

# ASEN 6519 Final Project REPORT

Kian Shakerin

TOTAL POINTS

(not graded)

QUESTION 1

1 Final Project Report

- 0 pts Correct

# Asteroid Exploration with Deep Reinforcement Learning

Kian Shakerin

*Ann & H.J. Smead Department of Aerospace Engineering Sciences  
University of Colorado Boulder  
Boulder, USA  
kian.shakerin@colorado.edu*

December 15, 2023

## Abstract

In this project we train a hopping agent to explore an asteroid with an unseen dynamical environment using Deep Reinforcement Learning. The agent actively explores the asteroid surface that has various reward states, representing areas of scientific interest, with the goal of reaching the terminal reward state, representing a final science site. The agent can select from a fixed action space to move itself through the environment. A spherical harmonic gravity model is used for the dynamics, and the coefficients are randomly varied at the start of each training episode. The agent starting location is also varied to simulate an uncertain landing site. Proximal Policy Optimization is used to train the agent. We train the agent on two cases which differ in the placement of the reward states to analyze the impact of reward shaping on policy performance. When the reward states are close to each other and the starting area, a near optimal policy is found for exploring the asteroid. When the reward states are more sparsely distributed about the asteroid, the policy converges to a local maximum. In both cases the agent successfully finds the terminal state in almost every episode. Furthermore we demonstrate that the learned policies are able to robustly and autonomously navigate the agent in unseen dynamical environments.

## 1 Introduction

Small celestial bodies, such as asteroids, comets, and moons, are of particular interest as scientific targets. They contain unique information about our solar system's evolution and may provide clues in the search for life beyond Earth [1]. These objects often have unique characteristics such as irregular spin poles, asymmetrical gravitational fields, and ejecta. Space operations about and on small bodies often take place in complex, varied, and evolving environments. These missions traditionally require significant prior knowledge of the target environment, face large operational burdens, and are inflexible to changing aims [2]. Pushing the autonomous capabilities of spacecraft is pivotal to enabling the next generation of space missions.

One area of interest for exploration is asteroid surfaces. Surface exploration would provide highly detailed information on things such as regolith composition and dynamics close to the body. Recent missions such as Hayabusa, Hayabusa 2, and OSIRIS-REs, have proved pivotal in our understanding of small bodies and pioneered methods for sample recovery and data gathering [3, 4, 5]. In particular, Hayabusa 2 deployed two small hopping launders to the surface of Ryugu with the goal of mapping the asteroids surface [4]. The concept of a hopping rover or lander was studied prior to Hayabusa 2 and has remained a popular topic for asteroid surface mobility research [6].

This mission concept still suffers from the issues listed previously, namely operating autonomously and uncertainties in the environmental dynamics on or near the asteroid surface. One approach to solving these problems is through Deep Reinforcement Learning (RL). This approach has been popular in the robotics community for some time for application in problems such as self driving cars and quadruped robots, and has gained popularity in aerospace research more recently. In this paper we present a simulated hopping rover trained via Deep RL to autonomously explore an asteroid with previously unseen environmental parameters.

## 2 Background

The application of Deep RL techniques to aerospace problems has gained popularity in recent years. Research in this area has shown interesting results and some promise in both enabling autonomy and developing systems that are capable of adjusting in real time to evolving dynamical systems environments. In a 2020 paper Guadet, Linares, and Furfaro present an application of Meta Reinforcement Learning to the descent and landing problem on small bodies [2]. Their aim is develop a guidance policy that can safely land the agent within specified tolerances. They train agents in several landing scenarios over a wide range of environmental parameters using Proximal Policy Optimization (PPO) modified with a recurrent layer in each of the networks [7]. Their results are mixed and there is no comparison to current state of the art guidance laws so it is difficult to draw conclusions. However, the work presented is interesting and provides an exploration of this relatively new area of research.

