REINFORCEMENT LEARNING FOR ATTITUDE CONTROL OF A SPACECRAFT WITH FLEXIBLE APPENDAGES

Conference Paper · September 2022

CITATIONS
0

READS
265

4 authors:

Ahmed Mahfouz
University of Luxembourg
20 PUBLICATIONS **44** CITATIONS
SEE PROFILE

Ayrat Valiullin
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Alexey Lukashevichus
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Dmitry Pritykin
Bureau 1440
42 PUBLICATIONS **181** CITATIONS
SEE PROFILE

IAC–22–C1.1.2x73341

REINFORCEMENT LEARNING FOR ATTITUDE CONTROL OF A SPACECRAFT WITH FLEXIBLE APPENDAGES

Ahmed Mahfouz
University of Luxembourg, Luxembourg, Ahmed.mahfouz@uni.lu

Ayrat Valiullin
Russian Federation, imjobjob@gmail.com

Alexey Lukashevichus
Russian Federation lukashevichus.aa@yandex.ru

Dmitry Pritykin
Bureau 1440, Russian Federation, dpritykin@rambler.ru

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.

keywords: satellite, flexible appendages, attitude control, reinforcement learning, proximal policy optimization

INTRODUCTION

Reinforcement learning has become an acknowledged framework to design autonomous control algorithms for systems that can be described as Markov Decision Process problems [1]. It has proven to perform well on a number of classical control problems and is currently being tried on real-life applications, i.e. unmanned aerial vehicle motion control systems. The essence of the reinforcement learning approach is in training an agent through interaction with the environment and learning from experience what is called a control policy, which amounts to a probability distribution of taking admissible control actions given the current state of the system. The agent that has learned a control policy is an equivalent of a state feedback control law implementation. One advantage of the reinforcement learning approach in the spacecraft attitude control loop is the ability of online learning, i.e enhancing the control strategy on the base of the accumulated experience during the operations. Improving and adapting the controller performance over time adds greater autonomy to the control system, which is important for space missions, especially those encountering uncertainties. Nowadays that the space mission design paradigm is shifted towards distributed space systems [2] with multiple spacecraft operating in groups, it may become possible to obtain greater amounts of training data and share it across

the group as different spacecraft explore different phase space regions [3].

So far the authors are aware of a few successful attempts of applying reinforcement learning to satellite attitude control [4, 5, 6]. These works are mainly focused on the control of a rigid body with a fixed dynamics, except [5] where a space environment simulator is employed as a training environment. All referenced works use the Proximal Policy Optimization (PPO) technique to train the control policy, except [5] that compares the PPO performance with the Soft-Actor Critic (SAC) algorithm. It is concluded that if the priority of the satellite is to be robust in sudden disturbances, PPO is the best algorithm, whereas SAC is better for fast attitude target. The latter is also the most comparable algorithm with the PID controller in terms of stability. An important conclusion made is that three is no need to re-tune reinforcement learning algorithms to obtain a good response, It is thus possible to complete the computation intensive learning while still on the ground to produce a control system, whose performance is comparable to conventional techniques. Furthermore, this system is able to continue to learn online, adapting its performance to the factors that had not been accounted for at design stages.

Following state of the art research, the PPO approach is used in this work as it is known for its learning stability and relative computational simplicity [6]. We start by formulating our problem in terms of Markov Decision Process setup, which implies presenting the system's state space S, the agent actions space A, the state-action transition function as the systems evolution operator (or the rule of proceeding to the next state given the current state and the choice of control action), and shaping the state-action reward function. The latter step is most crucial as it is the cumulative reward (reward along a trajectory) function that is optimised by the PPO algorithm in its search of the control policy. It is important to note that the resultant policy is not instantiated as a look-up table for all possible states, but is approximated by a neural network, which is trained to learn the control policy.

