR*, abs/1502.02259, 2015. URL https://arxiv.org/abs/1502.02259.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. Dropout Q-functions for doubly efficient reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xCVJMsPv3RT.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *International Conference on Robotics and Automation*, pp. 6023–6029, 2019. DOI: 10.1109/ICRA.2019.8794127. URL https://doi.org/10.1109/ICRA.2019.8794127.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6980.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *J. Artif. Intell. Res.*, 76:201–264, 2023. DOI: 10.1613/jair.1.14174.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/5d44ee6f2c3f71b73125876103c8f6c4-Paper.pdf.

Levente Kocsis and Csaba Szepesvári. Discounted UCB. In *2nd PASCAL Challenges Workshop*, 2006. URL https://www.lri.fr/~sebag/Slides/Venice/Kocsis.pdf.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=kfYxyvCYQ4.

Peilang Li, Umer Siddique, and Yongcan Cao. From explainability to interpretability: Interpretable reinforcement learning via model explanations. In *Reinforcement Learning Conference*, 2025. URL https://openreview.net/forum?id=kreQkWaOK5.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016. DOI: 10.48550/arXiv.1509.02971.

Yiheng Liu, Honglun Wang, Tiancai Wu, Yuebin Lun, Jiaxuan Fan, and Jianfa Wu. Attitude control for hypersonic reentry vehicles: An efficient deep reinforcement learning method. *Applied Soft Computing*, 123:108865, 2022a. ISSN 1568-4946. DOI: 10.1016/j.asoc.2022.108865.

Yuhan Liu, Guangfu Ma, Yueyong Lyu, and Pengyu Wang. Neural network-based reinforcement learning control for combined spacecraft attitude tracking maneuvers. *Neurocomputing*, 484: 67–78, 2022b. ISSN 09252312. DOI: 10.1016/j.neucom.2021.07.099.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Ping Lu. Predictor-corrector entry guidance for low-lifting vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 21(4):181:1–181:50, 2008.

Ronny Luss, Amit Dhurandhar, and Miao Liu. Local Explanations for Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7):9002–9010, jun 2023. ISSN 2374-3468. DOI: 10.1609/AAAI.V37I7.26081.

Ahmed Mahfouz, Ayrat Valiullin, Alexey Lukashevichus, and Dmitry Pritykin. Reinforcement learning for attitude control of a spacecraft with flexible appendages. In *International Astronautical Congress*. International Astronautical Federation, 2022. URL https://orbilu.uni.lu/handle/10993/53599.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Mił oś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample efficient continuous control. In *Advances in Neural Information Processing Systems*, volume 37, pp. 113038–113071. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/cd3b5d2ed967e906af24b33d6a356cac-Paper-Conference.pdf.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *International Conference on Robotics and Automation*, pp. 3803–3810, 2018. DOI: 10.1109/ICRA.2018.8460528.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *CoRR*, abs/1710.05941, 2017. URL http://arxiv.org/abs/1710.05941.

Paulo Rosa, José Fernandes Vasconcelos, Nicola Somma, Afonso Botelho, Giangregorio Tofanelli, Juan Ignacio Bravo, Robert Hinz, Joris Belhadj, Massimo Casasco, and Samir Bennani. Deep reinforcement learning based integrated guidance and control for a launcher landing problem. In *Sopot 2023*, 2023. DOI: 10.5270/esa-gnc-icatt-2023-145.

David K. Schmidt. *Modern Flight Dynamics*. McGraw Hill, 2011. ISBN 9780073398112.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

Sahil Sharma, Ashutosh Kumar Jha, Parikshit S Hegde, and Balaraman Ravindran. Learning to multi-task by active sampling. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1nZ1weCZ.

Joan Solà, Jérémie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. Technical Report IRI–TR–18–01, Institut de Robòtica i Informàtica Industrial, 2018.

Federico Spreng, Gregory Gelly, Philippe Vernis, S. Belin, Hans Strauch, V. Fernandez, and Alvaro Martinez Barrio. Robust skip entry guidance for accurate low earth orbit or lunar returns. In *8th International ESA Conference on Guidance, Navigation and Control Systems*, 2011.

Robert .F. Stengel. *Flight Dynamics*. Princeton University Press, 2022. ISBN 9780691220253.

Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation*. Wiley, 2003. ISBN 9780471371458.

Ruipeng Su, Fengge Wu, and Junsuo Zhao. Deep reinforcement learning method based on ddpg with simulated annealing for satellite attitude control system. In *Chinese Automation Congress*, pp. 390–395, 2019. DOI: 10.1109/CAC48633.2019.8996860.

Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. DOI: 10.15607/RSS.2018.XIV.010.

Vedant, James T. Allison, Matthew West, and Alexander Robin Mercantini Ghosh. Reinforcement learning for spacecraft attitude control. In *International Astronautical Congress*, 2019.

Philippe Vernis, Federico Spreng, Gregory Gelly, and Alvaro Martinez Barrio. Accurate skip-entry guidance for low to medium l/d spacecrafts return missions requiring high range capabilities. In *AIAA Guidance, Navigation and Control Conference*, 2011. DOI: 10.2514/6.2011-6649.

Bing Xiao, Haichao Zhang, Zhaoyue Chen, and Lu Cao. Fixed-time fault-tolerant optimal attitude control of spacecraft with performance constraint via reinforcement learning. *IEEE Transactions on Aerospace and Electronic Systems*, 59(6):7715–7724, 2023. DOI: 10.1109/TAES.2023.3292809.

Yan Zhen, Mingrui Hao, and Wendi Sun. Deep reinforcement learning attitude control of fixed-wing UAVs. In *International Conference on Unmanned Systems*, pp. 239–244, 2020. DOI: 10.1109/ICUS50048.2020.9274875.

# Supplementary Materials

*The following content was not necessarily subject to peer review.*

## A Action Space

| Action | Description | Unit | Limits |
|---|---|---|---|
| **Control Commands (Only in Pure RL or Additive Hybrid Control Mode)** | | | |
| $\Delta\delta_{e,\text{cmd}}(t)$ | Change ($\delta_{e,\text{cmd}}(t) - \delta_{e,\text{cmd}}(t-1)$) of symmetric deflection (both flaps trailing edge down). | rad | $\left[-\frac{15}{14}^\circ, \frac{15}{14}^\circ\right]$ |
| $\Delta\delta_{a,\text{cmd}}(t)$ | Change of antisymmetric deflection of both flaps. Right flap is deflected trailing edge down, left flap is deflected trailing edge up. | rad | $\left[-\frac{15}{14}^\circ, \frac{15}{14}^\circ\right]$ |
| $\tau_z(t)$ | The available four thrusters are fired in a group that produces a (yawing) torque about the body-fixed vertical (z) axis. The command is a torque demand that is converted to thruster opening times by low-level functions. | N | $[-300, 300]$ |
| **PID Gains (Only in Gain-Scheduling Hybrid Control Mode)** | | | |
| $f_{k_{p,\alpha}}$ | Factor for proportional gain of angle of attack error. | decibel | $[-6, 6]$ |
| $f_{k_{i,\alpha}}$ | Factor for integral gain of angle of attack error. | decibel | $[-6, 6]$ |
| $f_{k_{d,\alpha}}$ | Factor for derivative gain of angle of attack error. | decibel | $[-6, 6]$ |
| $f_{k_{p,\beta,\beta}}$ | Factor for proportional gain of yaw angle error to thruster command. | decibel | $[-6, 6]$ |
| $f_{k_{d,\beta,\beta}}$ | Factor for derivative gain of yaw angle error to thruster command. | decibel | $[-6, 6]$ |
| $f_{k_{p,\beta,\mu}}$ | Factor for proportional gain of bank angle error to thruster command. | decibel | $[-6, 6]$ |
| $f_{k_{d,\beta,\mu}}$ | Factor for derivative gain of bank angle error to thruster command. | decibel | $[-6, 6]$ |
| $f_{k_{p,\mu,\mu}}$ | Factor for proportional gain of bank angle error to flap command. | decibel | $[-6, 6]$ |
| $f_{k_{i,\mu,\mu}}$ | Factor for integral gain of bank angle error to flap command. | decibel | $[-6, 6]$ |
| $f_{k_{d,\mu,\mu}}$ | Factor for derivative gain of bank angle error to flap command. | decibel | $[-6, 6]$ |
| $f_{k_{p,\mu,\beta}}$ | Factor for proportional gain of yaw angle error to flap command. | decibel | $[-6, 6]$ |
| $f_{k_{d,\mu,\beta}}$ | Factor for derivative gain of yaw angle error to flap command. | decibel | $[-6, 6]$ |

## B Observation Space

| Observation | Description | Unit | Limits |
|---|---|---|---|
| **Environment Conditions** | | | |
| Altitude | Altitude above the WGS84 geoid. | m | $[0, 150k]$ |
| Mach number $Ma$ | Velocity as ratio of absolute speed and local speed of sound | unitless | $[0, 35]$ |
| Dynamic pressure $\bar{q}$ | Dynamic pressure depends on air speed $V_a$ and air density $\rho$ through $0.5V_a^2\rho$ | Pa | $[0, 10k]$ |
| **Commanded Aerodynamic Angles** | | | |
| The commands are generated using a model of the desired response characteristics (second-order transfer function) in the setpoint generator. Tracking the aerodynamic angles is the primary control objective. | | | |
| $\alpha_{\text{cmd}}(t)$ (angle of attack) | Requested angle between the direction of flight relative to air and the body-fixed frame in the vertical plane. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\beta_{\text{cmd}}(t)$ (sideslip angle) | Requested angle between the direction of flight relative to air and the body-fixed frame in the horizontal plane. The command can be assumed to be zero all of the time. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\mu_{\text{cmd}}(t)$ (bank angle) | Requested angle between the direction of lift and the vertical plane around the direction of flight relative to air. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| **Measured Angles and Angular Rates** | | | |
| $\alpha(t)$ (angle of attack) | Measured angle of attack. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\beta(t)$ (sideslip angle) | Measured angle of sideslip. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\mu(t)$ (bank angle) | Measured bank angle. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\alpha(t-1)$ | Last measured angle of attack. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\beta(t-1)$ | Last measured angle of sideslip. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\mu(t-1)$ | Last measured bank angle. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\dot{\phi}(t)$ (roll rate) | Angular velocity about the forward (x) axis of the vehicle. | rad/s | $[-10, 10]$ |
| $\dot{\theta}(t)$ (pitch rate) | Rate of change of the pitch angle. Corresponds to angular velocity about the intermediate y' axis in a standard Euler angle zyx-rotation sequence. | rad/s | $[-10, 10]$ |
| $\dot{\psi}(t)$ (yaw rate) | Rate of change of the yaw angle. Corresponds to angular velocity about the geodetic z axis in a standard Euler angle zyx-rotation sequence. | rad/s | $[-10, 10]$ |
| **Last Commands** | | | |
| $\delta_{\text{e,cmd}}(t-1)$ | Flap command from last step. See actions for details. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\delta_{\text{a,cmd}}(t-1)$ | Flap command from last step. See actions for details. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| **Baseline Control Commands (Only in Hybrid Control Mode)** | | | |
| $\delta_{\text{e,basecmd}}(t)$ | Flap command of baseline controller. See actions for details. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |

| | | | |
|---|---|---|---|
| $\delta_{\text{a,basecmd}}(t)$ | Flap command of baseline controller. See actions for details. | rad | $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ |
| $\tau_{\text{z,basecmd}}(t)$ | Thruster command of baseline controller. See actions for details. | N | $[-300, 300]$ |
| | **PID Error Components** | | |
| $e_\alpha(t)$ | Proportional error for angle of attack. | rad | $[-\pi, \pi]$ |
| $e_\beta(t)$ | Proportional error for sideslip angle. | rad | $[-\pi, \pi]$ |
| $e_\mu(t)$ | Proportional error for bank angle. | rad | $[-\pi, \pi]$ |
| $\int_0^t e_\alpha(\tau)d\tau$ | Integral error for angle of attack. | $\mathrm{rad\,s}$ | |
| $\int_0^t e_\beta(\tau)d\tau$ | Integral error for sideslip angle. | $\mathrm{rad\,s}$ | |
| $\int_0^t e_\mu(\tau)d\tau$ | Integral error for bank angle. | $\mathrm{rad\,s}$ | |
| $\frac{de_\alpha(t)}{dt}$ | Derivative error for angle of attack. | $\mathrm{rad/s}$ | |
| $\frac{de_\beta(t)}{dt}$ | Derivative error for sideslip angle. | $\mathrm{rad/s}$ | |
| $\frac{de_\mu(t)}{dt}$ | Derivative error for bank angle. | $\mathrm{rad/s}$ | |

# C Design of Reward Function

## C.1 Sensitivity Analysis of Reward Function

For various configurations of the control cost weights (flap cost weights $w_{\delta_e} = w_{\delta_a}$ and thruster cost weight $w_{\tau_z}$), we report the interquartile mean of absolute errors and control costs of the obtained controllers over ten randomly sampled evaluation trajectories after 2.4M training steps with one training run of MR.Q in Figure 7.

We see that increasing the flap cost weights $w_{\delta_e} = w_{\delta_a}$ has the desired effect of reducing the flap control cost without sacrificing tracking performance around a sweet spot. Good policies with low control costs are obtained in the range of $[0.03, 0.2]$. However, a weight of $0.2$ considerably reduces sample efficiency in comparison to lower values, indicating that it makes the learning task harder. Usually about $250k$ to $600k$ training steps are required to follow the trajectory completely during training. With $w_{\delta_e} = w_{\delta_a} = 0.2$, about $1M$ training steps were required. For a weight of $0.5$, $2.2M$ training steps are required. Although this analysis was done after the experiments under nominal conditions, $w_{\delta_e} = w_{\delta_a} = 0.05$ seems to be a good choice in hindsight as it balances sample efficiency, control cost, and control performance.

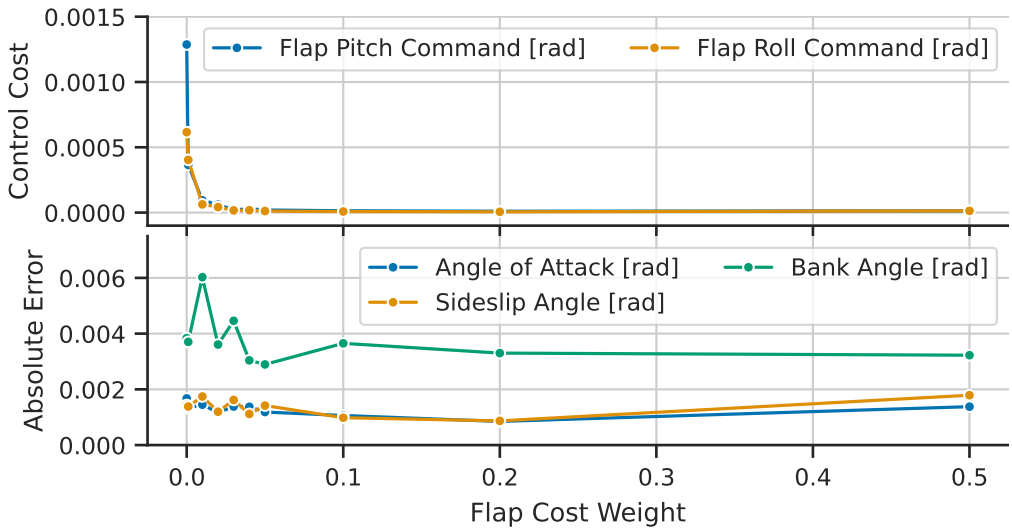
Similarly, increasing the thruster cost weight has the desired effect of reducing control cost. However, the errors of the aerodynamic angles decrease at the same time. The best possible weight is about $w_{\tau_z} = 1$ as increasing the weight further makes the learning problem so hard that it is not possible to follow the whole trajectory after $2.5M$ training steps.

## C.2 Monitoring Training

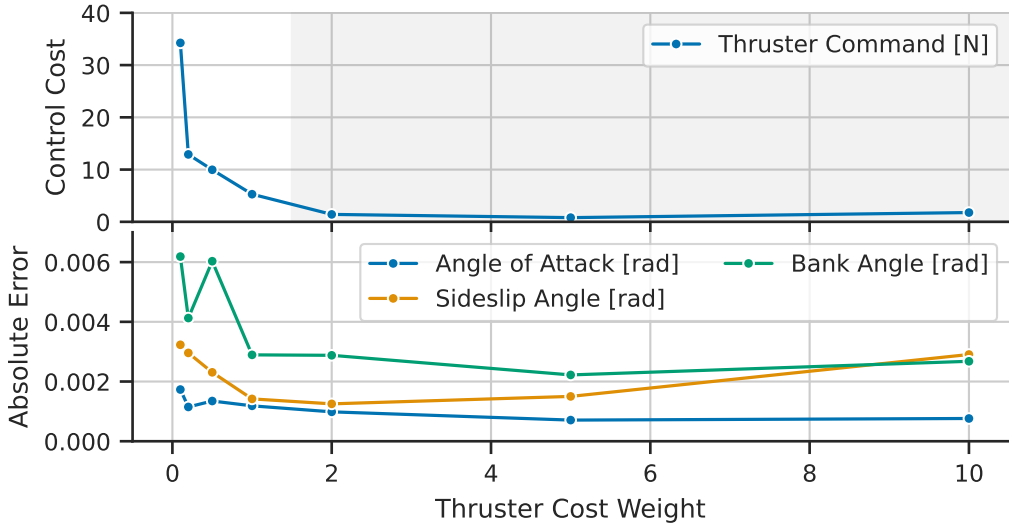
Since the maximum value of $r_t$ is 1, the maximum value of the expected return approximated by the value function is $\frac{1}{1-\gamma}$ under the discounted infinite horizon model, e.g., for $\gamma = 0.99$ the maximum return is 100 if we assume no control costs and that the attitude is tracked perfectly. Hence, we are able to directly track the performance of the policy in comparison to the optimum during training in algorithms based on DDPG (i.e., TD3, TD7, MR.Q), since the loss of the policy is the negative expected return approximated by the value function. The best possible value would be -100 under the condition of perfect approximation.

## C.3 Considered Reward Functions

In non-systematic, preliminary experiments we explored a wide range of reward functions and components of reward functions. We tried the following:




(a) Influence of flap cost weight on control cost and absolute error of aerodynamic angles.




(b) Influence of thruster cost weight on control cost and absolute error of aerodynamic angles. In the gray region, the policy was not able to follow the complete trajectory.

Figure 7: Sensitivity analysis of for weights in the reward function.

- Negative attitude cost: sum of squared differences between commanded and measured aerodynamic angles.
- Adding a penalty for thruster forces.
- Adding a penalty for flap angle changes.
- Adding a penalty for derivative of the aerodynamic angles. We do not use this component anymore because it is often required to change the aerodynamic angles considerably to track the commanded trajectory.
- Adding an extra penalty for large deviations from commanded angles (unsafe state penalty).
- Terminating an episode when the orientation is out of the safe range and adding a large penalty of -1000 or -8000 (stop penalty).

- Adding a reward for each step that the episode is not aborted (healthy reward), which results in significantly longer simulations, however, sometimes at the cost of less accurate tracking of the orientation.
- Replacing angle cost by forward reward, which rewards moving closer to the commanded attitude and is inspired by the reward function of the MuJoCo environments from Gymnasium.
- Initially, we set the weight of the control cost components to 0.001 and then increased it to 0.01, 0.1, 0.5, and 1. Higher weights lead to less switching between flap angles, less extensive use of thrusters, and generally less oscillation of commands. The current individual weights are a good compromise between tracking performance and avoiding excessive overcompensation of errors.
- We tried to remove the healthy reward and use a terminal cost instead, that rewards low altitude of the vehicle when the episode terminates. This did not lead to better results, even when evaluating policies obtained with the previous reward under the new reward.
- We tried to remove the healthy reward and use an attitude that is always positive and more positive when the attitude cost is lower. We call this the attitude reward and it is included in the final version of the reward function. The idea behind this is to substitute the healthy reward and the attitude cost component, since it is always better to survive longer and it is better to have a low attitude cost. The approach was successful, so we do not use any attitude cost, terminal cost, or stop penalty anymore.

