70th International Astronautical Congress, Washington D.C., United States, 21-25 October 2019.

IAC–19–C1.5.2

Reinforcement Learning for Spacecraft Attitude Control

Vedant[a*], James T. Allison[b], Matthew West[c], Alexander Ghosh[d]

[a]*Department of Aerospace Engineering, University of Illinois, United States,* vedant2@illinois.edu
[b]*Department of Mechanical Science and Engineering, University of Illinois, United States,* mwest@illinois.edu
[d]*Department of Industrial and Enterprise Systems Engineering, University of Illinois, United States,* jtalliso@illinois.edu
[d]*Department of Aerospace Engineering, University of Illinois, United States,* aghosh2@illinois.edu
[*] Corresponding Author

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.

Keywords: Attitude control, Reinforcement learning, Robust control, Machine learning, Artificial Intelligence, Adaptive control

Abbreviations

ACS Attitude Control System. 1–9

MDP Markov Decision Processes. 2, 4, 7

pdf probability distribution function. 2

POMDP Partially Observable Markov Decision Processes. 4

QRF Quaternion Rate Feedback. 2, 5

RL Reinforcement Learning. 2–9

1. Introduction

In this study, we aim to develop a framework which solves the general satellite attitude control problem. Spacecraft attitude control is the process of orienting a satellite toward a particular point in the sky, precisely and accurately. Most modern spacecraft offer active three-axis attitude control capability. Traditionally, satellite attitude control has been performed using several types of actuators, but the two main categories of Attitude Control Systems (ACSs) are momentum management and momentum exchange based devices. Momentum management based devices utilize external torques and hence can change the angular momentum of the satellite, such as attitude control thrusters and magnetic torque coils. Momentum exchange based devices produce torques by redistributing the angular momentum between satellite components, thus have no net external torques on

the satellite; this class of ACS include reaction wheel assemblies and control moment gyroscopes.

The pure attitude control problem, also known as the Euler rigid body rotation problem, has been studied for decades and several solutions exist [1, 2]. Despite this, the attitude control problem with realistic system constraints is a challenging problem for most current and future spacecraft missions. A key limitation of current control methods is to have state feedback control algorithms that guarantee stability and accuracy for realistic system constraints.

The current state-of-the-art solutions for attitude control problems split the ACS into two loops. An outer loop optimizes the performance of the system for some finite time horizon, using open-loop optimal control algorithms, such as Model Predictive Control (MPC) or Dynamic Programming (DP) based methods [3]. An inner loop tracks the trajectories obtained by the outer loop using state feedback-based control to perform the attitude control maneuvers. This provides a workaround for not having a global state feedback-based control systems, by finding trajectories that can be locally stabilized.

Reinforcement Learning (RL) has recently shown tremendous success in solving complex problems. RL is a method of finding the optimal response of a system, similar to that of dynamic programming methods, but without the "*curse of dimensionality*" [4, 5].

Most modern RL methods have been developed for discrete-time Markov Decision Processess (MDPs) [6]. All RL algorithms learn policies that provide a system with the action that leads to the best performance given the current state. Such a policy can be thought of as a surrogate state feedback control algorithm. RL has been demonstrated successfully for simple classical control problems, such as the inverted pendulum problem and the cart pole problem [7]. Figure 1 shows a conventional RL setup for control problems, where an agent interacts with an environment, and the actions of the agent produce feedback in the form of rewards and observations. The RL algorithm records the actions, observations, and rewards, and updates the agent, using various RL algorithms, at each epoch to maximize the expected reward.

All RL algorithms can be classified into two main categories: *value iteration* and *policy iteration* methods. Value iteration methods are generally more sample efficient, but work best with continuous state, discrete control type problems [8, 9]. Policy iteration methods can function for continuous space and continuous control type problems, but are generally

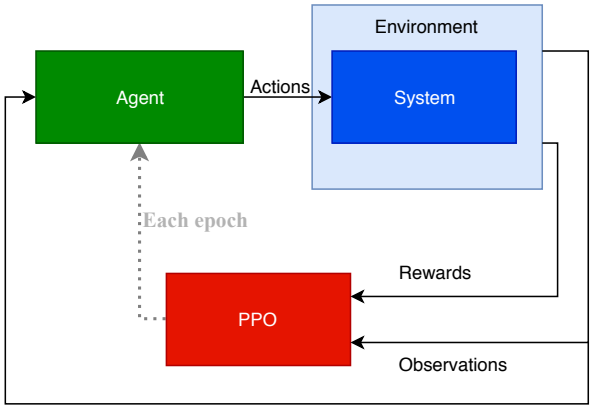

Fig. 1: RL setup for control problems

not as sample efficient [10]. The policy iteration based method, known as Proximal Policy Optimization (PPO) is considered in this study since the attitude control problem is a continuous control problem. Exploration of the search space in the PPO algorithm is performed by assuming probabilistic policies, where the actions taken for a given state is modeled using a Gaussian probability distribution function (pdf). The agent provides the mean action and standard deviation of that action, for an observation/state. A large standard deviation allows for more exploration, while a small standard deviation utilizes exploitation and also can be interpreted as a measure of how sure the agent is for a certain action.