Another work that combines Deep RL with small body exploration is the 2020 paper by Jiang, Zeng, Guzzetti, and You [8]. In this paper the authors investigate path planning for a hopping rover trained with Deep Q Learning. In their approach they constrain the rover to a square grid with an irregular surface and learn an optimal path across the terrain. The further explore improvements on DQN such as DDQN, Dueling Networks, and Prioritized Experience Replay to enhance their results. The authors demonstrate that they can successfully obtain optimal paths in new perturbed terrains. Again, this work shows interesting results and promising paths forward for future research.

## 3 Approach

### 3.1 Physical Environment and Dynamics

To simulate a semi-realistic asteroid environment, an approximate analog of the asteroid Bennu was developed. To simplify the simulation, I approximated Bennu's shape as a sphere. In the course of development a 2-dimensional model was first constructed and tested before expanding the environment to 3-dimensions. The physical parameters for Bennu used are listed in table 1.

Radius (km)	Gravitational Parameter ( $\mu$ ) ( $km^3 s^{-2}$ )
0.24503	$4.8904 * 10^{-9}$

Table 1: Bennu physical parameters used.

For the dynamics of this project, a spherical harmonics gravity model was implemented. This gravitational model is a good low-order approximation, considering the relatively spherical shape of the asteroid Bennu, and is a widely used model in aerospace applications [5]. I used zonal and tesseral components up to C30 to provide a semi-realistic representation of the gravitational environment around Bennu. The formulas for the spherical harmonic potential are presented in eqns 2 and the acceleration is presented in eqn 2.

$$\begin{aligned}
 U^e &= \frac{GM^*}{R_e^*} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{R_e^*}{r} \right)^{n+1} P_{nm}(\sin(\phi)) \begin{bmatrix} \cos(m\lambda) \\ \sin(m\lambda) \end{bmatrix} \begin{bmatrix} C_{nm}^e \\ S_{nm}^e \end{bmatrix} \\
 C_{nm}^e &= \frac{2}{M^*} \frac{(n-m)!}{(n+m)!} \int_M \left( \frac{r'}{R_e^*} \right)^n P_{nm}(\sin(\phi')) \cos(m\lambda') dm' \\
 S_{nm}^e &= \frac{2}{M^*} \frac{(n-m)!}{(n+m)!} \int_M \left( \frac{r'}{R_e^*} \right)^n P_{nm}(\sin(\phi')) \sin(m\lambda') dm'
 \end{aligned} \tag{1}$$

$$\vec{a}_{grav} = \frac{\partial U^e}{\partial \vec{r}} \tag{2}$$

The unnormalized spherical harmonic coefficients can be found in table 2 and were taken from Chesley et al, 2020 [9]. It should be noted that presenting the normalized values is a common practice in small body

dynamics research and gravity modelling so that convention is adopted for this paper. Prior to use in the dynamics model, these coefficients were unnormalized using the scheme devised by Kaula [10], presented in eqns 4.

Spherical Harmonic Coefficients	Normalized Value ( $km^3 s^{-2}$ )
C20	1.926e-2
C21	0
C22	3.06e-3
C30	-1.22e-3
S21	0
S22	-1.09e-3

Table 2: Table of normalized Spherical Harmonic Constants for the asteroid Bennu up to C30 and S22.

$$\begin{aligned}
 C_{n,m} &= N(n, m) \bar{C}_{n,m} \\
 S_{n,m} &= N(n, m) \bar{S}_{n,m} \\
 N(n, m) &= \left( \frac{(n-m)!(2n+1)(2-\delta_{0m})}{(n+m)!} \right), \delta_{0m} = 1 \text{ if } m = 0
 \end{aligned} \tag{3}$$

I used a very basic model for the hopping rover to simplify the dynamics calculations and overall simulation complexity. The hopper is assumed to be a point mass that does not interact with the asteroid surface in any way. The hopper can choose from a set of hop actions (see section 3.2) that each have the same fixed velocity magnitude. The hop magnitude was selected after fixing the lander mass to provide a balance between hop size and trajectory integration time. Stated alternatively, I balanced the hop magnitude so that the lander would return to the surface in a reasonably short time to both speed up training and keep things semi-realistic. The hopper parameters are listed in table 3.

