RL Based AOC Research

Reinforcement Learning for Spacecraft Attitude Control

- **Year**: October 2019
- **Abstract**: Reinforcement learning (RL) has recently shown promise in solving difficult numerical problems and has discovered non-intuitive solutions to existing problems. This study investigates the ability of a general RL agent to find an optimal control strategy for spacecraft attitude control problems. Two main types of Attitude Control Systems (ACS) are presented. First, the general ACS problem with full actuation is considered, but with saturation constraints on the applied torques, representing thruster-based ACSs. Second, an attitude control problem with reaction wheel based ACS is considered, which has more constraints on control authority. The agent is trained using the Proximal Policy Optimization (PPO) RL method to obtain an attitude control policy. To ensure robustness, the inertia of the satellite is unknown to the control agent and is randomized for each simulation. To achieve efficient learning, the agent is trained using curriculum learning. We compare the RL based controller to a QRF (quaternion rate feedback) attitude controller, a well-established state feedback control strategy. We investigate the nominal performance and robustness with respect to uncertainty in system dynamics. Our RL based attitude control agent adapts to any spacecraft mass without needing to re-train. In the range of 0.1 to 100,000 kg, our agent achieves 2% better performance to a QRF controller tuned for the same mass range, and similar performance to the QRF controller tuned specifically for a given mass. The performance of the trained RL agent for the reaction wheel based ACS achieved 10 higher better reward then that of a tuned QRF controller.
- **Algorithm**: PPO

Environment

- **Spacecraft Model**: Rigid body
- **Force Model**: The spacecraft is not affected by any external force

Observation Space

Two observation spaces are presented, in the first one (general ACS problem with full actuation) the state vectors s_t at any time t are a composition of the spacecraft attitude, represented by quaternion q_t and spacecraft angular velocity, represented by ω_t.

$$s_t = [q_t, \omega_t]$$

In the second one (attitude control problem with reaction wheel based ACS) the speed of each reaction wheel is added to the state vector (to make sure that the problem represents a MDP).

$$s_t = [q_t, \omega_t, \omega_{RWt}]$$

Action Space

It is not explicitly provided but **probably** a continuous action space with 3 components (control torque comand along the spacecraft principal axes) is used.

Reward Function

$$r(s_t, a_t) = -\alpha_q q_{err} - \alpha_\omega ||\omega_e||_2 - a_t - c$$

where:

- q_{err} : quaternion error between the current spacecraft orientation and the desired one.
- ω_e: difference between the current spacecraft angular velocity and the desired one.
- α_q, α_w: weights used to tune the system response.
- c: conditional reward:

$$c = \begin{cases} 200 : \text{if } q_{err} \leq q_\epsilon \\ 1000 : \text{if } q_{err} \leq q_\epsilon \text{ and } ||\omega_e||_2 \leq \omega_\epsilon \\ -10^3 : \text{if } q_{err} \geq q_l \text{ or } ||\omega_e||_2 \leq \omega_l \\ -10^4 : \text{if } q_{err} \geq 2q_l \text{ or } ||\omega_e||_2 \leq 2\omega_l \\ -10^3 : \text{if reaction wheels saturated} \\ 0 : \text{otherwise} \end{cases}$$

where $q_\epsilon, q_l, \omega_\epsilon, \omega_l$ are threshold parameters

> The term a_t was not explicitly defined but it is related to the control effort at time t.

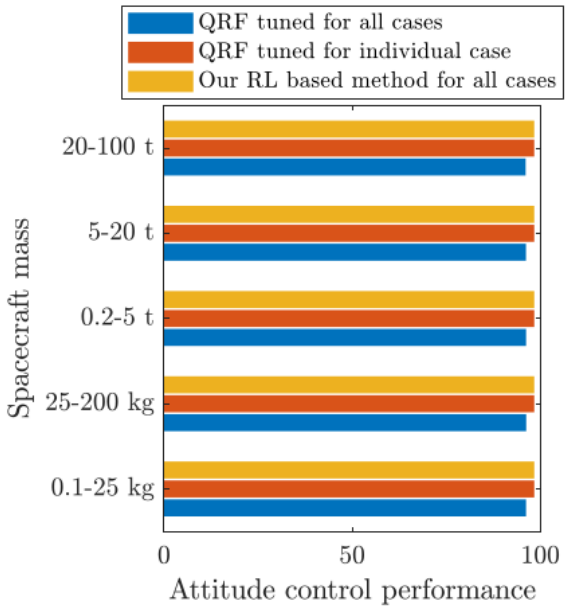
RL Algorithm

PPO algorithm with curriculum learning is used.

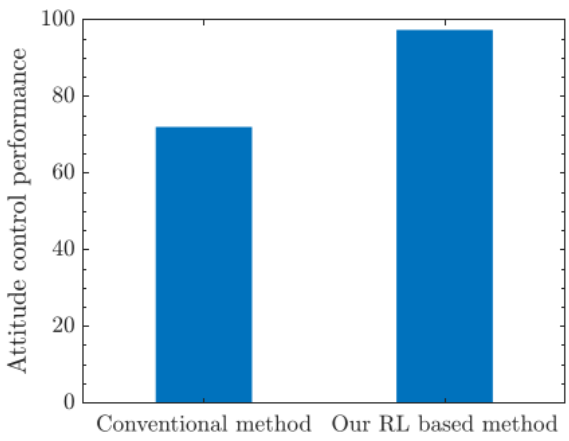
Hyperparameter	Value
Steps	600
dt	30 sec
Iterations	2000
Roll-outs	512
Epochs	256
Mini-batch	512
Layers	4 (Fully Connected)
Neurons per layer	7, 4, 4, 7

Results

The realized agent demonstrated that it could discover the attitude control solutions for an individual satellite, as well as for a family of satellites, without being informed of the mechanical properties of the satellite, with 2% performance benefit to a QRF controller tuned to have the best performance across the same mass range. The performance of the RL based attitude control is similar to QRF controllers that have been hand tuned for each mass case. The RL trained agent was tested for a mass variation in the range of 0.1 to 100,000 kg in the satellite mass, along with dimensional variation in the range of 0.1 m to 100 m for each side length, yielding a large variety of satellite physical properties.



For the higher fidelity reaction wheel based ACS, the RL agent had a performance metric of 97, a lead of 25 over the tuned QRF controller with a performance metric of 72.



> The performance is measured using the following relation:
>
> $$performance = \frac{1200}{1 - r_{average}} \cdot 100$$
>
> where $r_{average}$ is the average reward per step obtained.
> 1200 is the maximum reward value per step.

Such controllers and rapid learning-based techniques are promising strategies for a wide host of missions where the physical propertied of the satellite change unpredictably. Additionally, RL based attitude control algorithms can simplify development times and increase the reliability of ACS, since the same algorithm can operate for a large variety of missions.

Keywords

- Rigid body model
- PPO algorithm
- Curriculum learning

Adaptive Continuous Control of Spacecraft Attitude Using Deep Reinforcement Learning

- **Year**: August 2020
- **Abstract**: As modern and future space missions plan to explore diverse bodies across deep space, the ability to conduct successful spacecraft operations under uncertainty becomes increasingly apparent. A highly accurate and precise spacecraft attitude controller that is robust to the perturbation forces encountered in an uncertain dynamics environment is critical to the ongoing success of spaceflight in deep space. In this work, we present a framework for deriving an adaptive spacecraft attitude controller using deep reinforcement learning. The controller developed is shown to effectively perform large-angle slew maneuvers at industry-standard pointing accuracies. We find that the controller is capable of adapting in the presence of various disturbance torques unseen during training and is system-agnostic of the spacecraft being controlled, even when trained on one spacecraft configuration. Additionally, this study discusses the application specifics that yielded the reported results and discusses possible routes of expansion for future work.
- **Algorithm**: TD3

Environment

- **Spacecraft Model**: Rigid body
- **Force Model**: The spacecraft is not affected by any external force

Observation Space

Continuous observation space.

$$s = [q_e, \dot{q_e}, \omega]$$

where:

- q_e: quaternion error between the current spacecraft orientation and the desired one.
- $\dot{q_e}$: quaternion error derivative.
- ω: spacecraft angular velocity.

Action Space

Continuous action space.

$$a = [T]$$

where T is the vector of control torques (3 elements, one for each spacecraft principal axis). The maximum value (magnitude) for each component is 0.5 Nm.

Reward Function

The agent has evolving goals over time:

1. to slew the spacecraft to within the desired error tolerance of the desired orientation;
2. to stabilize the motion about that desired orientation and remain within the desired error tolerance for an extended period of time.

To reflect these evolving goals of the agent, two reward functions are utilized.
The first (r_1) is used from the beginning of the episode until the criterion $q_s \geq 0.999962$ is satisfied.

$$r_1 = \begin{cases} 0.1 \text{ if } q_{s,t} > q_{s,t-1} \\ -0.1 \text{ otherwise} \end{cases}$$

Upon reaching the criterion $q_s \geq 0.999962$, the reward function is then switched to r_2:

$$r_2 = \alpha q_e = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} q_1^2 \\ q_2^2 \\ q_3^2 \\ q_s^2 \end{bmatrix}$$

This reward function is maximized when the error angle is minimized, meant to incentivize the agent stabilizing about the desired orientation.

Additionally, a terminal reward of $+10$ is given if the terminal state satisifies $q_s \geq 0.999962$, 0 otherwise. A reward of -25 is given if the agent terminates the episode early (magnitude of angular velocity exceeds 0.5 rad/s).

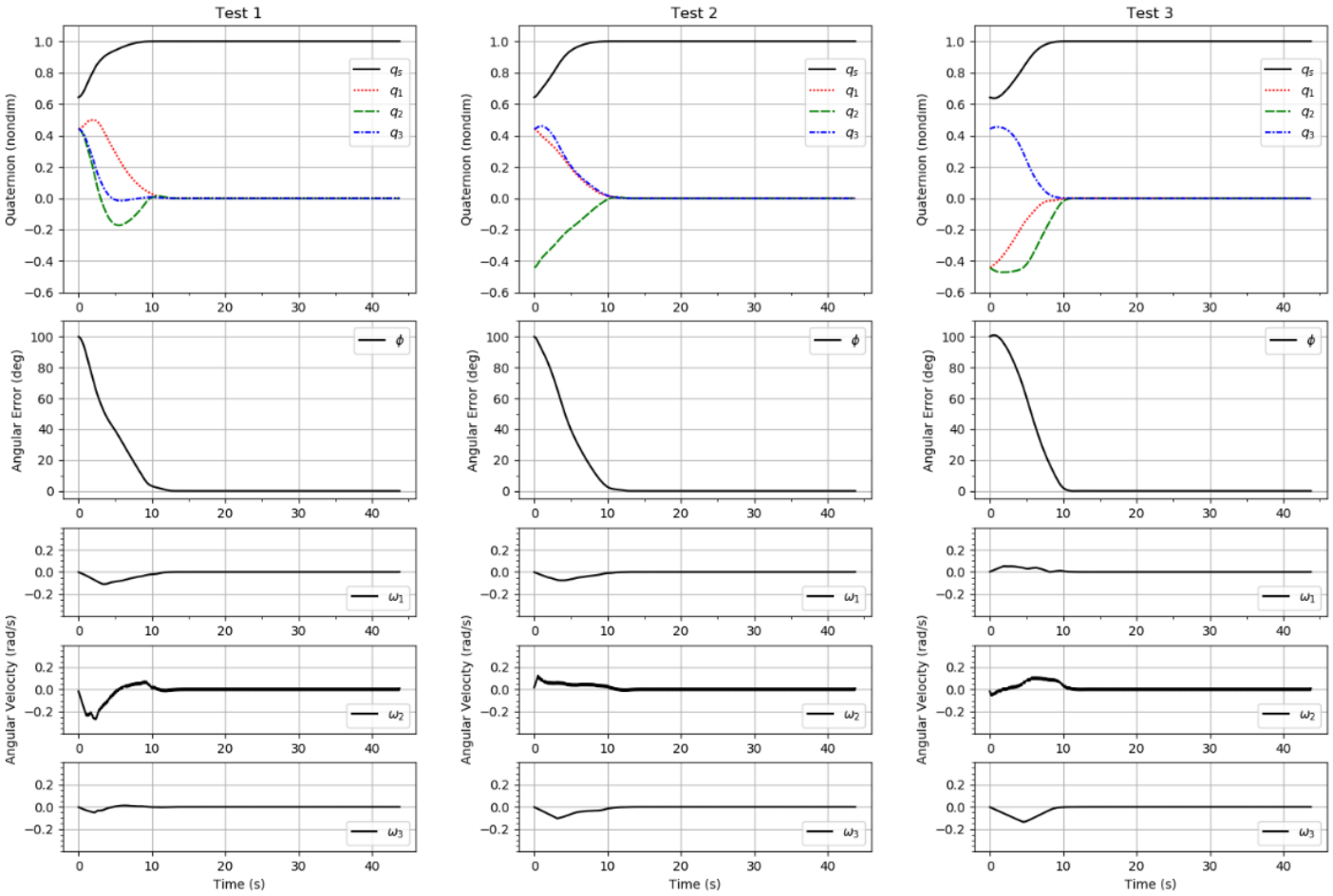
RL Algorithm

TD3 algorithm with frame skipping is used.