This study has two main parts. First, the attitude control problem is formulated for the RL algorithm. The RL algorithm will be trained for the simple attitude control problem, with the only constraints being actuator saturation limits. The RL algorithm is then trained for a family of spacecraft, based on an existing satellite bus, to have a robust algorithm that can work for a variety of missions. The results for the RL agent are compared against conventional control methods, such as the Quaternion Rate Feedback (QRF) controller. Next, the RL agent is trained for a momentum exchange based system with higher-fidelity models.

2. Methodology

The satellite attitude control problem is formulated as a discrete-time MDP, to utilize the PPO algorithm to obtain solutions. The time discretization of the dynamical system is a relatively simple step and has been performed for the satellite attitude control problem to use with Dynamic programming or

Discrete-time multiple shooting methods [11]. The satellite attitude control problem is an MDP if the state vectors s_t at any time t are a composition of the attitude, represented by quaternion q_t and angular velocity, represented by ω_t, in Eq. (1).

$$s_t = [q_t, \omega_t] \tag{1}$$

Given that the system starts with an initial angular velocity (Eq. (2)) and some initial orientation (Eq. (3)), the attitude control problem involves two objectives, which are dependent on each other. The first objective is to achieve a desired angular velocity (Eq. (4)), also known as slew rate, at a desired time (t_d). The second objective is to achieve a desired orientation (Eq. (5)), also known as points in space, at a desired time (t_d).

$$\omega(t_0) = \omega_0 \tag{2}$$
$$q(t_0) = q_0 \tag{3}$$
$$\omega(t_d) = \omega_d \tag{4}$$
$$q(t_d) = q_d \tag{5}$$

The same objectives can be stated in the target frame of reference by defining error states and setting them to zero, as depicted in Eq. (6) and Eq. (7). The transformation to the target frame of reference allows the solution of the attitude control problem from different states to the origin, and utilize the solutions for a family of problems that can be translated to the same initial states in the target space, quantified in Eq. (8) and Eq. (9). This change in reference frame reduces the search space considerably for the RL algorithm.

$$\omega_e(t_d) = \omega(t_d) - \omega_d = 0 \tag{6}$$
$$q_e(t_d) = q(t_d)q_d^* = [0, 0, 0, 1]^T \tag{7}$$
$$\omega_e(t_0) = \omega_0 - \omega_d \tag{8}$$
$$q_e(t_0) = q_0 q_d^* \tag{9}$$

For RL, the attitude control problem needs to be formulated as an unconstrained optimization problem. A simple way of accomplishing this is to enforce the constraints via penalties in the objective function [12]. In addition to including constraint penalties in the objective function, it is often desirable to include a control effort term in the objective. With this background, the following framework can be established:

$$r(s_t, a_t) = -\alpha_q q_{\text{er}} - \alpha_\omega ||\omega_e||_2 - a_t - c \tag{10}$$
$$q_{\text{err}} = |q_e(t) \cdot [0, 0, 0, 1]^T| - 1, \tag{11}$$

where α_q and α_ω are weights to tune the system response, and c is the conditional reward to include realistic constraints (Eq. (12). The magnitude for c ranges from 0–10^4; c is positive if the attitude and velocity are close to the desired targets, biasing the algorithm toward the targets. c is a large negative reward anytime the environment is reset due to poor agent performance (e.g., exceeding the maximum tumble rate for a satellite, or pointing 180° away from the target). The reason for the large negative reward and reset for slew and attitude is to bound the search space.

$$c = \begin{Bmatrix} 200 : \text{if } \mathrm{q_{err}} \leq \mathrm{q_\epsilon} \\ 1000 : \text{if } \mathrm{q_{err}} \leq \mathrm{q_\epsilon} \text{ and } ||\omega_\mathrm{e}||_2 \leq \omega_\epsilon \\ -10^3 : \text{if } \mathrm{q_{err}} \geq \mathrm{q_l} \text{ or } ||\omega_\mathrm{e}||_2 \leq \omega_\mathrm{l} \\ -10^4 : \text{if } \mathrm{q_{err}} \geq 2\mathrm{q_l} \text{ or } ||\omega_\mathrm{e}||_2 \leq 2\omega_\mathrm{l} \\ -10^3 : \text{if reaction wheels saturated} \\ 0 : \text{otherwise.} \end{Bmatrix} \tag{12}$$

Since the best reward per step is 1200 we also define a measure of attitude performance, which can be intepreted how close the reward per step is to 1200, defined in Eq. 13

$$\text{performance} = \frac{1200}{(1 - r_{\text{avrage}})}.100 \tag{13}$$

where, r_{avrage} is the average reward per step obtained. In all test cases, the best performance achievable is 100, with a higher number indicating a better performance.

