RL for Scientific Discovery: Examining Satellite Decision Making in the Orbit of Enceladus

Vanessa Bellotti and Tanay Nistala

1 Introduction

In the pursuit of advancing autonomous space exploration, this project endeavors to develop an innovative reinforcement learning model tailored for online learning within the confines of a simulated environment. The specific focus is on the detection of plumes, such as those found on Enceladus or analogous scientific phenomena on other planetary bodies. The model's multifaceted objectives encompass not only the identification of these plumes but also the dynamic guidance of a spacecraft towards their source for comprehensive analysis.

To meet these objectives, the reinforcement learning model must exhibit a capacity for continuous adaptation and learning from its interactions with the environment. In practical terms, this implies that the model should be capable of responding intelligently to unexpected scenarios, such as encountering new obstacles like ice geysers rather than water geysers. The model's ability to update its policy based on these novel situations ensures a robust and adaptive approach to navigating the spacecraft toward scientifically significant phenomena.

Moreover, the learning process extends to addressing potential challenges tied to sensor readings, including false positives or negatives triggered by extremes in surface temperature variations or areas of high radiation. The model's resilience is critical in not only refining its detection capabilities but also in safeguarding the spacecraft from potential damage. In essence, this project seeks to pioneer a reinforcement learning model that not only contributes to the scientific exploration of planetary bodies but also exemplifies a responsive and adaptable intelligence in the face of diverse and dynamic extraterrestrial environments.

2 Background Related Work

Dario Izzo et al. [4] explore optimality principles in spacecraft neural guidance and control, addressing fundamental challenges in autonomous spacecraft navigation. Their work delves into neural-guided decision-making processes, aiming to enhance the efficiency and adaptability of spacecraft systems. They discuss the transfer of burden and how these neural approaches in a sense alleviate it so onboard resources can focus on that which is vital to the mission and spacecraft maintenance systems.

Duncan Eddy and Mykel Kochenderfer [2] present a comprehensive approach using Markov Decision Processes (MDPs) for multi-objective satellite task planning. Their work focuses on the integration of MDPs to optimize decision-making processes in satellite missions, highlighting the significance of principled planning methodologies. They discuss the trade-off between timeliness and optimality and how critical it is to strike a balance between the two in spacecraft missions. This helped inform our decision-making with regards to the tradeoff and how we could account for it in our training and reward function.

Ashutosh Pandey et al. [5] propose a hybrid planning approach for decision-making in self-adaptive systems. By combining planning techniques, their work addresses challenges related to adaptability and decision-making under uncertainty, offering insights applicable to autonomous systems in dynamic environments.

Thom Badings et al. [1] present a nuanced perspective on decision-making under uncertainty, extending beyond traditional probabilistic models. Their work explores alternative approaches to decision-making, offering valuable insights into handling uncertainty in autonomous systems

3 Technical Approach / Methodology / Theoretical Framework

3.1 RL Environment

The reinforcement learning environment was set up using the Gym Python package to facilitate the setup of state and action spaces and the reward function, as well as to ease debugging. The state space was set up to include the current location of the agent as well as a concentration map that signifies the concentration of plumes around the agent. This produced a continuous state space, in contrast to the action space, which was set up as a discrete space of eight actions, corresponding to the cardinal and ordinal directions.

3.2 Simulation Environment

To accurately model the physics dynamics essential for simulating the satellite scientific discovery problem, we leveraged the PyBullet library. PyBullet provided a robust framework for simulating the physical interactions and dynamics within the simulated environment, enabling a realistic representation of the challenges associated with satellite navigation and scientific exploration, in this context around Encladus. The use of PyBullet facilitated the incorporation of gravitational forces, aerodynamic effects, and other physical phenomena integral to the accurate portrayal of a satellite's behavior in space. By harnessing the capabilities of PyBullet, we aimed to create a physics simulation environment that closely emulates the complexities of real-world satellite missions, thereby providing a foundation for training and evaluating reinforcement learning models tailored to the demands of scientific discovery in space exploration. Due to the challenges discussed in the relevant section of the paper, we leveraged related work to aid in our environment setup [6]

3.3 Reward Function

In the context of advancing autonomous space exploration and developing a reinforcement learning model for online learning in a simulated environment with a focus on plume detection, the reward function plays a crucial role in shaping the behavior of the spacecraft. The objective is not only to identify plumes but also to dynamically guide the spacecraft toward their source for comprehensive analysis. The reward function aims to capture the essence of intelligent adaptation to unexpected scenarios and challenges associated with sensor readings. We utilized a spare reard function [7] aimed at applying small negative rewards in most instances and large positive rewards when the goal state is reached.

