

Rocket Recycling via Reinforcement Learning

Alvin Mou

Worcester Academy, Worcester, MA, USA

mralvinmou@gmail.com

Abstract. Rocket recycling offers significant potential to reduce launch costs, increase launch frequency, and reduce space debris, as evidenced by SpaceX's success with rocket relandings. Traditional model-based control methods face challenges in handling the complex, non-linear dynamics inherent in the rocket recycling process. Reinforcement learning (RL), by contrast, provides a flexible, data-driven approach which is better suited to managing these complicate dynamic conditions. In this paper, we investigate two research problems in RL-based rocket recycling control: 1) how the granularity of control variables (e.g. thrust force levels) impacts the control performance; 2) how the environmental interference during training affects the robustness of RL controller. Our evaluation results demonstrate that increasing the granularity of control variables significantly slows the convergence of RL-based controller to a stable state. Moreover, introducing environmental interference during training—such as wind disturbances—improves both the efficiency and robustness of the RL-based controller. Our findings provide insights into how the RL-based controller can be more effectively used for rocket recycling tasks and highlight considerations that can enhance its applicability and robustness in dynamic environments.

Keywords: rocket recycling, reinforcement learning, convergence, robustness.

1. Introduction

Rocket recycling has shown its importance to astronomy since 2015 when SpaceX made its first successful rocket relanding [1]. Reusing the rocket provides several benefits: reducing the cost and complexity, allowing more frequent launches, and helping to reduce the space pollution debris. As to cost, unlike traditional rockets, it recycles the thruster, enabling it to be used for subsequent launches. However, how to precisely and safely control the recycling process of rockets is a challenging problem.

Traditional control methods, such as PID control [2] and model predictive control [3], are model-based and rely on accurately modeling system dynamics. However, due to the inherent difficulties in precisely modeling the dynamics of rocket systems, these traditional methods exhibit limited performance in the rocket recycling process [4]. To address this challenge, data-driven, learning-based approaches, such as reinforcement learning (RL), have been proposed [5-7]. RL-based controller for rocket recycling offers greater flexibility by not relying on a specific rocket model and has demonstrated promising performance in rocket recycling control.

In this paper, we study the problem of using RL-based controller for rocket recycling. Specifically, we investigate two key research problems: 1) how different granularities of the thrust forces and angles affect control performance, and 2) how environment interference impacts the rocket recycling control performance of the RL-based controller.

To address the first research question, we conduct experiments with varying granularities of control thrust forces and angles. For the second research question, we introduce a random variable to simulate environmental interference during rocket recycling, and we vary its magnitude to evaluate its impact on control performance.

Our simulation experiments leverage the platform developed by Zou in [8], inspired by SpaceX's Starship SN10. This platform simulates a simplified Starship model within a dynamic 2D plane environment, focusing on aspects of thrust vectoring and angular constraints. The project consisted of settings of gravitational and physics rules, rocket dynamics, and plotting systems. Training through episodes with a reward system that constantly reflects on the performance of the generated simulation. Having numerous maneuvers, it learns the best angular thrust and adjustment, finally achieving hovering or landing [9].

Our evaluation results yield two main findings. First, we observe that increasing the granularity of thrust forces and angles do not improve the performance of RL-based controller. Instead, it makes it slower for the controller to converge. Furthermore, we demonstrate that including environment perturbation during training can enhance the robustness of RL-based controller, leading to more stable control performance when faced with external interferences. Our findings offer valuable insights, which motivates further research into optimizing the RL-based controller for rocket recycling, particularly in developing strategies to its the control performance and robustness.

2. Preliminaries

In this section, we first describe the mathematical model used for simulating rocket recycling process. Then, we briefly introduce reinforcement learning.

2.1. Rocket Recycling

We define $s_t = [x_t, y_t, \theta_t]^T$ as the state of the rocket, where x_t is the x-axis location of the rocket at time step t , y_t is the y-axis location of the rocket at step t , and θ_t is the angle of the rocket at time step t .