As far as the environment model is concerned, a microsatellite in a circular low Earth orbit is considered. The spacecraft is acted upon by the gravitational torque, all other environmental disturbances are modeled as a random torque. We use the quaternion kinematics and Euler dynamical equations to model the attitude dynamics of the spacecraft. Rest-to-rest slew maneuvers from arbitrary initial orientations are studied with the invariable control objective of aligning the body frame of the spacecraft with its orbital frame (e.g. for nadir pointing). Attitude determination algorithms and the corresponding errors are not considered in this study. The first part of the study is about problem formulation, reward-shaping, and finally training the agent and comparing the obtained results with the state-of-the-art [4, 5, 6]. The second part of the study considers the same spacecraft with flexible appendages added to its structure. 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. The flexible elements model is incorporated into the system's dynamics following the approach proposed in [7]. 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. Again PPO is used to learn the control policy for rest-to-rest slew maneuvers in the extended system.

1. Proximal Policy Optimization

Any control system can be divided into two parts: a controller and a plant to be controlled, in other words these two parts can be referred to as mutually interacting agent and environment. The agent observes the environment, then chooses a control action and passes it to the environment, which results in the change of the system's state. It must be noted that while observing the environment the agent not only identifies the system's state, which is required to compute the control action for the next step. Additionally the agent at each step "senses" or acquires a reward for what it has done to arrive to the observed state. The reward mechanism is important for the agent to "become aware" of control objectives and best ways of satisfying them. This loop is iterated until a termination condition is satisfied. The described routine (Figure 1) forms the framework for reinforcement learning control techniques.

Reinforcement learning control problems are usually formulated in terms of Markov Decision Processes (MDP) consisting of:

- all possible states S of the system;
- all possible control actions A_s that the agent can choose while finding itself in the corresponding states;
- state-action transition function via the evolution operator $\hat{F}$.

The control policy π playing the role of a control law is defined as a conditional probability of outcomes from the set A_s given the state s:

$$\pi(\mathbf{a}|\mathbf{s}) = P(\text{choose action } \mathbf{a} \text{ in state } \mathbf{s}). \quad (1)$$

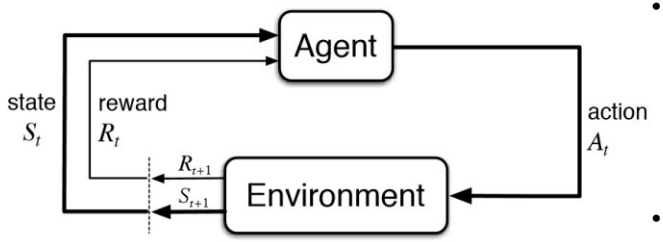

Fig. 1: Reinforcement learning loop [1]

The objective of reinforcement learning is to learn the policy π which maximizes the discounted reward R:

$$R_t(s_t) = \sum_{k=0}^{\mathsf{T}} \gamma^k r_{t+k}, \tag{2}$$

where $r_t \in \mathbb{R}$ is the immediate reward. The discounting factor $\gamma \in (0,1)$ reduces the contribution of long-term rewards thus stimulating the agent to learn receiving reward in the near future.

The value function $V(s_t)$ is the expected cumulative future reward for a given initial state:

$$V(s_t) = \hat{\mathbb{E}}(R_t(s_t)). \tag{3}$$

Using the value function, it is possible to construct an optimal policy. With smaller environments, it is possible to find the policy and value functions using dynamic programming but larger ones or those with continuous action space require more sophisticated methods. This is where the neural networks come into play. Adaptation of neural networks to solve reinforcement learning problems is usually referred to as "deep reinforcement learning". Basically, artificial neural networks are highly non-linear function approximators that require learning (or tuning their weights) with examples of possible inputs and correct outputs.

The method of reinforcement learning used in this study is Proximal Policy Optimization [8]. It uses training samples generated from the environment to iteratively approximate value and policy functions with neural networks.

2. Models of Spacecraft Attitude Dynamics

The goal of our study is to apply the reinforcement learning approach to spacecraft attitude control problems. Therefore, the models described in this section will play a role of environment (Fig. 1) in the reinforcement learning setup.