Due to the inter-dependence of the angular velocity and the attitude of a rigid body, the RL algorithm will have a difficult time discovering the solutions to the full attitude control problem. To mitigate this, a curriculum learning-based method is utilized. The environment starts with initial conditions close to the target states, and increases in difficulty as the agent learns the simpler problem. The difficulty of the problem is controlled by a variable termed "*hardness*" here. Hardness takes values between 0 to 1, where 1 is the requirements for a realistic system, and zero is the easiest version of the problem. In this study, a hardness of 0 indicates that the satellite is in the target state, and so the optimal action is to no torque.

In addition to the hardness variable to control the difficulty of the problem, the ACS in this test is given n time steps during a roll-out to achieve the target state, but if n steps were not sufficient to achieve the target state, the next rollout of n steps begins

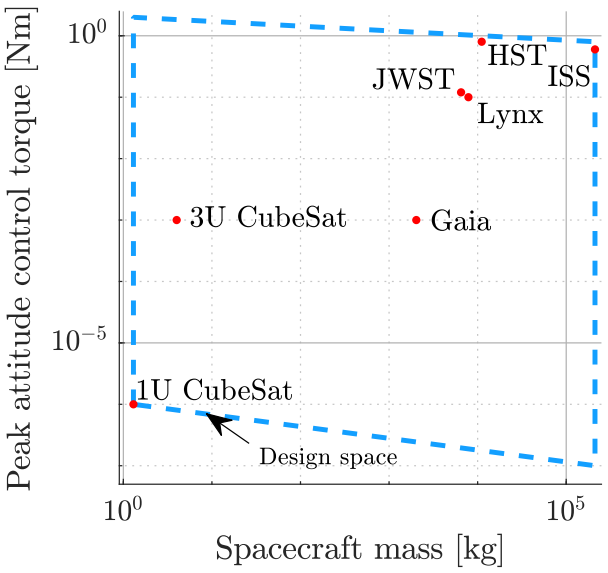

Fig. 2: Attitude control capability against spacecraft mass properties for various mission types

Spacecraft bus	Mass(kg)	Side length (m)
CubeSats	1.3-20	0.1-0.3
Microsatellites	20-200	0.5-1
Communication satellites	1000-5000	2-5
Deep space bus	1000-5000	5-20
Space observatory	10,000-20,000	4-15
Space station	100,000	20-100

Table 1: Physical properties of different pacecraft buses.

with the same states that the agent achieved in the previous roll-out.

3. Case Studies

One of the key objectives of this study is to obtain an attitude control agent that can be deployed to a broad family of spacecraft, irrespective of the actuator capability and satellite mass and moment of inertia. Such an attitude control method answers the problems faced by missions where the spacecraft capabilities change throughout a mission, such as the Asteroid Redirect Mission, Europa Clipper, etc. Additionally, a controller that can perform well across a wide variety of designs can then be used to solve optimal control co-design problems [13, 14]. To obtain such an agent, the spacecraft mass and attitude control authority are changed when each reset function is called. Obtaining a general attitude control agent allows using the same agent across multiple missions, which increases the reliability of the control algorithm. Figure 2 shows the range of different properties exhibited by different classes of satellite missions. The spacecraft properties for the RL agent is randomly chosen within the spacecraft design space enclosed by the convex hull indicated in Fig. 2. The blue region indicates the mass and peak attitude control actuator torques for ACS that have flight heritage [15]. Points within the region show examples of missions with vastly different requirements and capabilities [11, 16–22].

Hyper parameter	Value
Steps	600
dt	30 secs
Iterations	2000
Roll-outs	512
Epochs	256
Mini-batch	512
Layers	4 (Fully connected)
Neurons per layer	7, 4, 4, 7

Table 2: Hyper-parameter for tuning RL based satellite attitude control methods

To initialize random physical properties for the spacecraft, a scale integer is first randomly chosen. This integer determines if the physical properties are within the regime of nanosatellites, microsatellites, commercial satellite, or heavy satellite buses, seen in Table 1. Once the scale integer is chosen, a physical dimension for the spacecraft central bus is chosen that is appropriate to the spacecraft class, and a mass is assigned.