The state transition equations for the rocket are defined as follows:

$$\frac{dx_t}{dt} = v_{x,t} \quad (1)$$

$$\frac{dy_t}{dt} = v_{y,t} \quad (2)$$

$$\frac{d\theta_t}{dt} = v_{\theta,t} \quad (3)$$

$$\frac{d\phi_t}{dt} = v_{\phi,t} \quad (4)$$

$$\frac{dv_{x,t}}{dt} = a_{x,t} \quad (5)$$

$$\frac{dv_{y,t}}{dt} = a_{y,t} \quad (6)$$

$$\frac{dv_{\theta,t}}{dt} = a_{\theta,t} \quad (7)$$

$$a_{x,t} = f_{x,t} + w_t - \rho v_{x,t} \quad (8)$$

$$a_{y,t} = f_{y,t} + g - \rho v_{y,t} \quad (9)$$

$$a_{\theta,t} = \frac{f_{a,t}H}{2I} \quad (10)$$

$$f_{a,t} = -f_t * \sin(\phi_t) \quad (11)$$

$$f_{b,t} = f_t * \cos(\phi_t) \quad (12)$$

$$f_{x,t} = f_{a,t} * \cos(\theta_t) - f_{b,t} * \sin(\theta_t) \quad (13)$$

$$f_{y,t} = f_{a,t} * \sin(\theta_t) + f_{b,t} * \cos(\theta_t) \quad (14)$$

$$\rho = \frac{1}{\sqrt{\frac{250}{g}}} \quad (15)$$

where $v_{x,t}$ is the x-axis velocity of the rocket at time step t , $v_{y,t}$ is the y-axis velocity of the rocket at time step t , $a_{x,t}$ is the x-axis acceleration of the rocket at time step t , $a_{y,t}$ is the y-axis acceleration of the rocket at time step t , $a_{\theta,t}$ is the angular acceleration of the rocket at time step t , g is the gravitational acceleration, w_t is the acceleration along the x-axis due to external environment interference at time step t , f_t is the thrust force at time step t , ϕ_t is the thrust angle at time step t , $v_{\phi,t}$ is the velocity of thrust angle at time step t , H is the height of the rocket, I is the moment of inertia of the rocket, ρ is the drag coefficient during rocket recycling. The detailed rocket system dynamics are illustrated in Figure 1.

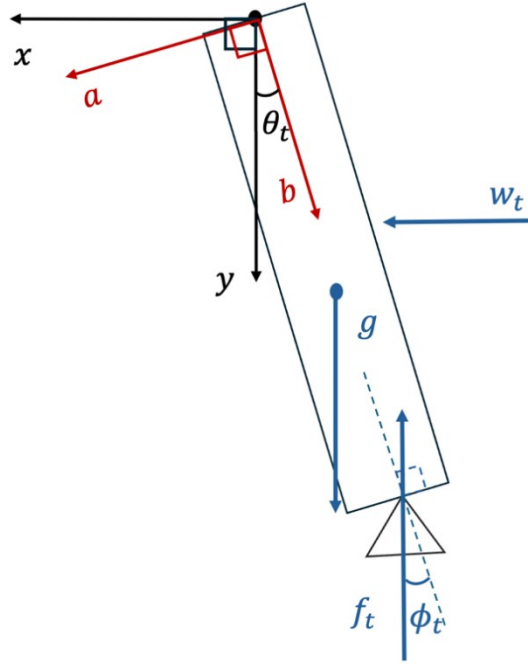


Figure 1. Rocket System Dynamics.