2.1 *Reference frames*

The following reference frames are used:

- The Orbital reference frame $\mathscr{F}^o$ with the origin at the center of mass of the satellite, z-axis pointing away from the center of the Earth, y-axis along the cross product of the satellite's center of mass position and velocity vectors, and x-axis completing the frame according to the right hand rule.
- The Body-fixed reference frame $\mathscr{F}^b$ with the origin at the satellite's center of mass. Its three axes coincide with the three principal axes of inertia of the satellite.

All vector transformations are given in the quaternion notation. A quaternion $\mathfrak{q}^{yx}$ is said to relate two reference frames $\mathscr{F}^x$ and $\mathscr{F}^y$. The representations of any given vector r in these frames are related by:

$$\mathbf{r}^y = \mathfrak{q}^{yx} \circ \mathbf{r}^x \circ \tilde{\mathfrak{q}}^{yx}, \tag{4}$$

where "$\circ$" denotes quaternion multiplication, $\tilde{\mathfrak{q}}$ is conjugate of q, and r^x is considered to be a quaternion with zero scalar part.

2.2 *Attitude Dynamics of a Spacecraft Equipped with Reaction Wheels*

2.2.1 *Rigid Spacecraft*

If a spacecraft is assumed to be a rigid body controlled by a set of reaction wheels, the rotational motion can be described by the Poisson's and Euler's equations [9].

The Poisson's kinematics equation expressed in quaternions is:

$$\dot{\mathfrak{q}}^{ob} = \frac{1}{2}\mathfrak{q}^{ob} \circ \Omega^b, \tag{5}$$

where $\mathfrak{q}^{ob}$ is the unit quaternion that transforms any vector from the $\mathscr{F}^b$ frame to the $\mathscr{F}^o$ frame, ω is the absolute angular velocity of the satellite (projected onto the body-frame). Ω^b is the spacecraft's angular velocity with respect to the $\mathscr{F}^o$ frame given by

$$\Omega^b = \omega^b - \mathfrak{q}^{bo} \circ \begin{bmatrix} 0 & \omega_0 & 0 \end{bmatrix}^{\mathsf{T}} \circ \tilde{\mathfrak{q}}^{bo}, \tag{6}$$

where ω_o is the mean motion of the spacecraft in orbit.

The rotational dynamics of a rigid satellite equipped with three reaction wheels, aligned with the axes of the $\mathscr{F}^b$ frame, can be described by

$$\begin{aligned} &\mathbb{J}^b\dot{\omega}^b + \omega^b \times (\mathbb{J}^b\omega^b + \mathbf{h}^b) = \mathbf{T}^b_{\text{ext}} + \mathbf{T}^b_{\text{ctrl}} + \mathbf{T}^b_{\text{dst}}, \\ &\dot{\mathbf{h}}^b = -\mathbf{T}^b_{\text{ctrl}} \end{aligned} \tag{7}$$

where $\mathbb{J}$ is the tensor of inertia of the spacecraft, $\mathbf{h}$ is the angular momentum of reaction wheels, $\mathbf{T}_{\text{ext}}$, $\mathbf{T}_{\text{ctrl}}$, and $\mathbf{T}_{\text{dst}}$ are respectively the external, control and disturbance torques acting on the satellite.

The external torques in the model used in this study are represented by the gravity-gradient torque given by:

$$\mathbf{T}_{\text{gg}} = 3\omega_0^2 \left(\mathbf{e}_o^b \times \mathbb{J} \mathbf{e}_o^b \right), \tag{8}$$

where $\mathbf{e}_o^b$ is the unit vector from the center of the gravity field to the body's center of mass.

The disturbance torque $\mathbf{T}_{\text{dst}}$ is modeled as a normally distributed random variable. The control action $\mathbf{T}_{\text{ctrl}}$ is computed for each control loop according to the algorithm described in the next section. The control goal in the subsequent numerical experiments is to align the body frame and the orbital frame, so that the required quaternion $\mathfrak{q}_{req}^{bo}$ and angular velocity Ω_{req}^b are

$$\mathfrak{q}_{req}^{bo} = [1,0,0,0]^\intercal, \quad \Omega_{req}^b = [0,0,0]^\intercal. \tag{9}$$