The attitude control problem with changing mass and inertial properties is not an MDP, but is instead a Partially Observable Markov Decision Processes (POMDP), but RL algorithms have shown good performance with solving POMDP [23], and hence the problem formulation for the changing mass property case is the same as that for the constant satellite mass property.

The RL based satellite attitude control agent is tested for two main cases:

1. Momentum management systems: ACS of satellites that utilize external torques, generated using thrusters or magnetic torque coils.

2. Momentum exchange systems: ACS of satellites

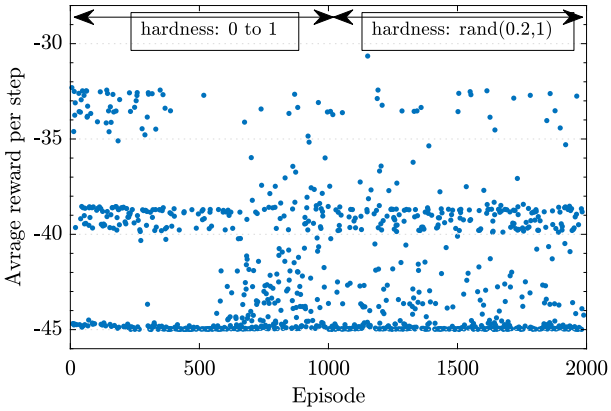

Fig. 3: Average reward using a tuned QRF controller for 180 kg satelllite (from Ball aerospace [25]).

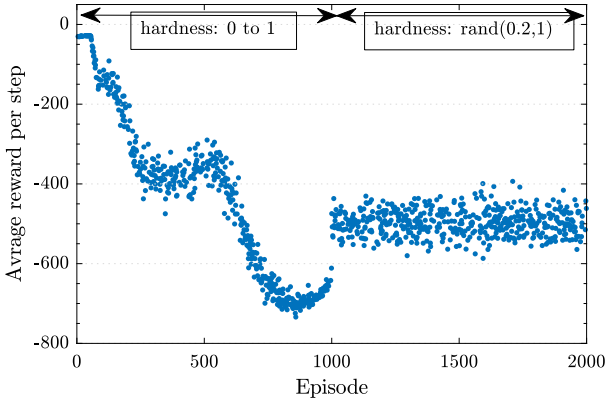

Fig. 4: Average reward using a tuned QRF controller for 180 kg satelllite (from Ball aerospace [25]) with Collin's Aerospace Reaction Wheel Assembly [26].

that use internal forces to change the attitude. Since no external torque is applied, such systems can only change the attitude and not the slew rate of a spacecraft for extended periods of time without the use of momentum management devices.

Both cases utilize a discrete-time system, with the control agent making decisions every 10 seconds. The hyper-parameters for each RL training are listed in Table 2.

4. Results and discussion

The system is simulated in the Mujoco physics engine [24]. The satellite is initialized as a rigid body. The satellite is connected to the world frame through a free joint, which is a joint with six degrees of freedom. The simulation environment has no gravitational, aerodynamic, or solar radiation pressure effects.

4.1 *QRF baseline*

As a baseline for comparison the average reward per step is presented for the Ball aerospace spacecraft bus [25] for the simple momentum management environment. The peak torque that can be applied for this simulation case is 10 mNm. The average reward for the QRF controller can be seen in Fig. 3; the data point near a hardness of 0 correspond to the spacecraft being at the desired target at the start of the simulation, hence the reward accrued is a large positive one. No other cases obtain a large positive reward, because the ACS uses torques to reach the target state, which results in negative rewards. The rewards for each case in Fig. 3 have been averaged for 512 roll-outs of the same hardness, to reduce the effect of random initial states. It can be seen that the average reward per step for the tuned QRF controller is between -45 and -30. The QRF controller for the higher fidelity environment utilizing reaction wheels from Collins aerospace, with the same satellite bus is presented in Fig. 4. The average reward per step is considerably lower than the simple control environment since the control algorithm saturates the reaction wheel while performing most of the trajectories. This is because saturating the reaction wheels and achieving the target attitude state is more optimal than not achieveing the target state.

4.2 *Momentum management based system*

The torques by the ACS are approximated by external torques acting on the rigid body, in the local (body) frame. Initial studies are performed with an ESPA ring class satellite bus by Ball aerospace [25]. The first 1000 episodes are simulated with a linearly increasing hardness variable, with episode 0 having a hardness of 0, and episode 999 a hardness of 1. Subsequent episodes are simulated with random hardness, chosen uniformly between 0.2 and 1. The reward obtained by the RL agents can be seen in Fig. 5 Each episode is simulated with a random satellite inertia and peak control torque capability, within the bounds shown in Fig.2. seen in the constantly varying reward received by the agent.