To simulate the rocket recycling process, we update the state of rocket at each time step using equations below:

$$x_{t+1} - x_t = v_{x,t} \Delta t + \frac{1}{2} a_{x,t} (\Delta t)^2 \quad (16)$$

$$y_{t+1} - y_t = v_{y,t}\Delta t + \frac{1}{2}a_{y,t}(\Delta t)^2 \quad (17)$$

$$\theta_{t+1} - \theta_t = v_{\theta,t}\Delta t + \frac{1}{2}a_{\theta,t}(\Delta t)^2 \quad (18)$$

$$\phi_{t+1} - \phi_t = v_{\phi,t}\Delta t \quad (19)$$

$$v_{x,t+1} - v_{x,t} = a_{x,t}\Delta t \quad (20)$$

$$v_{y,t+1} - v_{y,t} = a_{y,t}\Delta t \quad (21)$$

$$v_{\theta,t+1} - v_{\theta,t} = a_{\theta,t}\Delta t \quad (22)$$

where $x_{t+1} - x_t$ represents the change of x-axis location from time step t to $t + 1$, $y_{t+1} - y_t$ represents the change of y-axis location from time step t to $t + 1$, $\theta_{t+1} - \theta_t$ represents the change of rocket angle from time step t to $t + 1$, $\phi_{t+1} - \phi_t$ represents the change of thrust angle from time step t to $t + 1$, $v_{x,t+1} - v_{x,t}$ represents the change of x-axis velocity from time step t to $t + 1$, $v_{y,t+1} - v_{y,t}$ represents the change of y-axis velocity from time step t to $t + 1$, $v_{\theta,t+1} - v_{\theta,t}$ represents the change of rocket angle velocity from time step t to $t + 1$, and Δt is the duration of each time step.

2.2. Reinforcement Learning

Reinforcement Learning (RL) is a type of machine learning algorithm, where an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of reward based on the actions it takes, and it uses this feedback value to improve its future performance [10-11].

In this paper, RL is used to train a controller for rocket recycling process, allowing the rocket to return the target place in a dynamic and complex environment with factors such as gravity, air resistance, inertia, wind interference, etc. The RL-based controller can learn to automatically adjust the position of the rocket, based on feedback (i.e. rewards) linked to its decisions on thrust angle and force. We describe in detail all components of RL-based controller below.

1) State: We define the state in RL at time step t as $s_t = [x_t, y_t, \theta_t, v_{x,t}, v_{y,t}, v_{\theta,t}, \phi_t]$, including the x-axis location, y-axis location, rocket angle, x-axis velocity, y-axis velocity, rocket angle velocity, thrust angle at time step t .

2) Action: We define the action in RL at time step t as $a_t = [f_t, v_{\phi,t}]$, which represents the thrust force and the velocity of thrust angle at time step t . We further define the action set as A , which includes all the combinations of f_t and $v_{\phi,t}$. Note that since f_t and $v_{\phi,t}$ can have different granularities, the cardinality of A can change.

3) State transition: Given the state and action at time step t (i.e. s_t and a_t), the state at time step $t + 1$ (i.e. s_{t+1}) can be derived. To simulate it, we use Equations (1-22) described in Section 2.1 to compute s_{t+1} .

4) Reward: The reward r_t is defined as:

$$r_t = \begin{cases} 0, & \text{if } |\theta_t| > 0.5\pi \\ 0.25, & \text{if } d_{target} < d_t \leq 2d_{target} \\ 0.5, & \text{if } d_t \leq d_{target} \end{cases} \quad (23)$$

where $d_t = \sqrt{(x_t - x_{target})^2 + (y_t - y_{target})^2}$, (x_{target}, y_{target}) is the target return location during rocket recycling and d_{target} is an adjustable parameter indicating the acceptable error between target return location and the actual location of the rocket.

5) Policy: The policy in RL is a function, which takes as input s_t and outputs the probability for each action candidates. Formally, it is defined as $\Pr[a_t = a | s_t = s] = P_\alpha(s)$, where α is the parameter of the policy function, $\Pr[a_t = a | s_t = s]$ represents the probability of selecting action a_t as a given current state $s_t = s$.

Note that we use Actor-Critic algorithm [12] to train the RL-based control, which is one of the most widely used RL algorithms.

3. Design

In this section, we describe how to change the granularity of the control variable and environment interference for our experiments.