Hopper Mass (kg)	Hop Magnitude ( $km s^{-1}$ )
1.0	$6.0 * 10^{-5}$

Table 3: Hopper mass and jump magnitude.

### 3.2 Problem Definition

The problem is defined similarly to a Markov Decision process with a continuous state space, a discrete action space, a deterministic reward functions, and a generative transition function, though it does not strictly have the Markov property. There is no state or transition uncertainty. The state space is defined as the current location of the hopper at time  $t_k$  and the velocity vector of the hop at time  $t_{k-1}$ , which corresponds to the action  $a(t_{k-1})$  taken in the asteroid body frame. One can also think of the hop velocity as a control input. This is formally defined in eqn 4. The state space may take on any set of continuous positions on the asteroid's surface and any valid velocity defined by the hopper position and action set.

$$s(t_k) = [x(t_k) \quad y(t_k) \quad z(t_k) \quad \dot{x}(t_{k-1}) \quad \dot{y}(t_{k-1}) \quad \dot{z}(t_{k-1})]^T \tag{4}$$

The action space consists of 8 discrete hops that be taken. The hopper can move to the left, right, forward, and backward with two hop angles in each direction of  $30^\circ$  and  $70^\circ$ . The actions are listed in eqn 5. These actions are formally defined in the body frame of the agent, which I have selected to be the standard East-North Up (ENU) frame. The actions are defined in this frame and not the asteroid body frame to

maintain consistency.

$$a = \begin{cases} \text{Left: 70 deg} \\ \text{Left: 30 deg} \\ \text{Right: 70 deg} \\ \text{Right: 30 deg} \\ \text{Forward: 70 deg} \\ \text{Forward: 30 deg} \\ \text{Backward: 70 deg} \\ \text{Backward: 30 deg} \end{cases} \quad (5)$$

The transition function is deterministic and a function of the system dynamics. When an action is taken, the hop is integrated using the accelerations computed from the spherical harmonic gravity model until the lander intersects with the asteroid surface. This constitutes 1 step in the environment.

The reward function is a combination of negative rewards received for each hop taken and positive rewards for finding a reward state. The negative penalty for each hop is a linearly scaled function of the hop magnitude. Since the hop magnitude is fixed, the penalty is fixed at each step. The scale factor was experimented with during development and selected to balance with the possible positive rewards. There are 3 intermediate rewards states and one terminal reward state. Vectors are used to define the location of the rewards. A reward is gained when the angle between the hoppers position vector and that of the reward zone are within a certain threshold. This is designed to mimic an area of scientific interest on an asteroid in a real world mission. Once a reward is found in an episode, it cannot be found again, and the reward for that zone is set to zero. The reward function is defined in eqn 6, where  $r_i$  represents an intermediate reward state and  $r_f$  represents the terminal state.

$$R(s, a) = (-10^4 * |v|^2) + \begin{cases} 10 * i : \cos^{-1} \left( \frac{\mathbf{x} \cdot \mathbf{r}_i}{|\mathbf{x}| |\mathbf{r}_i|} \right) < 3 \text{ deg}, r_i \text{ for } i \in 1, 2, 3 \\ 100 : \cos^{-1} \left( \frac{\mathbf{x} \cdot \mathbf{r}_f}{|\mathbf{x}| |\mathbf{r}_f|} \right) < 5 \text{ deg} \\ 0 : \text{otherwise} \end{cases} \quad (6)$$

### 3.3 Training Setup

#### 3.3.1 Case Definitions

In this paper we present two separate cases in which we train the agent based on different reward locations.

1. (Case 1) Reward states are closely grouped and located relatively close to the hopper starting area.
2. (Case 2) Reward states are sparsely distributed about the asteroid.