2.2.2 *Spacecraft with flexible appendages*

For a spacecraft with flexible appendages such as large solar panels or flexible antennae we shall follow the approach outlined in [7]. The kinematics equations (5)-(6) are kept unchanged, whereas the rotational dynamics equations are subject to the following modification

$$\begin{aligned} &\mathbb{J}^b \dot{\boldsymbol{\omega}}^b + \boldsymbol{\omega}^b \times (\mathbb{J}^b \boldsymbol{\omega}^b + \mathbf{h}^b) = \mathbf{T}_{\text{ext}}^b + \mathbf{T}_{\text{ctrl}}^b + \mathbf{T}_{\text{dst}}^b - \mathbb{L}_P \ddot{\boldsymbol{\eta}}, \\ &\dot{\mathbf{h}}^b = -\mathbf{T}_{\text{ctrl}}^b, \\ &\ddot{\boldsymbol{\eta}} + \text{diag}\left(2\zeta_i \omega_i\right) \dot{\boldsymbol{\eta}} + \text{diag}\left(\omega_i^2\right) \boldsymbol{\eta} = -\mathbb{L}_P^\intercal \dot{\boldsymbol{\omega}}. \end{aligned} \tag{10}$$

This formulation includes the dynamics of the flexible appendages given mainly by the third equation in the set of (10), which has coupling terms with the first equation. The system now depends on the appendage flexible modes ω_i through three matrices: the stiffness matrix $\mathbb{K} = \text{diag}\left(\omega_i^2\right)$, the damping matrix $\mathbb{D} = \text{diag}\left(2\zeta_i \omega_i\right)$, and the modal participation factor matrix $\mathbb{L}_P$ at point P where the flexible solar panel is attached to the main body. For detailed description of the model and its derivation, please, refer to [7].

The external torques and the disturbance torques in the right-hand side of the first of the equations (10) are the same as in the rigid spacecraft model. So is the control goal, which is to stabilize the spacecraft in the orbital orientation described by (9).

2.2.3 *Model parameters*

The spacecraft moves in the 600 km altitude circular orbit. As only the central gravity field is considered the orbital motion is Keplerian and other orbital elements are not required for simulations.
The parameters required by the equations of rotational motion models are specified below. The values are given according to [10].

The inertia tensor of the rigid spacecraft:

$$\mathbb{J}^b = \begin{pmatrix} 75 & 1 & 2 \\ 1 & 40 & -1 \\ 2 & -1 & 80 \end{pmatrix} \text{ kg} \cdot \text{m}^2. \tag{11}$$

The control actuation is carried out by a set of three orthogonal reaction wheels, each can produce the torque of $|\dot{h}| \leq \tau_{max} = 0.05$ N, and its angular momentum is limited by $h_{max} = 0.5$ Nms.

The modal participation factor matrix for the flexible model

$$\mathbb{L}_P = \begin{pmatrix} 0 & 12.5 & 0 \\ -3.84 & 0 & 0 \\ 0 & 2.51 & 0 \end{pmatrix},$$

this matrix is derived based on the inertia tensor of the solar panel and the point where it is attached to the spacecraft's main body [10].

Flexible mode's frequencies

$$(\omega_1, \omega_2, \omega_3) = (5.6, 19.3, 35.4) \text{ rad/s}$$

and flexible mode damping coefficients

$$(\zeta_1, \zeta_2, \zeta_3) = (0.005, 0.005, 0.005).$$

3. Reinforcement Learning Problem Setup

3.1 *Attitude Control Problem as a Markov Decision Process*

Formulating the control problem in terms of a Markov Decision Process (MDP) requires us to specify the state-space, action-space, the state-transition rule and the reward. They are as follows:

- The state is a 10d vector the spacecraft attitude quaternions q, the components of angular velocity $\boldsymbol{\omega}$ and the angular momentum of reaction wheels $\mathbf{h}$. In case of the model with flexible appendages six additional states $\boldsymbol{\eta}$ and $\dot{\boldsymbol{\eta}}$ introduced by (10) are appended to 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,

- The action space is

  $$A_s = m \cdot 10^{-i} \cdot T_{max} \quad \text{for } i \in \{1,2,3\},$$
  $$m = \{m \colon m = -1 + 0.04n,\ n \in \{0,1,2,\ldots,50\}\},$$

  where T_{max} is the maximum allowable torque by the actuators.