It can be seen from Fig. 6 that the simulation starts with an easy case, where the satellite is already at the target state. Here the agent learns quickly that the optimal action is to not produce any torques. As the hardness increasese, the optimial action is more

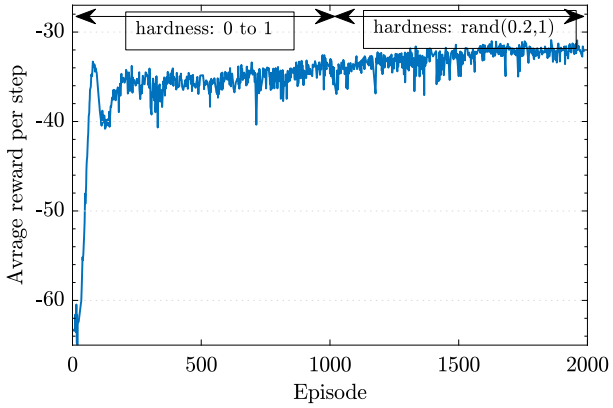

Fig. 5: Average rewards obtained by RL based attitude control agent (from Ball aerospace [25])

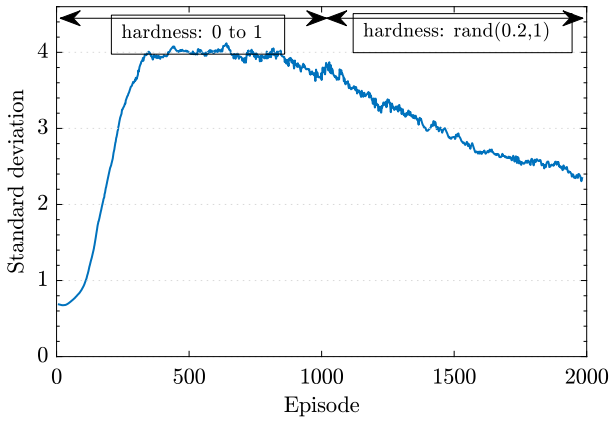

Fig. 6: Standard deviations of the probabilistic actions performed by the RL based attitude control agent

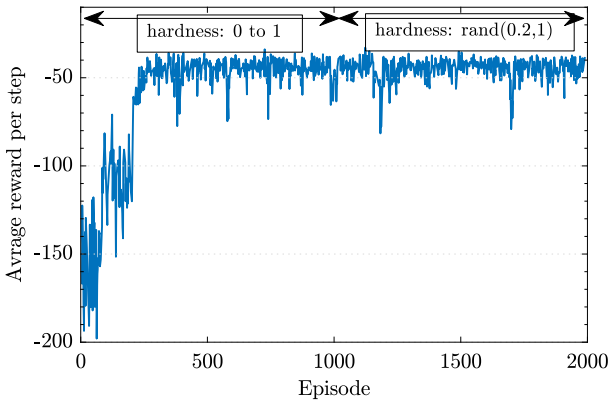

Fig. 7: Average rewards obtained by RL based attitude control agent for different spacecraft classes

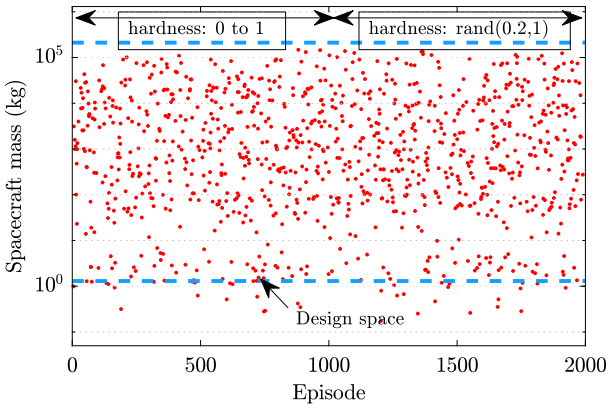

Fig. 8: Mass variation of spacecraft for the RL runs shown in Fig. 7

complicated, and the reward obtained can never be as high as that for the easier environment, as seen in Fig. 7. The results obtained for the changing mass attitude control agent is siilat to OpenAI's *Learning Dexterity* [27] study, where a robotic had was supposed to change the orientation of an arbitrary object t a desired pose, this is similar to changing the attitude of a spacecraft of different physical properties.

Figure 6 shows the standard deviation of the actions produced by the agent. A smaller standard deviation indicates that the agent is sure of the response for the given state. Once the agent has learned the response for the maximum hardness case, a decrease in the standard deviation can be seen. This indicates that the agent is more certain of the action to be taken to maximize the reward.