# D Simulation Details

## D.1 Common Configuration

**Initial conditions:** The vehicle starts at the outer atmosphere of Earth with a fixed altitude of about $93\,\mathrm{km}$ and fixed velocity of approximately Mach 26.75. The dynamic pressure is low with approximately $54\,\mathrm{Pa}$ in this state.

**Flight path:** To avoid overfitting to a specific flight path, we randomize the trajectory given by the guidance system in each episode by setting two parameters $\gamma$ and $\Delta\chi_{\max}$ (see Table 5). The flight path angle $\gamma$ determines the desired speed of descent. The maximum deviation from the flight course $\Delta\chi_{\max}$ controls bank angle reversals. These reversals are necessary because the bank angle of the vehicle influences not just the speed of descent, but also the direction of the flight. In order to follow a straight path, the bank angle has be reversed when $\Delta\chi$ exceeds the given threshold. This has the effect of modifying the environment dynamics and reward function from the perspective of the agent because it modifies the commanded aerodynamic angles that are part of the state space as well as the PID error components that are part of the observation space.

## D.2 Nominal Conditions

Under nominal conditions, the initial attitude, vehicle mass, the vehicle's inertia tensor, and the actuation are constant as summarized in Table 5.

## D.3 Dynamics Randomization

In comparison to nominal conditions, we additionally randomize the initial attitude for each episode according to Table 5 and the context vector contains the vehicle's mass, inertia tensor, and flap actuator bandwidth.

To model uncertainty in the actuation of the flaps, we modify the flap actuator bandwidth $\omega_0$. Lower values make controlling the vehicle harder. The baseline controller becomes unstable for values under $14\,\mathrm{rad/s}$.

The initial mass is varied in a predefined range according to Table 5. The vehicle's mass defines its gravitational force and, hence, influences the evolution of the flight path. The flight path then defines

Table 5: Configuration of simulation.

| Variable | Nominal | Distribution (Nominal) | Distribution (Dynamics Randomization) |
|---|---|---|---|
| | | Trajectory Parameters | |
| $\gamma$ | $-1.0°$ | $\mathcal{U}(-1.1°, -0.9°)$ | $\mathcal{U}(-1.1°, -0.9°)$ |
| $\Delta\chi_{\max}$ | $3.25°$ | $\mathcal{U}(1.5°, 5°)$ | $\mathcal{U}(1.5°, 5°)$ |
| | | Initial Attitude | |
| $\alpha(0)$ | $45.024°$ | constant | $\mathcal{U}(\alpha_{0,\text{nominal}} - 5°, \alpha_{0,\text{nominal}} + 5°)$ |
| $\beta(0)$ | $0.046°$ | constant | $\mathcal{U}(\beta_{0,\text{nominal}} - 5°, \beta_{0,\text{nominal}} + 5°)$ |
| $\mu(0)$ | $61.141°$ | constant | $\mathcal{U}(\mu_{0,\text{nominal}} - 5°, \mu_{0,\text{nominal}} + 5°)$ |
| | | Inertia | |
| $m_0$ | $1640\,\text{kg}$ | constant | context, $m_0 \in [1312, 1968]$ kg |
| $\boldsymbol{I}$ | $\text{diag}\begin{pmatrix}492\\2247\\2358\end{pmatrix}\text{kg}\cdot\text{m}^2$ | constant | context, see main text |
| | | Actuation | |
| $\omega_0$ | $30\,\text{rad/s}$ | constant | context, $\omega_0 \in [12, 30]$ rad/s |

the commanded aerodynamic angles, as it is required to increase or decrease the lift to compensate for the change in mass.

To generate physically plausible inertia tensors, we define perturbations of the nominal inertia tensor with the following procedure. Since the nominal inertia tensor is diagonal, we apply a perturbation directly to the principal moments and then rotate the inertia tensor. To modify the principal moment, we use a vector $\boldsymbol{a} \in [-0.1, 0.1]^3$, and to modify the rotation, we use a rotation vector $\boldsymbol{\omega} = \theta\hat{\boldsymbol{\omega}} \in \mathbb{R}^3$ with an angle $\theta \in [-10°, 10°]$ and axis $\hat{\boldsymbol{\omega}} \in S^2 \subset \mathbb{R}^3$. With the exponential map $\text{Exp} : \mathbb{R}^3 \to SO(3)$ (Solà et al., 2018), we compute the rotation matrix from the rotation vector so that the nominal inertia is perturbed with

$$\boldsymbol{I} = \text{Exp}(\boldsymbol{\omega}) \cdot \text{diag}(\mathbf{1}_3 + \boldsymbol{a}) \cdot \boldsymbol{I}_{\text{nominal}} \cdot \text{Exp}(\boldsymbol{\omega})^T,$$

where $\text{diag}(\boldsymbol{x})$ transforms a vector $\boldsymbol{x} \in \mathbb{R}^3$ to a diagonal matrix. We generate a random set of rotation axes by sampling from a 3D standard normal distribution $\mathcal{N}(\mathbf{0}_3, \boldsymbol{I}_{3\times3})$ and normalizing the vector.

# E Hyperparameters of RL Algorithms

| | Hyperparameter | MR.Q | TD7 | TD3 | SAC |
|---|---|---|---|---|---|
| Common | Discount factor $\gamma$ | 0.99 | 0.99 | 0.99 | 0.99 |
| | Replay buffer capacity | 2M | 2M | 2M | 2M |
| | Mini-batch size | 256 | 256 | 256 | 256 |
| | Target update frequency $T_{\text{target}}$ | 250 | 250 | - | - |
| | Target update rate $\tau$ | - | - | $5 \cdot 10^{-3}$ | $5 \cdot 10^{-3}$ |
| MR.Q | Dynamics loss weight $\lambda_{\text{Dynamics}}$ | 1 | - | - | - |
| | Reward loss weight $\lambda_{\text{Reward}}$ | 0.1 | - | - | - |
| | Terminal loss weight $\lambda_{\text{Terminal}}$ | 0.1 | - | - | - |
| | Activation loss weight $\lambda_{\text{pre-activ}}$ | $1e-5$ | - | - | - |
| | Encoder horizon $H_{\text{Enc}}$ | 5 | - | - | - |
| | Multi-step returns horizon $H_Q$ | 3 | - | - | - |
| TD3 | Target policy noise $\sigma$ | $\mathcal{N}(0, 0.2^2)$ | $\mathcal{N}(0, 0.2^2)$ | $\mathcal{N}(0, 0.2^2)$ | - |
| | Target policy noise clipping $c$ | $(-0.3, 0.3)$ | $(-0.5, 0.5)$ | $(-0.5, 0.5)$ | - |
| | Initial random exploration steps | 10k | 25k | 25k | 5k |
| | Exploration noise | $\mathcal{N}(0, 0.2^2)$ | $\mathcal{N}(0, 0.1^2)$ | $\mathcal{N}(0, 0.2^2)$ | - |
| | Policy delay | 1 | 2 | 2 | 1 |
| LAP | Probability smoothing $\alpha$ | 0.4 | 0.4 | - | - |
| | Minimum priority | 1 | 1 | - | - |
| Value Network | Optimizer | AdamW[1] | Adam[2] | Adam | Adam |
| | Learning rate | $3e-4$ | $3e-4$ | $3e-4$ | $1e-3$ |
| | Hidden dim | 512 | 256 | 256 | 256 |
| | Activation function | ELU[3] | ELU | ReLU | ReLU |
| | Weight initialization | Xavier uniform[4] | LeCun normal[5] | LeCun normal | LeCun normal |
| | Bias initialization | 0 | 0 | 0 | 0 |
| | Gradient clip norm | 20 | - | - | - |
| Policy Network | Optimizer | AdamW | Adam | Adam | Adam |
| | Learning rate | $3e-4$ | $3e-4$ | $3e-4$ | $3e-4$ |
| | Hidden dim | 512 | 256 | 256 | 256 |
| | Activation function | ReLU[6] | ReLU | ReLU | Swish[7] |
| | Weight initialization | Xavier uniform | LeCun normal | LeCun normal | LeCun normal |
| | Bias initialization | 0 | 0 | 0 | 0 |
| Encoder Network | Optimizer | AdamW | Adam | - | - |
| | Learning rate | $1e-4$ | $3e-4$ | - | - |
| | Weight decay | $1e-4$ | - | - | - |
| | $\mathbf{z}_s$ dim | 512 | 256 | - | - |
| | $\mathbf{z}_{sa}$ dim | 512 | 256 | - | - |
| | $\mathbf{z}_a$ dim | 256 | 256 | - | - |
| | Hidden dim | 512 | 256 | - | - |
| | Activation function | ELU | ELU | - | - |

[1] Loshchilov & Hutter (2019)
[2] Kingma & Ba (2015)
[3] Clevert et al. (2016)
[4] Glorot & Bengio (2010)
[5] Klambauer et al. (2017)
[6] Glorot et al. (2011)
[7] Ramachandran et al. (2017)

| | | | | | |
|---|---|---|---|---|---|
| Encoder (cont.) | Weight initialization | Xavier uniform | LeCun normal | - | - |
| | Bias initialization | 0 | 0 | - | - |
| | Reward bins | 65 | - | - | - |
| | Reward range | $[-10, 10]$ (effective: $[-22\text{k}, 22\text{k}]$) | - | - | - |
| SAC | Entropy optimizer | - | - | - | Adam |
| | Entropy learning rate | - | - | - | $1\text{e}-3$ |

While the MR.Q algorithm is a theoretically sound approach (Fujimoto et al., 2025), we had to modify it slightly in comparison to its original implementation. In the original implementation, the dynamics loss compares the latent state computed from unrolled dynamics $z_{sa}^T W_p \in \mathbb{R}$ to the latent target state obtained by applying the state encoder $f_\omega$ to a state $s'$ from the replay buffer $f_\omega(s') = z_{s'}$, in which the state encoder applies layer normalization and ELU, hence, limits the output to $z_{s'} \in \mathbb{R}_{\geq -1}$. Initially, we had this as a bug in our implementation of MR.Q, but we found that removing ELU from the last layer of the state encoder enhances performance. Hence, our implementation does apply layer normalization without ELU after the last layer of the state encoder, which resembles the behavior of the state encoder in TD7 (Fujimoto et al., 2023), in which AvgL1Norm is supposed to protect from monotonic growth of the features.