The approximate reward locations in each case were selected intentionally to analyze the impact of reward shaping on policy performance, however the exact location of the rewards was chosen arbitrarily. The reward locations for each case are listed in table 4.

Reward	Case 1	Case 2
$r_1$	(0.3, 0.2, 0.9)	(0.3, -0.3, 0.7)
$r_2$	(0.3, 0.1, 0.7)	(0.5, 0.3, 0.4)
$r_3$	(0.4, 0.1, 0.5)	(0.7, -0.2, -0.2)
$r_f$	(0.6, 0.1, 0.4)	(1, 1, -0.5)

Table 4: Vectors defining the intermediate and terminal reward states for each case.

### 3.3.2 Training

To build the simulation environment and train the hopping agent, this project leverages existing RL and Deep RL frameworks that are publicly available. In particular, the Gymnasium API (CITE) was selected for defining the environment and Stable Baselines 3 was selected for its implementation of PPO (cite). These pieces of software are compatible and allow for extensive customization. A custom Gymnasium environment was developed based on the physical parameters, dynamics, and problem definitions previously discussed. A policy was then trained using the off-the-shelf implementation of PPO from Stable Baselines 3, directly interfacing with the custom Gymnasium environment. PPO was chosen for this project because it is a state-of-the-art Deep RL algorithm and for its use in prior aerospace research incorporating Deep RL [2]. Specifically, the Stable Baselines 3 implementation was used with default parameters.

The policy was trained for 100000 steps in the environment. One step in the environment is equivalent to one hop. The number of steps per episode was limited to 100, after which the episode would terminate and a new one would begin. The default batch size was 64 episodes. The policy was evaluated on 100 test episodes. The policies in both cases were evaluated at 50000 and 100000 steps. The policy in case 2 was further evaluated at 200000 steps. No fixed seed was used in the training or testing. No specific policy or heuristic was used to aid the agent in exploring the environment.

At the start of each episode the gravitational environment of the system was randomly varied. This allowed the agent to learn a policy that was robust to a range of uncertain dynamical parameters. At the start of each episode, the gravitational parameter was randomly sampled from a normal distribution with mean  $4.8904 \times 10^{-9} \text{ km}^3\text{s}^{-2}$  and sigma of  $\pm 10\%$ . Similarly, the spherical harmonic constants were randomly sampled from a normal distributions with means equal to the unnormalized value of those values listed in table 2, with sigmas of  $\pm 10\%$ . The starting location of the hopper was also varied to simulate an uncertain landing. The x and y positions of the lander were randomly sampled from a normal distributions with zero means and sigmas of 5 meters.

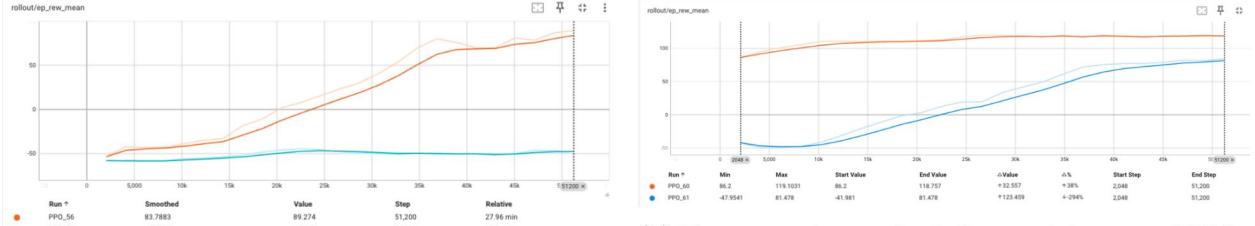
## 4 Results

The results of the policies for both cases, evaluated at 50000 and 100000 steps are shown in table 5.