Hyperparameter	Value
Discount (γ)	0.99
Batch size	100
Exploration noise	0.1
Exploration noise clip	0.5
Policy delay	2
Polyak averaging proportion	0.995
Learning rate	3e-4 ⇨ 1e-6
Hidden Layers	2 (Fully Connected)
Neurons per layer	400, 300

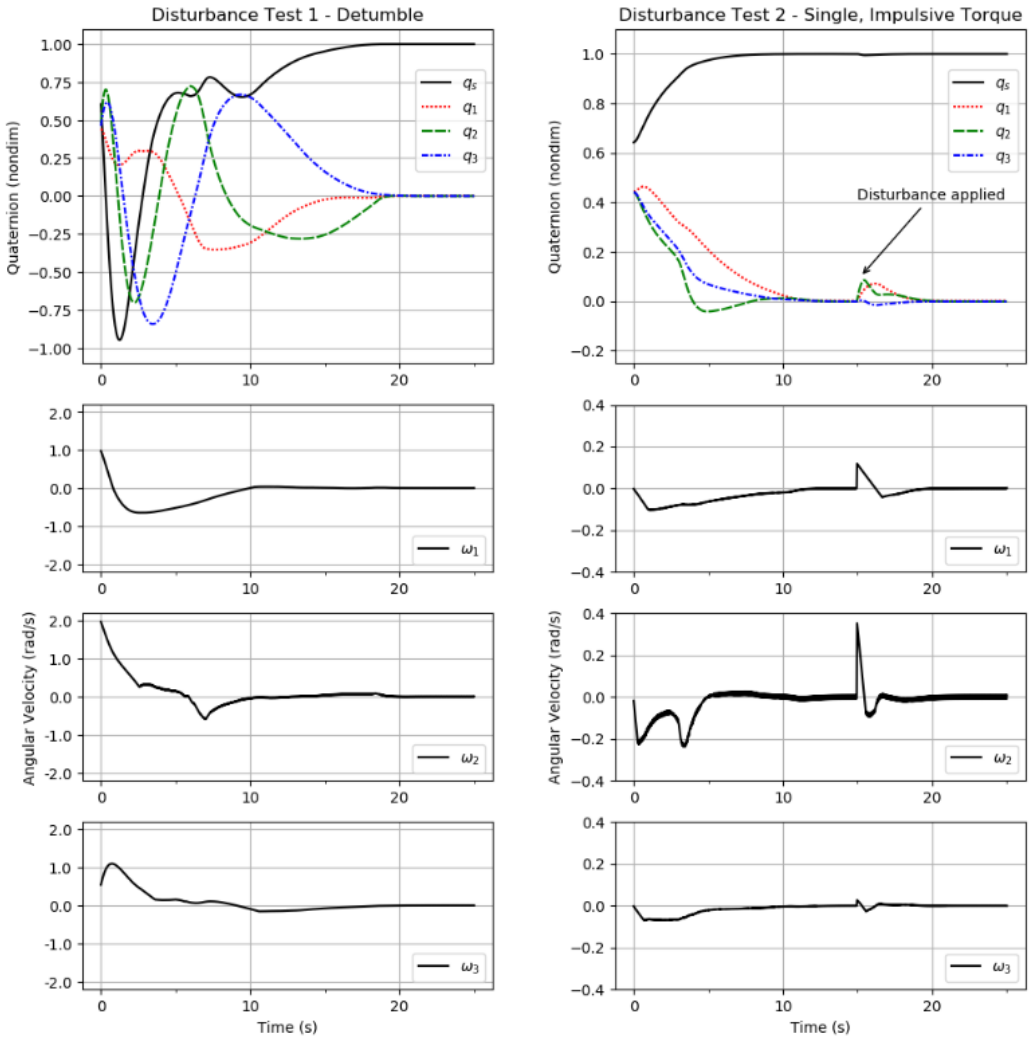
Results

Test 1: Three slew maneuvers were considered with no perturbation torques included.



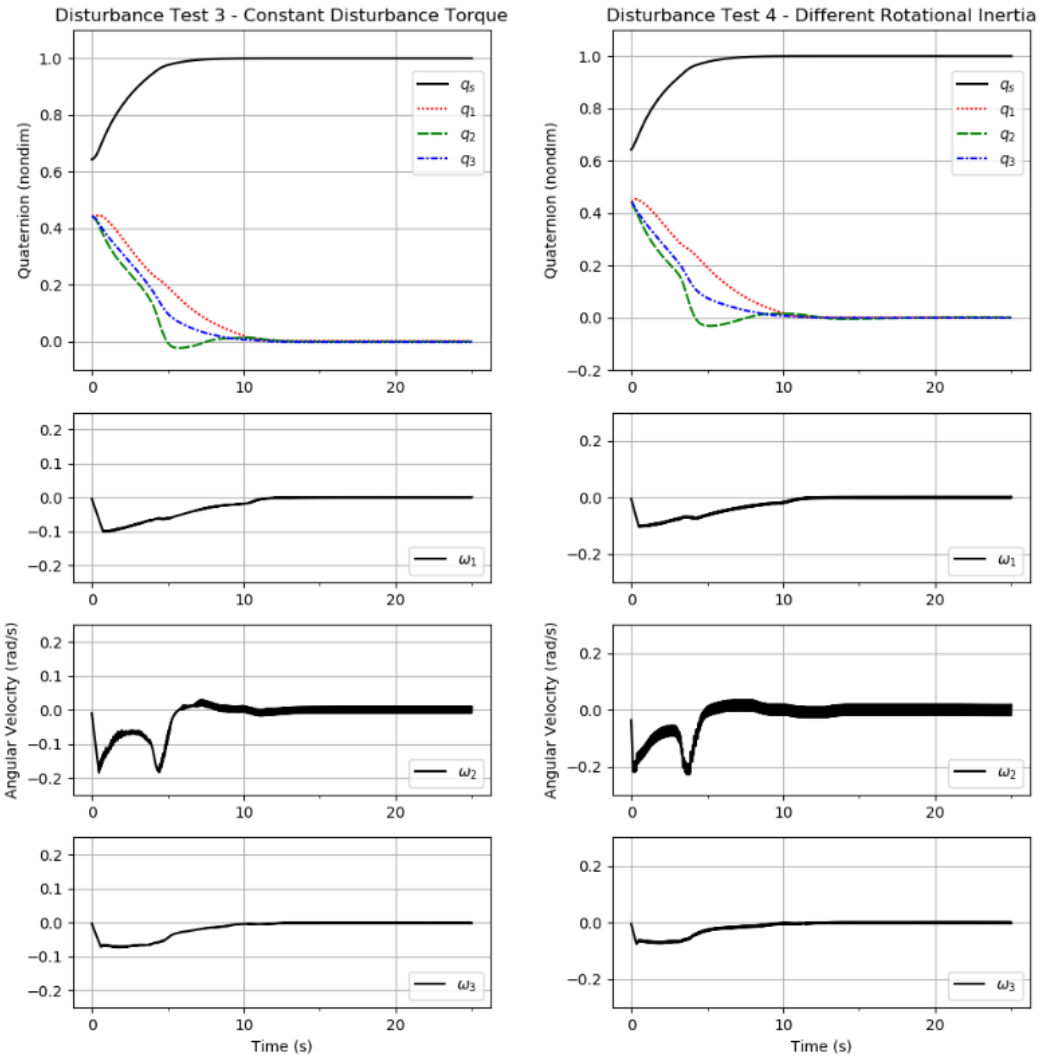
Tests 2 and 3

- Spacecraft detumble (initial spacecraft angular velocity: $\omega = [1, 2, 0.5]$ rad/s).
- Single impulsive disturbance $T_{dist} = [5, 2, 1]$ Nm at $t = 15$ s.



Tests 4 and 5

- Constant disturbance torque $T_{dist} = [0.0341, 0.0038, -0.0379]$ Nm
- Different rotational inertia: the rotational inertia tensor is decreased elementwise by a factor of two.



Keywords

- Rigid body model
- TD3 algorithm
- Frame skipping

Autonomous Spacecraft Attitude Control Using Deep Reinforcement Learning

- **Year**: October 2020
- **Abstract**: While machine learning and spacecraft autonomy continue to gain research interest, significant work remains to be done in efficiently applying modern machine learning techniques to problems in spaceflight. This study presents a framework for deriving a discrete neural spacecraft attitude controller using reinforcement learning, a paradigm of machine learning, without the need for high-performance computing. The developed attitude controller is an approximately time-optimal solution to a highly constrained control problem, able to achieve well above industry-standard pointing accuracies. Control examples are also presented of the agent performing large-angle spacecraft slews in the developed simulation environment and future extensions of
  this work are discussed.
- **Algorithm**: PPO

Environment

- **Spacecraft Model**: Rigid body
- **Force Model**: The spacecraft is not affected by any external force

Observation Space

Continuous observation space.

$$s = [q_e, \dot{q}_e, \omega]$$

where:

- q_e: quaternion error between the current spacecraft orientation and the desired one.
- $\dot{q_e}$: quaternion error derivative.
- ω: spacecraft angular velocity.

Action Space

Discrete action space.

$$a = [T]$$

where T is the vector of control torques (3 elements, one for each spacecraft principal axis).
The control problem is discretized by only allowing the agent to control near-impulsive torques on any of the three principle axes in the spacecraft body frame to three magnitudes: 0.5 Nm, 0.05 Nm, or 0.005 Nm.