## F Hyperparameters of Task Scheduling Algorithms

For task scheduling, we sample a discrete set of contexts and create one replay buffer per context. For each update, we first randomly sample the replay buffer and then a batch of samples from this replay buffer. This ensures that we continue training each task even though a task is not selected anymore.

In comparison to the original SMT (Cho et al., 2024), we do not perform any network resets, which does not seem to be necessary in MR.Q. Furthermore, we do not learn a task encoding.

| | Hyperparameter | Active MT | SMT |
|---|---|---|---|
| Common | RL algorithm | MR.Q | MR.Q |
| | Number of tasks $\lvert\mathcal{T}\rvert$ | 50 | 50 |
| | Training steps | $15M$ | $15M$ |
| | Scheduling interval | 1 episode | 1 episode |
| | Replay buffer size per task | $\max\left(200k, \frac{2M}{\lvert\mathcal{T}\rvert}\right)$ | $\max\left(200k, \frac{2M}{\lvert\mathcal{T}\rvert}\right)$ |
| D-UCB | Upper bound for task selection reward $r_{\max}$ | $10.8k$ | - |
| | Discount factor $\gamma_{\mathcal{T}}$ | 0.95 | - |
| | Padding function strength $\xi$ | $1\text{e}-4$ | - |
| SMT | Stage 1 budget $B_1$ | - | $12.75M$ |
| | Stage 2 budget $B_2$ | - | $2.25M$ |
| | $\kappa$ | - | 0.7 |
| | $K$ | - | 8 |
| | Threshold for unsolved tasks $m$ | - | $-100$ |
| | Threshold for solved tasks $M$ | - | $10k$ |
| | Reset interval | - | No resets |

# G Ablation Studies

We perform ablation studies to find out why MR.Q works in this application. To evaluate the effect of algorithm components and hyperparameters in the MR.Q algorithm, we train for $2.4M$ steps in the single task setting with one seed, and measure the accumulated reward over ten evaluation episodes. Although the results of these studies are not statistically robust with only one training seed, we expect the conclusions to be correct, since MR.Q is a robust algorithm with mostly consistent behavior across random seeds and the results are unambiguous.

## G.1 Effect of Policy Activation Penalty

| | Activation loss weight $\lambda_{\text{pre-activ}}$ | | | |
|---|---|---|---|---|
| | 0 | $10^{-6}$ | $10^{-5}$ (default) | $10^{-4}$ |
| Metric | IQM $\pm$ interquartile std. dev. | | | |
| Return | 10587 $\pm$ 166 | 10467 $\pm$ 163 | 10492 $\pm$ 181 | 10570 $\pm$ 193 |
| Flap Roll Cmd. [rad] | 1.6e-5 $\pm$ 3.1e-5 | 1.6e-5 $\pm$ 10e-5 | 2.0e-5 $\pm$ 5.0e-5 | 2.1e-5 $\pm$ 6.0e-5 |
| Flap Pitch Cmd. [rad] | 0.8e-6 $\pm$ 2.5e-5 | 1.4e-5 $\pm$ 8.7e-5 | 1.2e-5 $\pm$ 3.7e-5 | 1.0e-5 $\pm$ 4.7e-5 |
| Thruster Cmd. [N] | 2.97 $\pm$ 3.03 | 4.31 $\pm$ 3.6 | 5.29 $\pm$ 4.69 | 4.60 $\pm$ 4.42 |

Similarly to the control cost of the reward function, we expected that the activation loss of the policy forces the policy to reduce the magnitude of control commands. Other algorithms do not have this component and learn policies that often result in bang-bang control. However, setting $\lambda_{\text{pre-activ}} = 0$ does not decrease the performance. The activation penalty is not contributing to the considerable performance improvement of MR.Q in comparison to TD3 and TD7.

## G.2 Effect of Model-Based Representation Learning

| Encoder loss | $\lambda_{\text{Dynamics}}$ | $\lambda_{\text{Reward}}$ | $\lambda_{\text{Terminal}}$ | Return (IQM $\pm$ interquartile std. dev.) |
|---|---|---|---|---|
| No loss | 0 | 0 | 0 | 2987$\pm$277 |
| No reward loss | 1 | 0 | 0.1 | 10455$\pm$212 |
| No dynamics loss | 0 | 0.1 | 0.1 | 10630$\pm$193 |
| Default | 1 | 0.1 | 0.1 | 10492$\pm$181 |

The model-based representation learned by the encoder is the main difference of MR.Q in comparison to TD3 and TD7. Deactivating the loss for the encoder completely, i.e., using random features, results in a performance similar to TD3 and TD7, which confirms that the model-based representation of MR.Q is important for the application. Interestingly, deactivating either the dynamics loss or the reward loss has a negligible effect on the final performance. Note that these results are specific for this application, in which a useful and meaningful reward can be computed in every step. The behavior might be different for environments with sparse reward.

## G.3 Effect of Horizon for Multi-Step Returns in Critic Loss

| Multi-step returns horizon $H_Q$ | 1 | 3 (default) | 5 | 10 |
|---|---|---|---|---|
| Return (IQM) | 3184 | 10492 | 9250 | 4483 |

As EVE-TD3 (Liu et al., 2022a) uses $H_Q = 30$, we are interested in the effect of the hyperparameter in this application with MR.Q. The horizon controls the bias-variance trade-off in the target value for the action-value function (see also Schulman et al. (2016) for a discussion in the context of advantage

function estimation). The results demonstrate that the multi-step return target for the critic is a main reason for the good performance of MR.Q in this application. Furthermore, MR.Q's default value of $H_Q = 3$ is also an optimum in this application. Considering the control frequencies of $14\,\mathrm{Hz}$ in our controller and $50\,\mathrm{Hz}$ in the controller of Liu et al. (2022a) (control step size of $20\,\mathrm{ms}$), we use a horizon of ca. $0.214\,\mathrm{s}$ and Liu et al. (2022a) use a horizon of $0.6\,\mathrm{s}$ in simulation time, which is in the same order of magnitude.

## H Code

Although we cannot make the code for the simulation and experiments publicly available, the source code for all reinforcement learning algorithms used in our experiments is available at `https://github.com/mlaux1/rl-blox` (version 0.5.7, JAX version 0.7.0, flax version 0.11.1).

For statistical comparison of RL algorithms, we rely on the open-source library rliable of Agarwal et al. (2021) available at `https://github.com/google-research/rliable`.

## I Computation Time

One run with 10 million time steps in the single task setting takes about 48 h with MR.Q and TD7, 24 h with TD3, and 30 h with SAC. In total, we analyzed 10 training seeds and two control architectures (pure RL and additive hybrid) for each algorithm, resulting in $(48 + 48 + 24 + 30) \times 10 \times 2 = 3000$ hours of wall-clock training time to generate the learning curves. However, the runs could be parallelized. In addition, we used another script to run the performance evaluation of the stored checkpoints.

# J Application-Specific Performance Analysis Under Nominal Conditions

## J.1 Tracking Performance Over Full Trajectory

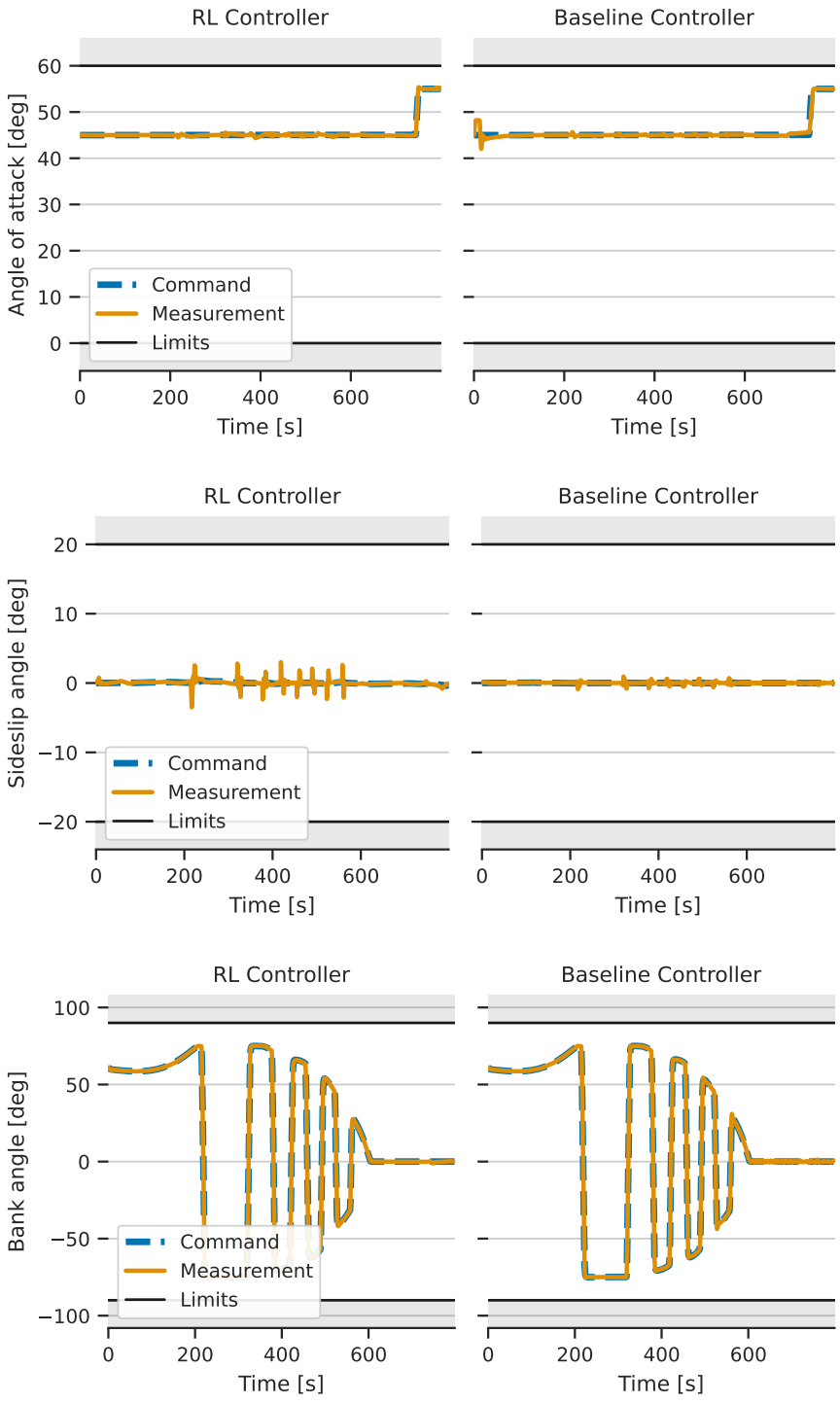



Figure 8: Comparison of commanded and measured aerodynamic angles for pure RL controller based on MR.Q trained under nominal conditions and baseline controller.

