Formulating an Asteroid Exploration Mission as a Markov Decision Process

Kian Shakerin
Department of Aerospace Engineering Sciences
University of Colorado Boulder
Boulder, USA
ASEN 5264: Final Project
kian.shakerin@colorado.edu

***Abstract*—This project explores the formulation of an small body mission as a Markov Decision Process (MDP). We develop the problem as a spacecraft orbiting about the asteroid Bennu with the goal of reaching a final science orbit while accomplishing set of intermediary science objectives in varying orbits about the central body. We define our MDP and spacecraft states using Orbital Elements. We formulate negative rewards associated with each action taken as the Δv required to transition between the states in our MDP. By solving the MDP we determine a sequence of actions that represent a near-optimal path for performing mission operations about our target asteroid.**

I. Introduction

Missions to small celestial bodies, such as asteroids and comets often take place in complex, varied, and evolving environments. Small bodies contain unique information about our solar system's evolution and may provide clues in the search for life beyond Earth [1]. The exploration of these objects traditionally requires significant prior knowledge of the target environment, faces operational burdens, and is inflexible to changing conditions or new scientific aims [2]. Advancing the autonomous capabilities of spacecraft is an important aspect of furthering the exploration of scientifically valuable small bodies within the solar system and has been identified by NASA as an important area of research interest [3]. This project attempts to address one key component of the exploration problem: high level mission execution and decision making.

II. Background

High-level mission tasks often occur at fixed "science-orbits" or target regions about small bodies. The current mission paradigm is to use human-in-the-loop operations to perform these tasks. This process can be slow and cumbersome and often requires large teams of scientists and engineers to carry out intensive analysis work before executing these tasks. While this process has worked well in the past, it may be difficult to employ on more expansive and complex future missions.

One interesting approach to automating future missions is by leveraging decision making techniques such as MDPs, Partially Observable MDPs, and Reinforcement Learning to perform high-level tasks. An example of such an approach, and some of the inspiration for this project, is the paper "Autonomous Imaging and Mapping of Small Bodies Using Deep Reinforcement Learning" by Chan and Agha-mohammadi [4]. In this paper the authors frame a small-body imaging mission as a Partially Observable Markov Decision Process, solve it using Reinforcement Learning Techniques, and demonstrate favorable results compared to state-of-the-art spacecraft autonomy systems such as AutoNav. In this work we use an MDP to represent the sequential decision making process an autonomous spacecraft might need to go through to accomplish a set of mission goals.

III. Problem Formulation

A. Dynamics and Orbital Elements

For this project we simulate a spacecraft on a mission to the asteroid Bennu. Bennu was the target of the OSIRIS-REx mission, an ongoing asteroid sample return mission led by NASA [2]. One of the main aims of the OSIRIS-REx mission, and indeed one of the key aims of modern space exploration, is to better understand the Solar System history through analysis of samples gathered from small bodies. Bennu is representative of other scientifically interesting targets and provides a realistic small body about which we frame this project. Some information about Bennu that is used in our project is listed below:

- Target Body: Bennu
  - Mass = $7.329 * 10^{10}$ kg
  - Equatorial Radius = 0.245023 km
  - Gravitational Parameter (μ) = $4.8904 * 10^{-9} km^3 s^{-2}$

As previously mentioned, operations about small bodies occur in complex dynamic environments. Mission designers will typically model the dynamics of of these systems to include effects such as polyhedral gravity models, thirds-body perturbations, and SRP. For this work we use simplified two-body problem dynamics to model the movement of our spacecraft around the central body. We assume the mass of our spacecraft to be zero compared to the mass of the central body, allowing us to calculate orbits in this model. Our dynamical system is then derived from Newton's 3rd Law, equation (1) and our equation of motion becomes equation (2). In this system we also assume the mass of the orbiting body, our spacecraft, is significantly less than the central body and thus becomes zero.

$$F_1 = -F_2 = -G\frac{m_1 m_2}{r^2}\frac{\mathbf{r}}{\dot{\boldsymbol{r}}} \tag{1}$$

$$\ddot{r} = -\frac{\mu}{r^3}\mathbf{r} \tag{2}$$

By solving the equations of motion above, we can then derive the orbital motion of our spacecraft. Leveraging laws of conservation we derive the Orbital Elements and represent the motion of our spacecraft in Orbital Element Space. While the OE representation maps directly to Cartesian space, it can be used to more easily represent orbits. The orbital elements are listed below and can be seen visually represented in figure 1.