3.1. Varying the Granularity of Control Variables

We implement three different setups for the thrust force f_t , with varying level of granularity. We describe them in detail as follows:

1) Coarse granularity (default): Under the default setup, the thrust force f_t has three options: $\{0.2g, 1.0g, 2.0g\}$, corresponding to thrust levels at 0.2, 1.0, and 2.0 times of gravitational acceleration g . The thrust angle velocity $v_{\phi,t}$ also has three options: $\{-\frac{30}{180}\pi, 0, \frac{30}{180}\pi\}$. In total, there are 9 combinations for control variable in default setup, which means the action in RL at time step t as $a_t = [f_t, v_{\phi,t}]$ has 9 choices.

2) Medium granularity: Under this setup, we consider a more refined range of options for the thrust force f_t and thrust angle velocity $v_{\phi,t}$. Specifically, f_t is adjusted in increments of 0.5 times of gravitational acceleration g , spanning from 0 to $2g$ (i.e. 5 different choices); $v_{\phi,t}$ is adjusted in increments of $\frac{15}{180}\pi$, ranging from $-\frac{30}{180}\pi$ to $\frac{30}{180}\pi$ (i.e. 5 different choices). In total, there are 25 combinations for control variable in this setup, which means the action in RL at time step t as $a_t = [f_t, v_{\phi,t}]$ has 25 choices.

3) Fine granularity: This setup offers the most fine-granular options for the thrust force f_t and thrust angle velocity $v_{\phi,t}$. Specifically, f_t is adjusted in increments of 0.2 times of gravitational acceleration g , spanning from 0 to $2g$ (i.e. 11 different choices); $v_{\phi,t}$ is adjusted in increments of $\frac{5}{180}\pi$, ranging from $-\frac{30}{180}\pi$ to $\frac{30}{180}\pi$ (i.e. 13 different choices). In total, there are 143 combinations for control variable in this setup, meaning the action in RL at time step t as $a_t = [f_t, v_{\phi,t}]$ has 143 choices.

3.2. Varying the Environmental Interference

To simulate the environment interference during rocket recycling process, we introduce a random external environmental interference, which will impact the rocket's state at each time step. As already mentioned in Section 2.1, we define w_t as the acceleration along the x-axis due to external environment interference at time step t , and we assume that w_t is a time-independent random variable satisfying uniform distribution between $-w$ and w . This addition creates a more challenging environment for the rocket during hovering and landing, allowing us to train the RL-based controller for better robustness.

Specially, we implement four different setups to simulate varying levels of environmental interference:

1) Default Setup: By default, there is no external environmental interference.

2) Small Interference: We set $w=0.1$, which means w_t is a random variable uniformly distributed between -0.1 and 0.1.

3) Medium Interference: We set $w = 0.2$, which means w_t is a random variable uniformly distributed between -0.2 and 0.2 .

4) Large Interference: We set $w = 0.5$, which means w_t is a random variable uniformly distributed between -0.5 and 0.5 .

These setups can help us understand how different levels of environmental interference can affect the RL-based controller's performance during rocket recycling.

4. Evaluation

In this section, we present our evaluation results under different setups described in Section 3. We conduct experiments using the simulation platform developed by Zou in [1], and our goal is to maximize the reward of RL-based controller (see Section 2.2 for detailed definition of reward). Moreover, we train the RL-based controller for 20000 episodes for each setup and report how the reward (i.e. average performance of the controller) changes during training.

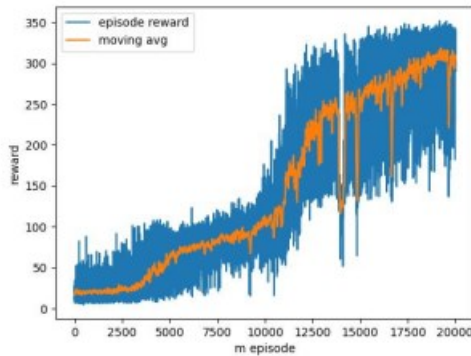
4.1. Results Under Different Granularities of Control Variables

As demonstrated in Figure 1, increasing the granularity of control variables (i.e. thrust force and thrust angle velocity) will make it harder for the controller to converge.