The next result is for a varying satellite mass, and peak attitude control torque. A random value is chosen from the design space depicted in Fig. 2.

Figure 8 shows the variety of masses simulated for the RL training run, the spacecraft properties were randomly sampled from the design space defined in Fig. 2. It can be see that the RL ACS agent achieved similar performance to the best QRF controllers without the need for explicit re-tuning for each spacecraft. The RL agent training results are seen in Fig. 7. The standard deviation of the probabilistic actions taken by the RL agent for the varying satellite case can be seen in Fig. 9

4.3 *Momentum exchange based systems*

To simulate momentum exchange based ACS, the Mujoco environment was modified to simulate a rigid-body, the satellite bus, with 3 rotating disks of certain inertias connected to the center body using a

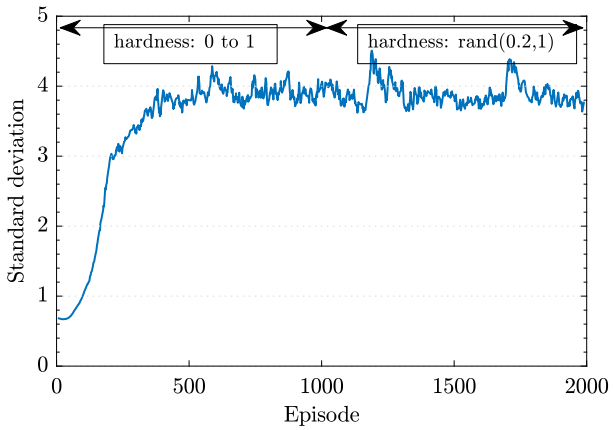

Fig. 9: Standard deviations of the probabilistic actions performed by the RL based attitude control agent

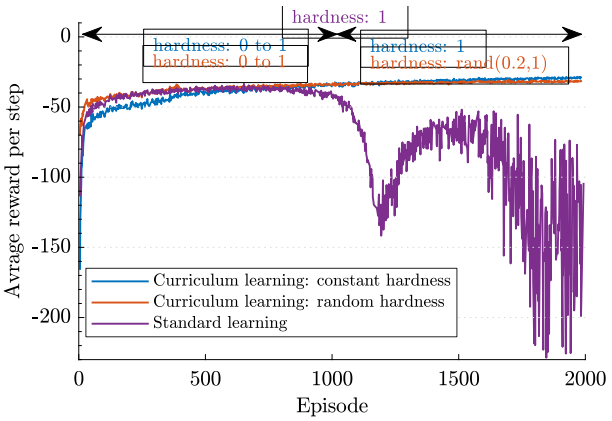

Fig. 11: Average rewards obtained by RL based attitude control agent for Reaction Wheel Assembly based ACS.

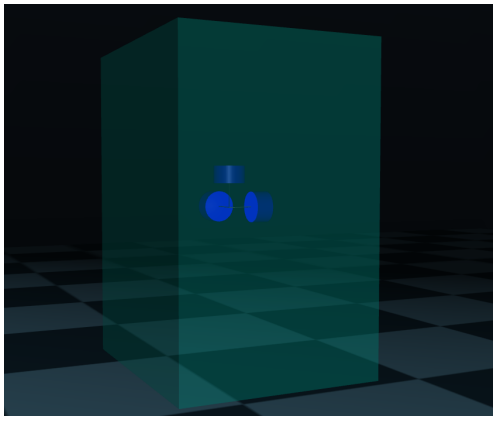

Fig. 10: Mujoco environment rendering of the satellite with a 3-axis reaction wheel assembly (blue).

single degree of freedom joint, seen in Fig. 10. This setup closely simulates a reaction wheel system. Mujoco allows modeling actuation of revolute joints using motors and joints with friction, the reaction wheel actuation was modeled using such motors. The satellite bus for the initial studies was the same Ball Aerospace micro-satellite bus with Collin's Aerospace Reaction Wheel Assembly [26].

All simulation parameters were kept identical from the simple actuator test, except the torque production numbers and maximum momentum storage capabilities. These numbers were assigned according to the data-sheet for Collins RWAs [26]. Additionally, the speeds of each reaction wheel was provided to the control agent, this was to make sure that the problem represents a MDP.