Reward Function

$$r_t = \begin{cases} r_a + 9 \text{ if } \phi \leq 0.25^\circ \\ r_a \text{ otherwise} \end{cases}$$

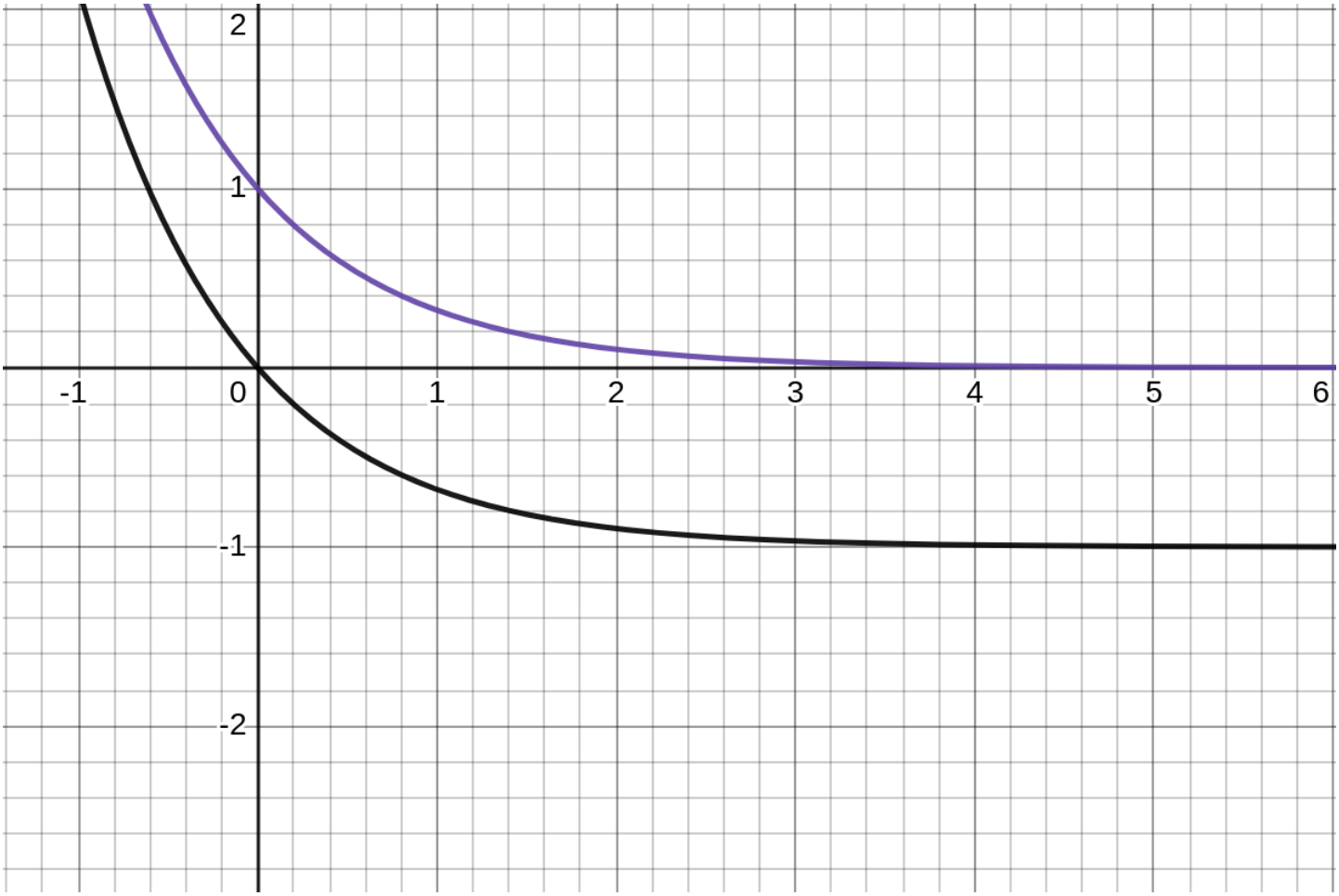
where:

- ϕ: angular error between current spacecraft attitude and desired one.
- r_a:

$$r_a = \begin{cases} \exp(\frac{-\phi}{0.14 \times 2\pi}) \text{ if } q_{s,t} > q_{s,t-1} \\ \exp(\frac{-\phi}{0.14 \times 2\pi}) - 1 \text{ otherwise} \end{cases}$$

The scalar constant of 0.14 in the denominator of the argument is a horizontal scaling factor, chosen so the function returns a magnitude of about 0.001 when $\phi = 180^\circ$.

The following plot shows the shape of r_a.



The entire reward formulation can be stated as a negative exponential function, with an added bonus if the agent is within our specified goal criterion of $\phi \leq 0.25^\circ$.

A terminal reward of $+50$ is given if the spacecraft terminal state satisifies $\phi \leq 0.25^{\circ}$, 0 otherwise. A reward of -50 is given if the magnitude of the spacecraft angular velocity exceeds 0.5 rad/s.

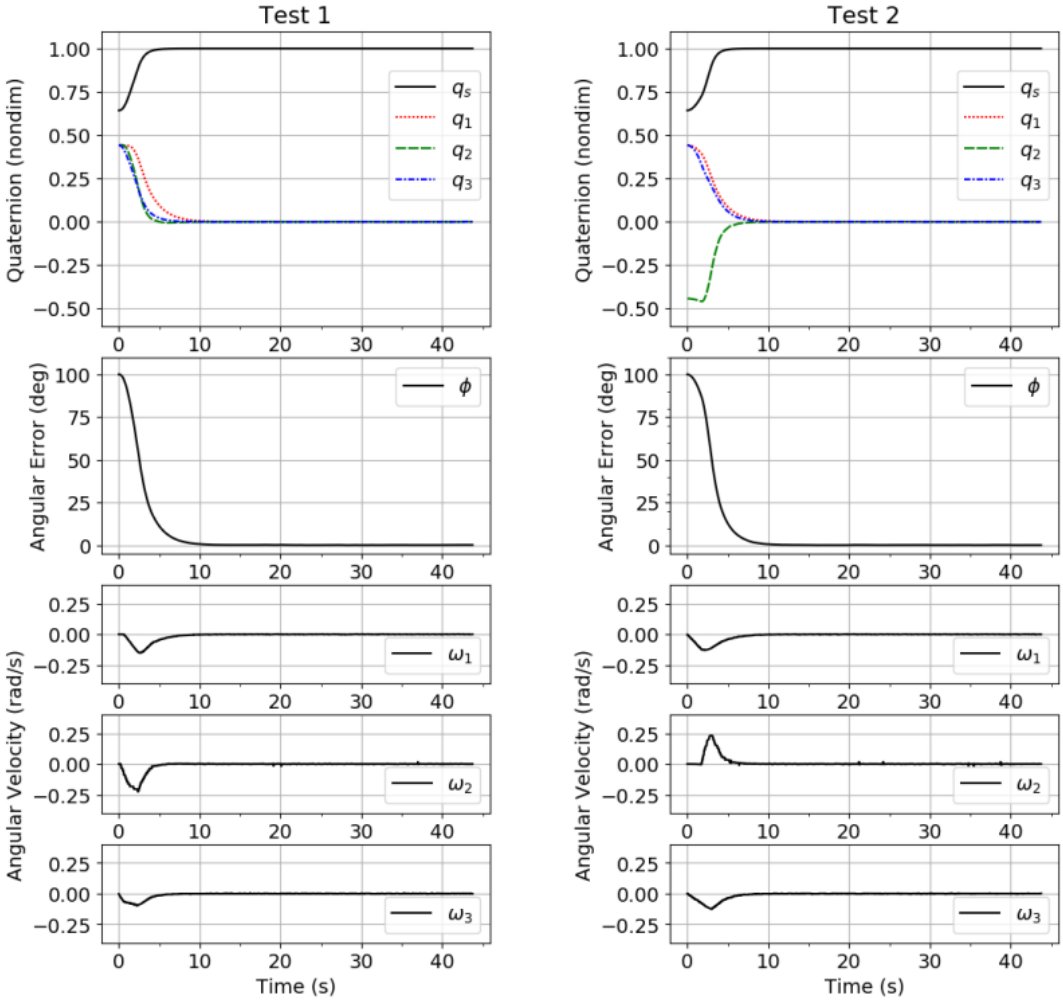
RL Algorithm

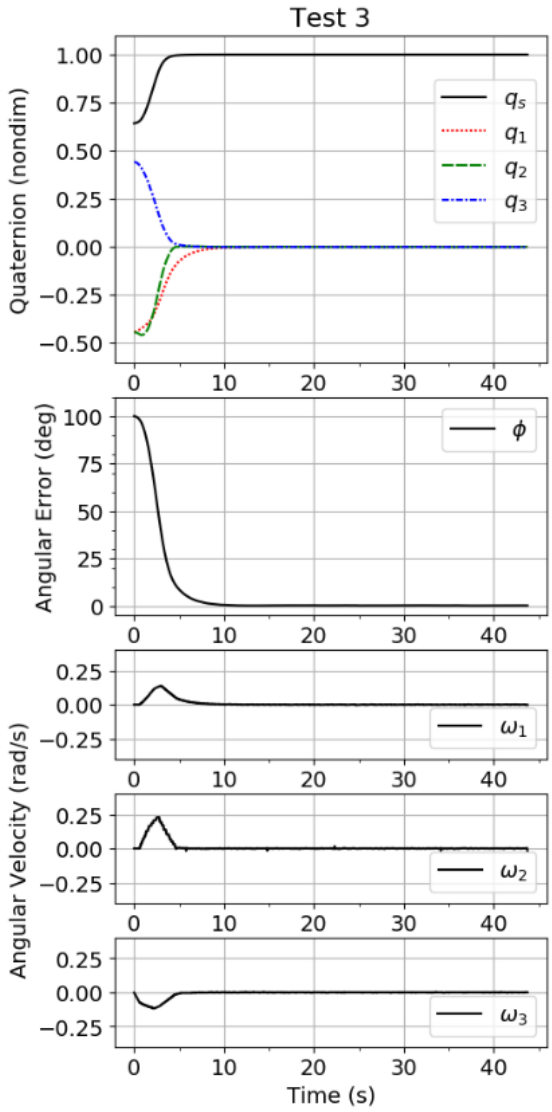
TD3 algorithm with frame skipping is used.

Hyperparameter	Value
Discount (γ)	0.99
Batch size	128
Minibatch size	30
Clipping parameter (ϵ)	0.2
Entropy coeff.	0.0001
Learning rate	3e-4 ⇨ 1e-5
Hidden Layers	2 (Fully Connected)
Neurons per layer	400, 300

Results

Test	Axis of Rotation	Angle (°)	Quaternion
1	[0.57735, 0.57735, 0.57735]	100	[0.44228, 0.44228, 0.44228, 0.64279]
2	[0.57735, -0.57735, 0.57735]	100	[0.44228, -0.44228, 0.44228, 0.64279]
3	[-0.57735, -0.57735, 0.57735]	100	[-0.44228, -0.44228, 0.44228, 0.64279]





Keywords

- Rigid body model
- PPO algorithm
- Frame skipping

Model-Free Attitude Control of Spacecraft Based on PID-Guide TD3 Algorithm

- **Year**: December 2020
- **Abstract**: This paper is devoted to model-free attitude control of rigid spacecraft in the presence of control torque saturation and external disturbances. Specifically, a model-free deep reinforcement learning (DRL) controller is proposed, which can learn continuously according to the feedback of the environment and realize the high-precision attitude control of spacecraft without repeatedly adjusting the controller parameters. Considering the continuity of state space and action space, the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm based on actor-critic architecture is adopted. Compared with the Deep Deterministic Policy Gradient (DDPG) algorithm, TD3 has better performance. TD3 obtains the optimal policy by interacting with the environment without using any prior knowledge, so the learning process is time-consuming. Aiming at this problem, the PID-Guide TD3 algorithm is proposed, which can speed up the training speed and improve the convergence precision of the TD3 algorithm. Aiming at the problem that reinforcement learning (RL) is difficult to deploy in the actual environment, the pretraining/fine-tuning method is proposed for deployment, which can not only save training time and computing resources but also achieve good results quickly. The experimental results show that DRL controller can realize high-precision attitude stabilization and attitude tracking control, with fast response speed and small overshoot. The proposed PID-Guide TD3 algorithm has faster training speed and higher stability than the TD3 algorithm.
- **Algorithm**: TD3

Environment

- **Spacecraft Model**: Rigid body
- **Force Model**: The spacecraft is not affected by any external force in the **first** training process, then in the **second** training process, a sinusoidal disturbance is considered.

Observation Space

Continuous observation space.

$$s = [p, \omega, p_e, \omega_e]$$

where:

- p: spacecraft attitude angle vector.
- ω: spacecraft angular velocity vector.
- p_e: difference between the current spacecraft attitude angles and the desired ones.
- ω_e: difference between the current spacecraft angular velocity and the desired one.