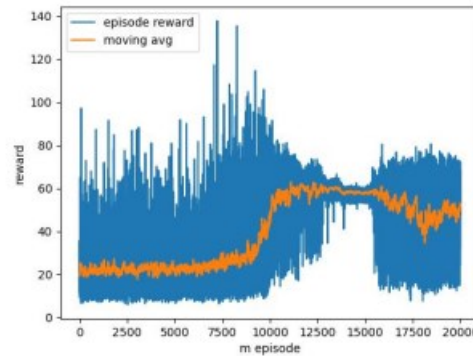
Specifically, Figure 1 (a) shows the simulation results when default setup (i.e. coarse granularity) is used for control variables. It serves as a baseline that can be used to compare with other setups.

Figure 1 (b) shows the results when control variables with medium granularity is used for RL training. The thrust force varies from 0 to $2g$ with an interval of $0.5g$, and the thrust angle velocity varies from $-\frac{30}{180}\pi$ to $\frac{30}{180}\pi$, with an interval of $\frac{15}{180}\pi$. We observe that the reward stops increasing at training episode 10000 and converges to a local optimal state with value 60. Compared to our baseline which achieves a reward of 300, the training of RL-based controller with more medium granularities is considered failed. This says, with the given granularity of thrust force and thrust angle velocity, the RL-based controller cannot make the rocket recycled successfully.

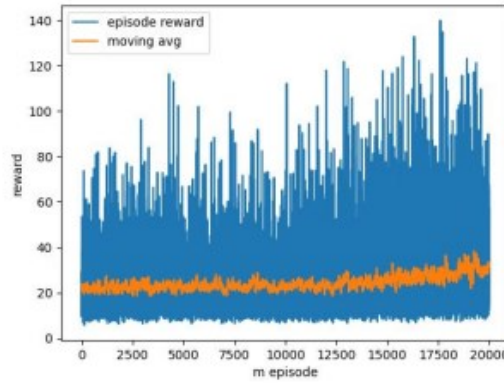
Figure 1 (c) further shows the simulation results when control variables with fine granularity is used for training the RL-based controller (see Section 3.1 for detailed definition of fine granularity). The thrust force varies from 0 to $2g$ with an interval of $0.2g$, and the thrust angle velocity varies from $-\frac{30}{180}\pi$ to $\frac{30}{180}\pi$, with an interval of $\frac{5}{180}\pi$. As illustrated in Figure 1 (c), the reward of RL-based controller fails to converge to a high value, only staying at the value of 30. Compared to baseline whose reward goes until 300, the training of this model with finer granularity of control variable is also considered as a failure, meaning with the given thrust force and thrust angle velocity options, the RL-based controller cannot be trained well.



(a) Coarse granularity



(b) Medium granularity



(c) Fine granularity

Figure 2. Training results with different granularities of control variables.

4.2. Results Under Various Environmental Interference

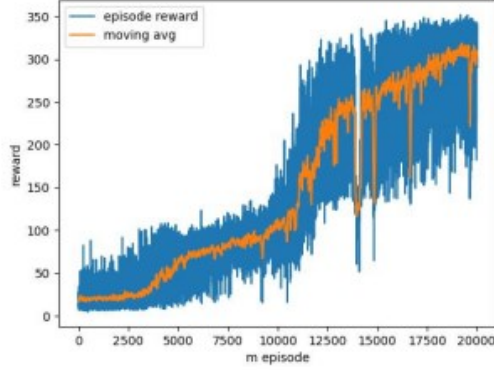
In Figure 3, we demonstrate that adding environment perturbation during training can make the controller more robust. More specifically, Figure 3 (a) shows the reward of RL-based controller when the default setup (i.e. no environmental interference) is used in training. Figure 2 (b)-(d) show the reward of RL-based controller with small, medium, and large environmental interference during training.