Objective: The primary objective of the reward function is to incentivize the spacecraft to efficiently detect plumes and navigate towards their source for comprehensive analysis. The reward signal is crafted to encourage adaptive learning in the face of unexpected scenarios and challenges associated with sensor readings.

3.3.1 Components of the Reward Function:

Proximity to Plume Source: The reward is influenced by the spacecraft's distance to the nearest plume source. Closer proximity to an unvisited plume results in a positive reward. The reward magnitude increases with proximity, fostering efficient plume detection.

Exploration and Avoidance: The model is incentivized to explore new plumes intelligently. Upon discovering a plume with a concentration above a threshold (e.g., 0.8), the spacecraft is rewarded positively. However, revisiting a previously explored plume incurs a penalty to discourage redundant exploration.

3.4 Proximal Policy Optimization

The application of Proximal Policy Optimization (PPO) [3] to the spacecraft navigation and plume detection problem is motivated by its inherent advantages in stability, sample efficiency, and suitability for continuous control tasks. In the pursuit of advancing autonomous space exploration, where the spacecraft must intelligently navigate and detect plumes within a simulated environment, the stability of PPO becomes particularly advantageous. PPO addresses issues related to policy optimization instability, mitigating the risk of large policy updates that could lead to divergent training. This stability is crucial for ensuring the reliable learning of a spacecraft control policy.

Additionally, the sample efficiency of PPO is a significant asset, especially in scenarios where data collection or simulation is resource-intensive. Given the computational expense associated with training reinforcement learning models for spacecraft navigation, PPO's ability to achieve good performance with fewer samples is valuable. Furthermore, the suitability of PPO for continuous action spaces aligns seamlessly with the nature of spacecraft control, where actions involve continuous variables such as thrust and torque.

The adaptive learning rates employed by PPO contribute to its effectiveness in handling varying dynamics in the environment. This adaptability ensures that the model can adjust its step sizes during training, enhancing resilience to changes in the simulated environment. As the spacecraft navigation problem inherently involves a trade-off between exploration and exploitation, PPO's design allows it to balance these aspects effectively. This is crucial for the spacecraft to intelligently explore the environment, discover new plumes, and subsequently transition to exploitation for optimal navigation toward known plumes.

Moreover, PPO's compatibility with simulated environments, a common practice in training reinforcement learning models, adds to its suitability for the spacecraft navigation problem. Simulated environments offer a controlled setting for training where real-world experimentation may be impractical. PPO's incremental updates and adaptability make it well-suited for online learning, a key focus of this project where the model must continuously adapt to new information and scenarios.

In conclusion, the selection of PPO for the spacecraft navigation and plume detection problem is grounded in its stability, sample efficiency, adaptability to continuous action spaces, and compatibility with simulated environments. These characteristics collectively contribute to the efficacy of PPO in addressing the challenges posed by this complex and dynamic space exploration task.

4 Challenges

We encountered many challenges throughout this course of this phase of our project, from the environment setup to oscillations of the drone within the environment to difficulties detecting past the first plume to inconsistencies in performance.

The first challenge was setting up the PyBullet and Gym setup, especially the first of the two softwares. Specifically, when we placed a satellite into the simple environment we created, it would not maintain orbital speed and would crash into the planet we had mocked in the environment. This was a large issue in that it prevented us from setting up the scenario we intended to train on, and there was seemingly no explanation in the open documentation. In order to resolve this, the only path forward seemed to involve coding in all of the physics logic that the physics simulation should have inherently. In order to circumvent this, we had to utilize a drone based environment [6] and refactor it for our purposes. Mark Moussa was very helpful in our efforts around this. Thus far, in order to try and mitigate other difficulties we are experiencing, we have been trying to experiment with the Gaussian, and particularly the alpha, of the plume concentration to make it better suited for our purposes.