- a: Semi-Major Axis
- e: Eccentricity
- i: Inclination
- ω: Longitude of the Ascending Node
- ω: Argument of Periapsis
- f: True Anomaly

B. MDP Formulation

In this project we formulate a simple asteroid exploration mission as a Markov Decision Process (MDP). We define our MDP (S, A, T, R, γ) in the following manner:

- S: $s_k = \{a_k, e_k, i_k, \Omega_k, \omega_k\}$
- A: $\{a_{up}, a_{down}, e_{up}, e_{down}, i_{up}, i_{down}, \Omega_{up}, \Omega_{down}, \omega_{up}, \omega_{down}\}$
- T: P(s, a) $->$ s' = 0.99 ; s otherwise
- R: R(s, a, s') = R(s) + Cost(s, s')
- γ: 0.95

While the physical representation of a spacecraft in a mission scenario occupies a continuous state space, we discretize the problem into individual orbits represented by Orbital Elements. We have chosen the state space representation to be a 5-tuple (a, e, i , Ω, and ω). We choose to keep the true anomaly of each orbit fixed at 0 because varying the true anomaly does not change the orbit represented, only the location along the orbit.

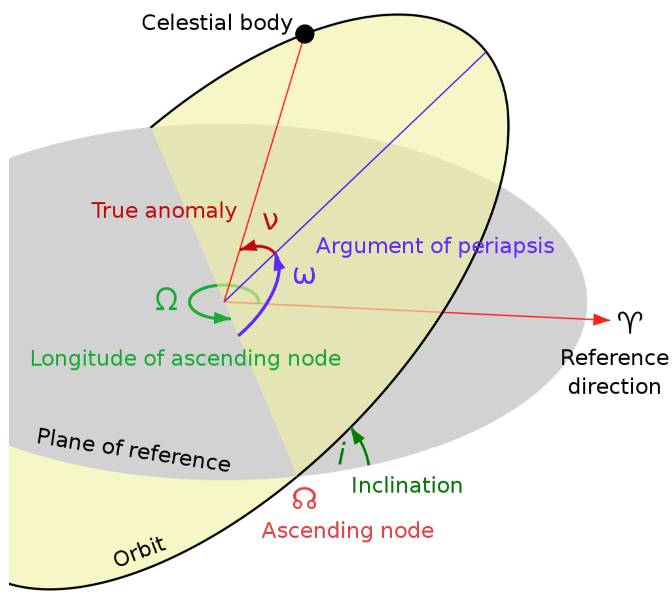

Fig. 1. Diagram showing the Orbital Elements [4].

To better model a space mission we decided to incorporate the Δv required for maneuvering between states as part of the reward structure. We model these as costs that are calculated for a state, action, new state set.

$$\Delta v_i = \frac{2\sin(\frac{\Delta i}{2})(1+e)na}{\sqrt{1-e^2}\cos(\omega)}, n = \sqrt{\frac{\mu}{a^3}} \tag{3}$$

$$\Delta v_\omega = 2\sqrt{\frac{\mu}{a(1-e^2)}}\sin\left(\frac{\Delta\omega}{2}\right) \tag{4}$$

$$\Delta v_{a,e,\Omega} = v_{s'} - v_s \tag{5}$$

To compute the cost for transferring between states when the action chosen includes changing semi-major axis (a), eccentricity (e), and right ascension of the ascending node (Ω), we use the solution to Lambert's problem. This is represented in equation (5) but the full solution is not presented here. Lambert's solution gives the initial and final velocities needed for a transfer between two points in Cartesian space given some time of flight under 2-body dynamics. We use the time of flight derived from the minimum energy solution of Lambert's problem and then compute the initial and final velocities needed for the transfer. We use the Universal Variables implementation of Lambert's solution as presented in Fundamentals of Astrodynamics and Applications by Vallado [7] We convert our state space from Orbital Elements to Cartesian coordinates before computing the Δv cost for taking one of the action above using Lambert's solution. For changes in inclination we use equation (3) and for changes to the argument of periapsis we use equation (4).

Δv costs were scaled by a multiplicative factor of 10^4. This was done to bring the them within the same order of magnitude as the rewards. Performing transfers in the gravitational environment of Bennu required relatively small Δv.

IV. Approach and Results

A. Level 1

The approach for our Level 1 (minimum working example) problem formulation involved a simplification of the MDP described in the previous section. In this approach we:

- Use a reduced state space: S = {a, i}
  - a $\in$ [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] km
  - i $\in$ [-50, -40, -30, -20, -10, 10, 20, 30, 40, 50] deg
- Use a reduced action space: $A = \{a_{up}, a_{down}, i_{up}, i_{down}\}$
- Use Δv for costs
- Use value iteration to find the value function at each state
- Use policy iteration to extract the optimal policy for each state
- Initial State = (10km, 50deg) — Terminal State = (1km, -20deg)

The problem setup is a 10x10 grid world where the x-axis represents changes in the orbital inclination of the spacecrafts orbit and the y-axis represents change sin the semi-major axis. To solve the MDP we use simple Value Iteration and then policy iteration to find the optimal action for each state. We start with our spacecraft in state (9,9), which maps to (a=10km, i=50deg) and take the optimal sequence of actions to reach the end state (0,3), which maps to (a=1km, i=-20deg). In figure 2 we can see the optimal action for each state. We start in an orbit that has high inclination and a large semi-major axis. Positive and negative rewards have been placed in intermediary orbits to simulate a set of secondary science objectives. From the initial state the spacecraft first decreases the semi-major axis of it's orbit, then decreases the inclination, and so on, using the optimal set of actions to reach the end state. The spacecraft visits the intermediary reward states, per the policy. Figure 3 shows the orbits the spacecraft transitions between, a representation in Cartesian space of the states and actions taken to reach the end goal state.