In Figure 3 (b), we observe that adding a random variable with value between -0.1 and 0.1 affects the rocket's position throughout the whole process of landing. At the 10000th episode, the moving average for the reward reaches 200. Then, at the 15000th episode, the moving average reward reaches 300 at the 20000th episode. Finally, the moving average reward converges to 350. By contrast, for the baseline in Figure 3 (a), the reward only reaches 100 at the 10000th episode, and it converges to a value of 300 until the 20000s episode. This indicates when external environmental interference is added during training, the RL-based controller can converge faster and achieve better control performance.

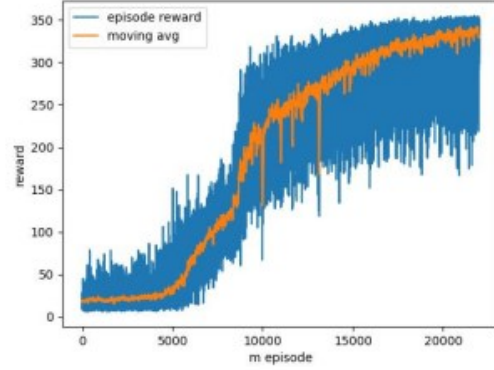
Figure 3 (c) shows the reward of RL-based controller with random interference ranging from -0.2 to 0.2. It can affect the rocket's position more than the setup in Figure 3 (b), since the rocket will receive a bigger span of possible environmental interference. From the figure, we can clearly tell the moving average reward starts increasing before the 5000s episode and raises to 250 moving average rewards at the 10000s episode. By contrast, for the standard model, at the 10000th episode, it only has a moving average reward of 100.

Figure 3 (d) shows the reward of RL-based controller with random interference ranging from -0.5 to 0.5, interfering with the rocket's position during relanding even more. This figure shows the moving average reward starts climbing steeply from around the 2500s episode, while the baseline setup in Figure 3 (a) only starts climbing up faster at around the 11000s episode. The reward in Figure 3 (d) also increases earlier compared with Figure 3 (b) and (c). Moreover, at around the 12500th episode, the RL-based controller can already reach a moving average reward of 300. By contrast, for the baseline model in Figure 3 (a), the reward is only 200 at the 12500th episode and it takes 17500 episodes to reach 300. The maximum average reward in Figure 3 (d) also reaches almost 350.

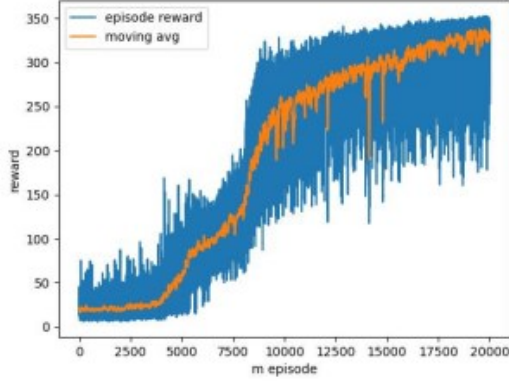
In summary, we can conclude that adding external environmental interference during training can improve both the control performance (i.e. maximizing reward) and the convergence speed of RL-based controller.



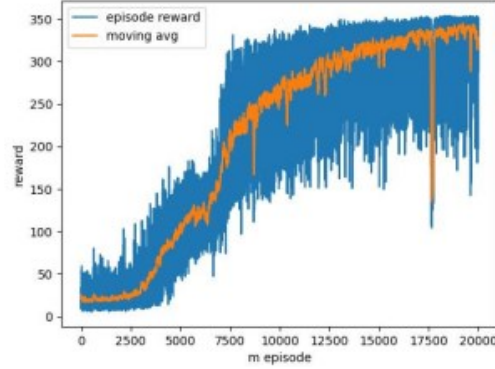
(a) No interference



(b) Small interference



(c) Medium interference



(d) Large interference

Figure 3. Training results when varying the wind interference.