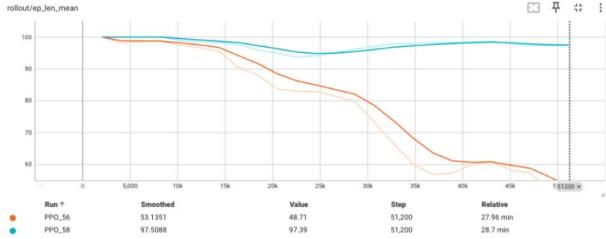
	Mean Reward (50000 steps)	Mean Reward (100000 steps)	Mean Reward (200000 steps)
Case 1	$5 \pm 34.86$	$121 \pm 19.56$	-
Case 2	$-58 \pm 6.69$	$81 \pm 9.6$	$93 \pm 7.19$

Table 5: Mean reward evaluated after 50000, 100000, and 200000 training steps. Rewards computed based on 100 test episodes with randomly selected gravitational parameters and lander starting position. Case 1 was not evaluated at 200000 training steps as it had already converged to a near optimal solution at 100000 training steps.

The policy in case 2 was trained for and evaluated at 200000 steps to see if it improved significantly beyond the 100000 step mark. After 50000 steps a good policy was learned in case 1 that was finding the reward states and the terminal state on a semi-regular basis. The policy in case 2, however, did not learn anything meaningful by this training checkpoint and was achieving a near worst-case reward. After 100000 training steps the policy in case 1 was achieving a near optimal score and reliably finding all reward states in a minimal number of hops in almost every episode. The policy in case 2 significantly improved between steps 50000 and 100000 and was achieving a relatively high mean reward of 81 at the second training checkpoint. However, it should be noted that this policy was not finding all the reward states, and instead taking a short path to the terminal state in almost every episode. I trained the policy in case 2 to see if it would improve and after an additional 100000 training steps it did improve the mean reward achieved but was still only finding the terminal state, a sub-optimal result.

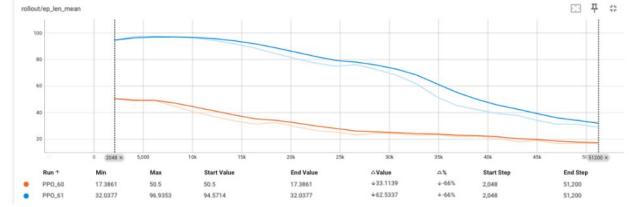


(a) Mean reward per episode from training steps 0 - 50000.



(c) Mean episode length (steps in the environment) from training steps 0 - 50000.

(b) Mean reward per episode from training steps 50000 - 100000.



(d) Mean episode length (steps in the environment) from training steps 50000 - 100000.

Figure 1: Mean episode length and Case 1 is shown in orange and case 2 is shown in blue.

The mean episode length and reward for both cases are presented in fig 1. Case 1 is shown in orange and case two is shown in blue. These plots contrast the learning rate and performance of the policies in each cases. They highlight the impacts that reward shaping, in this case through placement, can have on policy performance and success. For case 2 the policy performed poorly and consistently for almost half the training before it began to learn and perform better. In contrast, the policy in case 1 learned in a somewhat linear fashion before plateauing near the optimal solution.

## 5 Discussion

The results of the policies trained in case 1 and 2 present interesting behavior and highlight the impact of reward placement and reward shaping. With the rewards placed both closely to each other and relatively close to the lander starting area, PPO was able to locate the rewards and find a shortest path between them after a relatively small number of training steps. Intuitively this makes sense as the number of steps needed before finding a reward state are smaller the closer the reward is to the agent. With a random policy at the start of training, proximity matters.

The opposite of this trend was seen in case 2, when the rewards were sparsely distributed about the asteroid. Up to the first training checkpoint of 50000 steps, the policy in case 2 aimlessly explored the asteroid, occasionally finding a reward state, but for the most part it received a mean episode reward close to the lowest possible value. However, it was able to eventually find the terminal state and the policy learned to head directly towards it. This is not the global optimum but a local maxima that provided a fairly decent mean reward per episode of around 81. After training for another 100000 steps the policy in case 2 refined its movements to reach the terminal state faster but did not learn a new strategy to find any of the other states. I believe this is due to the fact that the rewards were simply spread too far apart and the algorithm was not incentivized to explore the asteroid surface for extra rewards once it had found the terminal state. This result further highlights the importance of reward shaping when Deep RL is applied to aerospace problems.