Figure 11 shows the average reward obtained by the RL attitude control agent for all the 2000 training episodes. Three runs are shown, *blue and orange lines* show rewards per step for agents trained in the curriculum learning setup, while *magenta line* run shows the reward per step for a standard RL setup. It is seen that for this problem the rate of learning initially is similar to that from curriculum learning, but the algorithm seems to have significant issues midway through the training and the average rewards drop significantly. The other two curves show training under curriculum learning setup, both agents learn the optimal policy quickly, and have a consistent increasing average reward per step, showing that the learnt policy is stable. The orange line learns slightly slower then the blue line, this is probably because the *hardness* of the environment keeps changing randomly post curriculum (non-ergodic environment), and hence the optimal policy keeps changing, and learning a changing policy is generally harder. The blue curve has a constant *hardness* of one after episode 999 and this makes the environment ergodic for the rest of the episodes.

Figure 12 shows standard deviation of the control agents for the reaction wheel ACS case obtained from the three different training regimes. Three runs are shown, *blue and orange lines* show standard deviation for agents trained in the curriculum learning setup, while *magenta line* run shows the standard deviation for the agent for a standard RL setup. Both the agents that were trained using curriculum have a low standard deviation which is constantly decreasing. A decreasing standard deviation is an indicator that the optimal policy has been discovered and the agent now is sure of the action to make for a given state. The

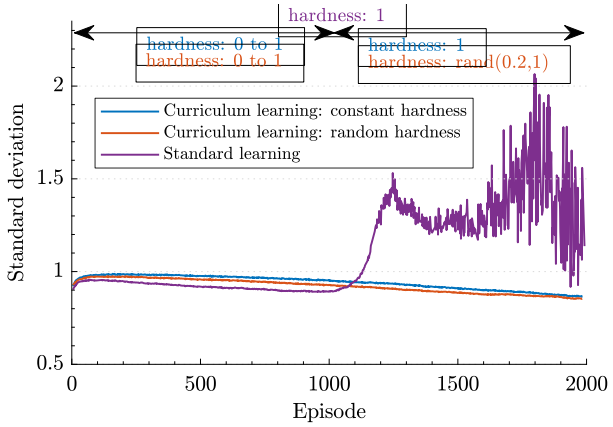

Fig. 12: Standard deviations of the probabilistic actions performed by the RL based attitude control agent for Reaction Wheel Assembly based ACS.

agent trained using standard, non-curriculum based methods, shows a sharp increase in the standard deviation once it encounters the run with a significantly lower average reward, indicating that the agent needs to explore more to converge to an optimal policy.

5. Conclusion

A general RL based attitude control agent was trained and presented. The training utilizes a curriculum learning based approach since the attitude control problem is nonlinear. Because of this nonlinearity, using conventional RL techniques to achieve target states would be infeasible.

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 as seen in Fig. 13.

The performance of the RL based attitude control is similar to QRF controllers that have been hand tuned for each mass case, seen in Fig. 13. 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 0.1m 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 (Eq. 13), a lead of 25 over the tuned QRF controller with a performance metric of 72, as seen in Fig. 14.

Such controllers and rapid learning-based tech-

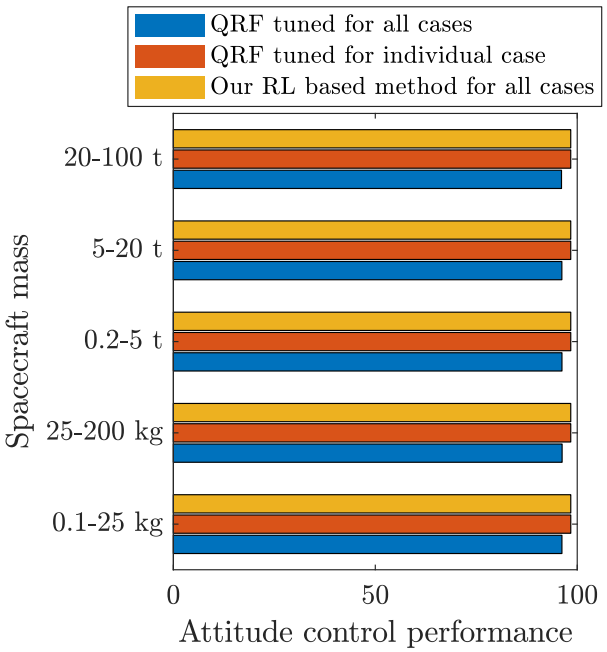

Fig. 13: Performance of our RL based attitude control agent vs. QRF controllers.