5. Conclusion

In this paper, we explore the application of reinforcement learning (RL)-based controllers in rocket recycling, focusing on the effects of control variable granularity and environmental interference on performance. Our main objective in this paper is to determine how these factors influence the convergence and robustness of the RL-based controller.

From our evaluation, we observed that increasing the granularity of control variables, such as thrust force and angle velocity, does not enhance the performance of the RL-based controller. On the contrary, higher granularity slows down the convergence and reduces the overall reward. This finding highlights the challenges of training with detailed control parameters and suggests that coarser control may be more effective for such applications.

Additionally, our results demonstrate that incorporating environmental interference during training can significantly enhance the robustness of the RL-based controller and lead to faster convergence and higher rewards. For instance, training with small to large environmental interference allowed the RL-based controller to achieve a maximum reward of approximately 350, surpassing the baseline's 300.

These findings underscore the potential of RL-based controllers in rocket recycling when trained under varied environmental conditions. The inclusion of external environmental perturbations during training can be a strategic approach to improve both control performance and robustness. Future research can explore how to design methods that select better granularity for control variable and add more

adaptive environment interference to further optimize the performance and efficiency of RL-based controllers for rocket recycling.

References

- [1] Space Economy Institute (2023). The Role of Reusable Rockets in Reducing the Cost of Access to Space. <https://spaceeconomyinstitute.com/2023/08/24/the-role-of-reusable-rockets-in-reducing-the-cost-of-access-to-space/>
- [2] Dokumaci, Korhan, M. Timur Aydemir, and Metin U. Salamci. "Modeling, PID control and simulation of a rocket launcher system." *2014 16th International Power Electronics and Motion Control Conference and Exposition*. IEEE, 2014.
- [3] Wang, Jinbo, Naigang Cui, and Changzhu Wei. "Optimal rocket landing guidance using convex optimization and model predictive control." *Journal of guidance, control, and dynamics* 42, no. 5 (2019): 1078-1092.
- [4] Xue, Shuai, Zhaolei Wang, Hongyang Bai, Chunmei Yu, and Zian Li. "Research on Self-Learning Control Method of Reusable Launch Vehicle Based on Neural Network Architecture Search." *Aerospace* 11, no. 9 (2024): 774.
- [5] Li, Wenting, Xiuhui Zhang, Yunfeng Dong, Yan Lin, and Hongjue Li. "Powered Landing Control of Reusable Rockets Based on Softmax Double DDPG." *Aerospace* 10, no. 7 (2023): 590.
- [6] Jin, Guangyin, Jincan Huang, Yanghe Feng, Guangquan Cheng, Zhong Liu, and Qi Wang. "Addressing the Task of Rocket Recycling with Deep Reinforcement Learning." In *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, pp. 284-290. 2018.
- [7] Rodriguez, Juan. "Recyclable Rocket Modeling and Control in Complex Weather Conditions." (2023).
- [8] Zhengxia Zou (2021). Github Repository: Rocket-recycling with Reinforcement Learning. <https://github.com/jiupinjia/rocket-recycling>
- [9] Sebastian, R., Joshua, P., Corey, N., (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. <https://www.mdpi.com/2078-2489/11/4/193>
- [10] Shin, Joohyun, Thomas A. Badgwell, Kuang-Hung Liu, and Jay H. Lee. "Reinforcement learning—overview of recent progress and implications for process control." *Computers & Chemical Engineering* 127 (2019): 282-294.
- [11] Li, Yuxi. "Deep reinforcement learning: An overview." *arXiv preprint arXiv:1701.07274* (2017).
- [12] Han, Minghao, Lixian Zhang, Jun Wang, and Wei Pan. "Actor-critic reinforcement learning for control with stability guarantee." *IEEE Robotics and Automation Letters* 5, no. 4 (2020): 6217-6224.