Action Space

Continuous action space.

$$a = [T]$$

where T is the vector of control torques (3 elements, one for each spacecraft principal axis).The maximum value (magnitude) for each component is 1 Nm.

Reward Function

$$r_t = r_1 + r_2 + r_3$$

where:

$$r_1 = -|||[p_e, \omega_e]|||_1 - a^2$$

$$r_2 = \frac{1}{||p_e||_1 + 0.01}(|||[p_e, \omega_e]||_{-\infty} \leq 0.1)$$

$$r_3 = -100(||p||_\infty \geq 4 \text{ or } ||\omega||_\infty \geq 4)$$

The smaller the error, the greater the reward of r_1 . The second term in r_1 represents energy consumption. When the attitude of the spacecraft is very close to the target attitude, the error change is small, and the change of r_1 will be very insignificant. Therefore, the continuous reward r_2 is introduced to increase the reward gradient when the absolute value of each error component is less than 0.1, so as to guide the attitude angle to approach the target value quickly and accurately. r_3 is a discrete reward, which can control the attitude angle not to exceed the range and increase training speed.

RL Algorithm

In order to speed up the training speed and improve the convergence stability of the algorithm, a PID-Guide TD3 algorithm is proposed. The core idea of the PID-Guide TD3 algorithm is as follows. In the current state s, two actions are generated by the action network and PID controller, respectively. Then, the critical network is used to evaluate the two actions; the action with higher value will be actually executed.

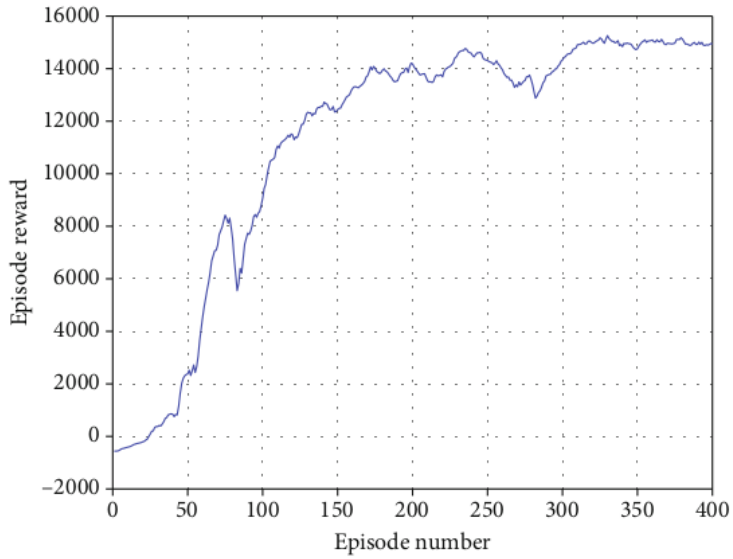
Hyperparameter	Value
Max Episodes	400
Max steps per episode	200
Replay buffer size	250
Policy net. learning rate	0.0003
Critic net. learning rate	0.001
Discount (γ)	0.99
Exploration noise	0.1
Exploration noise clip	0.5
Policy delay	3
Soft update rate	0.01
Hidden Layers	2 (Fully Connected)

Hyperparameter	Value
Neurons per layer	250, 250

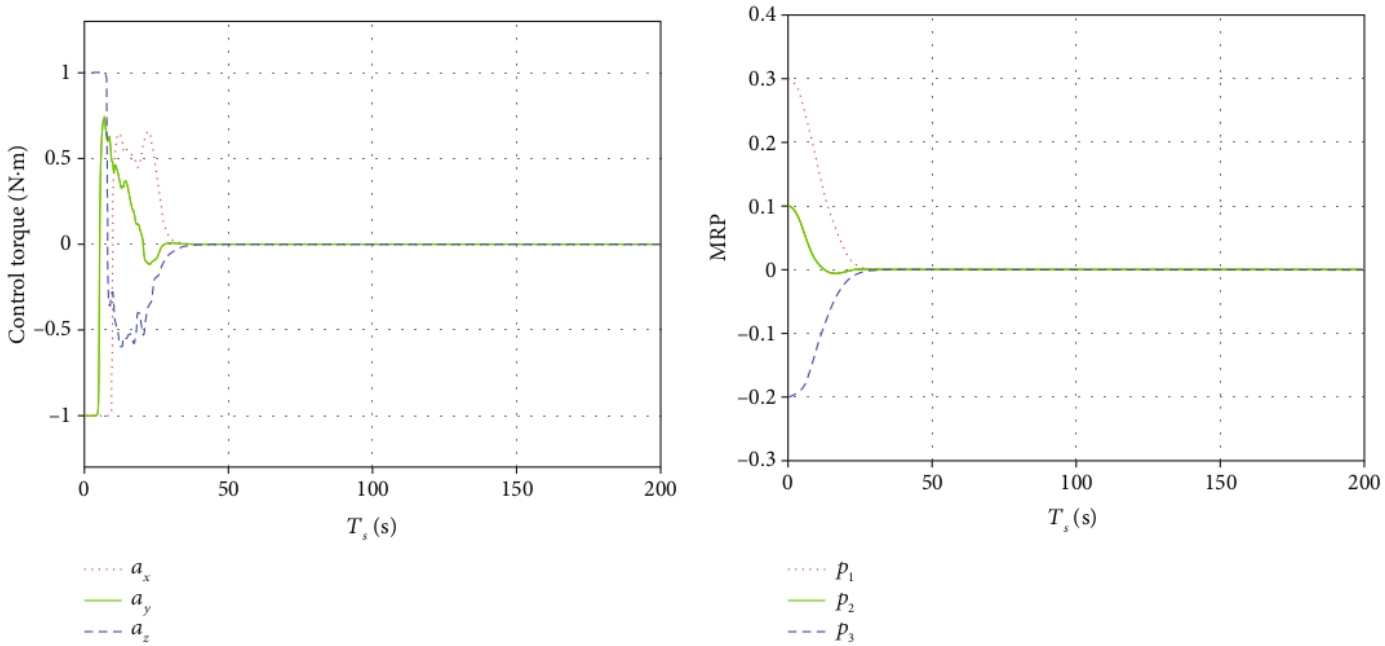
Results

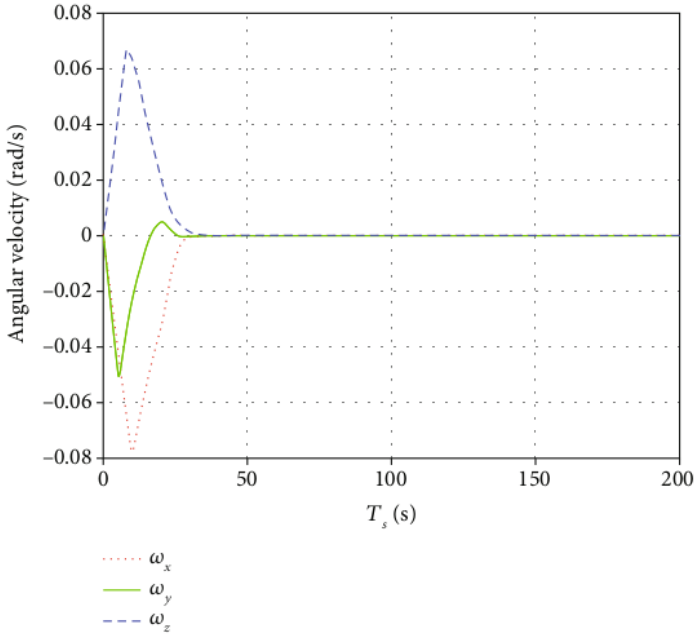
Case 1: in the ideal environment without external disturbances, the agent is trained to realize the attitude stabilization control and attitude tracking control of spacecraft, respectively.

