Underactuated Attitude Control with Deep Reinforcement Learning

Introduction

Autonomy is a key challenge for future space exploration endeavors. Deep Reinforcement Learning holds the promises for developing agents able to learn complex behaviors simply by interacting with their environment. This work investigates the use of Reinforcement Learning for satellite attitude control applied to two working conditions: the nominal case, in which all the actuators (a set of 3 reaction wheels) are working properly, and the underactuated case, where an actuator failure is simulated randomly along one of the axes. In particular, a control policy is implemented and evaluated to maneuver a small satellite from a random starting angle to a given pointing target. In the proposed approach, the control policies are implemented as Neural Networks trained with a custom version of the Proximal Policy Optimization algorithm, and they allow the designer to specify the desired control properties by simply shaping the reward function. The agents learn to effectively perform large-angle slew maneuvers with fast convergence and industry-standard pointing accuracy.

Environment Formulation as MDP

The simulation environment has been modelled to fit a Markov Decision Process structure as follows:

State space

The environment state space, and consequently the agent input, is constituted by the current attitude of the simulated spacecraft, i.e. the orientation quaternion plus the angular rates along the three axes, and the RWs speeds.

$$s = < q_0, q_1, q_2, q_3, w_x, w_y, w_z, rw_x, rw_y, rw_z >$$

Action space

The action space is a 3-dimensional vector representing the torques command for the RWs.

$$a_t = < M_x, M_y, M_z >$$

Reward function

The design of the reward function is a fundamental part of the model since it expresses the objective that the agent maximizes. We tested various approaches and thoroughly considered their impact on the agent performance and behavior and finally come up with a composite reward structure.

- If the angular speed along any of the 3 axes exceeds a certain threshold the reward is fixed to -1, no matter what the orientation of the satellite is. This ensures that the agent is incentivized to learn how to detumble the satellite.
- In case the angular rate of the satellite is within the threshold the reward is:

$$r = r_{target} - r_{penalty} + r_{bonus}$$

where r_{target} is a reward that is inversely proportional to the distance (in terms of the angle) of the satellite to the target orientation, $r_{penalty}$ adds an energy cost forcing the agent to be more efficient and, finally, r_{bonus} gives the agent an additional reward at convergence.

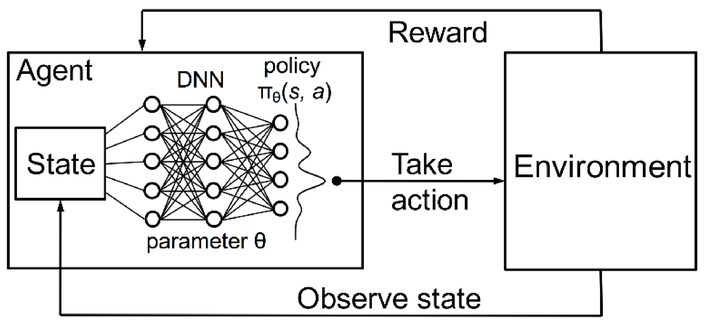


Figure 1
DRL agent interaction loop with the environment

Results and main remarks

The results shown are from the final control agents evaluated in10'000 episodes. To assess challenging maneuvers the starting angle is randomly drawn as $\theta \in [144°, 180°]$ about any rotation axis from the initial orientation. The observed horizon is of 1600 steps (corresponding to 800s), and the accuracy threshold depends on whether there is a failure or not. In Figure 3 an example from the nominal condition is presented as an example of how smooth and efficient is the control maneuver executed starting from the maximum distance towards a precision of more than 0.01 rad (red threshold in the graph). Figure 3 shows a control maneuver with a failure simulated along the X axis, and it can be seen that a relatively small residual periodic drift occurs after convergence. In correspondence of the angular drift the torques applied are visibly higher as the reaction wheels spin to restore the target position. Finally, in Figure 4, an example of angular curves show that over 10K evaluations, the controller (trained for the failure on the X axis) is able to always keep the satellite below a 0.25 rad precision in the worst cases, and on average with 0.1 rad pointing accuracy.