Let us note that our numerical experiments showed that using finer mesh for the action space has little to no improvement compared to the one listed above let alone being more computationally expensive.

- The state transition $\hat{F}$ is given by the system of equations (5) and (7) for the rigid spacecraft problem or equations (5) and (10) for the flexible spacecraft problem.

One of the most important parts in reinforcement learning is constructing the reward function which makes the agent learn the preferred policy. The candidate reward functions are in general heuristic, and there is no rule of thumb when it comes to choosing one. Different reward functions varying in complexity as well as in rewarding criteria have been tried, but before the chosen reward functions can be introduced, the following definitions are introduced.

- The error angle, φ, is defined as,

$$\varphi = 2 \cdot \arccos\left(q_{e,0}\right), \qquad \varphi \in [-\pi, \pi]$$

with $q_{e,0}$ being the scalar part of the error quaternion.

- The absolute error angle difference, $\Delta\varphi$, is defined as,

$$\Delta\varphi = |\varphi| - \left|\varphi_{\text{prev}}\right|, \qquad \Delta\varphi \in [-\pi, \pi]$$

where φ_{prev} is the error angle from previous step (which can be expressed through the current state information, i.e. current error angle and angular velocity).

The goal of the control system is to drive φ to zero, while the best $\Delta\varphi$ the agent can achieve is $\Delta\varphi = -\pi$. A good reward function encourages positive steps towards the control goal in terms of the decrease in the error angle and the right direction of the angular velocity. Another thing a reward function may take into account is discouraging overshoots which lead to long settling times. This may also be done by punishing wrong directions of angular velocity. Suppression of long transient processes is especially essential in the problem of the flexible spacecraft, where oscillations in the flexible appendage are easily excited if special care is not taken. With this in mind we have constructed and tested a number of rewards and the four finalist reward functions are presented as follows,

$$\begin{aligned} R_1(\varphi, \Delta\varphi) &= \beta_1(\Delta\varphi) \cdot \tau_1(\varphi), \\ R_2(\varphi, \Delta\varphi) &= \beta_1(\Delta\varphi) \cdot \tau_2(\varphi), \\ R_3(\varphi, \Delta\varphi) &= \beta_2(\Delta\varphi) \cdot \tau_1(\varphi), \\ R_4(\varphi, \Delta\varphi) &= \beta_2(\Delta\varphi) \cdot \tau_2(\varphi), \end{aligned} \tag{12}$$

where the scalar-valued functions, $\beta_1(\Delta\varphi)$, $\beta_2(\Delta\varphi)$, $\tau_1(\varphi)$, and $\tau_2(\varphi)$ are defined as

$$\begin{aligned} \beta_1(\Delta\varphi) &= \begin{cases} 0.5, & \Delta\varphi > 0 \\ 1, & \text{otherwise} \end{cases}, \\ \beta_2(\Delta\varphi) &= e^{-0.5(\pi+\dot{\varphi})}, \\ \tau_1(\varphi) &= e^{(2-|\varphi|)}, \\ \tau_2(\varphi) &= \frac{14}{1+e^{2|\varphi|}}. \end{aligned}$$

To put the introduced reward functions into perspective, the surface plots in figures 2 and 3 depict the shapes of reward functions $R_1(\varphi, \Delta\varphi)$ and $R_4(\varphi, \Delta\varphi)$ (see equation (12)).