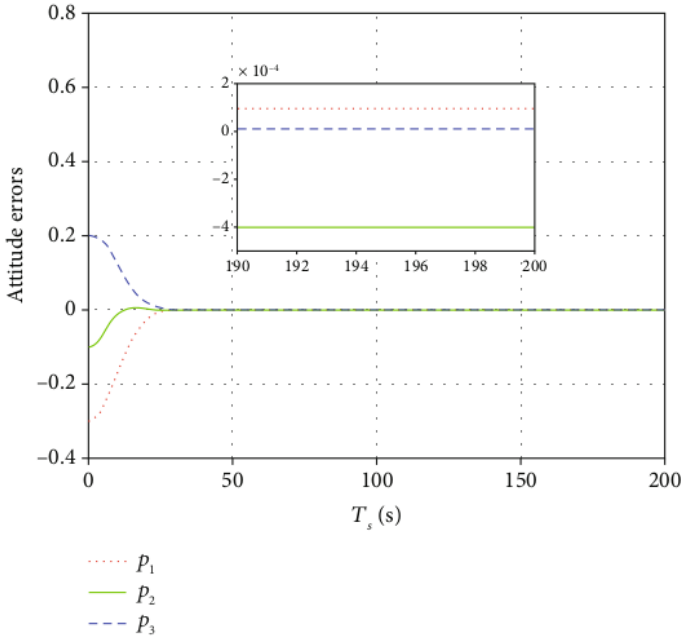
LEARNING PROCESS

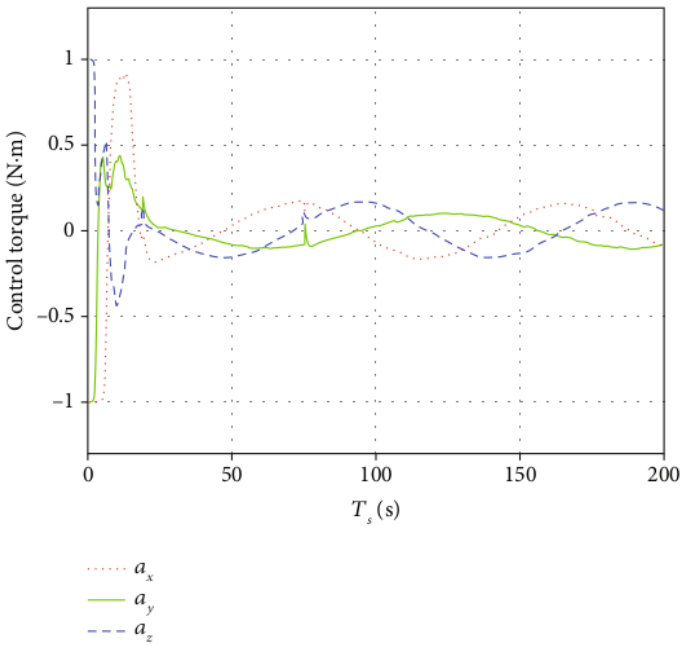


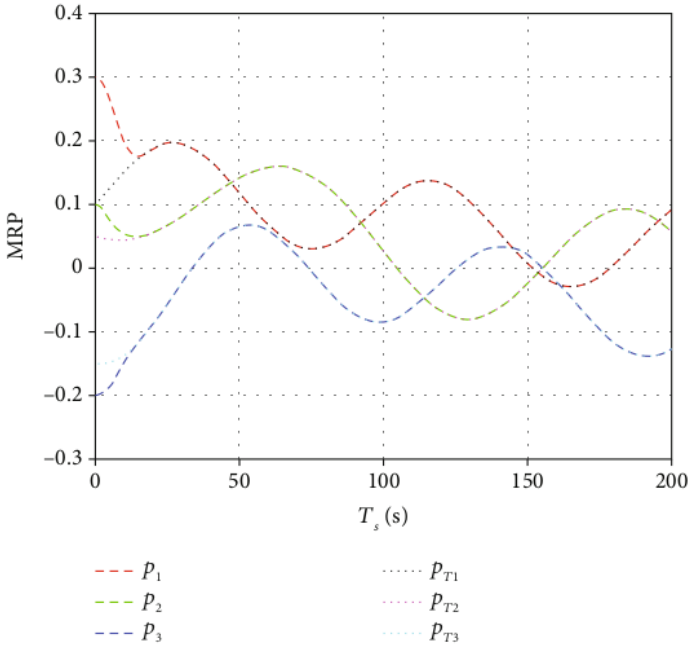
TEST RESULTS

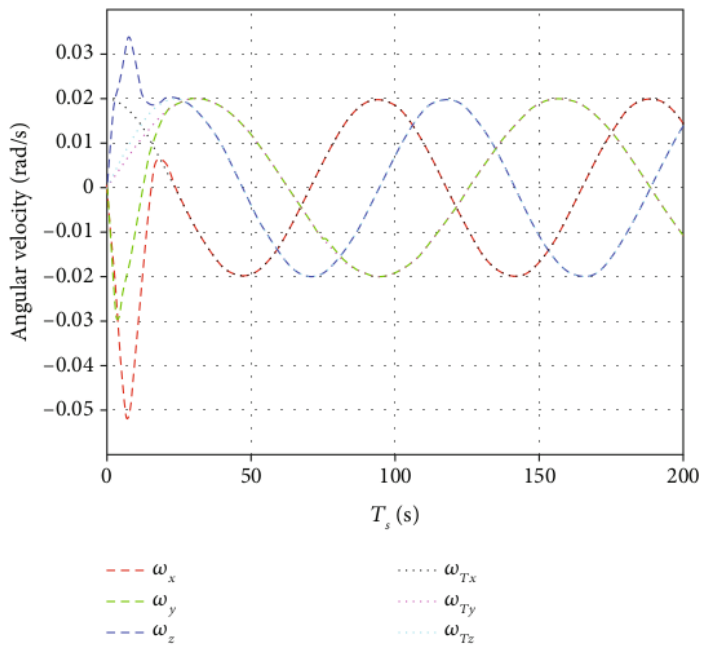


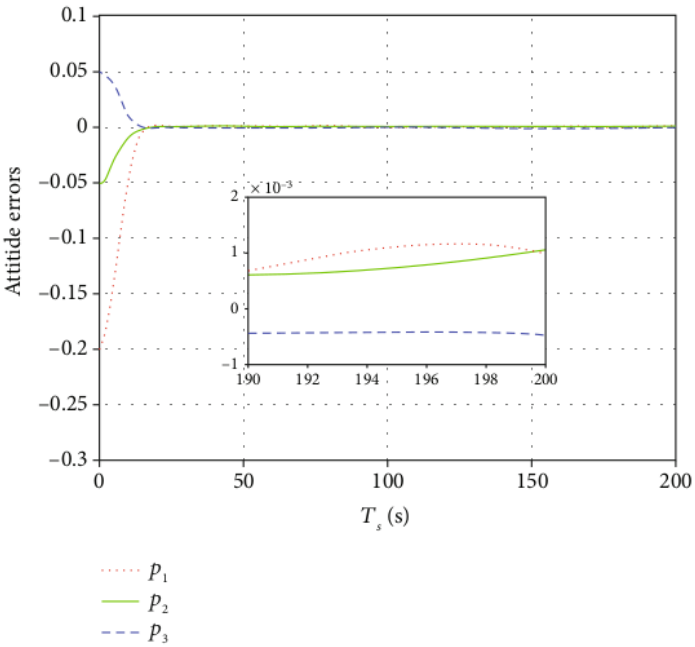

Angular velocity (rad/s)
0.08
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
−0.08
0
50
100
150
200
T_s (s)
ω_x
ω_y
ω_z


Attitude errors
0.8
0.6
0.4
0.2
0
−0.2
−0.4
× 10⁻⁴
2
0
−2
−4
190
192
194
196
198
200
0
50
100
150
200
T_s (s)
p_1
p_2
p_3


Control torque (N·m)
1
0.5
0
−0.5
−1
0
50
100
150
200
T_s (s)
a_x
a_y
a_z


MRP
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
0
50
100
150
200
T_s (s)
p_1
p_2
p_3
p_{T1}
p_{T2}
p_{T3}

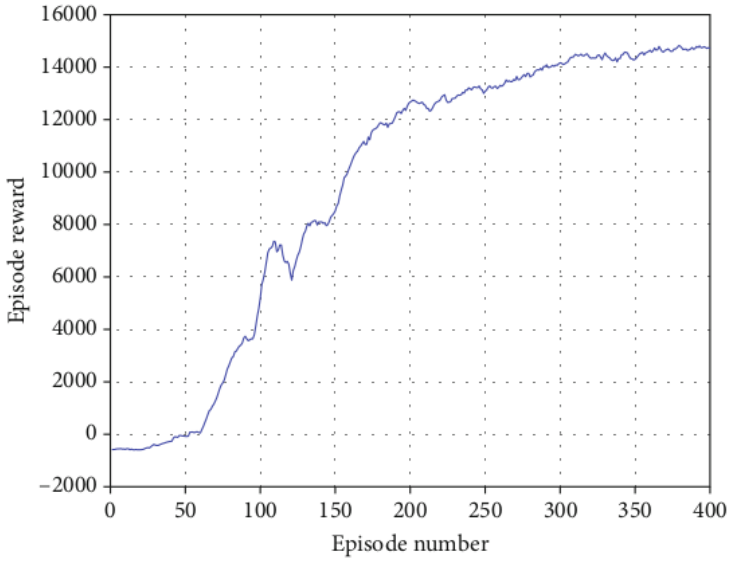

Angular velocity (rad/s)
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
−0.04
−0.05
0
50
100
150
200
T_s (s)
ω_x
ω_y
ω_z
ω_{Tx}
ω_{Ty}
ω_{Tz}


Attitide errors
0.1
0.05
0
−0.05
−0.1
−0.15
−0.2
−0.25
−0.3
× 10⁻³
2
1
0
−1
190
192
194
196
198
200
0
50
100
150
200
T_s (s)
p_1
p_2
p_3

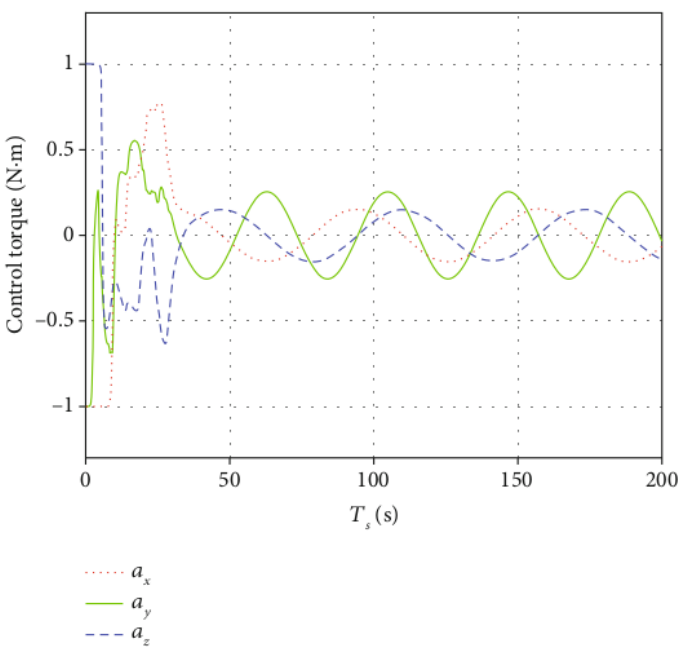
Case 2: on the basis of Case 1, the existence of unknown disturbance torques is considered.

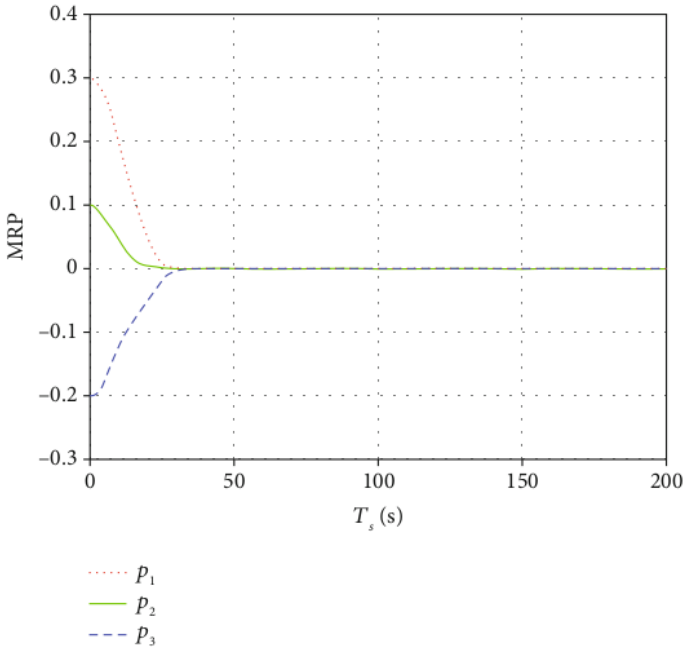
$$d(t) = [0.15\cos(0.1t), 0.25\cos(0.15t), 0.15\cos(0.1t)]\ \mathrm{Nm}$$