### J.2 Tracking Performance at Critical Points

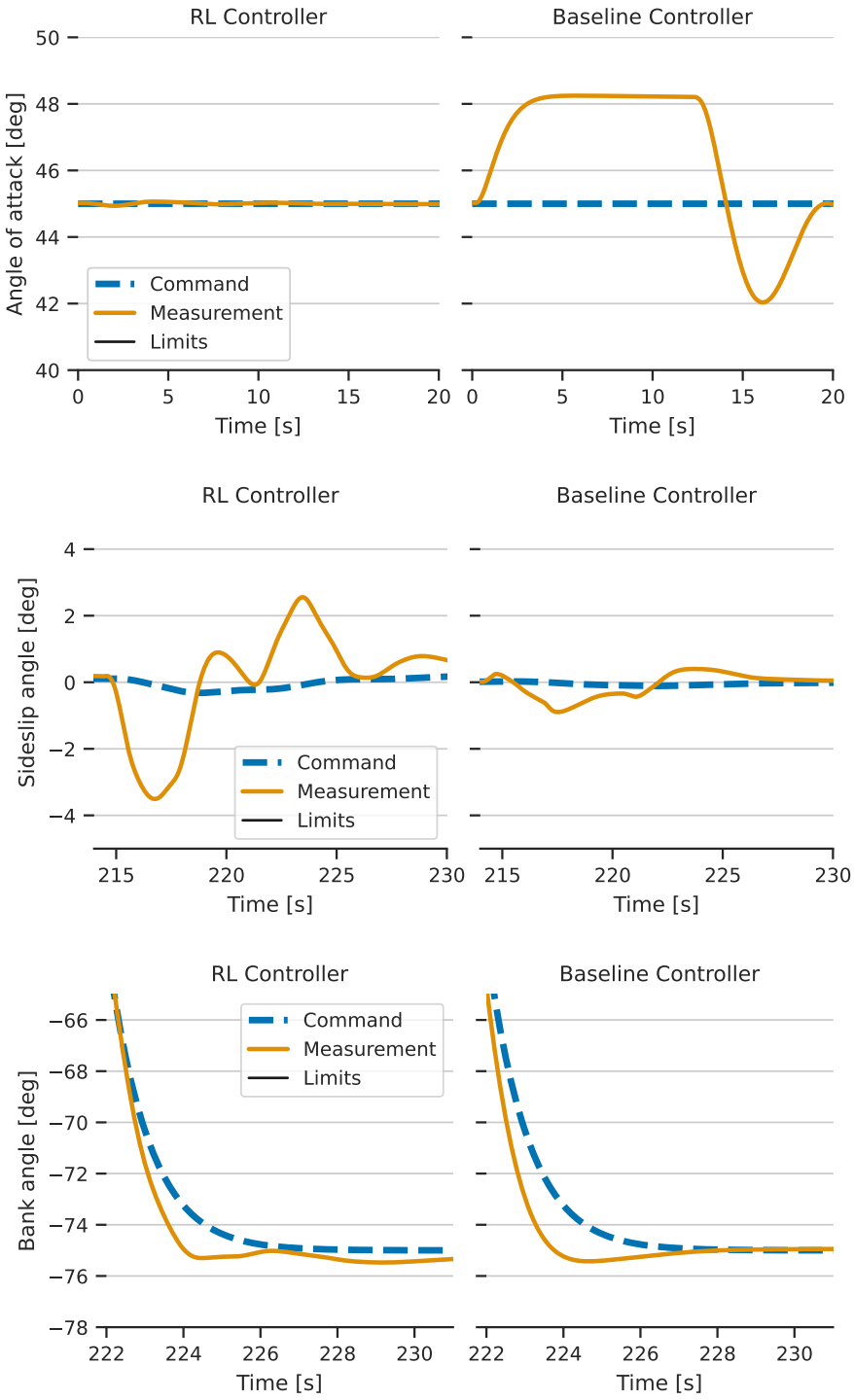



Figure 9: Comparison of commanded and measured aerodynamic angles for pure RL controller based on MR.Q trained under nominal conditions and baseline controller. Zoomed in at critical points of the trajectory.

### J.3 Commands

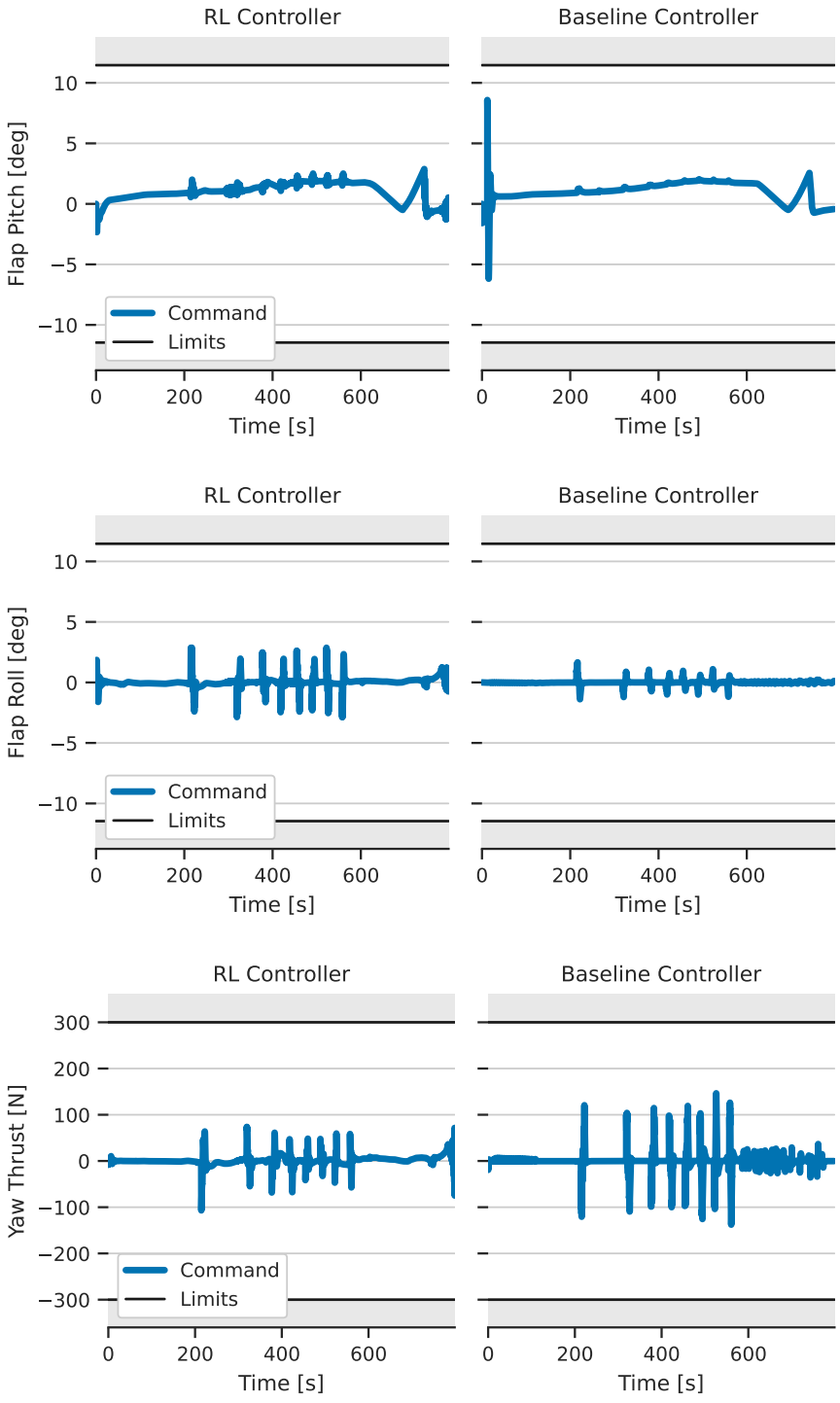



Figure 10: Control commands for pure RL controller based on MR.Q trained under nominal conditions and baseline controller.

### J.4 Error Distribution

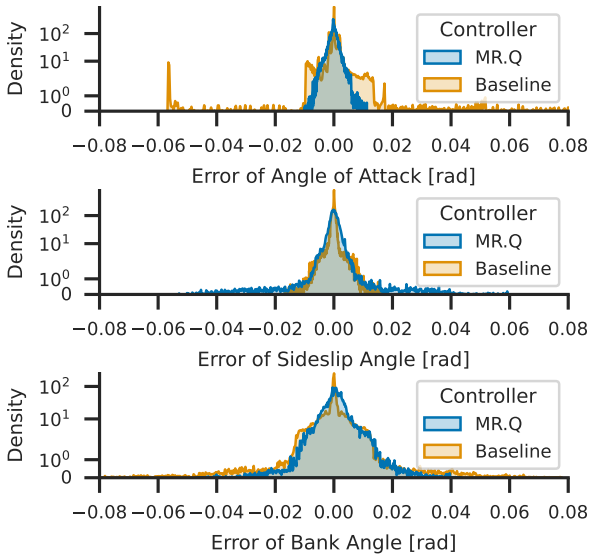



Figure 11: Distributions of aerodynamic angle errors. The results are obtained for the best policy and 10 evaluation seeds. A kernel density estimation was applied to the sampled errors. Note that density is plotted on a logarithmic scale to highlight the tails. The peak of the baseline controller around 0 is more pronounced on a linear scale.