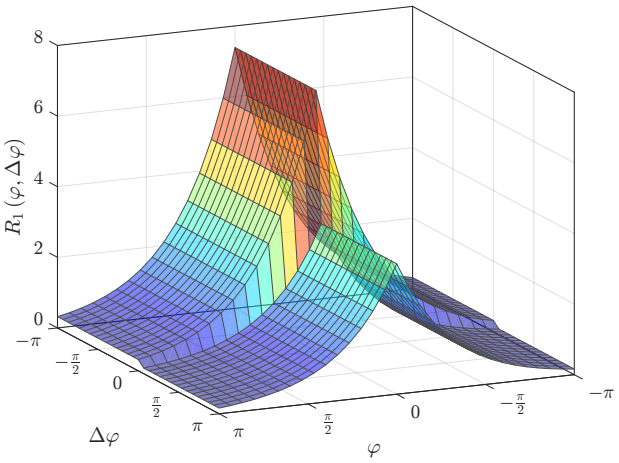

Fig. 2: Shape of the reward function $R_1(\varphi, \Delta\varphi)$

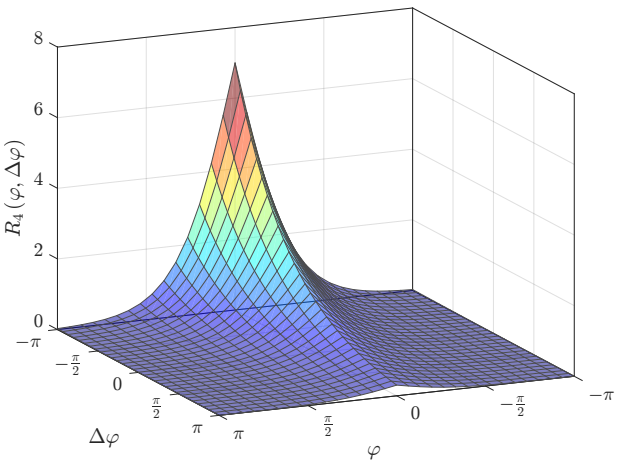

Fig. 3: Shape of the reward function $R_4(\varphi, \Delta\varphi)$

It can be seen either from the set of equations (12) or from figures 2 and 3 that the goal of the four reward func-

tions is to incentify actions that lead to small error angles, moreover, the reward would be much greater if the action is causing the error angle to evolve from a higher to a lower value.

4. Results and Discussion

The proposed reward functions (12) were initially used to train four different policies for the a rigid satellite with the attitude-related parameters in 2.2.3. The agent was trained on a computational node that comprises 8 parallel CPUs for a total of 12500 epochs (500 batches with 25 episodes each) for each policy which took on average 10 hours per policy. By that time, the minimum batch reward had plateaued for all the trained policies. After the training process ended for each of the four reward functions, it was clear that the wining reward function is $R_4(\varphi, \Delta\varphi)$. Table 1 presents the final ranking of the reward functions based on the mean RMS angle error for a batch of 50 episodes.

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

Table 1: Reward functions performance for the four proposed reward functions

The evolution of the minimum reward in a batch, which comprises 25 episodes, for the winning policy (i.e. the one which is trained over $R_4(\varphi, \Delta\varphi)$) is illustrated in Fig. 4.

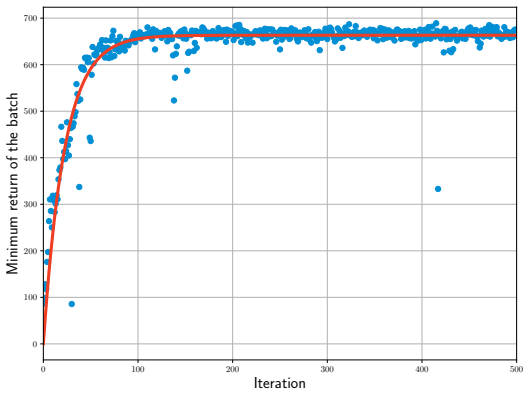

Fig. 4: Reward evolution for an agent using the reward function $R_4(\varphi, \Delta\varphi)$

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

Table 2: Moment of Inertia Multiplication Factors (MoI-MF) for the test satellites

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, see equation (11)) by a multiplication factor. Table 2 delineates the six different Moment of Inertia Multiplication Factors (MoIMF), where the value of M in the table is diag$(0.5, 0.2, 1.67)$.