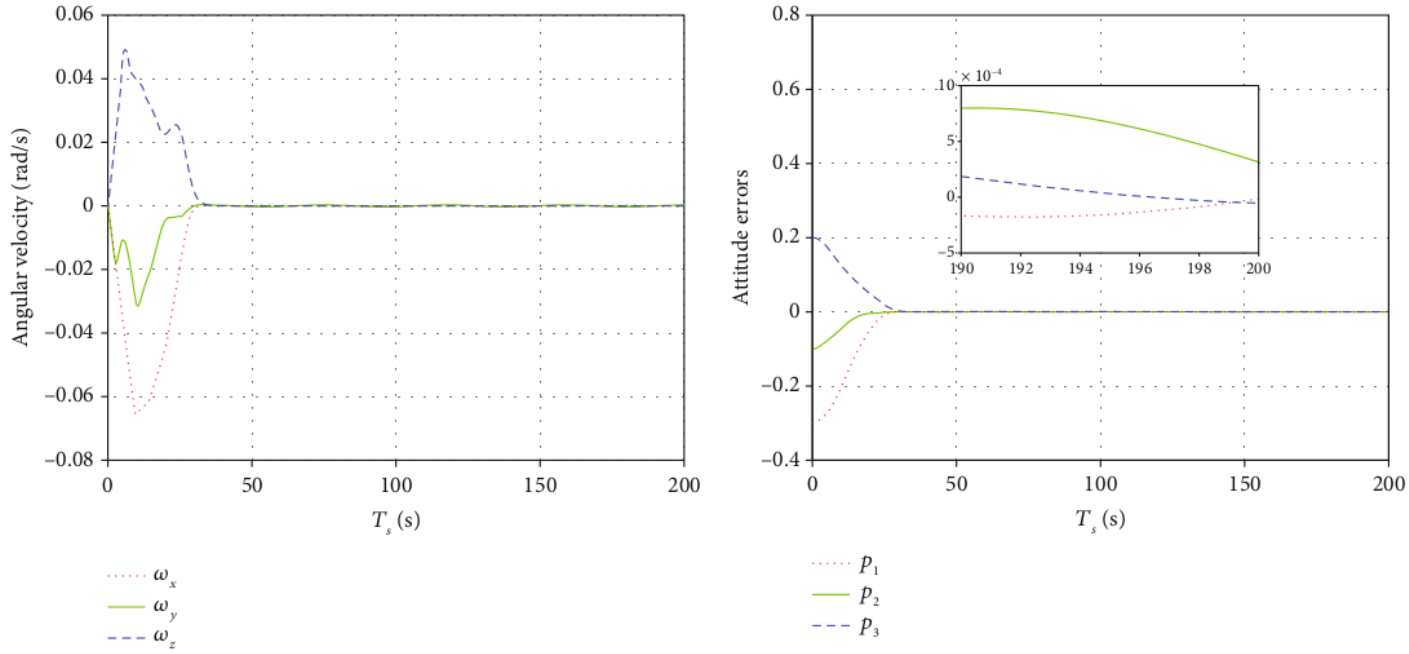
LEARNING PROCESS



TEST RESULTS



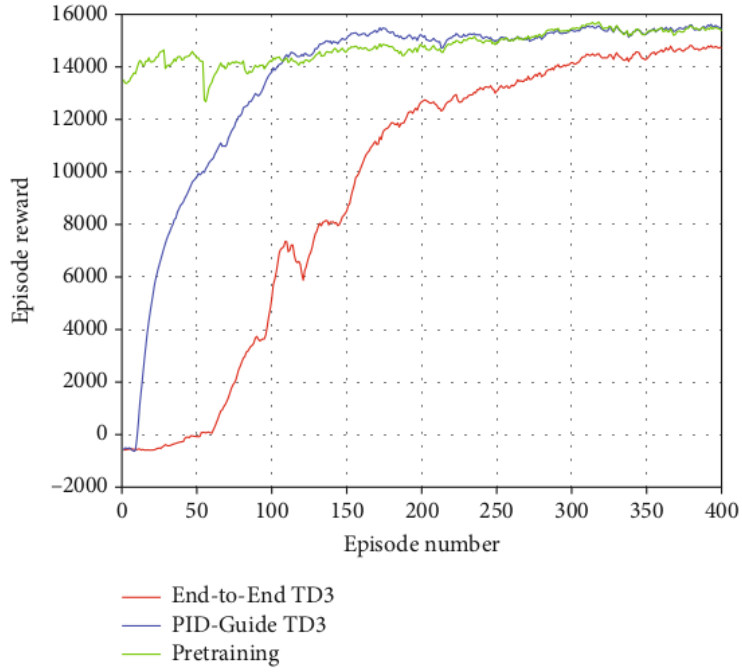




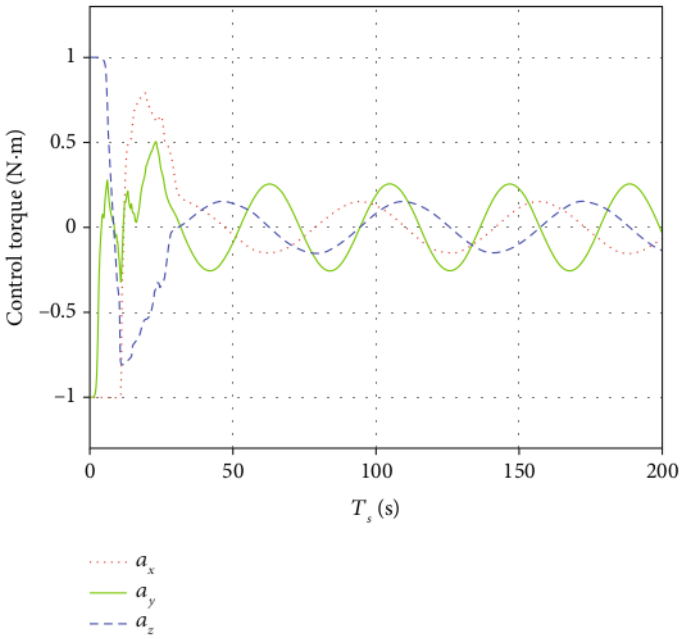
Case 3: on the basis of Case 2, the PID-Guide TD3 algorithm is used to accelerate the training speed and convergence stability.

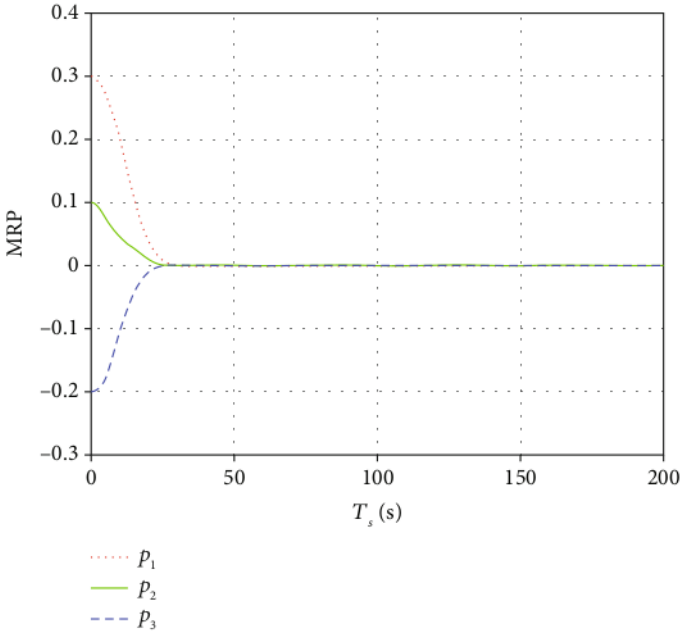
In order to verify the superiority of the PID-Guide TD3 algorithm, the training speed and stability performance of PID-Guide TD3, End-to-End TD3, and pretraining/fine-tuning method are compared, respectively.

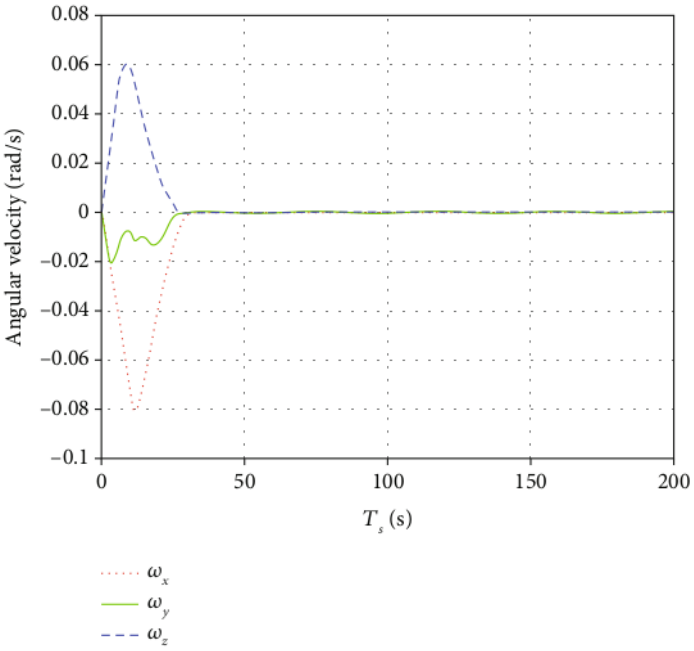
LEARNING PROCESS

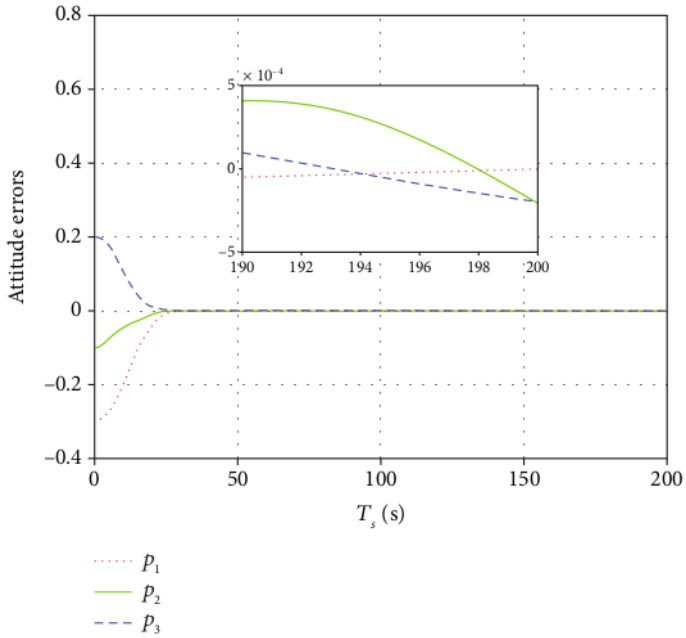


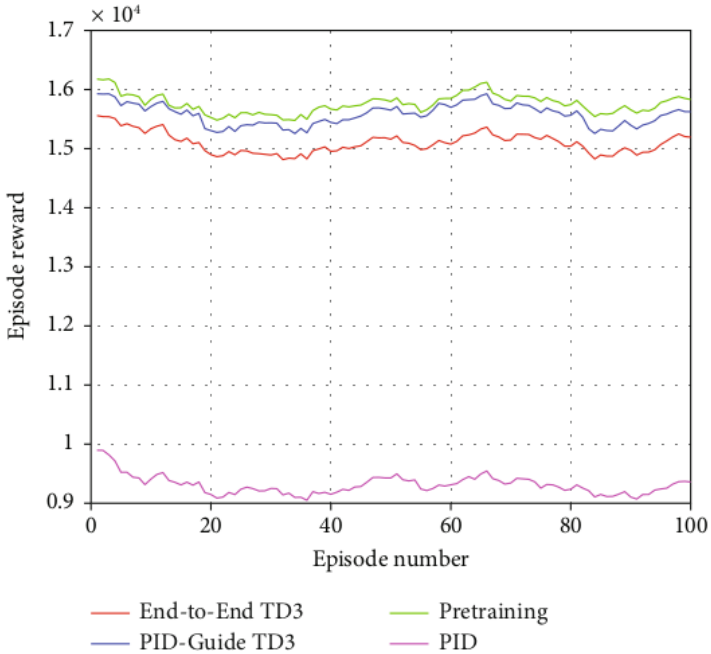
TEST RESULTS


Control torque (N·m)
1
0.5
0
−0.5
−1
0
50
100
150
200
T_s (s)
a_x
a_y
a_z


MRP
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
0
50
100
150
200
T_s (s)
p_1
p_2
p_3


Angular velocity (rad/s)
0.08
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
−0.08
−0.1
0
50
100
150
200
T_s (s)
ω_x
ω_y
ω_z


Attitude errors
0.8
0.6
0.4
0.2
0
−0.2
−0.4
0
50
100
150
200
T_s (s)
× 10⁻⁴
5
0
−5
190
192
194
196
198
200
p_1
p_2
p_3


× 10⁴
Episode reward
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0
20
40
60
80
100
Episode number
End-to-End TD3
PID-Guide TD3
Pretraining
PID

The pretrained agent only needs a few episodes of training to adapt to the new environment and at the same time avoid the occurrence of dangerous states in the process of exploration.
The benefits of pretraining extend beyond merely quick convergence, since pretraining can improve model robustness and uncertainty. Therefore, the pretrained DRL controller can be deployed to the spacecraft and then fine-tune parameters on orbit, instead of training from scratch.

Keywords

- Rigid body model
- TD3 algorithm
- PID-Guide

Reinforcement Learning for Attitude Control of a Spacecraft with Flexible Appendages

- **Year**: September 2022

- **Abstract**: This study explores the reinforcement learning (RL) approach to constructing attitude control strategies for a LEO satellite with flexible appendages. Attitude control system actuated by a set of three reaction wheels is considered. The satellite is assumed to move in a circular low Earth orbit under the action of gravity-gradient torque, random disturbance torque, and oscillations excited in flexible appendages. The control policy for rest-to-rest slew maneuvers is learned via the Proximal Policy Optimization (PPO) technique. The robustness of the obtained control policy is analyzed and compared to that of conventional controllers. The first part of the study is focused on problem formulation in terms of Markov Decision Processes, analysis of different reward-shaping techniques, and finally training the RL-agent and comparing the obtained results with the state-of-the-art RL-controllers as well as with the performance of a commonly used quaternion feedback regulator (Lyapunov-based PD controller). We then proceed to consider the same spacecraft with flexible appendages added to its structure. Equations of excitable oscillations are appended to the system and coupling terms are added describing the interactions between the main rigid body and the flexible structures. The dynamics of the rigid spacecraft thus becomes coupled with that of its flexible appendages and the control strategy should change accordingly in order to prevent actions that entail excitation of oscillation modes. Again PPO is used to learn the control policy for rest-to-rest slew maneuvers in the extended system. All in all, the proposed reinforcement learning strategy is shown to converge to a policy that matches the performance of the quaternion feedback regulator for a rigid spacecraft. It is also shown that a policy can be trained to take into account the highly nonlinear dynamics caused by the presence of flexible elements that need to be brought to rest in the required attitude. We also discuss the advantages of the reinforcement learning approach such as robustness and ability of online learning pertaining to the systems that require a high level of autonomy.