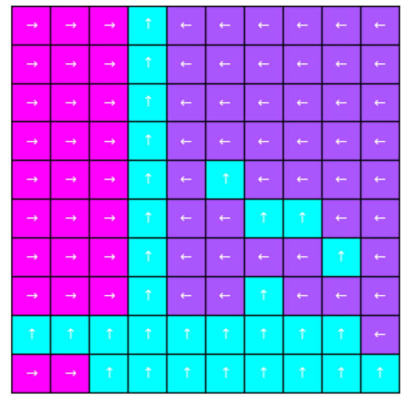

Fig. 2. Optimal actions to take in each state of the Level 1 MDP. Actions are also color coded for visualization purposes.

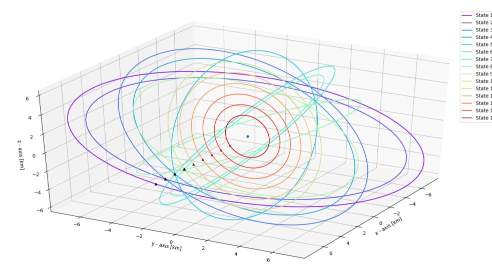

Fig. 3. Solution states of the Level 1 MDP. The ordered list of states maps the color of the orbit to the sequence of state transitions. The black triangles represent the discrete Orbital Element states mapped to Cartesian space. Each triangles represents one state.

B. Level 2

The approach for our Level 2 (main approach) problem formulation involved the MDP described in the previous section. In this approach we:

- Use the MDP described above with states built from these parameters:
  - a $\in$ [1.75, 3, 4, 5] km
  - e $\in$ [0, .2, .4, .6]
  - i $\in$ [-45, 0, 45] deg
  - $\Omega \in$ [0, 45, 90] deg
  - $\omega \in$ [0, 22.5, 45] deg
- Use MCTS
- Initial State = (5, .6, 45, 90, 45) — Terminal State = (1.75, .2, 0, 0, 0)

In our Level-2 approach, the state and action space becomes significantly larger than in the Level

1 approach. By allowing the eccentricity (e), right ascent As we move from a 2D state space representation to a 5D state space and from 4 possible actions to 10, we quickly begin to suffer from the curse of dimensional. Even with relatively few discrete parameters in each dimension, the problem grows exponentially. We also added transition uncertainty to our transition model. Each time an action is taken there is a 1-percent chance the spacecraft stays in the same state. This is a fairly simplistic way to represent an issue such as a missed burn.

Using Value Iteration to solve the problem becomes slow and impractical. For this reason we turn to Monte Carlo Tree Search (MCTS), an online algorithm for solving MDPs. With MCTS we do not need to find the value function at each state which is beneficial given our state and action space size. The algorithm does not need any domain knowledge and generally has good computational performance [5]. The MCTS algorithm implemented is taken from the lecture notes and textbook and uses the following approach:

- Search: Traverse the tree (explored state space), starting at the root node. We use a Upper Confidence Bound (UCB) exploration policy to select the actions from various states.
- Expand: When we reach a state that has not been previously explored we expand our tree at this new child node. We do this for each action in our action space.
- Rollout: From the new child node we perform a roll-out policy with a random search to generate an estimate of the value function at that node.
- Backup: We use information gathered to update value estimates and visit counts across nodes in the tree.
- MCTS Parameters:
  - Max Rollout Steps: 200
  - Tree Depth: 50
  - Discount: 0.95
  - UCB Exploration Constant: 100

After some tuning of the MCTS parameters and the MDP state space, we were able to generate policies that looked similar the one shown in 4. We were able to find policies that explored the state space and stepped through intermediary science orbits which were assigned positive rewards before ultimately finding the terminal state. The terminal state in this case is similar to the final science orbit of OSIRIS-REx. Comparing figure 4 with figure 5 we can see that the solution state space is significantly smaller than the full state space. It should be noted that the path (set of state, action pairs) taken by the spacecraft in figure 4 does include some repeat visits to states. At times there were issues regarding local maxima and balancing Δv costs that led to this behavior and resulted in the spacecraft transitioning repeatedly between two states. We can see in figure 4 that the spacecraft starts in a larger orbit and progressively transitions down in semi-major axis towards the goal. On the way the spacecraft transition some more inclined and eccentric orbits before then passing through reward states closer to the central body and finally reaching the gaol science orbit.