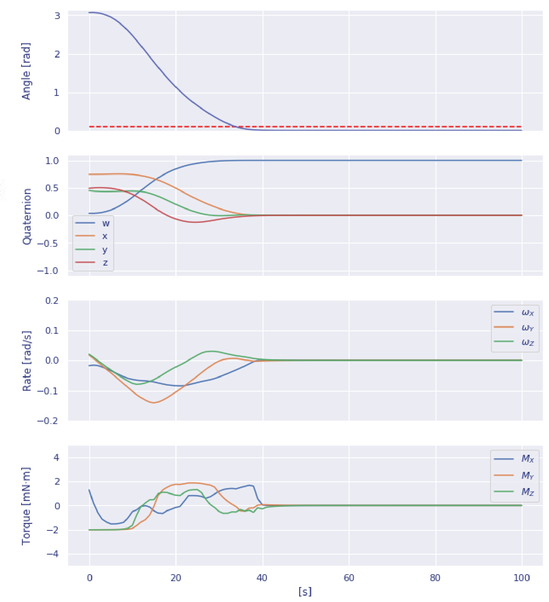


Figure 2
Control behaviour with fault on X (blue) axis

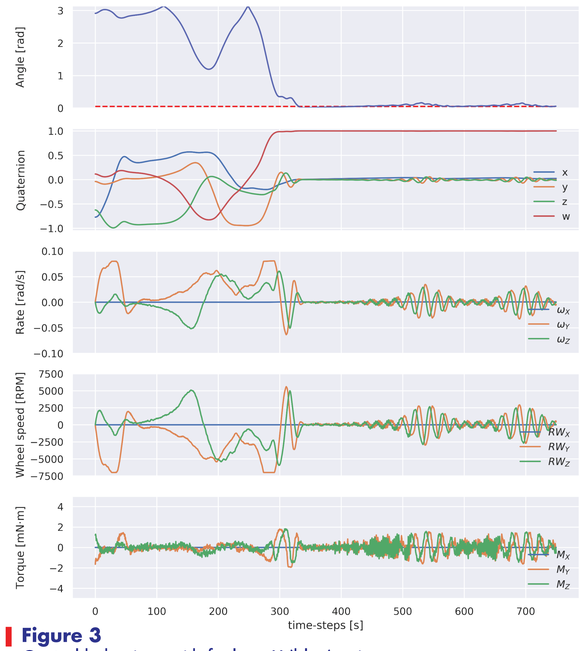


Figure 3
Control behaviour with fault on X (blue) axis

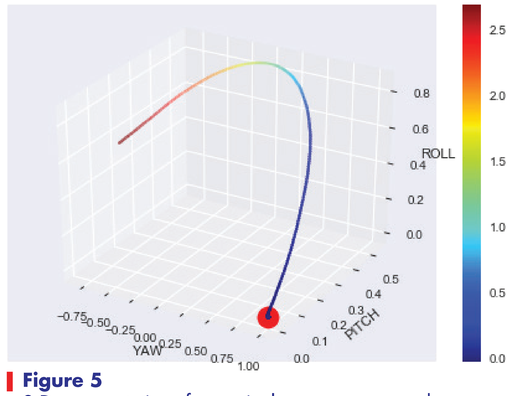


Figure 5
3-D representation of a nominal manoeuvre around the quaternion sphere

Final remark

The RL Agent can be trained to solve the attitude control problem both in nominal, and underactuated conditions (for asymmetrical platforms).

The Reward function can be shaped to naturally add the following constraints:

- Pointing accuracy threshold
- Minimize actuators energy consumption
- Avoiding RWs saturation
- Avoiding tumbling conditions (body rate speed limit)

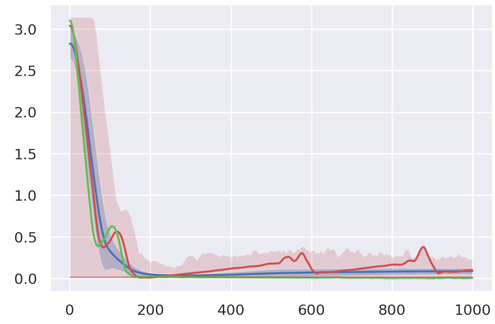
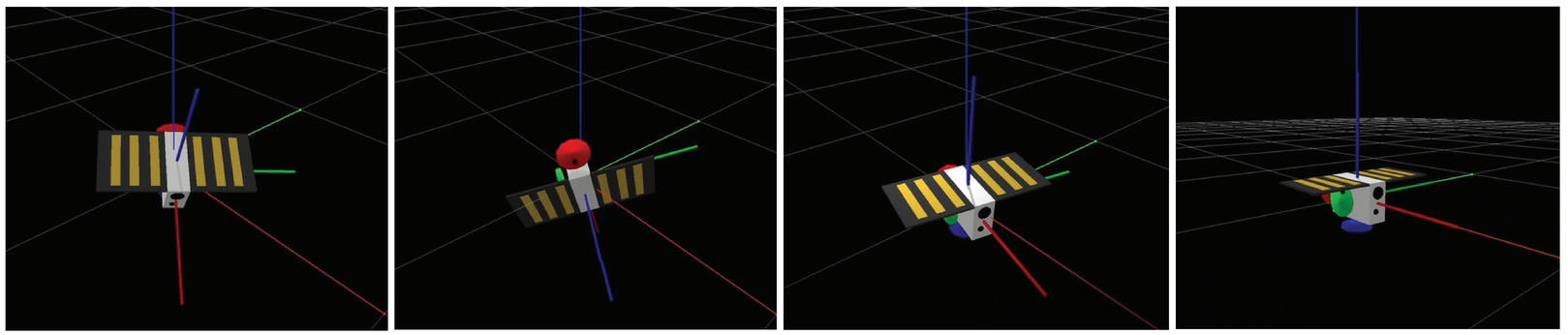


Figure 3
: Angular evolution of 10'000 simulations with failure on X axis. Red, blue and green represent the worst, average and best trajectories

Results and main remarks



1. Initial position
2. Maneuver
3. Maneuver
4. Target achieved

Matteo El Hariry
R&D Machine Learning Engineer
matteo.elhariry@argotecgroup.com