The performance of the winning policy is benchmarked against the that of a simple Lyapunov based PD controller [11]. The results of this experiment for the six satellites are depicted in the following Fig.5. It is quite

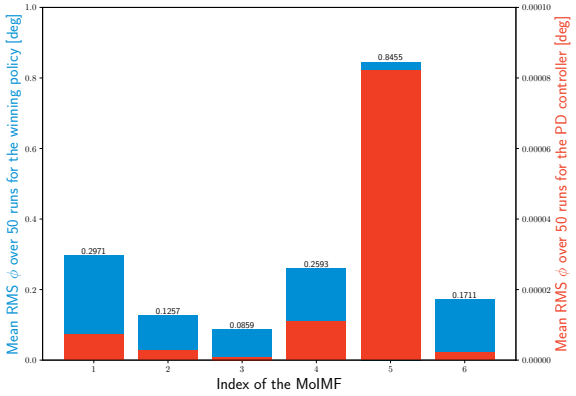

Fig. 5: Performance of the winning policy against that of a PD controller for a rigid satellite

interesting to see that the controlling policy still performs adequately even when it is controlling a completely different satellite.

The characteristic behaviour of the controlled rigid spacecraft attitude for an arbitrary initial condition as expressed through a set of Euler angles is shown in the Fig. 6.

As for the flexible spacecraft problem, the winning reward function, $R_4(\varphi, \Delta\varphi)$, was used to train an agent, again for 500 batches, each batch comprising 25 episodes. The evolution of the minimum reward in a batch for that policy is illustrated in Fig. 7. It is interesting to notice that

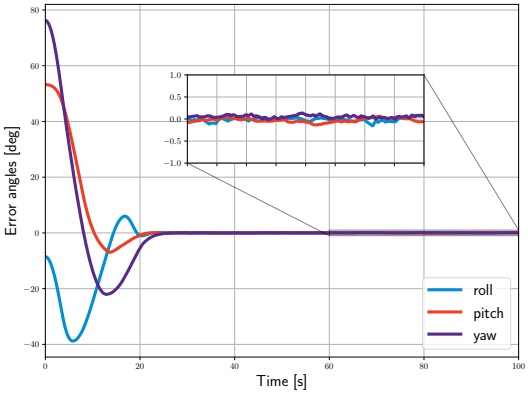

Fig. 6: Euler angles for a single episode of the rigid SC controlled by the winning policy

although the system of a satellite with flexible appendages is more complex than a rigid satellite, it is clear by comparing Figures 4 and 7 that the agent which controllers a satellite with flexible appendages reaches maximum possible reward in less amount of iterations.

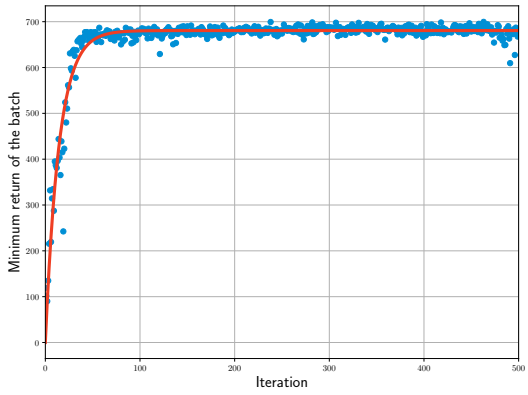

Fig. 7: Reward evolution for an agent using the reward function $R_4(\varphi, \Delta\varphi)$

The winning policy was once again used to control six different satellites that have different moment of inertia tensors. The performance of the control policy against the six satellites is depicted in Fig.8.

Finally, the characteristic behaviour of the controlled rigid spacecraft attitude for an arbitrary initial condition as expressed through a set of Euler angles is shown in the following Fig. 9.