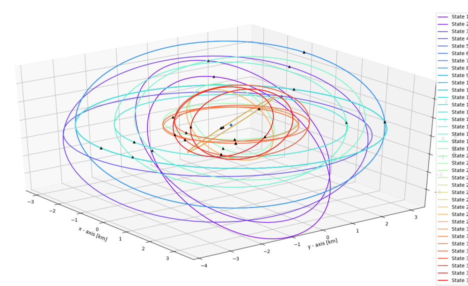

Fig. 4. Solution states of an example MCTS run on the Level 2 MDP. The ordered list of states maps the color of the orbit to the sequence of state transitions. The black triangles represent the discrete Orbital Element states mapped to Cartesian space. Each triangles represents one state.

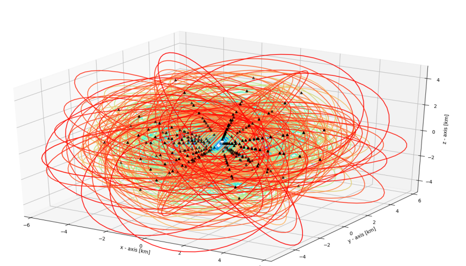

Fig. 5. Full state space for the Level 2 MDP. This figure is included to show the contrast between the full state space and the solution space.

V. Discussion and Conclusion

Through this project we were able to demonstrate how an asteroid exploration mission can be modeled as a discrete set of states and actions represented by a Markov Decision Process. We began by limiting our state space by only varying the semi-major axis and the inclination of the orbits the spacecraft was allowed to be in (ie. the state space). Using value iteration and policy iteration we were able to determine the optimal set of actions for traversing the state space from an initial orbit to a final science orbit. Next we we expanded our state space to include variations in eccentricity, right ascension of the ascending node, and argument of periapsis. To solve this new MDP with a larger state and action space, we turned to Monte Carlo Tree Search. Using MCTS we were able to find a policy for traversing the state space that visited secondary orbits representing lesser science objectives before finally reaching the goal state.

Some of the main issue encountered in this work came from setting up the reward and cost structure of the MDPs. In the classic, 2D grid-world representation that was used for the Level 1 approach, placing positive and negative rewards is relatively straightforward. However, in the 5D state space of our Level 2 MDP, reward shaping became significantly less intuitive. Both the placement and scale of the rewards became an issue when formulating the MDP. Scaling was also important because the Δv costs structure associated with transitioning between states became more complex and if the costs of transitions were not scaled correctly against the potential rewards then issues would arise with the MCTS solution. Similarly, the MCTS often ended up finding local maxima and transitioning between the same two states when the costs and rewards were not balanced correctly. Selecting viable parameters for the MCTS algorithm also posed a challenge because it was difficult to get a grasp on how the trees were expanding in the higher-dimensional state and action space.

VI. Future Work

While this project yielded interesting results, there is much potential future work that can be done. Some immediate future work that could be done include developing heuristic roll-out policies for the MCTS, implementing other algorithms for solving MDPs such as Q-Learning, and investigating methods for better reward shaping. The Level 3 stretch goal proposed is the next step in this line of work. That approach would include a more realistic dynamical model combined with a POMDP formulation of the problem. This would allow us to better capture the dynamics and uncertainties in the problem space. Similarly, placing constraints on Δv use or using it as a parameter to optimize could also lead to interesting new avenues of research.

Acknowledgment

All work was done by the author, Kian Shakerin. The author grants permission for this report to be posted publicly.

References

[1] B. Gaudet et al., "Adaptive Guidance and Integrated Navigation with Reinforcement Meta-Learning." (2020)

[2] D.S. Lauretta, et al. "OSIRIS-REx: Sample Return from Asteroid (101955) Bennu" (2017)

[3] Starek, Joseph et al., (2016). Spacecraft Autonomy Challenges for Next-Generation Space Missions. Lecture Notes in Control and Information Sciences. 460.

[4] Chan, D. M., Agha-Mohammadi, A. A. (2019). Autonomous Imaging and Mapping of Small Bodies Using Deep Reinforcement Learning. IEEE Aerospace Conference Proceedings, 2019-March

[5] W.C.Lasunncty (2010) Digram illustrating and explaining various terms in relation to orbits of celestial bodies. [Online].Available:https://commons.wikimedia.org/wiki/File:Orbit1.svg

[6] Browne, C. B., et al. (2012). A Survey of Monte Carlo Tree Search Methods. IEEE Transactions on Computational Intelligence and AI in Games.

[7] Vallado DA McClain WD. Fundamentals of Astrodynamics and Applications. New York: McGraw-Hill Companies; 1997.