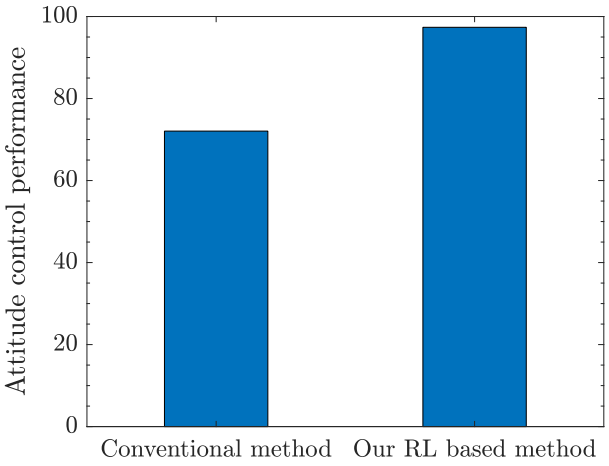

Fig. 14: Performance of RL agent vs tuned QRF controller for a 200 kg scale satellite, RL based method has a 25% improvement over the QRF controller.

niques 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.

6. Acknowledgements

This material is based upon work partially supported by the National Science Foundation under Grant No. CMMI-1653118.

References

[1] A. Deprit, "Free rotation of a rigid body studied in the phase plane," *American Journal of Physics*, vol. 35, no. 5, pp. 424–428, 1967.

[2] O. Bogoyavlensky, "Euler equations on finite dimensional lie algebras arising in physical problems," *Communications in Mathematical Physics*, vol. 95, no. 3, pp. 307–315, 1984.

[3] Vedant and A. R. M. Ghosh, "Dynamic programming based attitude trajectories for underactuated control systems," in *41st Annual AAS Rocky Mountain Section Guidance and Control Conference, 2018*, pp. 191–202, Univelt Inc., 2018.

[4] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[5] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2017.

[6] C. White, *Markov decision processes*. Springer, 2001.

[7] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[9] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[11] collin, E. Kroeker, and A. R. M. Ghosh, "Optimal attitude determination and control system design for laice and cubesail," in *27th AAS/AIAA Space Flight Mechanics Meeting, 2017*, pp. 3021–3033, Univelt Inc., 2017.

[12] G. Di Pillo and L. Grippo, "Exact penalty functions in constrained optimization," *SIAM Journal on control and optimization*, vol. 27, no. 6, pp. 1333–1360, 1989.

[13] D. R. Herber and J. T. Allison, "Nested and Simultaneous Solution Strategies for General Combined Plant and Control Design Problem," *ASME Journal of Mechanical Design*, vol. 141, p. 011402, Jan. 2019. 10.1115/1.4040705.

[14] H. K. Fathy, J. A. Reyer, P. Y. Papalambros, and A. Ulsov, "On the coupling between the plant and controller optimization problems," in *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, vol. 3, pp. 1864–1869, IEEE, 2001. 10.1109/ACC.2001.946008.

[15] S. R. Starin and J. Eterno, "Attitude determination and control systems," 2011.

[16] G. Matticari, G. Noci, P. Siciliano, G. Colangelo, and R. Schmidt, "Cold gas micro propulsion prototype for very fine spacecraft attitude/position control," in *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 4872, 2006.

[17] G. Beals, R. Crum, H. Dougherty, D. Hegel, and J. Kelley, "Hubble space telescope precision pointing control system," *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 2, pp. 119–123, 1988.

[18] L. Meza, F. Tung, S. Anandakrishnan, V. Spector, and T. Hyde, "Line of sight stabilization of james webb space telescope," 2005.

[19] N. Bedrossian, S. Bhatt, M. Lammers, and L. Nguyen, "Zero-propellant maneuver [tm] flight results for 180 deg iss rotation," 2007.

[20] A. Siahpush and J. Gleave, "A brief survey of attitude control systems for small satellites using momentum concepts," 1988.

[21] E. Silani and M. Lovera, "Magnetic spacecraft attitude control: a survey and some new results," *Control Engineering Practice*, vol. 13, no. 3, pp. 357–371, 2005.

[22] J. A. Gaskin, D. Swartz, A. A. Vikhlinin, F. Özel, K. E. Gelmis, J. W. Arenberg, S. R. Bandler, M. W. Bautz, M. M. Civitani, A. Dominguez, *et al.*, "Lynx x-ray observatory: an overview," *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 5, no. 2, p. 021001, 2019.

[23] T. Jaakkola, S. P. Singh, and M. I. Jordan, "Reinforcement learning algorithm for partially observable markov decision problems," in *Advances in neural information processing systems*, pp. 345–352, 1995.

[24] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, Oct 2012.

[25] "Ball aerospace: Bcp small sats." `https://www.ball.com/aerospace/markets-capabilities/capabilities/spacecraft-space-science/satellites`. Accessed: 2019-08-28.

[26] "Colllins aerospace: Reaction wheels specsheet." `https://www.rockwellcollins.com/Products-and-Services/Defense/Platforms/Space/RSI-12-Momentum-and-Reaction-Wheels.aspx`. Accessed: 2019-08-28.

[27] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *arXiv preprint arXiv:1808.00177*, 2018.