### J.5 Distribution of Control Cost

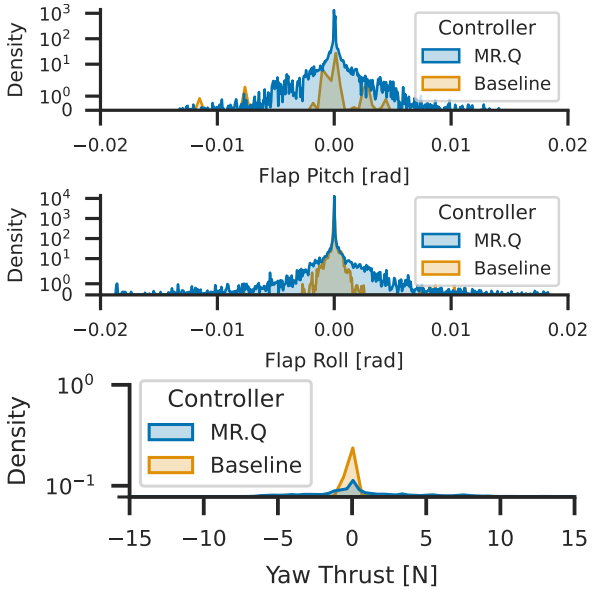



Figure 12: Distributions of control commands. The results are obtained for the best policy and 10 evaluation seeds. A kernel density estimation was applied to the sampled control commands. Note that density is plotted on a logarithmic scale to highlight the tails. The peak of the baseline controller around 0 is more pronounced on a linear scale.

## K Error Percentiles After Training With Dynamics Randomization

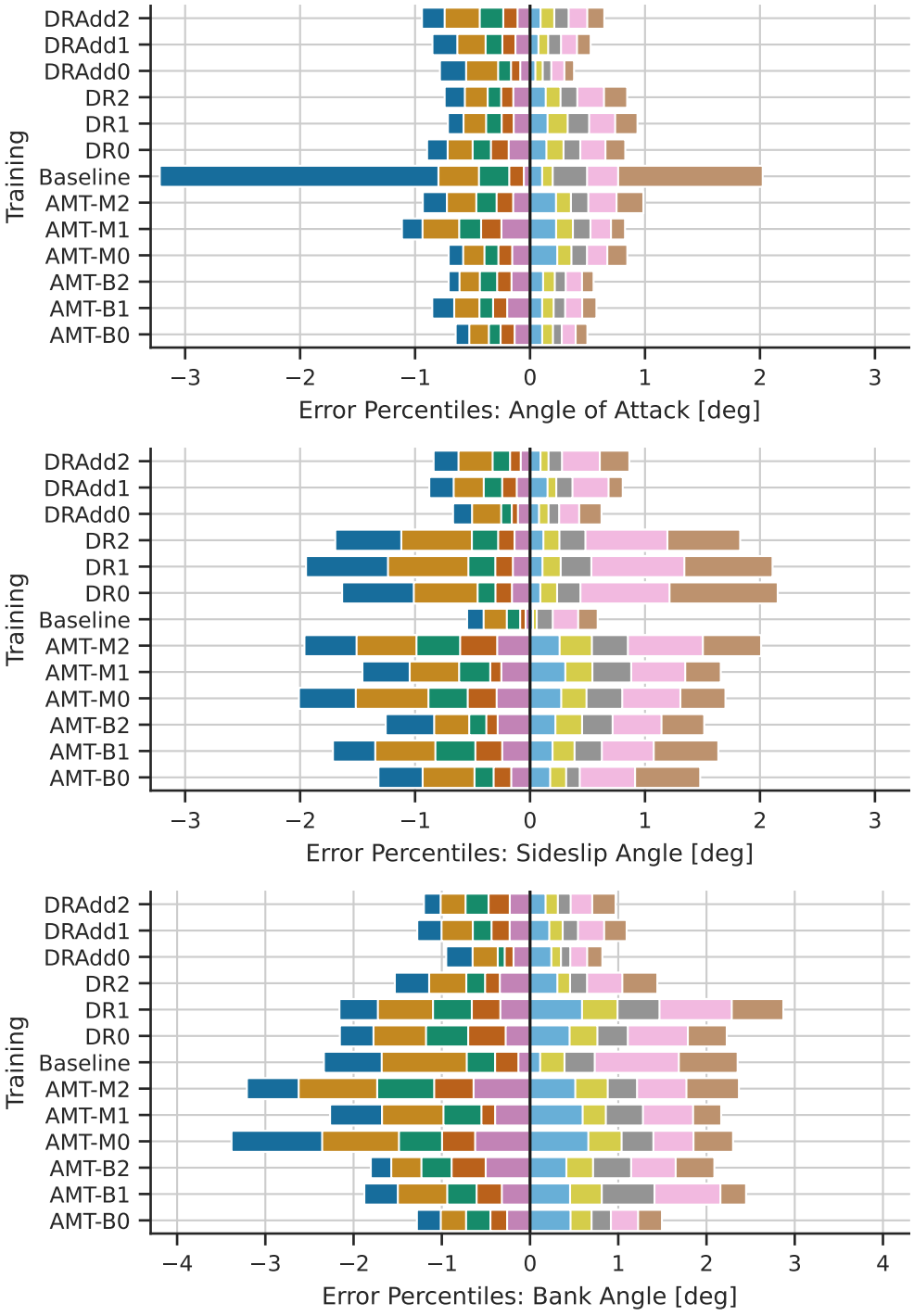



Figure 13: Distribution of aerodynamic angle errors evaluated over 100 test contexts. The overlapping bars correspond to the following percentiles of the distribution: 1, 2, 5, 10, 20, 80, 90, 95, 98, 99.

## L Domain-Specific Performance Analysis

Table 10: Evaluation of robustness of controllers with domain-specific metrics. Cell background color indicates best values or values better than the baseline (only if success rate is >0%). Appendix K contains more details on the error distributions.