Our drone was also experiencing wide oscillations in the environment during training episodes. This posed a challenge as it was not immediately obvious if the issue was due to the reward function and training or due to the simulation itself. The current belief is that it is due to the simulation, and specifically due to the ration between the PyBullet frequency and the control frequency. This is currently being modified still in order to ensure better results on our agent learning. Furthermore, we tried states that only had one plume (and generated another on reset by time out or discovery) as well as multi-plume environments, and the agent would get confused when multiple plumes existed at once as to which one it should pursue. This issue was amplified by the oscillations because its sense of plume concentration would vary widely between each swing it performed movement wise, thus meaning it did not have enough time to actually learn where to go.

Our issue with the agent only detecting the first plume and not finding subsequent plumes was interesting because it seemed to have a "phantom path" that it followed, every episode gravitating back to the coordinates that used to contain a plume but no longer did.

Given that the plume locations are randomly generated, the performance of the agent can vary largely between episodes and between training runs. For instance, on one episode the plumes were generated very close to the origin and the agent performed very well, but in the next episode, the agent never reached a single plume despite dozens being generated.

5 Evaluation

The effectiveness of the developed reinforcement learning model, trained using the Proximal Policy Optimization (PPO) algorithm, is assessed through a comprehensive evaluation process. The objective is to gauge the model's performance, its ability to generalize to diverse scenarios, and its adaptability to unforeseen challenges. The evaluation metrics encompass both quantitative measures and qualitative assessments to provide a holistic view of the model's capabilities.

5.1 Quantitative Metrics:

Average Reward: Measure the average reward achieved by the spacecraft over multiple episodes. A higher average reward indicates successful learning and effective navigation toward plume sources.

Exploration vs. Exploitation: Analyze the balance between exploration and exploitation. Evaluate whether the model efficiently explores the environment to discover new plumes initially and transitions to exploitation for optimal navigation.

Adaptability to Challenges: Introduce specific challenges in the environment, such as new obstacles or extreme sensor readings. Assess the model's adaptability by monitoring changes in its behavior and reward accumulation in response to these challenges.

Training Stability: Examine the stability of the training process. Ensure that the model converges to a stable policy and does not exhibit erratic behavior during training.

5.2 Qualitative Assessments:

Visual Inspection: Visualize the trajectories of the spacecraft over episodes. Observe whether the model exhibits intelligent exploration, successfully identifies plumes, and navigates toward their sources.

Adaptive Policy: Evaluate the adaptability of the learned policy. Introduce novel scenarios or tweak environmental parameters to assess whether the model updates its policy intelligently and in a manner aligned with scientific objectives.

Handling Unseen Scenarios: Introduce scenarios not encountered during training to assess the model's generalization capabilities. Evaluate whether the spacecraft can handle new plume configurations or environmental conditions.

5.3 Safety and Robustness:

Avoidance of Damage: Assess the model's ability to avoid damage by monitoring its response to extreme environmental conditions or unexpected obstacles. A resilient model should prioritize safety while pursuing scientific objectives.

Collision Avoidance: Evaluate the effectiveness of the model in avoiding collisions. Ensure that the spacecraft navigates through the environment without unnecessary collisions, indicating a well-trained and robust policy.

Data Collection Efficiency: Assess the efficiency of data collection. Evaluate whether the spacecraft optimally allocates resources to explore regions with higher scientific significance. Future work could implement a consideration of the spacecraft's fuel economy, especially since this kind of mission would not allow for refueling.

References

[1] Thom Badings, Thiago D. Simão, Marnix Suilen, and Nils Jansen. Decision-making under uncertainty: Beyond probabilities, 2023.

[2] Duncan Eddy and Mykel Kochenderfer. Markov decision processes for multi-objective satellite task planning. In *2020 IEEE Aerospace Conference*, pages 1–12, 2020.

[3] John Schulman et al. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[4] Dario Izzo, Emmanuel Blazquez, Robin Ferede, Sebastien Origer, Christophe De Wagter, and Guido C. H. E. de Croon. Optimality principles in spacecraft neural guidance and control, 2023.

[5] Ashutosh Pandey, Gabriel A. Moreno, Javier Cámara, and David Garlan. Hybrid planning for decision making in self-adaptive systems. In *2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pages 130–139, 2016.

[6] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P. Schoellig. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7512–7519, 2021.

[7] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.