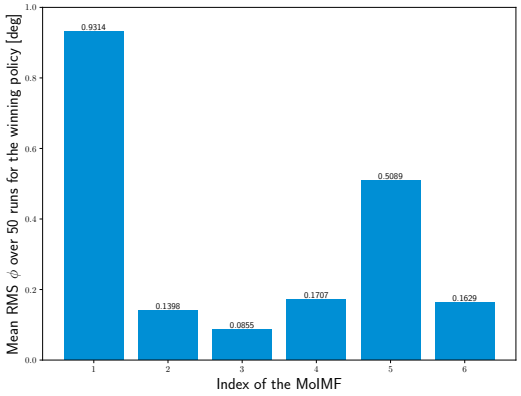

Fig. 8: Performance of the winning policy against that of a PD controller for a satellite with flexible appendages

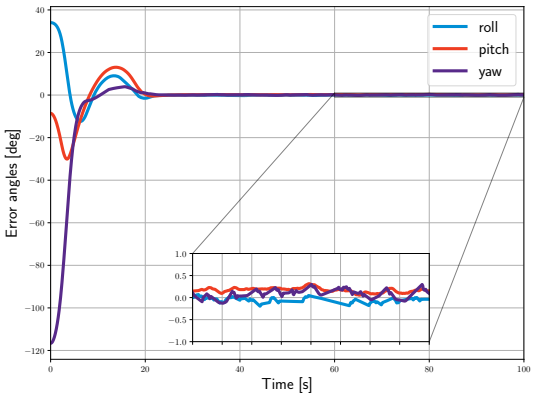

Fig. 9: Euler angles for a single episode of the SC with flexible appendages controlled by the winning policy

Conclusion

This work presents a reinforcement learning approach to formulating the spacecraft attitude control problem and obtaining the solution to this problem in the form of a control policy. All in all, the proposed approach is shown to converge to a policy that resembles the performance of the quaternion feedback regulator for a rigid spacecraft. It is also shown that for the same problem setup which has been used for the rigid spacecraft, a policy can be learned to takes into account the highly nonlinear dynamics caused by the presence of flexible elements that need to be brought to rest in the required attitude. 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. We believe that it is in

this direction that the advantage of the RL approach lies, because it better adapts to the environments with higher degree of uncertainty, such as the problem of a flexible spacecraft attitude control. It is usually quite difficult to identify the flexible modes and correctly incorporate them into the control loop. Furthermore, the dynamics of the spacecraft's flexible parts can change over time due to material degradation, and the ability of RL agent to learn from experience can be used to make the control policy track any such changes.

ACKNOWLEDGMENT

This work is supported by the Luxembourg National Research Fund (FNR) – AuFoSat project, BRIDGES/19/MS/14302465.

REFERENCES

[1] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[2] D. Cellucci, Nick B. Cramer, and Jeremy D. Frank. *Distributed Spacecraft Autonomy*.

[3] Seongin Na, Tomáš Krajník, Barry Lennox, and Farshad Arvin. Federated reinforcement learning for collective navigation of robotic swarms, 2022.

[4] F. Vedant, J.T. Allison, M. West, and A. Ghosh. Reinforcement learning for spacecraft attitude control. In *Proceedings of the International Astronautical Congress, IAC*, volume 2019-October, 2019.

[5] Vanessa Tan, John Leur Labrador, and Marc Caesar Talampas. Mata-rl: Continuous reaction wheel attitude control using the mata simulation software and reinforcement learning. In *Proccedings of 35th Annual Small Satellite Conference*, 2021.

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

[7] Daniel Alazard, Christelle Cumer, and Khalid Tantawi. Linear dynamic modeling of spacecraft with various flexible appendages and on-board angular momentums. In *7th International ESA Conference on Guidance, Navigation & Control Systems (GNC 2008)*, pages 1–14, Tralee, IE, 2008.

[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[9] Bong Wie and Peter M. Barba. Quaternion feedback for spacecraft large angle maneuvers. *Journal of Guidance, Control, and Dynamics*, 8(3):360–365, 1985.

[10] I A Courie, Francesco Sanfedino, and Daniel Alazard. Worst-case pointing performance analysis for large flexible spacecraft. *ArXiv*, abs/2106.01893, 2021.

[11] Bong Wie, Haim Weiss, and Ari Arapostathis. Quaternion feedback regulator for spacecraft eigenaxis rotations. *Journal of Guidance, Control, and Dynamics*, 12(3):375–380, 1989.