| Training | Seed | Success | Median | 90-Perc. | 95-Perc. | 98-Perc. | Median | 90-Perc. | 95-Perc. | 98-Perc. | Median | 90-Perc. | 95-Perc. | 98-Perc. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\lvert e_\alpha \rvert$ [deg] | | | | $\lvert e_\beta \rvert$ [deg] | | | | $\lvert e_\mu \rvert$ [deg] | | | |
| Baseline | - | 79% | 0.05 | 0.46 | 0.67 | 2.96 | 0.02 | 0.20 | 0.36 | 0.57 | 0.06 | 0.72 | 1.43 | 2.35 |
| DR (Add.) | 0 | 93% | 0.05 | 0.23 | 0.34 | 0.59 | 0.07 | 0.25 | 0.41 | 0.65 | 0.17 | 0.42 | 0.59 | 0.87 |
| DR (Add.) | 1 | 93% | 0.07 | 0.37 | 0.56 | 0.83 | 0.07 | 0.30 | 0.53 | 0.85 | 0.15 | 0.60 | 0.84 | 1.11 |
| DR (Add.) | 2 | 94% | 0.07 | 0.33 | 0.47 | 0.71 | 0.11 | 0.39 | 0.60 | 0.84 | 0.17 | 0.60 | 0.84 | 1.19 |
| DR | 0 | 100% | 0.12 | 0.47 | 0.63 | 0.86 | 0.09 | 0.45 | 0.88 | 1.87 | 0.23 | 1.14 | 1.63 | 2.19 |
| DR | 1 | 99% | 0.10 | 0.45 | 0.62 | 0.84 | 0.10 | 0.53 | 1.06 | 2.03 | 0.34 | 1.31 | 1.82 | 2.57 |
| DR | 2 | 98% | 0.10 | 0.39 | 0.55 | 0.80 | 0.09 | 0.49 | 0.95 | 1.76 | 0.25 | 0.69 | 0.98 | 1.49 |
| RR | 0 | 98% | 0.15 | 0.45 | 0.58 | 0.75 | 0.20 | 0.68 | 0.96 | 1.42 | 0.30 | 1.21 | 1.52 | 1.98 |
| RR | 1 | 96% | 0.25 | 0.71 | 0.82 | 0.95 | 0.26 | 0.97 | 1.21 | 1.70 | 0.57 | 1.37 | 1.78 | 2.36 |
| RR | 2 | 95% | 0.10 | 0.33 | 0.43 | 0.58 | 0.15 | 0.46 | 0.73 | 1.27 | 0.26 | 0.87 | 1.11 | 1.35 |
| SMT | 0 | 0% | 0.05 | 0.27 | 0.43 | 0.67 | 0.23 | 0.76 | 1.24 | 1.80 | 0.40 | 0.94 | 1.48 | 2.57 |
| SMT | 1 | 0% | 0.08 | 0.43 | 0.62 | 1.00 | 0.18 | 1.07 | 1.79 | 2.90 | 0.28 | 2.04 | 2.90 | 3.62 |
| SMT | 2 | 0% | 0.17 | 0.56 | 0.90 | 1.36 | 0.23 | 1.13 | 1.78 | 2.96 | 0.38 | 1.60 | 2.28 | 3.52 |
| AMT-M | 0 | 94% | 0.15 | 0.45 | 0.59 | 0.76 | 0.23 | 0.84 | 1.27 | 1.85 | 0.49 | 1.43 | 1.89 | 2.73 |
| AMT-M | 1 | 100% | 0.17 | 0.57 | 0.75 | 1.01 | 0.21 | 0.75 | 1.10 | 1.59 | 0.40 | 1.14 | 1.64 | 2.20 |
| AMT-M | 2 | 94% | 0.15 | 0.49 | 0.67 | 0.96 | 0.19 | 0.92 | 1.35 | 1.99 | 0.52 | 1.41 | 2.08 | 2.90 |
| AMT-B | 0 | 96% | 0.08 | 0.32 | 0.43 | 0.59 | 0.13 | 0.46 | 0.78 | 1.39 | 0.33 | 0.85 | 1.04 | 1.41 |
| AMT-B | 1 | 96% | 0.11 | 0.38 | 0.51 | 0.74 | 0.17 | 0.72 | 1.10 | 1.69 | 0.30 | 1.17 | 1.71 | 2.27 |
| AMT-B | 2 | 99% | 0.10 | 0.37 | 0.50 | 0.66 | 0.20 | 0.62 | 0.91 | 1.41 | 0.36 | 1.19 | 1.51 | 1.95 |
| | | | $\lvert \Delta\delta_{e,max} \rvert$ [deg] | | | | $\lvert \Delta\delta_{a,max} \rvert$ [deg] | | | | $\lvert \tau_z \rvert$ [N] | | | |
| Baseline | - | 79% | 0.00 | 0.01 | 0.01 | 0.06 | 0.00 | 0.01 | 0.03 | 0.06 | 0.31 | 16.11 | 71.70 | 98.1 |
| DR (Add.) | 0 | 93% | 0.00 | 0.10 | 0.21 | 0.37 | 0.00 | 0.07 | 0.12 | 0.21 | 5.49 | 18.25 | 53.15 | 89.0 |
| DR (Add.) | 1 | 93% | 0.02 | 0.26 | 0.43 | 0.69 | 0.01 | 0.13 | 0.22 | 0.38 | 8.46 | 25.73 | 50.76 | 88.2 |
| DR (Add.) | 2 | 94% | 0.00 | 0.21 | 0.38 | 0.66 | 0.00 | 0.15 | 0.26 | 0.43 | 6.24 | 31.23 | 60.88 | 93.0 |
| DR | 0 | 100% | 0.02 | 0.43 | 0.56 | 0.71 | 0.01 | 0.27 | 0.42 | 0.63 | 6.15 | 23.85 | 33.44 | 51.1 |
| DR | 1 | 99% | 0.01 | 0.39 | 0.60 | 0.84 | 0.00 | 0.21 | 0.34 | 0.54 | 6.28 | 30.23 | 49.73 | 72.8 |
| DR | 2 | 98% | 0.02 | 0.48 | 0.66 | 0.86 | 0.01 | 0.26 | 0.46 | 0.88 | 4.95 | 23.49 | 39.33 | 67.0 |
| RR | 0 | 98% | 0.09 | 0.56 | 0.68 | 0.83 | 0.04 | 0.31 | 0.46 | 0.65 | 10.95 | 37.90 | 57.59 | 100.6 |
| RR | 1 | 96% | 0.13 | 0.66 | 0.79 | 0.92 | 0.09 | 0.50 | 0.64 | 0.78 | 15.56 | 56.70 | 75.46 | 107.0 |
| RR | 2 | 95% | 0.06 | 0.55 | 0.75 | 0.95 | 0.04 | 0.36 | 0.48 | 0.63 | 10.26 | 37.39 | 55.20 | 85.7 |
| SMT | 0 | 0% | 0.04 | 0.85 | 1.01 | 1.06 | 0.02 | 0.32 | 0.54 | 0.93 | 3.31 | 53.46 | 76.66 | 126.8 |
| SMT | 1 | 0% | 0.08 | 0.70 | 0.84 | 0.96 | 0.03 | 0.47 | 0.71 | 1.06 | 3.23 | 47.38 | 78.06 | 127.8 |
| SMT | 2 | 0% | 0.16 | 0.80 | 0.90 | 0.98 | 0.14 | 1.06 | 1.07 | 1.07 | 17.23 | 69.22 | 102.72 | 152.5 |
| AMT-M | 0 | 94% | 0.03 | 0.35 | 0.55 | 0.82 | 0.03 | 0.37 | 0.53 | 0.72 | 15.22 | 47.83 | 71.20 | 115.6 |
| AMT-M | 1 | 100% | 0.05 | 0.47 | 0.60 | 0.76 | 0.04 | 0.33 | 0.45 | 0.61 | 13.24 | 54.50 | 74.81 | 103.8 |
| AMT-M | 2 | 94% | 0.21 | 0.77 | 0.91 | 1.03 | 0.14 | 0.67 | 0.91 | 1.06 | 14.74 | 64.18 | 90.91 | 139.2 |
| AMT-B | 0 | 96% | 0.01 | 0.30 | 0.43 | 0.58 | 0.01 | 0.25 | 0.40 | 0.62 | 7.69 | 30.31 | 49.38 | 89.6 |
| AMT-B | 1 | 96% | 0.12 | 0.65 | 0.79 | 0.95 | 0.08 | 0.42 | 0.56 | 0.77 | 10.37 | 47.69 | 71.30 | 107.5 |
| AMT-B | 2 | 99% | 0.11 | 0.62 | 0.78 | 0.96 | 0.05 | 0.34 | 0.50 | 0.72 | 16.36 | 51.86 | 69.21 | 100.8 |

## M Individual Learning Curves Under Nominal Conditions

Plots throughout the paper show the IQM for the returns, which remove outliers, e.g., failed runs. Here we plot individual learning curves for training under nominal conditions without outlier filtering (experiments from Section 4.1.1). For each training run, we average the performance over ten evaluation episodes. For comparison, we plot the performance of the baseline controller, no control, and the IQM of all training runs. Individual learning curves for MR.Q indicate that the training process is slightly unstable, but all runs surpass the baseline controller at some point.

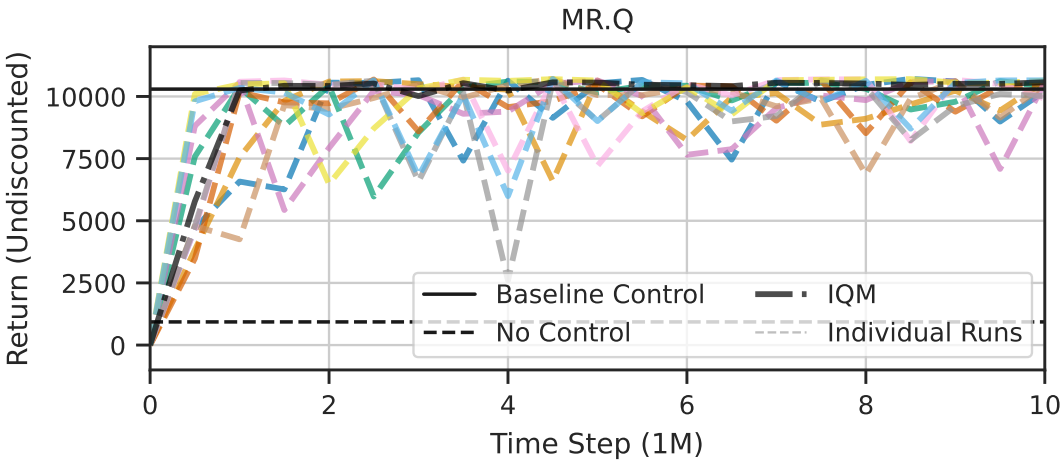



Figure 14: Individual learning curves of MR.Q (Only RL).

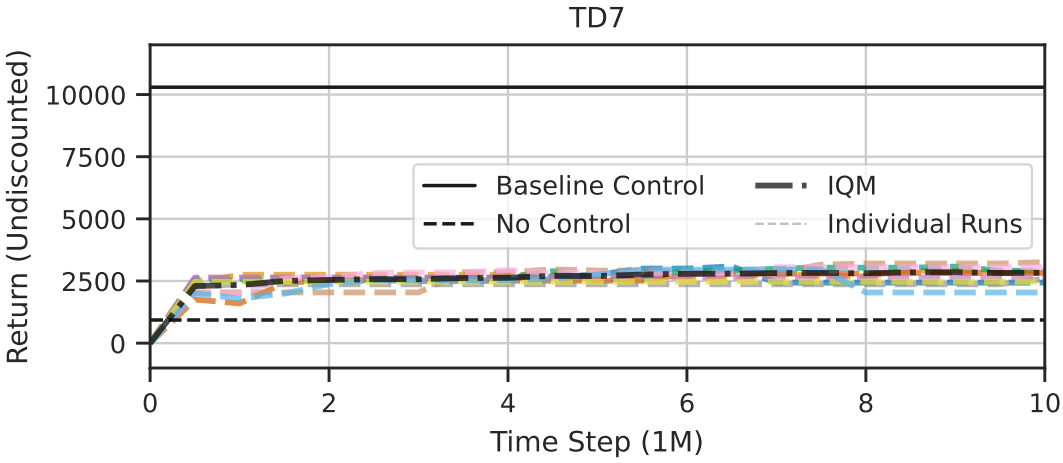



Figure 15: Individual learning curves of TD7 (Only RL).

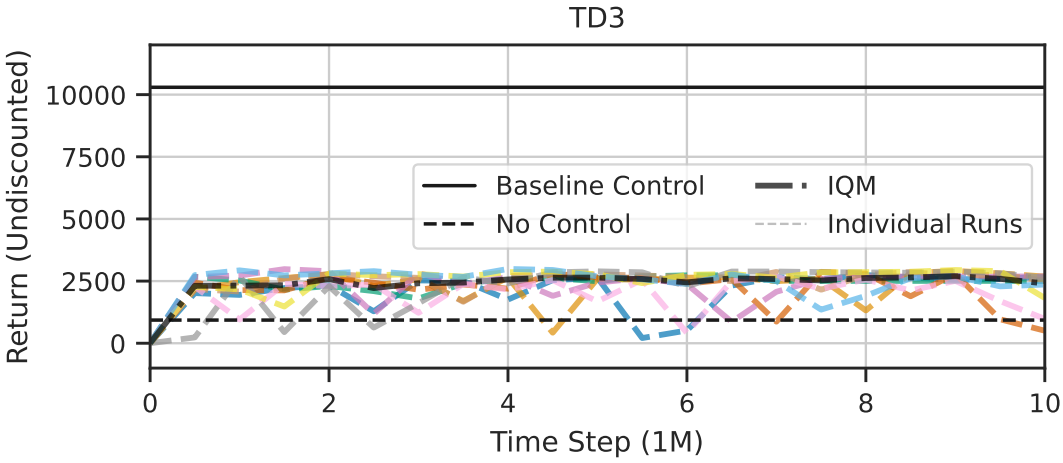



Figure 16: Individual learning curves of TD3 (Only RL).