- **Algorithm**: PPO

Environment

- **Spacecraft Model**: Rigid body (first training process) and rigid body with flexible appendages (second training process)
- **Force Model**: The spacecraft is affected by the gravitational torque (LEO circular orbit), all other environmental disturbances are modeled as a random torque.

Observation Space

Continuous observation space.

$$s = [q, \omega, h]$$

where:

- q: spacecraft attitude quaternion vector.
- ω: spacecraft angular velocity vector.
- h: reaction wheels angular momentum vector.

In case of the model with flexible appendages six additional states η and $\dot{\eta}$ are considered in the state vector of the dynamical system simulating the environment. The agent, however, is not made aware of this change and keeps the 10-dimensional state vector.

Action Space

Discrete action space.

$$a = m \cdot 10^{-i} \cdot T_{max} \text{ for } i \in \{1, 2, 3\}$$

$$m = \{m : m = -1 + 0.04n, n \in \{0, 1, 2, ..., 50\}\}$$

where Tmax is the maximum allowable torque by the actuators.

Reward Function

Several rewards functions are tested:

$$R_1(\phi, \Delta\phi) = \beta_1(\Delta\phi) \cdot \tau_1(\phi)$$

$$R_2(\phi, \Delta\phi) = \beta_1(\Delta\phi) \cdot \tau_2(\phi)$$

$$R_3(\phi, \Delta\phi) = \beta_2(\Delta\phi) \cdot \tau_1(\phi)$$

$$R_4(\phi, \Delta\phi) = \beta_2(\Delta\phi) \cdot \tau_2(\phi)$$

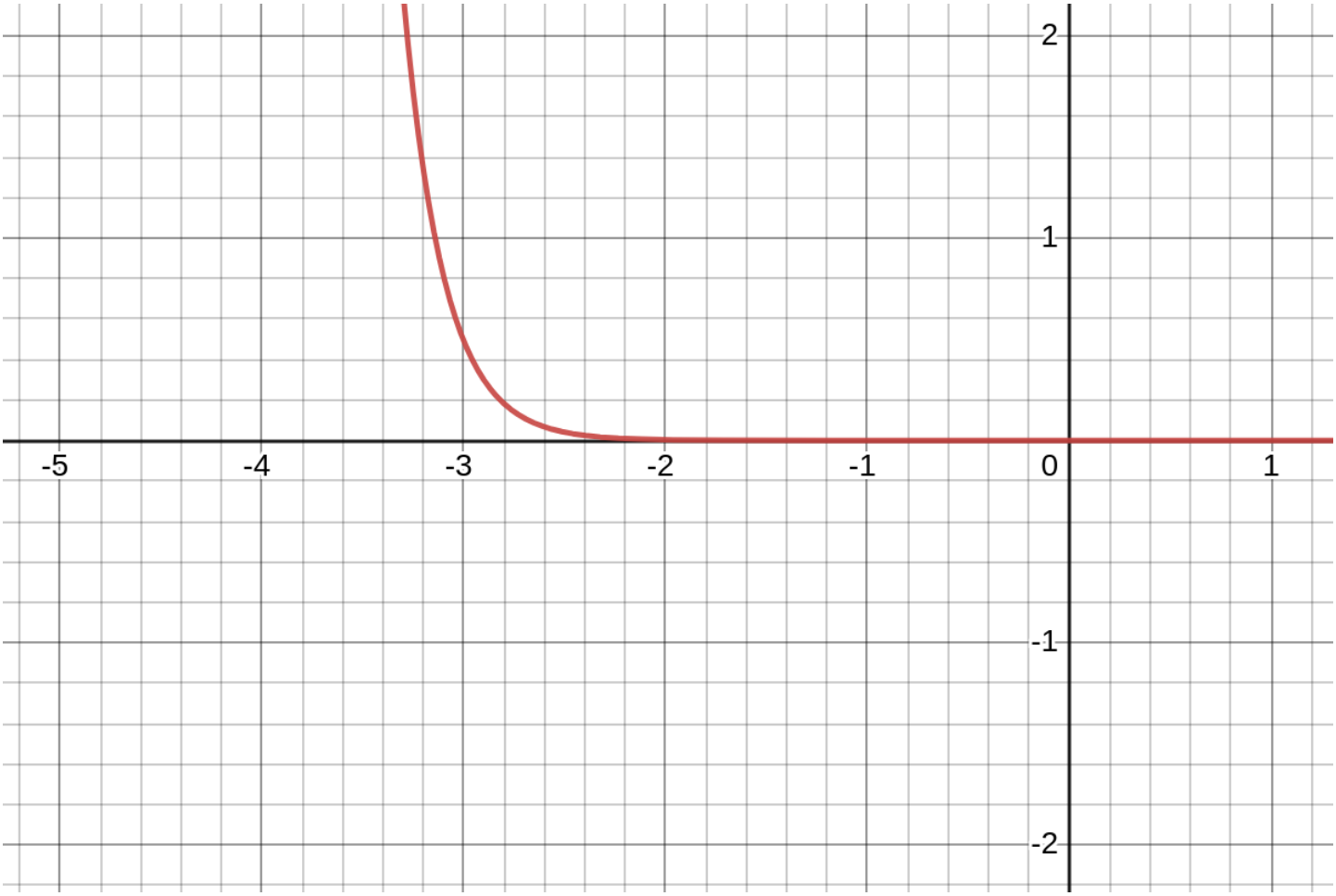
where:

- ϕ: angular error
- $\Delta\phi_t = |\phi_t| - |\phi_{t-1}|$

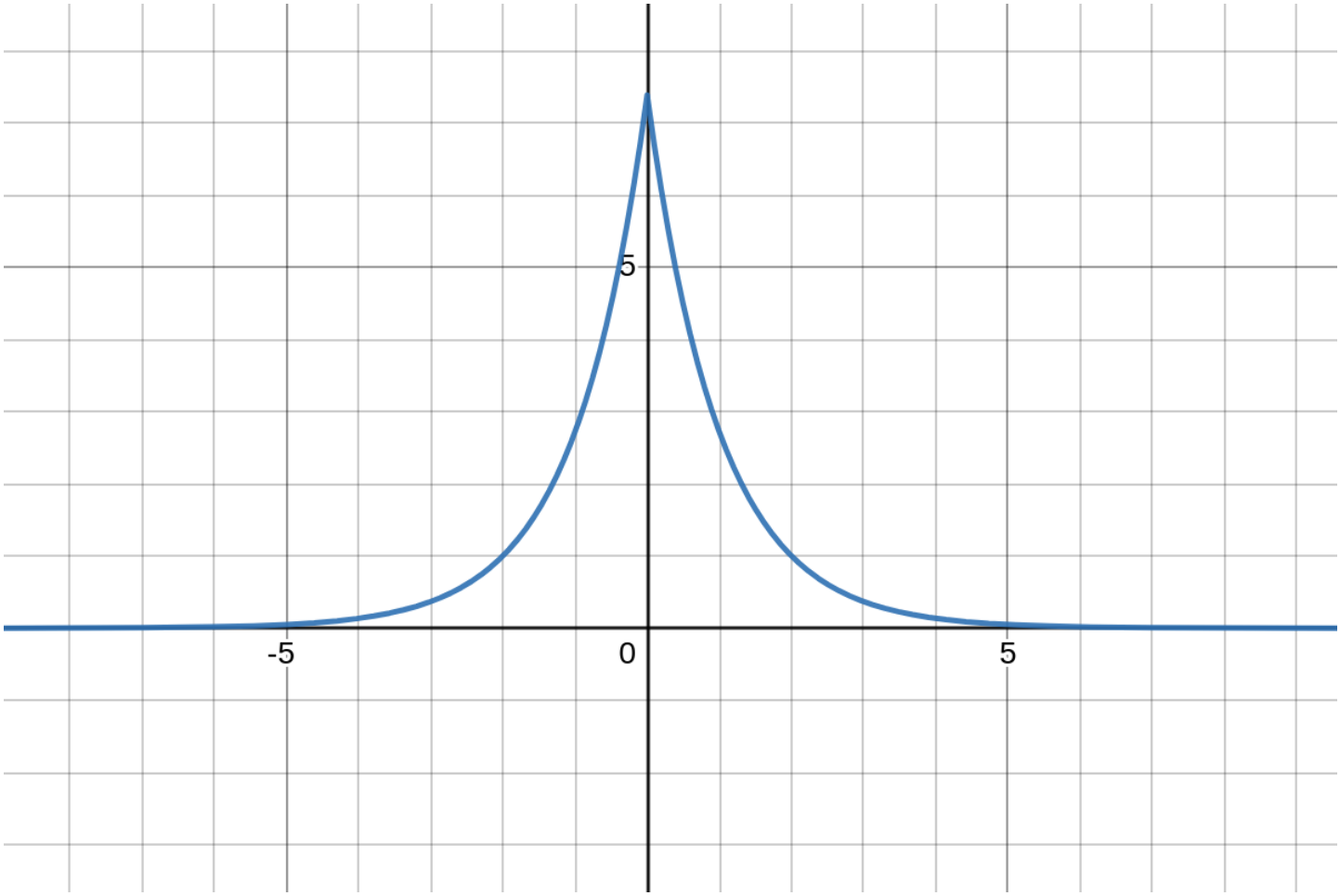
and

$$\beta_1(\Delta\phi) = \begin{cases} 0.5 \text{ if } \Delta\phi > 0 \\ 1 \text{ otherwise} \end{cases}$$