In both cases the trained policies successfully learned to autonomously explore in a range of gravitational environments. This is reflected in the mean rewards gained in testing on randomized gravitational parameters. More work needs to be done to quantify how well the policies could adapt to new environments. Similarly, I think there needs to be more work done to explore the validity of using Deep RL for problems such as these, with comparisons to state of the art autonomy and path planning algorithms. However, I believe this work provides some interesting initial results and hints that Deep RL may be useful in solving problems in future asteroid exploration missions.

## 6 Future Work

As alluded to in previous sections, there is significant room for future work to be done on this problem. The first area of future work is in the simulation environment and dynamics. Some immediate improvements to the environment could include the use of a real asteroid shape model. Similarly, increasing the fidelity of the dynamics through the incorporation of solar radiation pressure, higher order spherical harmonic coefficients, or an alternative gravity model such as polyhedral would make the simulation environment more robust and accurate. The second area of future work would be in the problem formulation itself. It would be interesting to look into an expanded or potentially continuous action space to provide the agent more flexibility. Similarly, a more scientifically accurate placement of reward states based on the asteroid of interest would better represent a real-world mission. The addition of a heuristic policy could be also be beneficial to help guide the exploration. Expanding the pool of test cases would also be an interesting path to see how well a trained policy can generalize. A third area of future work, and one that I find particularly interesting, is leveraging this framework for gravity estimation. This could be approached through classical estimation techniques or potentially through Physics Informed Neural Networks (PINNs) [11]. This area of research is highly relevant to future asteroid missions. A fourth, and final (for this paper), area of future work could be in reformulating the problem as a coordinated distributed tasking problem with multiple hoppers. As can be seen there are a myriad of potential future exploration paths based on this work.

## 7 Conclusion

In this work I present the application of Deep Reinforcement Learning to asteroid exploration with a hopping rover. A simplified asteroid environment based on Bennu was developed and spherical harmonic gravity was utilized for the system dynamics. A model of a small hopping rover was developed to explore this environment with the aim of finding goal states representing scientifically valuable sites on the asteroid. This environment was developed in the Gymnasium framework and interfaced with the Stable Baselines 3 PPO implementation. We trained policies using PPO in two cases of reward distributions and compared the performance. The results showed that reward shaping plays an important role in training and success. Furthermore, we successfully developed an agent capable of robust autonomous exploration under unseen environmental parameters. This work explored the potential of leveraging Deep RL to solve interesting and challenging aerospace problems.

## 8 Additional Requirements

### 8.1 Implementation Details

This project was implemented by myself only. This project was developed in Python. I implemented the custom environment, physical model, problem definition, and dynamics from scratch. I used the Stable Baselines 3 PPO implementation off-the-shelf.

### 8.2 Release

The authors grant permission for this report to be posted publicly.