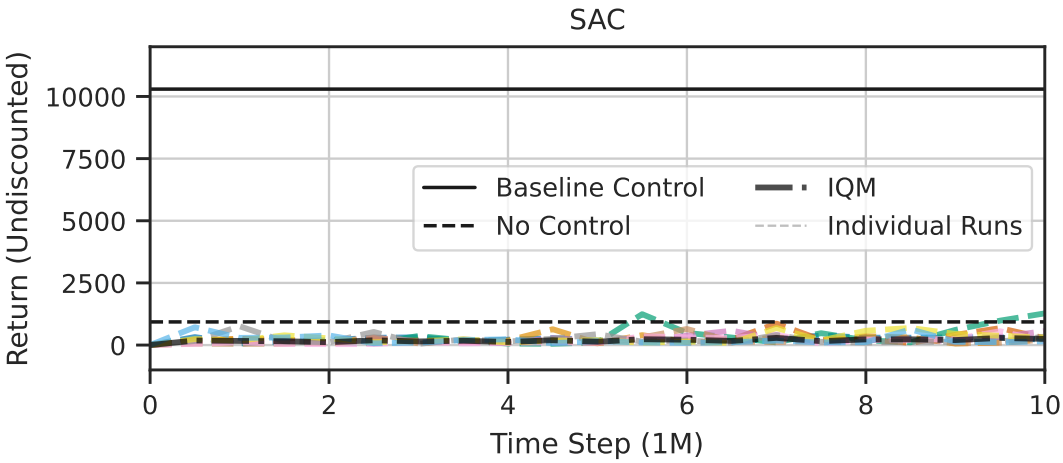



Figure 17: Individual learning curves of SAC (Only RL).

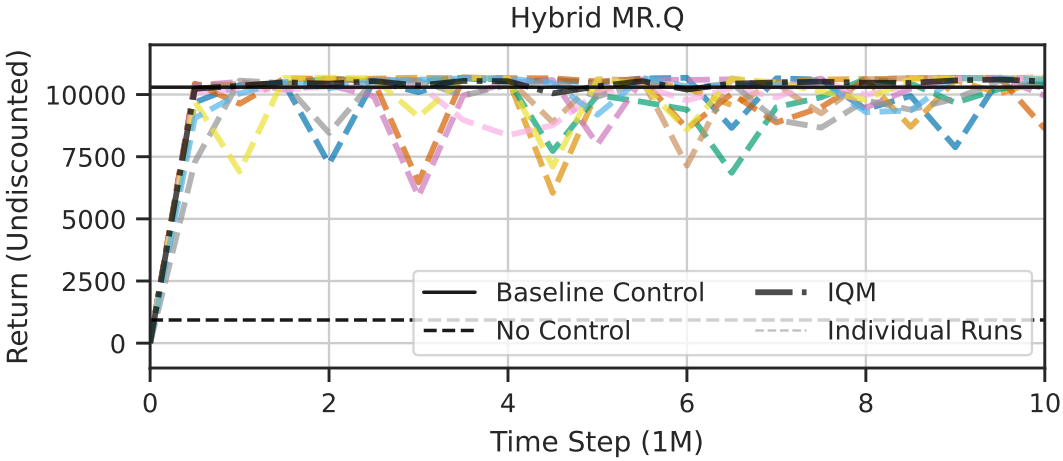



Figure 18: Individual learning curves of MR.Q (Additive Hybrid).

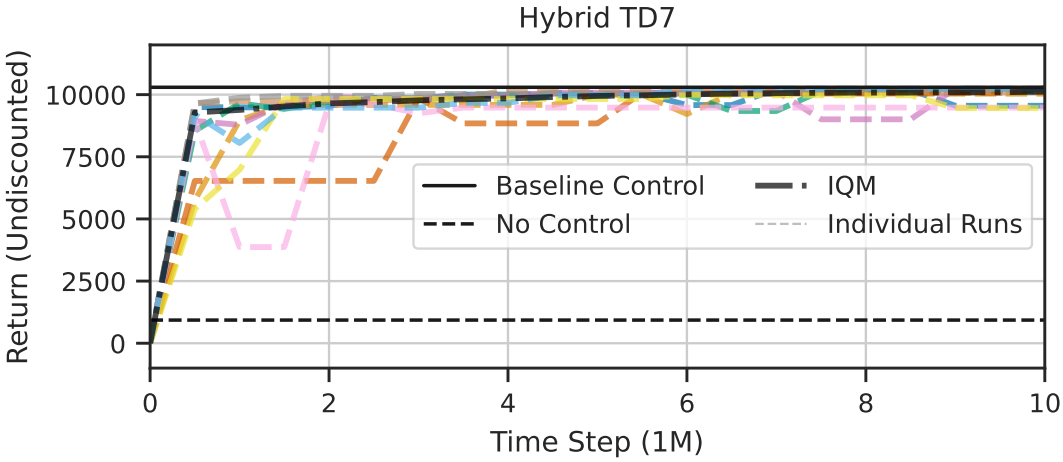



Figure 19: Individual learning curves of TD7 (Additive Hybrid).

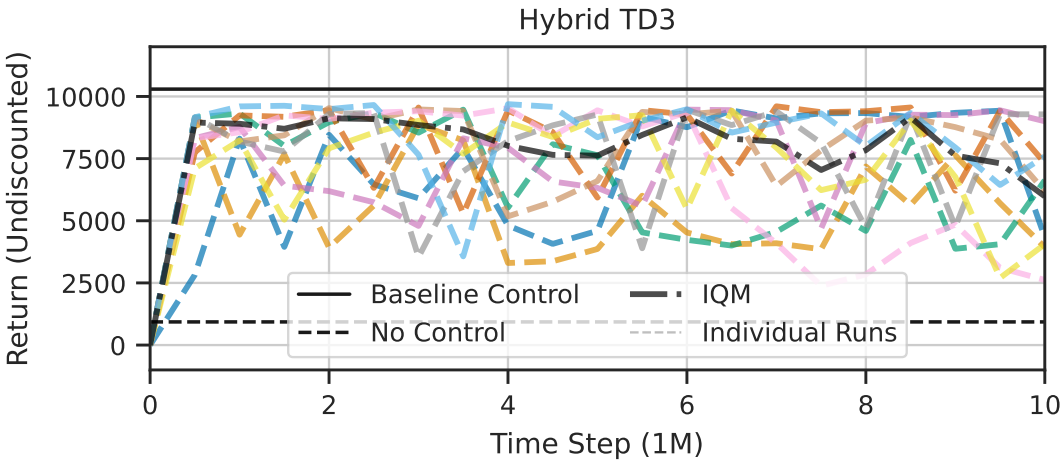



Figure 20: Individual learning curves of TD3 (Additive Hybrid).

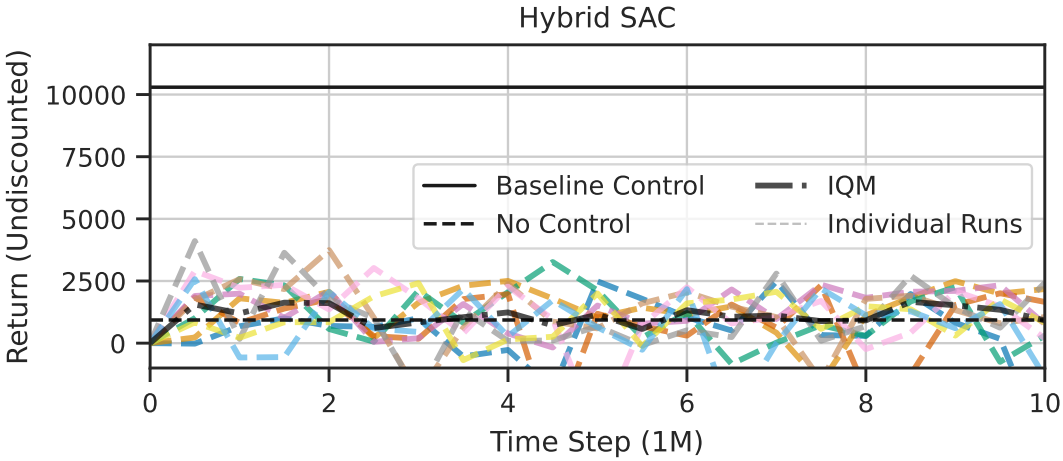



Figure 21: Individual learning curves of SAC (Additive Hybrid).

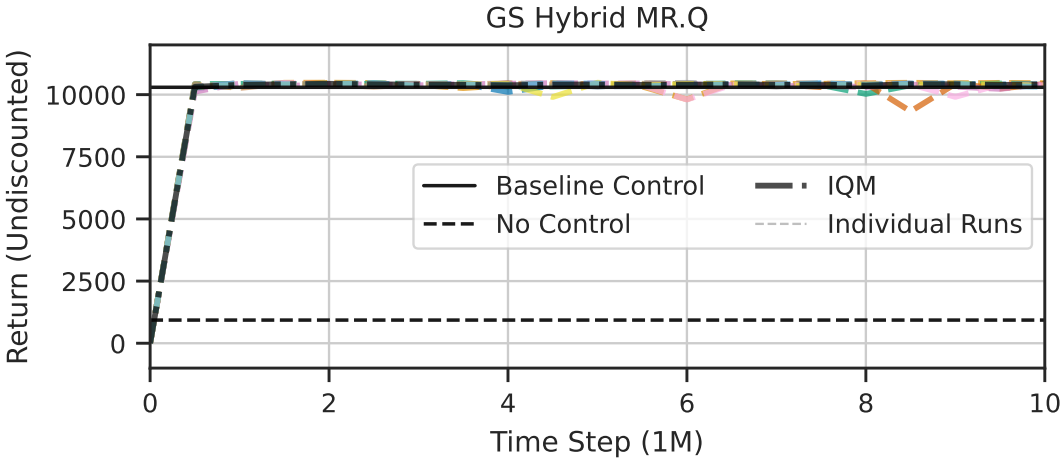



Figure 22: Individual learning curves of MR.Q (Gain-Scheduling Hybrid).