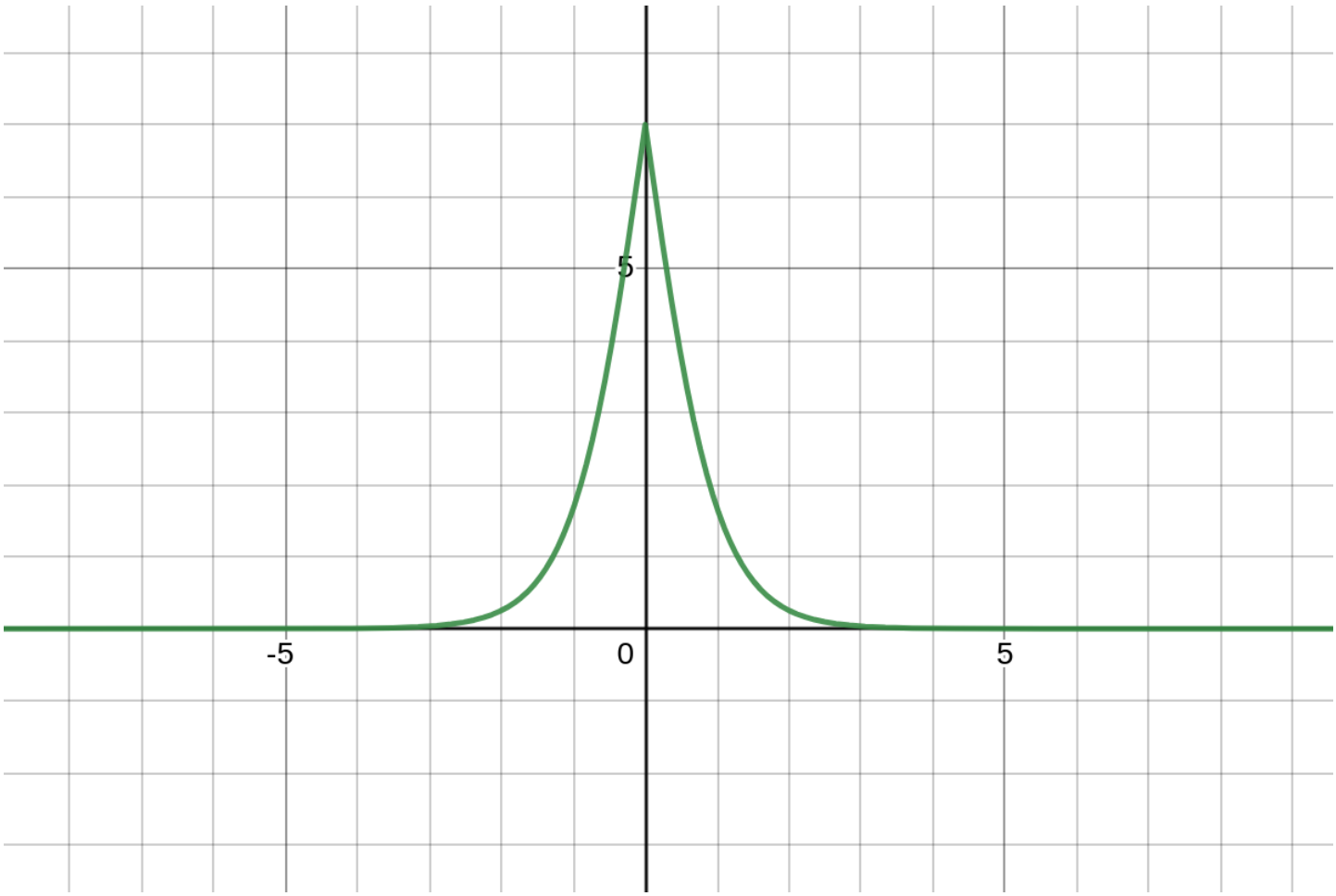
$$\beta_2(\Delta\phi) = \exp(-0.5(\pi + \dot{\phi}))$$



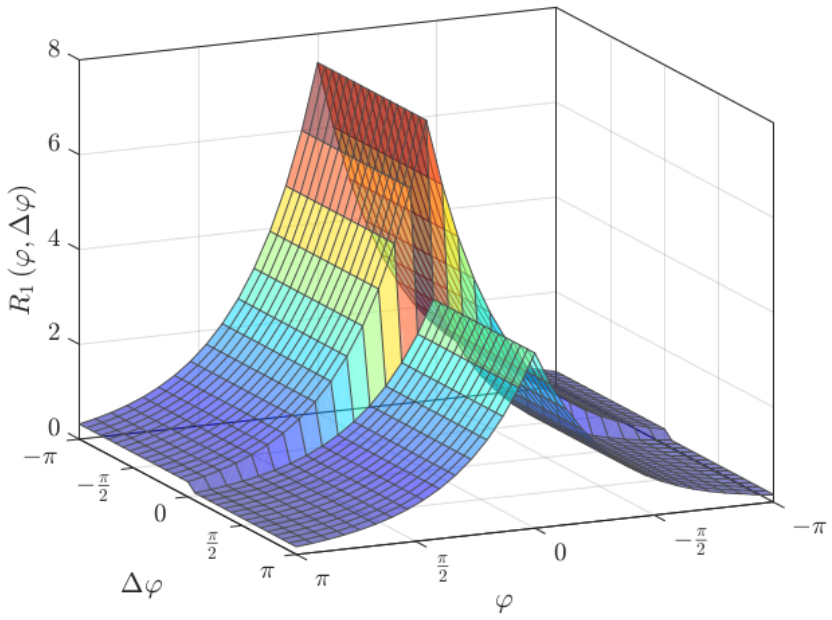
$$\tau_1(\phi) = \exp(2 - |\phi|)$$



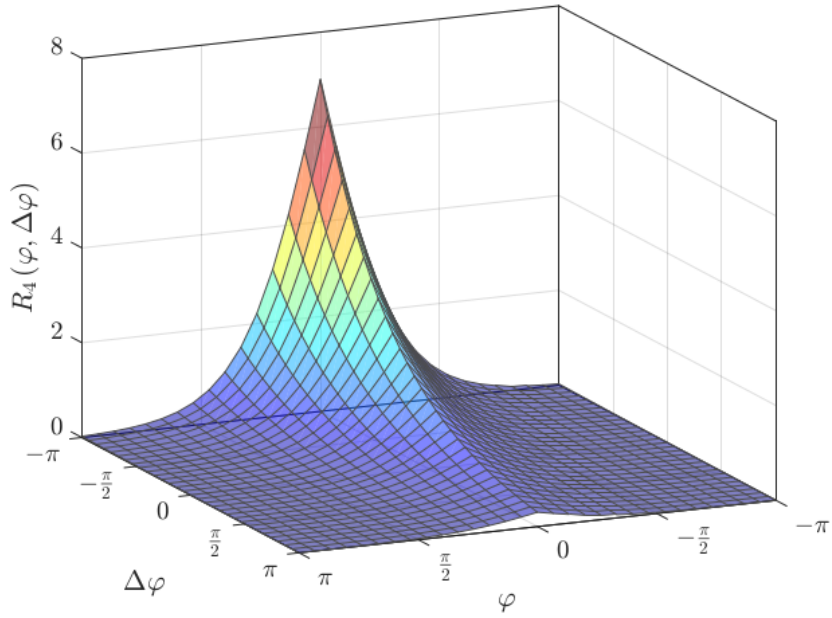
$$\tau_2(\phi) = \frac{14}{1 + \exp(2|\phi|)}$$



Shape of reward R_1



Shape of reward R_4



Results

RIGID SPACECRAFT

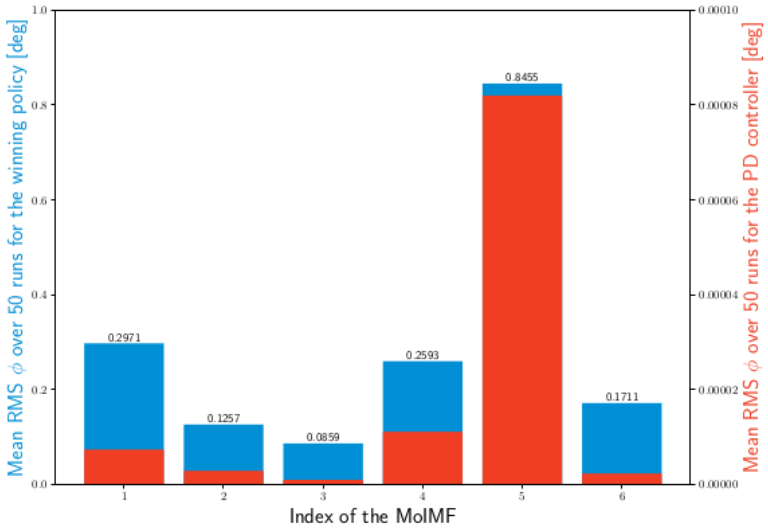
The proposed reward functions were initially used to train four different policies for a rigid spacecraft, the winning reward was R_4

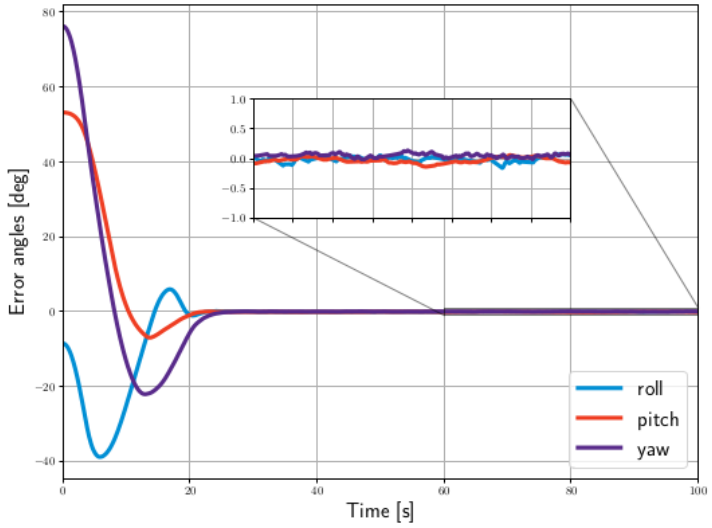
Order	Reward function	Mean RMS φ [deg]
1	$R_4(\varphi, \Delta\varphi)$	0.086
2	$R_3(\varphi, \Delta\varphi)$	0.153
3	$R_2(\varphi, \Delta\varphi)$	0.16
4	$R_1(\varphi, \Delta\varphi)$	0.29

To fully assess the performance of a certain policy, it is put under testing against six different satellites. The moment of inertia matrix of each of these six satellites would differ from that of the original satellite (against which the policy is trained) by a multiplication factor *MoIMF*.

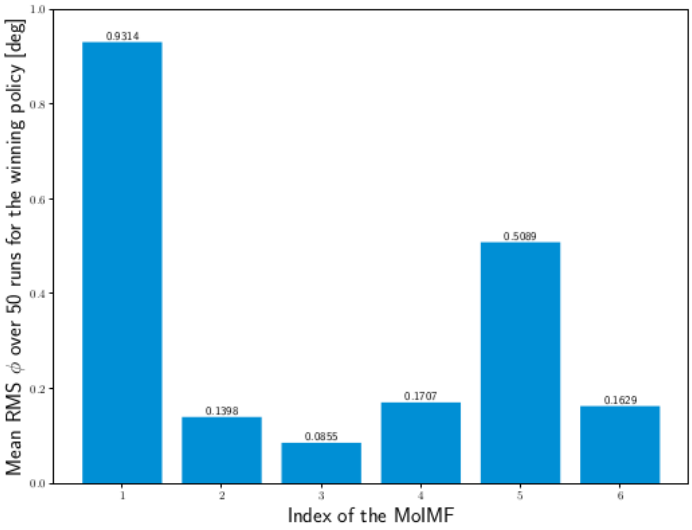
MoI-MF index	1	2	3	4	5	6
MoI-MF value	0.1	0.5	1	2	5	M

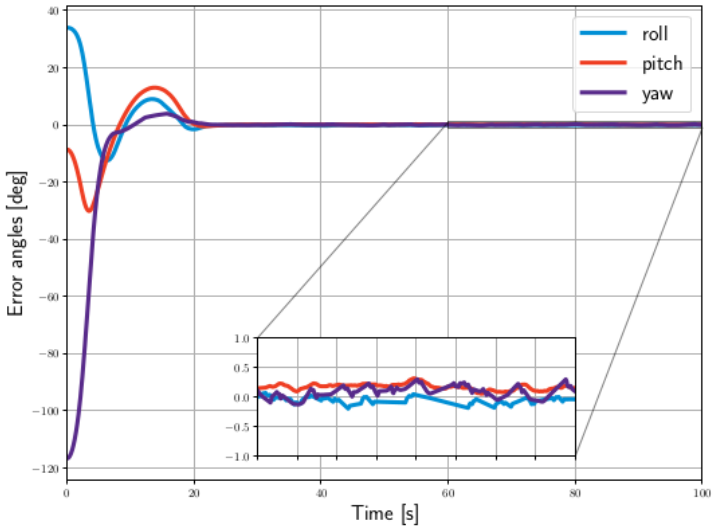
The performance of the winning policy is benchmarked against the that of a simple Lyapunov based PD controller.





FLEXIBLE SPACECRAFT





It is interesting to note that the same algorithms learns control policies for two different environments (i.e rigid and flexible spacecraft) with equal success.

Keywords

- Spacecraft with flexible appendages
- TD3 algorithm