## References

- [1] D. S. Lauretta, S. S. Balram-Knutson, E. Beshore, W. V. Boynton, C. D. d'Aubigny, D. N. DellaGiustina, H. L. Enos, D. R. Golish, C. W. Hergenrother, E. S. Howell, C. A. Bennett, E. T. Morton, M. C. Nolan, B. Rizk, H. L. Roper, A. E. Bartels, B. J. Bos, J. P. Dworkin, D. E. Highsmith, D. A. Lorenz, L. F. Lim, R. Mink, M. C. Moreau, J. A. Nuth, D. C. Reuter, A. A. Simon, E. B. Bierhaus, B. H. Bryan, R. Ballouz, O. S. Barnouin, R. P. Binzel, W. F. Bottke, V. E. Hamilton, K. J. Walsh, S. R. Chesley, P. R. Christensen, B. E. Clark, H. C. Connolly, M. K. Crombie, M. G. Daly, J. P. Emery, T. J. McCoy, J. W. McMahon, D. J. Scheeres, S. Messenger, K. Nakamura-Messenger, K. Righter, and S. A. Sandford, “Osiris-rex: Sample return from asteroid (101955) bennu,” pp. 925–984, 10 2017.
- [2] B. Gaudet, R. Linares, and R. Furfaro, “Adaptive guidance and integrated navigation with reinforcement meta-learning,” *Acta Astronautica*, vol. 169, pp. 180–190, 4 2019. [Online]. Available: <http://arxiv.org/abs/1904.09865> <http://dx.doi.org/10.1016/j.actaastro.2020.01.007>
- [3] J. Kawaguchi, A. Fujiwara, and T. Uesugi, “Hayabusa—its technology and science accomplishment summary and hayabusa-2,” *Acta Astronautica*, vol. 62, pp. 639–647, 5 2008.
- [4] S. ichiro Watanabe, Y. Tsuda, M. Yoshikawa, S. Tanaka, T. Saiki, and S. Nakazawa, “Hayabusa2 mission overview,” pp. 3–16, 7 2017.
- [5] D. J. Scheeres, J. W. McMahon, A. S. French, D. N. Brack, S. R. Chesley, D. Farnocchia, Y. Takahashi, J. M. Leonard, J. Geeraert, B. Page, P. Antreasian, K. Getzandanner, D. Rowlands, E. M. Mazarico, J. Small, D. E. Highsmith, M. Moreau, J. P. Emery, B. Rozitis, M. Hirabayashi, P. Sánchez, S. V. W. 10, P. Tricarico, R.-L. Ballouz, C. L. Johnson, M. M. A. Asad, H. C. M. Susorney, O. S. Barnouin, M. G. Daly, J. A. Seabrook, R. W. Gaskell, E. E. Palmer, J. R. Weirich, K. J. Walsh, E. R. Jawin, E. B. Bierhaus, P. Michel, W. F. Bottke, M. C. Nolan, H. C. Connolly, D. S. Lauretta, and R. Team, “The dynamic geophysical environment of (101955) bennu based on osiris-rex measurements,” *Nature Astronomy*. [Online]. Available: <https://doi.org/10.1038/s41550-019-0721-3>
- [6] T. Yoshimitsu, T. Kubota, I. Nakatani, T. Adachi, and H. Saito, “Micro-hopping robot for asteroid exploration,” *Acta Astronautica*, vol. 52, pp. 441–446, 1 2003.
- [7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017.
- [8] J. Jiang, X. Zeng, D. Guzzetti, and Y. You, “Path planning for asteroid hopping rovers with pre-trained deep reinforcement learning architectures,” 2020. [Online]. Available: <https://doi.org/10.1016/j.actaastro.2020.03.007>
- [9] S. R. Chesley, A. S. French, A. B. Davis, R. A. Jacobson, M. Brozović, D. Farnocchia, S. Selznick, A. J. Liounis, C. W. Hergenrother, M. C. Moreau, J. Pelgrift, E. Lessac-Chenen, J. L. Molaro, R. S. Park, B. Rozitis, D. J. Scheeres, Y. Takahashi, D. Vokrouhlický, C. W. Wolner, C. Adam, B. J. Bos, E. J. Christensen, J. P. Emery, J. M. Leonard, J. W. McMahon, M. C. Nolan, F. C. Shelly, and D. S. Lauretta, “Trajectory estimation for particles observed in the vicinity of (101955) bennu,” *Journal of Geophysical Research: Planets*, vol. 125, 9 2020.
- [10] W. M. Kaula, “Theory of satellite geodesy, blaisdell publ,” *Co., Waltham, Mass*, vol. 345, 1966.
- [11] J. Martin and H. Schaub, “Physics-informed neural networks for gravity field modeling of small bodies,” *Celestial Mechanics and Dynamical Astronomy*, vol. 134, 10 2022.

1 Final Project Report

- 0 pts Correct